فهرست

بخش اول

روشهای مختلف برنامه نویسی روی اینترنت

فصل اول:
برنامه نویسی کاربردی روی سرویس دهندهٔ وب

۱-۱ مقدمه

ارتباط در عصر حاضر و در شروع قرن بیست و یکم الفبای زندگی صنعتی و مدرن و متمدن جوامع مترقی این کره خاکی را تشکیل می دهد. اهمیت ارتباطات و در معنایی ساده تر ، تبادل اطلاعات بین جوامع بشری و انسانها که تشکیل دهندهٔ جوامع بشری می باشند و در عصر امروز آنقدر مهم و حائز اهمیت است که به زعم بزرگان علم ، در جهان امروز اگر کسی خود را بی نیاز از تبادل اطلاعاتی بداند در حقیقت دچار توهمی بزرگ از یک محیط پررمز و واقعیت گشته و در حقیقت از آمادگی لازم برای ورود به قرن بیست و یکم برخوردار نیست و از قافلهٔ علم بشری به شدت عقب مانده است.

نظریات ارتجاعی در محدود کردن جوامع بشری در دستیابی به اطلاعات محکوم به شکست هستند پیشرفت علم وفن در جهان امروز آنقدر سریع و شتابان است که هرگز هیچ کشور مقتدری در دنیا خود را از دریافت اخبار و اطلاعات مربوط به مراکز دیگر یا کشور دیگربی نیاز نمی داند. حتی کشورهای در حال توسعه ، یا حتی کشورهای فقیر به تناوب سود می برند. در جهان متنوع و رنگارنگ ما قومیتها و مذاهب و فرهنگها جوامع مختلف به از بین رفتن منازعات قومی و فرهنگی مخاصمات دینی منجر می شود و افکار ملل دنیا بطرز بی سابقه ای به یکدیگر نزدیک می گردد. علم انحصاری و تکنولوژی که سابق بر این منحصراً در اختیار کشورهای پیشرفته صنعتی قرار داشت از انحصار آنها خارج می گردد و در سطح جهان گسترده می شود و ملل مختلف از نعمات آن بهره مند می شوند.

اگر حادثه ای در نقطه ای از جهان روی داد با مخابره خبر آن، در عرض کمتر از یک صدم ثانیه به سراسر جهان، همهٔ مردم دنیا به کمک این قسمت از کره خاکی می شتابند و همه اینها ممکن نیست مگر به وسیله تکنولوژی ارتباط و اطلاعات. گسترش سیستمهای اطلاعاتی در سر تا سر جهان از قبیل ماهواره ها سیستمهای مایکروویو ، سیستمهای اطلاعات کامپیوتری و غیره .... جهان بزرگ ما را تبدیل به یک دهکدهٔ کوچک کرده است، بطوری که هر فرد از هر ملیتی در دورترین فاصلهٔ کرهٔ خاکی می تواند در آن واحد با دیگری ارتباط برقرار کند و هر اتفاقی هر چقدر کوچک و بی اهمیت توسط سیستمهای پیشرفته تبادل اطلاعات در عرض صدم ثانیه به دورترین فاصله از آن نقطه مخابره می شود ، گو اینکه فاصله ها در جهان ما از بین رفته و بعد جغرافیایی کرهٔ زمین و چه بسا فضای کیهانی تبدیل به مسافتی کوتاه شده است.

۱-۲ تاریخچهٔ اینترنت

برای درک اساسی و بنیادین یک علم و جهت گیری به سمت جنبه های علمی آن دانستن تاریخ و علل بوجود آمدن آن ضروری می نماید ، لذا نویسنده ابتدا به تشریح تاریخ این علم می پردازد. تولد ارتباطات کامپیوتری تاریخ

جالبی دارد و آن به رقابت بین دو ابر قدرت قرن بیستم یعنی اتحاد جماهیر شوروی و ایالات متحدهٔ آمریکا مربوط می شود. همانطور که می دانید اولین ماهواره مصنوعی ساخت دست بشر در سال ۱۹۵۷ به نام اسپوت نیک توسط شوروی به فضا پرتاب گردید ، درست از همین سال علم تبادل اطلاعات کامپیوتری بوجود آمد.

ماهواره ها در ارتباط مخابراتی و جاسوسی و به طور کلی جذب اخبار و ارقام تواناییهای فراوانی دارند. این توانائی و قابلیت کاملاً مورد توجه دانشمندان آمریکایی و بطور کلی نظام آمریکایی بود. نظامی که در این زمینه در آن سالها دارای عقب ماندگی محسوسی از شوروی بود لذا مراکز تحقیقاتی بطور اعم و مراکز نظامی بطور اخص در آمریکا مأمور شدند تا با توجه به قابلیت و توانایی موشکهای شوروی در پرتاب ماهواره ها به فضا در نتیجه حمل بمب اتمی توسط موشک و پرتاب آن به سمت شهرهای آمریکا و همچنین توانایی گسترده مخابراتی شوروی سیستمی را طراحی کنند که اگربه فرض یکی از شهرهای آمریکا توسط بمبهای اتمی نابود شد ، سیستمی موجود باشد و اطلاعات موجود در کامپیوترهای این شهر را قبل از نابودی به شهر دیگرمنتقل کند. دانشمندان و محققان در پنتاگون (وزارت دفاع آمریکا) موفق به طراحی سیستمی شدند که قابلیت انتقال اطلاعات مثلاً ازطبقهٔ دوم پنتاگون اتاق ٤٠٢ را به طبقهٔ چهارم اتاق ٩٤٤ و سایر طبقات و اتاقهای این وزارتخانه، داشت. یعنی دو کاربر یا چند نقطهٔ مختلف این سازمان توانایی تبادل اطلاعات بین یکدیگر و بین یک کامپیوتر مرکزی را داشتند و همچنین می توانستند توسط این سیستم به تبادل نامه بپردازند که این سیستم انتقال نامه به پست الکترونیکی[1] معروف شد. اما یادآوری این نکته ضروری می نماید که در سیستمهای مدرن امروزی و سیستمهایی که در آینده طراحی خواهد شد انتقال نامه به یک موضوع پیش پا افتاده و بسیار ساده تبدیل خواهند شد. در سیستمهای آینده ، انسان خود انتخابگرمی باشد و آنچه که او اراده کند که انجام دهد فقط با یک کامپیوتر و یک مودم و یک خط تلفن در منزل یا محل کار او امکان پذیر می شود.

در آینده می توان فیلمهای سینمایی را در خانه دید به خرید اجناس مورد نیاز روزانه از طریق سرویس Remote Shopping پرداخت کارهای اداره را از طریق کامپیوتر در منزل انجام داد. در آن زمان دیگر رادیو و تلویزیون و سینماها نیستند که برای ما تعیین و تکلیف کنند که چه برنامه ای ببینیم ، بلکه آنچه که خود اراده کنیم که ببینیم ، بر روی صفحهٔ مانیتور کامپیوتر ظاهر می شود دیگر در کتابخانه های عریض و طویل به دنبال این کتاب و آن کتاب با کتابدار چانه نمیزنیم بلکه هر کتاب که مورد علاقهٔ ما باشد تنها با فشار دکمه ای بر روی صفحهٔ مانیتور ظاهر می شود و می توانیم صفحات آن را ورق بزنیم و مطالعه کنیم. عبور ومرور در شهرهای بزرگ کاهش می یابد آلودگی زیست محیطی شهرهای بزرگ کم می شود و انسانها عمر خود را در خیابانهای پر رفت و آمد و پشت راهبندانهای سنگین تلف نمی کنند وقت برای تفریح ، تحقیق مطالعه و کارهای فرهنگی وعلمی و غیره فراهم می شود. بازارهای بورس توسعه می یابد و در نتیجه گسترش و پیشرفت اقتصاد ، جوامع بشری به رفاه می رسند.

آیا این زمان سطح بهداشت در کرهٔ زمین به صورت امروز باقی می ماند؟ یا ریشهٔ همهٔ بیماریها به علت همفکری و تبادل اطلاعات در سرتا سر دنیا و در نتیجه پیشرفت علوم بهداشتی از بین می رود؟

آیاجائی برای سانسور اخبار از طریق دیکتاتور در دنیا باقی می ماند؟ یا دموکراسی به طرز چشمگیری در تمام کشورها فراگیر می شود؟ پاسخ با خواننده است.

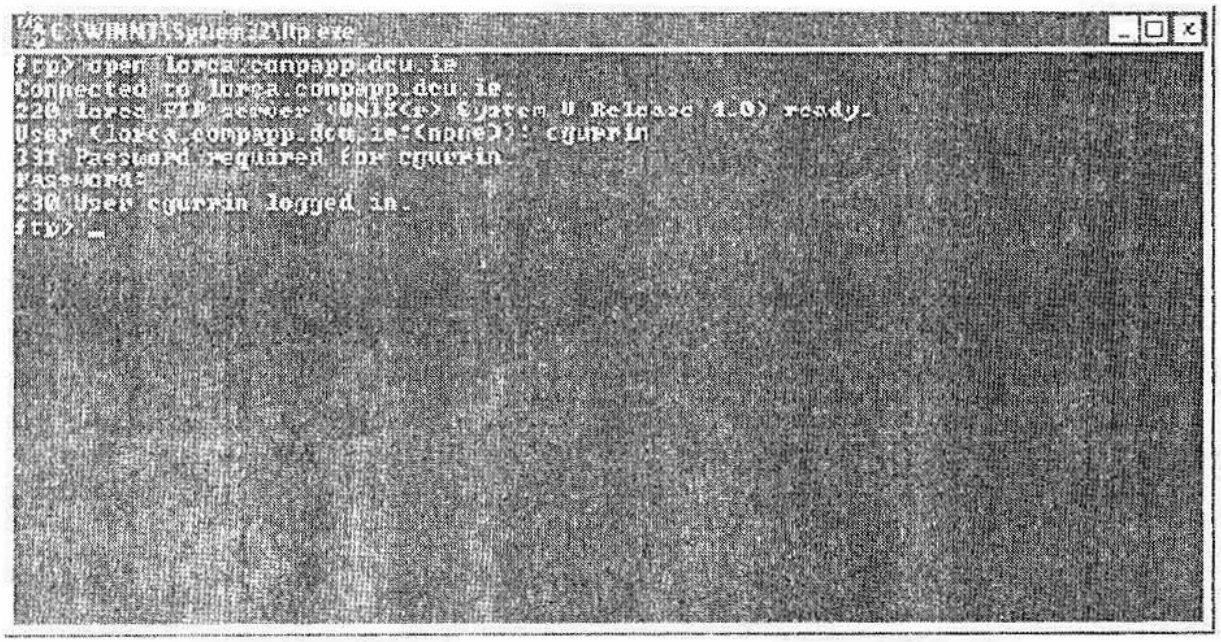
اما برگردیم به تاریخچهٔ خودمان : سیستمی که در قسمت تاریخچه تعریف کردید در ابتدا به نام آرپا[1] مشهور شد مأموریت اصلی و نهایی آرپا تحقیق و اتصال کامپیوترهای دانشگاه و مراکز نظامی از طریق بستر مخابراتی به نحوی بود که چندین کاربر بتواند در یک محیط ارتباطی یا خط ارتباطی با هم شریک شوند. هدف ، ایجاد شبکه هایی بود که در آن اطلاعات که همان داده های کامپیوتری می باشند. بتوانند از نقطه ای به نقطهٔ دیگر بروند و تمام شبکه های محلی در نقاط مختلف به یکدیگر متصل شوند.

البته در ابتدا هدف آرپا ایجاد شبکه ای مانند اینترنت نبود و فقط یک اقدام احتیاطی در مقابل حملهٔ احتمالی موشکهای اتمی دوربرد اتحاد شوروی بود. در اوایل ۱۹۷۳ یعنی زمانی که سیستمهای کامپیوتری بزرگ[2] در بازار بودند و هنوز خبری از کامپیوترهای شخصی نبود آرپا که با افزوده شدن انگلیس (DEFENCE) به آژانس پروژه های پژوهشی پیشرفته دفاعی به DARPA تغییر نام داده بود شروع به کار بر روی پروژهٔ جدیدی نمود که پروژه به هم مرتبط سازی نامیده می شد هدف از این پروژه یافتن راهی برای متصل ساختن شبکه ها به یکدیگر بود البته باید توجه داشت که هر یک از این شبکه ها برای جابجائی اطلاعات خود از روشهای متفاوتی استفاده می کردند.وقتی روش لینک کردن یا مرتبط ساختن کامپیوترهای شخصی پیش می آمد صاحبان شبکه ها می توانستند ار طریق تجهیزات خاصی موسوم به دروازه ها[3] شبکه های خود را به هم وصل کنند که البته ارتباط بین شبکه ها احتیاج به پروتکلهای مناسب داشت.

در سال ۱۹۶۲ پاول باران در مقاله ای تحت عنوان روی شبکه های ارتباطی توضیحاً به تشریح شبکه های PACKET SWITCH پرداخت ، در این روش داده ها به قطعات و بسته های کوچکتری خرد می شوند و هر بسته شبیه یک نامهٔ پستی شامل آدرس فرستنده و گیرنده است و می تواند از هر مسیری به مقصد برسد. در مقصد بسته ها مجدداً یک پارچه می شوند و به فرم کامل تحویل مقصد می شوند.

در ۱۹۶۹ دولت ایالت متحده چهار کامپیوتر را با استفاده از تکنولوژی PACKET SWITCH در ایالتهای کالیفرنیا و یوتا به هم متصل کرد این شبکه خوب کار کرد ، کاربران این کامپیوترها توانستند تقریباً همزمان به دیگر کاربرها پیام بفرستند و فایل به اشتراک بگذارند ، این پروژه همان آرپا نام گرفت اما یک کلمه جدید به انتهای آن اضافه شد و به صورت (ARP ANET) درآمد با گذشت زمان، کامپیوترها و کاربران جدید در سایتهای دولتی و دانشگاهی به آن اضافه شدند در سال ۱۹۷۰ کامپیوترهای میزبان استفاده از پروتکل کنترل شبکه NCP را شروع کردند و یک سال بعد تعداد گره های این پروژه به ۱۵ و تعداد میزبانهای آن به۳۲ عدد رسید در همین سال شخصی به نام تایلون سیستم نامه رسان الکترونیکی را برای یک شبکه توزیع شده ابداع نمود در ۱۹۷۳ کشورهای بریتانیا و

نروژ به ARPANET متصل شدند در سال ۱۹۷٤ دکتر رابرت متکالف نیز نظریه خود را در مورد اینترنت ارائه داد در همین سال سیرف و باب کان جزئیات پروتکل TCP را ارائه دادند کمپانی BBN نیز نسخهٔ تجاری آرپانت به اسم رتل بنت را ارائه کرد. از اواسط دههٔ ۱۹۷۰ تا ۱۹۸۰ شبکه های کوچکی از تکنولوژی آرپانت استفاده می کردند تصمیم گرفتند تا به صورت شبکه ای باهم کار کنند آنها آرپانت را به عنوان هسته انتخاب کردند و شروع به ارتباط از طریق خطوط استیجاری نمودند در سال ۱۹۸٦ سرعت این شبکه ٥٦ کیلو بیت در ثانیه بود سرانجام در سال ۱۹۹۰ آرپانت تغییر نام کرد و نام اینترنت بر روی آن گذاشته شد.در آن زمان از سیستمهای یونیکس در یک محیط خط فرمانی[1] برای استفاده از امکانات اینترنت استفاده می شد. با دستوراتی مثل Telnet و Ftpبرای اتصال و استفاده از سایر امکانات اینترنت استفاده می شد که لازم بود هر بار کاربر شناسه کاربری و رمز عبور[2] وارد نماید.



شکل ۱-۱ اینترنت متنی روی سیستمهای UNIX

۱-۳ تاریخچهٔ وب

ویژگی اصلی اینترنت این است که هر نوع کامپیوتری صرف نظر از مشخصات سخت افزاری و سیستم عامل با رعایت یک مجموعه استاندارد می تواند به کامپیوترهای دیگری که آن استانداردها را رعایت می کند وصل شود.

جنبه های مختلف اینترنت در طول سالیان تکامل بسیاری یافته است اما مسئله اساسی که تا این اواخروجود داشت دشواری استفاده از اینترنت برای افراد غیر حرفه ای بود. راه حلی در اوایل ۱۹۹۳ پیدا شد هنگامی که شیوهٔ موسوم به تارجهان گستر یا WWW استفاده از شبکه را برای هر کسی ممکن ساخت.

در سال ۱۹۸٤ دومتخصص فیزیک در دو نقطه از اروپا GENEVAوCERN روی یک پروژهٔ مشترک فیزیک کار می کردند لازم بود به جدیدترین و به روزترین اطلاعات در رابطه با تحقیقات همدیگر دسترسی داشته باشند،

روشی را ابداع کردند که از این طریق می توانست مستندات خود را روی اینترنت به اشتراک بگذارند این روش را وب نامیدند. این مستندات که بعدا" Page یا صفحه نامیده شد می توانست توسط یک نرم افزار به نام مرورگر[1] نمایش داده شود و در این مستندات کلمات کلیدی وجود داشتند که کاربر با انتخاب آنها می توانست به یک صفحهٔ دیگرروی اینترنت متصل شود بدون اینکه هیچگونه رمز عبور و شناسه کاربری وارد کند.

به سیستمی که روی اینترنت امکان به اشتراک گذاشتن صفحات را می داد WWW [2] تارجهان گستر می گویند.

مزایای وب

۱- بدون وارد کردن رمز عبور و شناسهٔ کاربری کاربران می توانند مستندات خود را به اشتراک بگذارند.

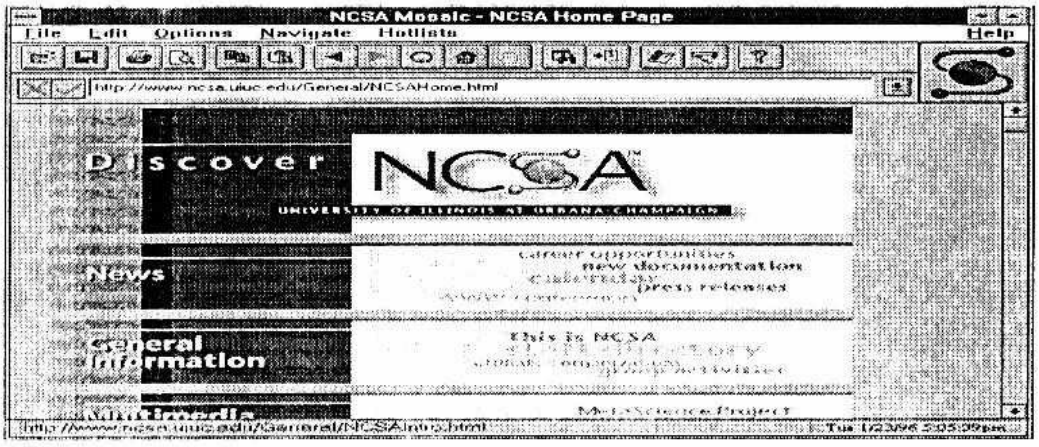
۲- لزومی به دانستن اینکه سند در کجای اینترنت قرار دارد وجود نداشت (یعنی محل قرار گرفتن صفحه[3] روی اینترنت از دید کاربر پنهان است)

۳- امکان استفاده از فرا متن[4] به جای متن های معمولی وجود داشت.

وب گرافیکی

در سال ۱۹۹۳ یکی از دانشجویان دانشگاه illinois به نام Andreessen ایدهٔ رابط کاربر گرافیکی را روی وب مطرح کرد که به نام مرورگرهای گرافیکی وب معرفی شدند. و برای آن یک مرورگر به نام Mosaic نوشت این نرم افزار اولین مرورگر گرافیکی روی اینترنت بود.

Andreessen بعدا" شرکت Netscape Navigator را تاسیس کرد که 80% کاربران اینترنت امروزه از این مرورگر استفاده می کنند. در این مرورگر امکان استفاده از تصاویر ، تصاویر متحرک و متن وجود داشت به این مستندات که می تواند علاوه بر متن و پیوند ، دارای صدا و تصاویر متحرک نیز باشند را فرامتن گویند.



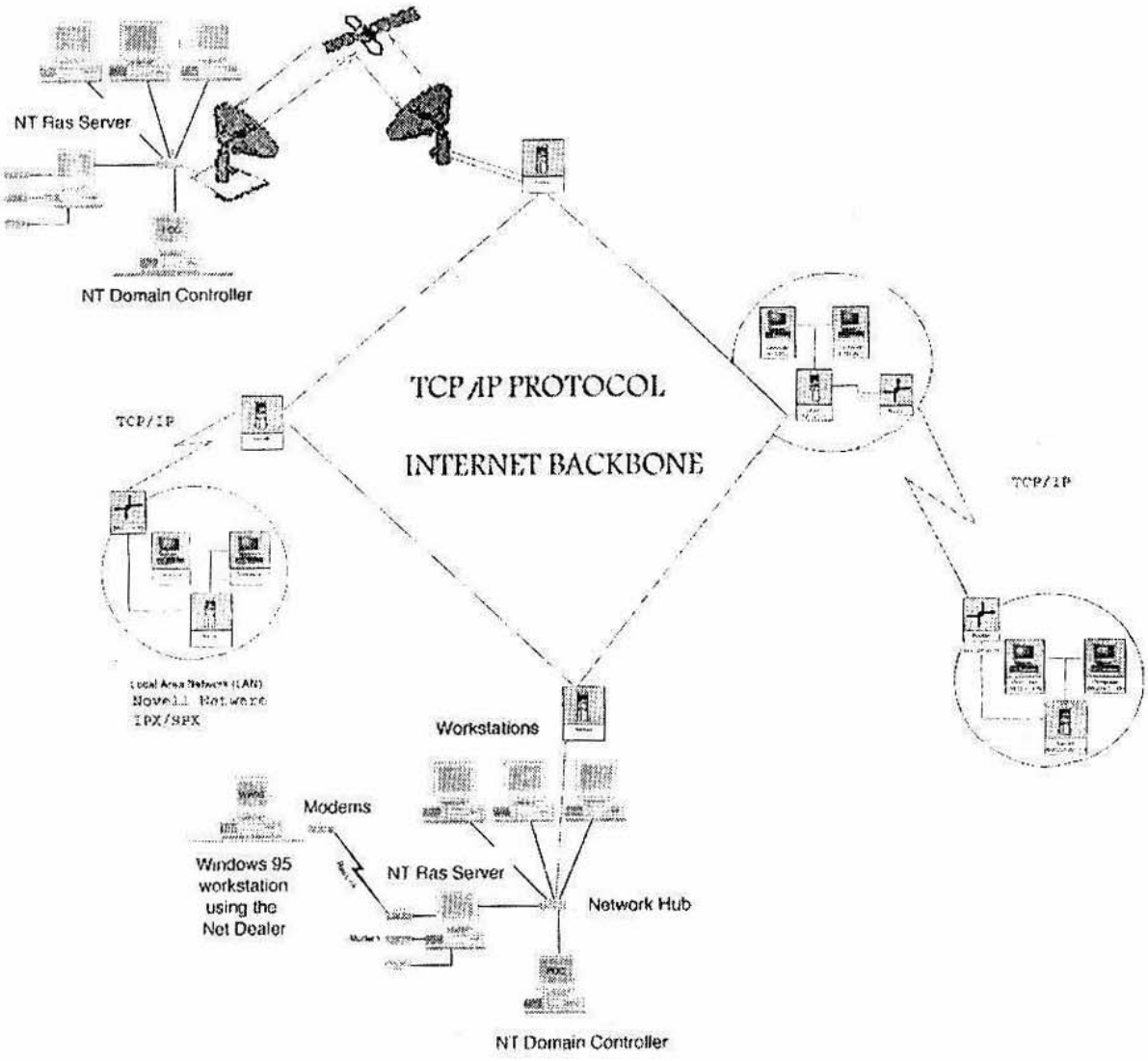
شکل۲-۱ شمایی از Mosaic

تعریف وب

مجموعه ای از پروتکلها که بین سرویس گیرنده و سرویس دهنده وب[1] مطرح می شود تا اینکه مستندات قابل اشتراک روی شبکه اینترنت باشند وب گویند.

۱-۴ ساختار داخلی اینترنت و زیرشبکه های آن

اینترنت شبکه ای در حد WAN متشکل از تعدادی شبکهٔ کوچکتر LAN می باشد که به این شبکه های کوچکتر زیرشبکه های اینترنت یا Subnet گوییم. اینترنت شبکه ای از زیرشبکه ها می باشد.



شکل ۳-۱ ساختار اینترنت شبکه های مختلف

ستون فقرات[1] اینترنت

خطوط پرسرعتی که بیشتر فیبرنوری یا خطوط ماهواره ای با پهنای باند بالا مـی باشـند کـه کامپیوترهـای میزبـان اینترنت که عملیات مدیریت کاربران روی اینترنت را برعهده دارند بـه هـم متصـل مـی کننـد سـتون فقـرات اینـترنت گویند.

روشهای ارتباط ماهواره ای با اینترنت

جهت برقراری ارتباط از طریق ماهواره به اینترنت به دو صورت می توان عمل کرد این دو روش عبارتند از:

۱- متقارن

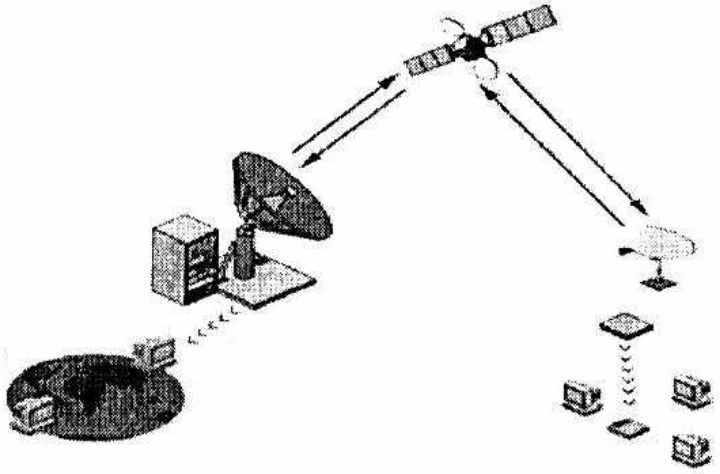
دراین روش هم ارسال اطلاعات وهم دریافت اطلاعات از طریق خط ماهواره ای انجام می گیرد.

۲- نامتقارن

در این روش ارسال که بیشتر اوقات کمتر از دریافت می باشد با خط تلفن[2] انجام می شود و دریافت اطلاعات که مقدار آن بیشتر می باشد از طریق ماهواره انجام خواهد شد.

سرویس دهندهٔ خدمات اینترنتISP[3]

بعضی از زیرشبکه های اینترنت عملیات و خدمات اتصال به اینترنت برای کاربران فراهم می کنند به این شرکتها یا زیرشبکه ها ISP گویند. البته ISP ها خودشان امکانات وب و صفحات وب نیز دارند. و دارای آدرس اینترنتی مـی باشند.



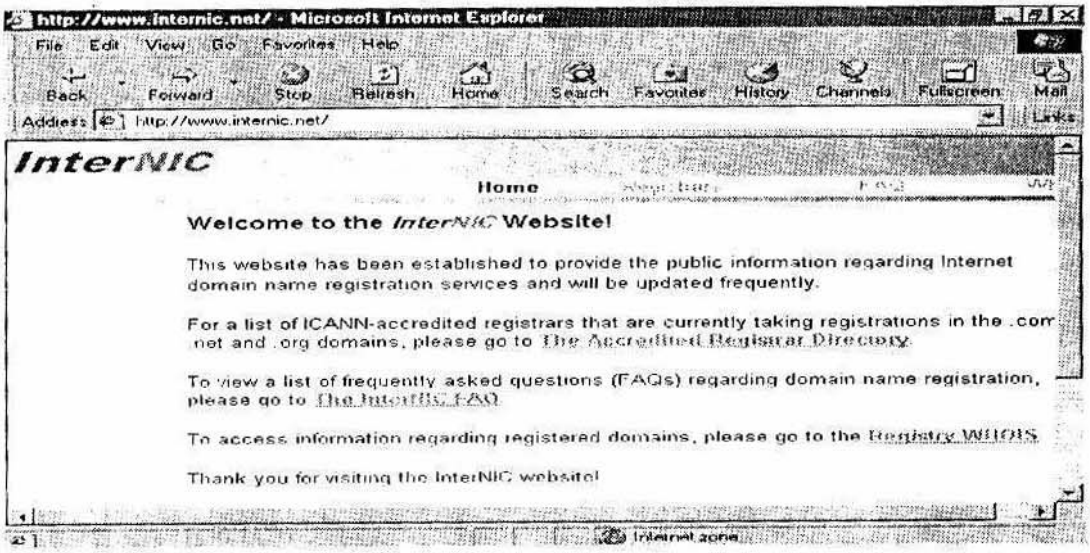
شکل ٤-١ ارتباط متقارن با اینترنت از طریق ماهواره

۱-۵ حوزه[1] و شرکتهای ثبت حوزه

هر ISP روی اینترنت نیاز به یک نام اینترنتی دارد که توسط آن در اینترنت شناخته می شود مثلاً Cnn.Com یا Yahoo.Com و.... به این اسامی حوزه گوییم.

هر سایت ISP جهت ثبت حوزه باید با یکی از شرکتهای ثبت حوزه ارتباط برقرار کند. این شرکتها که درواقع سه تا از Host های اصلی اینترنت را دارا می باشند عبارتند از: Internic ، Apnic ، Rips که برای ارتباط با آنها روی اینترنت می توان آدرس آنها را بصورت زیر وارد کرد.

www. internic . net
www.Apnic . net
www. Rips . net



شکل ۵ -۱ سایت Internic در اینترنت

این سه شرکت دارای سه DNS[2] اصلی اینترنت می باشند که عملیات تبدیل اسامی اینترنتی به آدرس IP توسط آنها انجام می شود.

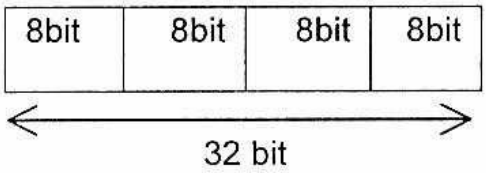
هنگامی که شما یک آدرس اینترنتی مثلاً www.altavista.com را می نویسند ابتدا یک درخواست به DNS ISP شما ارسال می شود. DNS یک بانک اطلاعاتی دارد ، اگر در آنجا معادل آدرس IP یافت شود به کاربر (مرورگر) ارسال می شود اگر نباشد با یکی از شرکتها DNS های اصلی اینترنت تماس اینترنتی برقرار شده و معادل IP آن بدست آمده و بعد بسته روی شبکه ارسال خواهد شد. هزینهٔ ثبت حوزه آن برای دو سال اول ۱۰۰$ و سالهای بعد هر سال ۵۰$ می باشد.

۱-۶ آدرسهای IP

شبکهٔ اینترنت یک شبکهٔ TCP/IP می باشد و در یک شبکهٔ TCP/IP هر کامپیوتر شبکه چه سرویس گیرنده و سرویس دهنده وچه تجهیزات بین شبکه[1] مثل Router دارای یک شناسهٔ منحصر بفرد می باشد که به آن آدرس IP گویند.

آدرس IP یک عدد ۳۲ بیتی می باشد که از ٤ عدد ۸ بیتی تشکیل می شود.

مثال : 194.224.1.2



هر آدرس IP دو بخش دارد :

۱ . Net ID (مشخص کنندهٔ یک زیرشبکهٔ اینترنت)

۲ . Host ID (مشخص کنندهٔ یک کامپیوتر درزیرشبکه)

تعریف سایت وب

یک سایت وب از یک یا چند صفحه تشکیل می شود که به هم پیوند شده اند.

Home page , Welcome page

اولین صفحه یا صفحهٔ آغازین هر سایت وب را Welcome page گویند که با نوشتن نام حوزه ظاهر می شود. مثلاً با نوشتن آدرس www.cnn.com صفحهٔ اول این سایت ظاهر خواهد شد.

Home page مجموعه صفحات مربوط به یک شخص، شرکت یا مؤسسهٔ خاص از یک سایت وب را گویند. مثلاً Home page مربوط به یک دانشگاه یا یک شخص خاص در سایت وب Neda .

اطلاعات موجود در یک Welcome page

Welcome page از آنجا که اولین صفحه می باشد که برای کاربر ظاهر می گردد از این جهت باید بسیار جذاب و گیرا باشد طوری که کاربر را برای دیدن سایر صفحات آن سایت تشویق کند.

بهتر است صفحهٔ اول یک سایت خیلی سنگین نباشد تا سریع بار گردد. اطلاعات موجود در یک Welcome page بهتر است مطالب زیر باشد.

۱- فهرست یا جدولی از عناوین و مطالب موجود در سایت.

۲- دارای آرم و یا شناسهٔ آن مؤسسه باشد (هم می تواند بصورت زمینه کمرنگ زیر متن باشد و هم می تواند دربالای صفحه قرار گیرد).

۳- دارای تعدادی پیوند به صفحات دیگر مربوط به آن صفحهٔ خاص باشد

*بهتر است در هر صفحه یک مسیر بازگشت به Welcome page وجود داشته باشد.

۱-۷زبانهای نشانه ای[1]

زبانهای نشانه ای عوامل اولیه تولید صفحات وب می باشند این زبانها دارای تعدادی دستورالعمل یا برچسب[2] می باشند که به سند اضافه شده و به مرورگر طریقهٔ نمایش سند را از نظر فونت و رنگ و اندازه و سایر موارد نشان می دهند.

معروفترین زبان نشانه ای زبانHTML (Hyper Text Markup Language) است. که زبان نشانه ای فرامتن می باشد. زبانهای VRML (Virtual Reality Modeling Language)وXML (Extensibl Markup Language) نیز امروزه برای صفحات وب استفاده می شوند. زبان HTML بطور مفصل تر شرح داده خواهد شد.

VRML نیز یک زبان مدل سازی برای وب می باشد که برای شبیه سازی محیطها و اجسام می تواند استفاده شود از طریق آن می توان محیطهایی مثل یک اتاق یا یک هواپیما و خانه و... را شبیه سازی کرد که کاربر بتواند در آن محیط حرکت کرده و ارتباطی متقابل[3] با محیط داشته باشد. باید توجه کرد که در این حالت کامپیوتر کاربر باید قدرتمند باشد زیرا به طور مرتب در حال Render کردن تصاویر سه بعدی می باشد تا محیطی کاملاً واقعی را برای کاربر شبیه سازی نماید. فایلهای VRML دارای پسوند WRL می باشند که توسط مرورگرهای معمولی قابل نمایش نیستند برای اینکه یک مرورگر بتواند این فایل را نمایش دهد نیاز به نصب یک برنامهٔ اتصالی[4] خاص می باشد که امکان نمایش این فایلها را فراهم می کند یکی از این Plugin ها Cosmo Player می باشد.

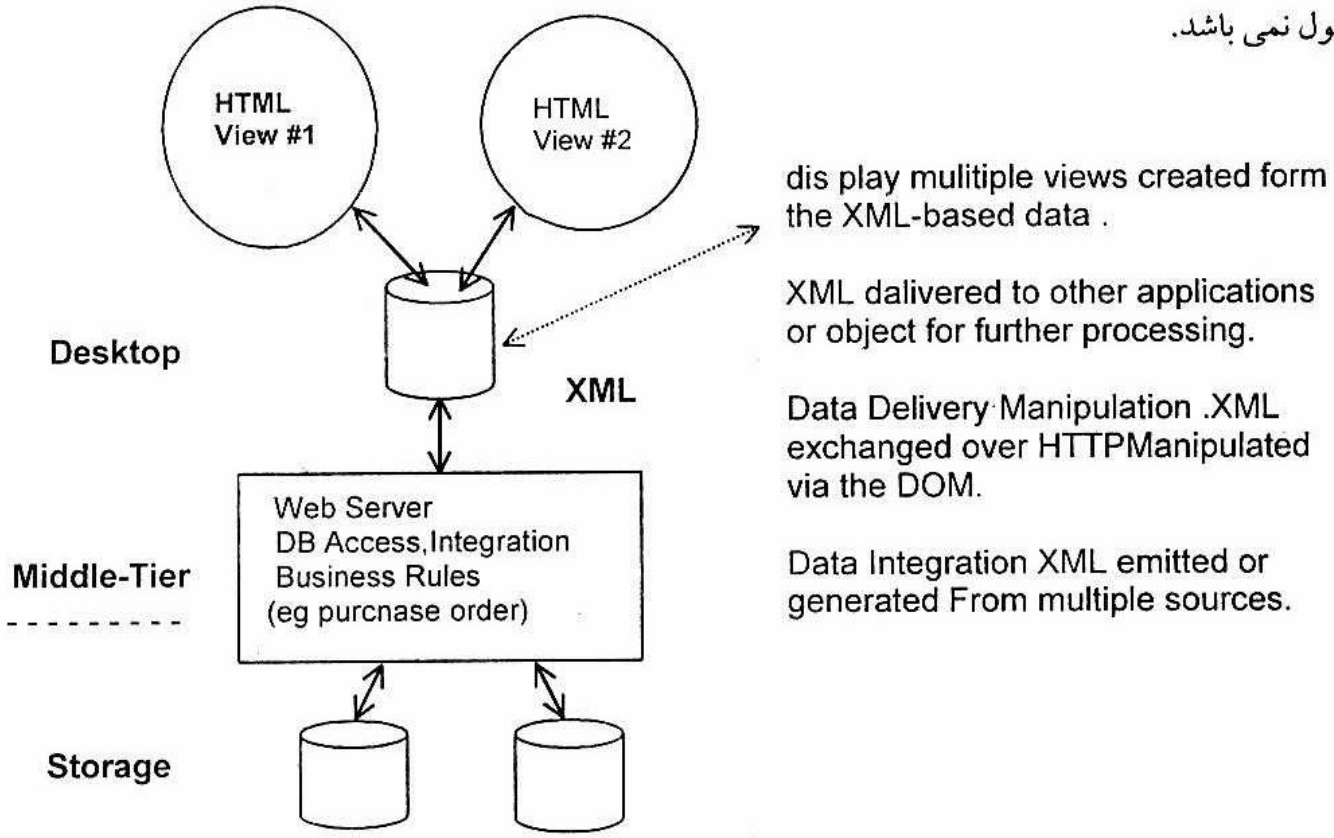
کاربردهای مبتنی بر XML

XML یک زبان مبتنی بر Tag بر اساس (SGML) است که برای نقل انتقالات دنیای وب بهینه سازی شده است. XML ، Layout داده برای یک سند مفروض را تعریف می کند؛ اما ویژگی هایی را که به طور معمول هنگام استفاده از HTML تعریف می شوند را از خود بروز نمی دهد. XML امکان اشتراک داده ای بین کاربردهای مختلف را بر روی سکوهای سخت افزاری مختلف فراهم می آورد. XML نیازی به تبعیت از مشخصه های HTML ندارد.

یکی از مهمترین برتریهای XML در آن است که به نحو بسیار مؤثری محتوای یک صفحه وب را از چگونگی نمایش آن جدا نموده است. به همین دلیل هنگامی که محتوا نیاز به روزرسانی داشته باشد اما نیاز به ذخیرهٔ طولانی مدت نداشته باشد استفاده از مدل ارائه شده XML ایده آل است.

ایده اصلی که XML بر آن بنا شده است چندان جدید نیست: داده باید به فرم استاندارد مبادله شود. چنین فرآیندی در کاتالوگهای کالاها، قراردادها و تقاضاهای خرید در عمل به وفور رخ می دهد. درک می نمایند. مزیت

دیگر XML در آن است که شرکتها و افراد مستقل با سادگی بیشتری می توانند روی یک فرم معمول اسنادی به توافق برسند، در حالی که حصول چنین توافقی بر روی ساختارهای داده ای پیشین (Back-end) به سادگی قابل حصول نمی باشد.



شکل ٦-١ معماری XML

به وسیلهٔ XML یک توسعه دهنده می تواند داده را از طریق یک ساختار داده ای منطقی[1] که مستقل از تمام پیاده سازیهای بخش پیشین[2] می باشد مورد استفاده قرار دهد یا آنها را خلق نموده و تغییر دهد. منابع داده ای چندگانه[3] می توانند داده را به یک ساختار XML واحد تغذیه نمایند. چنین فرایندی امکان تجمع سیستمهای متفرق را تحت یک سقف فراهم می آورد. از سوی دیگر از آنجا که XML در قالب متنی ارائه می شود تجمع سیستمهای متفرق می تواند بر روی وب از طریق HTTP منتقل گردد.

به عبارتی شاید مهمترین دلیل پدید آمدن XML اعمال نوعی ساختارگرایی شیء گرا به صفحات HTML بوده باشد.

مثال زیر چگونگی اختلاف دیدگاهها در مواجهه با یک سند HTML و XML را نشان می دهد:

مثال ١ :

```
<HEAD>
<TITLE> Printer</TITLE>
</HEAD>
```

```
<BODY>
<H1> Computer source is wizbang 3000 dot matrix printer</H1>
Features:
40 Pages per Minute
$200.00
10 1bs.
</BODY>
```

حال اگر این برنامه را بخواهیم بصورت XML بنویسیم داریم:

```
<?XML Version="1.0"?>
<MANUFACTURER>Computer source
    <PRODUCT>
        <CLASS>Printer
        <TYPE> dot Matrix</TYPE>
        </CLASS>
    <NAME>Wizbang3000</NAME>
        <EEATURES>
        <SPEED Units="ppm">40</SPEED>
        <QUALITY Units="dpi">60</QUALITY>
        </EEATURES>
        <PRICE Units="USD">
        <RETAIL>200</RETAIL>
        <WHOLESALE>110</ WHOLESALE>
         <PRICE>
    <WEIGHT  Units="1bs">10</WEIGHT>
    </PRODUCT>
</MANUFACTURER>
```

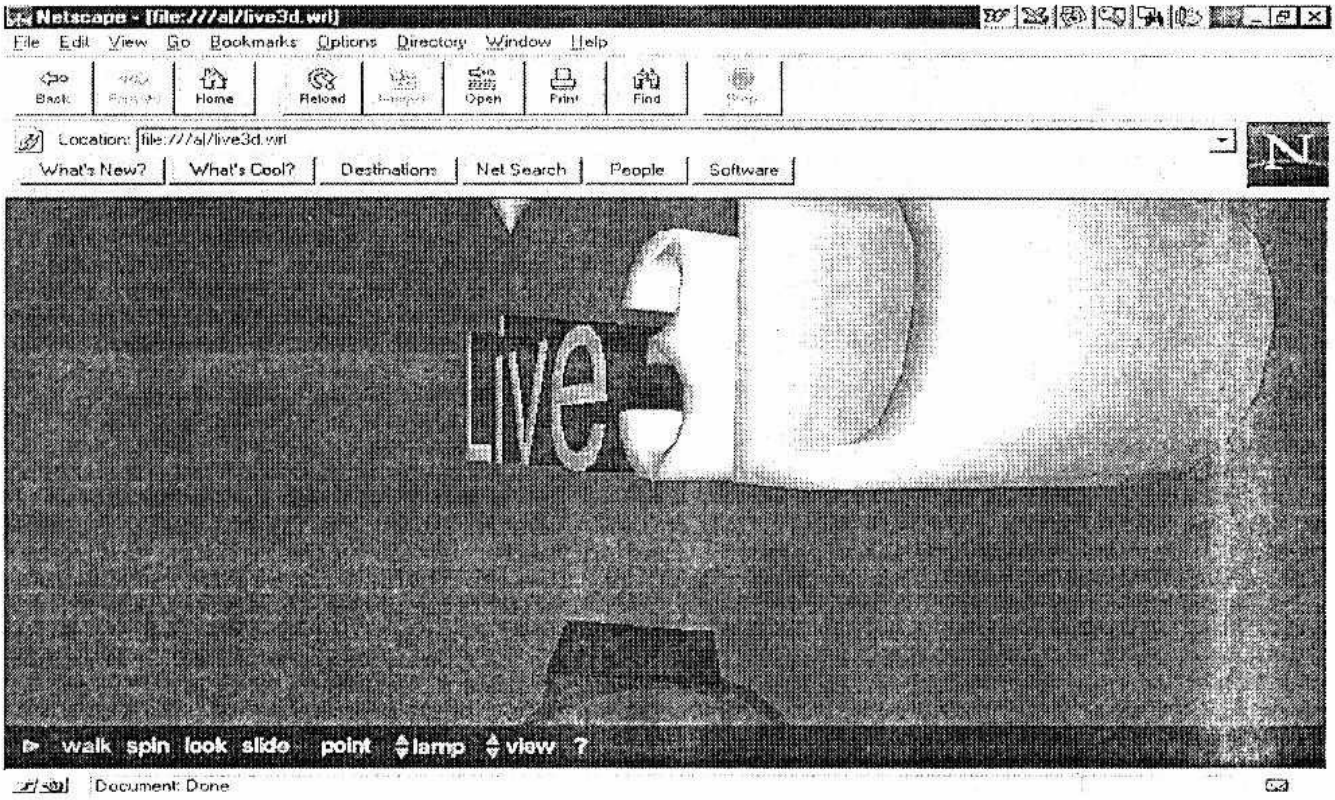
در مثال اول در ارتباط با محصولی مانند چاپگر توضیحاتی در زبان HTML ارائه شده است تشریح همان محصول از XML نیز در مثال دوم آمده است.

همانگونه که در این مثالها آشکار است XML بر تشریح محتوای سند استوار است در حالی که HTML ساختار سند و Layout را مدنظر قرار می دهد، در عین حال هر دو زبان از روشی مبتنی بر Tag ها برای تشریح نامها و ویژگیها استفاده می نمایند.

مثال فوق به خوبی ماهیت ساختارگرایی شیءگرای XML را آشکار می سازد. در این مثال <Product> فرزند <Manufacturer> است. چنین تقسیم بندی از نظر ساختاری کاملاً عقلانی است به جهت آنکه اگر کارخانه ای دارای محصولات متعدد بود می توان حجم کدهای نوشته شده را به نحو در خور توجه ای کاهش داده و علاوه بر آن خوانایی برنامه را افزایش داد.

از دیگر ویژگیهای منحصر به فرد XML آن است که می توان دیدگاه های داده ای (Data Views) متفاوتی را بسته به نیاز مشتریان تعریف نمود و آنگاه برای هر یک از مشتریان کاتالوگ های متفاوتی ارائه نمود.

XML برتریهای فراوانی (نسبت به HTML) دارد به گونه ای که در آینده نزدیک تا حدود بسیاری جای HTML را خواهد گرفت. اما نمی توان از نظر دور داشت که در حال حاضر XML توسط مرورگرهای بسیار کمی حمایت می شود. (از جمله Internet Explorer5) لذا XML در واقع بیشتر طرحی برای آینده است.



شکل ۷-۱ نمایش یک فایل VRML

۸-۱ URL (Uniform Resource Locator)

همانطور که گفته شد هر کامپیوتر سرویس دهندهٔ وب دارای یک نام اینترنتی منحصر بفرد روی اینترنت می تواند باشد که به آن حوزه گوییم. هر صفحه روی اینترنت نیز دارای یک نام اینترنتی منحصر بفرد می باشد که به آن URL گوییم. به عنوان مثال

http://www. cnn.com /sample / main.htm

باید توجه کرد که آدرس www.cnn.com به یک آدرس زیرشاخهٔ خاص روی سرویس دهنده اشاره دارد که به آن زیرشاخهٔ ریشه گویند در سیستمهای مختلف زیرشاخهٔ ریشه متفاوت می باشد در سیستم Windows NT این زیرشاخه بصورت پیش فرض عبارت است از:

C:\Inetpub\www Root\

اجزای تشکیل دهندهٔ URL

۱- پروتکل مورد استفاده جهت انتقال سند می تواند (http ://, ftp ://, gopher://, ...) باشد.

۲- سرویس دهندهٔ وب یا حوزه مثلاً www.yahoo.com

۳- فهرست روی سرویس دهندهٔ وب

٤- نام صفحهٔ html .

http : // www. sample.com / samples / sampele.html

↓ ↓ ↓ ↓

transfer protocol type Specific host computer Directory on hast Web page filename

باید توجه کرد که اگر نام فایل ذکر نگردد سرویس دهندهٔ وب بصورت پیش فرض به دنبال فایلهای index.htm و یا Default.htm بسته به سیستم عامل خواهد گشت و اگر پیدا کند، آنها را ارسال می نماید.

پسوندهای[1] حوزه

این پسوندها هم می تواند مشخص کنندهٔ کشور مربوطه و یا نوع سازمان مربوطه باشند مثلاً :

پسوند	معنی
Com	Commercial
Edu	Education
Gov	Government
Org	Organization
Net	Network provider
AC	Academic
Ca	Canada
Au	Australia
Ir	Iran
.	.
.	.

باید توجه کرد که ممکن است این پسوندها با هم ترکیب نیز گردند مثلاً www.iau.edu.ir

۱-۹ پروتکلهای اینترنت

اگر شما قبلاً از شبکهٔ وب استفاده می کردید خواه ناخواه از پروتکل TCP/IP استفاده می نمودید. بنابراین پروتکل TCP/IP اصلی ترین پروتکل شبکه اینترنت است. TCP/IP بر روی LAN هایی که بر اساس توپولوژی اترنت و Tokenring بسته شده اند کار می کند و همچنین قابلیت ایجاد سرویس برای مشترکین Dial-Up را نیز دارد. اما قبل از اینکه به ادامهٔ بحث بپردازیم لازم است کمی در مورد چگونگی کار کرد پروتکل TCP/IP توضیح داده شود پروتکل TCP/IP دارای چهار لایه می باشد، که وظایف هر لایه مشخص است. سیستم حوزه بندی در اینترنت به کاربران امکان دسترسی ساده تر به اینترنت را می دهد گر چه آدرسهای اینترنت برای یادآوری آسان نیستند اما استفاده از آنها بسیار ساده تر از روش کامپیوتر ها برای اشاره به همان محل است، کامپیوترها در اینترنت از یک سیستم مبتنی بر اعداد برای آدرس و حتی دسترسی به گره ها استفاده می کنند. سیستم آدرسهای اسمی که در

بالا مورد بحث قرار گرفت به آدرس عددی تبدیل می شوند و سپس دسترسی صورت می گیرد این کار اندکی شبیه ذخیره کردن شماره تلفنها در یک حافظه تلفن با استفاده از اسم صاحبان تلفن است. در این حالت برای گرفتن شماره تلفن یک شخص خاص کافیست اسم وی را یافته و دکمهٔ مربوطه را فشار دهید و نیازی نیست که شمارهٔ وی را بدانید این همان راهی است که پروتکل اینترنت طبق آن عمل می کند.

سیستمهای کامپیوتری در اینترنت یک جدول تبدیل اسم به عدد دارند و از آن برای فرستادن اطلاعات به مقصد مناسب استفاده می کنند. در اینترنت اجزاء متفاوتی وجود دارند شبکه های شرکتی ، آموزشی ، شبکه های منطقه ای و ایستگاههای کار اختصاصی ، تمام این اجزاء از طریق ستون فقرات با خطوط تلفنی یا اختصاصی به یکدیگر متصل شده اند. می بینید که بدون یک سیستم آگاه از اسامی و آدرسها غیر ممکن خواهد بود که اطلاعات را به محل دلخواه فرستاد. مدیریت این کار توسط مسیر یابهای کامپیوتری که در اینترنت مشغول به کار هستند و ضمن نظارت بر نقل و انتقال به مطالعهٔ آدرسهای بسته ها می پردازند انجام می شود. پروتکل اینترنت داده ها را با گروه بندی آنها بصورت یک بسته از نقطه ای به نقطه ای دیگر می فرستد. یک بسته عبارت است از یک مجموعه از کاراکترها که کمتر از 1500 بیت است که همهٔ آنها به یک مقصد می روند به عبارت دیگر همهٔ آنها دارای آدرس اینترنت مشابه هستند. بسته ها به روش نردبانی از یک سری کامپیوتر می گذرند تا در نهایت به مقصد برسند. در روش آدرس دهی عددی ، هر آدرس شامل حداکثر چهار عدد است که با نقطه از یکدیگرجدا شده اند و همهٔ آنها از 256 کمترند. مثلاً خدمتکار Archie (Archie.Unl.Edu) دارای آدرس IP به صورت 129.93.1.14 است. آنرا اینطور می خوانند 129 نقطه 93 نقطه 1 نقطه 14 . برخی از اعداد در این آدرس مشخص می کند که کدام شبکه در اینترنت باید بسته را دریافت کند. وقتی این شبکه بسته را گرفت یک مسیریاب دیگربسته را تا مقصد بعدی می فرستد و الی آخر تا کامپیوتر مشخص در انتها بسته را بگیرد. در آنجا بسته شکسته شد یا در یک فایل ذخیره می شود یا روی مانیتور نمایش داده می شود.

OSI Layers	TCP/IP Layels	TCP/IP Protocols & Components	
Application	Application	Application Protocols	
Presentation			
Session			
Transport	Transport	TCP	UDP
Network	Internet	IP	ARP
		ICMP	RARP
Data Link	Network Interface	Your Network Sofware & Network Card	
Physical	Hardware		

جدول ۱-۱ لایه های مختلف مدل TCP/IP در مقایسه با OSI

این سیستم مبتنی بر اعداد برای صحبت کردن کامپیوتر با یکدیگر بدون اشتباه کردن آدرس و بدون سر در گم شدن در سیستمهای شبکه ضروری است. اما برای ما سیستم اسمی راحتر است و برای دسترسی ها به منابع اینترنت کافی است. با این حال یادگرفتن مبانی پروتکل اینترنت برای فهمیدن اینکه اطلاعات چطور این سر و اینترنت سر می روند خوب است. دلیل دیگری برای یاد گرفتن نکاتی در مورد پروتکل اینترنت این است که ماشین تبدیل نام به آدرس در اینترنت BIND بعضی اوقات خراب می شود (Berkeley Internet Name Domain) در این حالت است که یا مجبورید منتظر باشید تا خدمتکار اسم دوباره فعال شود و یا اینکه از سیستم عددی آدرس در کامپیوتر خود استفاده کنید.

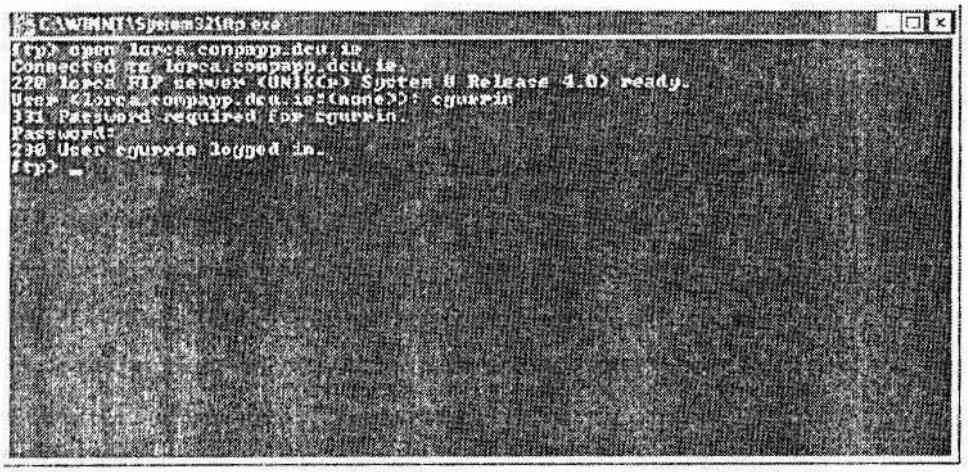
پروتکل اصلی اینترنت TCP/IP است ولی در هر کدام از لایه های آن پروتکلهای مختلفی مطرح می شوند اصلی ترین پروتکلها در لایه کاربرد عبارتند از FTP،Gopher،Http .

پروتکل انتقال فایل[1] FTP

این پروتکل جهت انتقال فایل بکار می رود. از همان ابتدای ایجاد اینترنت در آن وجود داشت و بعضی سایتها این خدمات را ارائه می دادند. دستوری در آنها وجود داشت به نام ftp که همانند شکل می توانستیم از طریق آن به سایتهای ftp متصل شده و خدمات ftp بگیریم.

وظیفهٔ اصلی و خصوصیت اصلی FTP این است که از طریق آن می توان به سایتهایی که دارای شناسهٔ کاربری می باشند، از طریق شناسهٔ کاربری anonymous و رمز عبور آدرس پست الکترونیکی ، متصل شد و از امکانات آنها استفاده کرد. در مرورگرهای گرافیکی نیز می توانید در بخش آدرس از آدرسهای ftp مثل زیر استفاده کرد.

ftp :// ftp.vax.orgیاftp ://ftp.dec.com



شکل ۸-۱ صفحهٔ مربوط به دسترسی به یک سایت FTP

در این صورت خود مرورگر بصورت یک میانجی عمل کرده و بصورت anonymous با کمترین دسترسی ارتباط شما را با سایت برقرار می کند User ID سؤال نمی کند و فهرست ریشهٔ سایت ftp را نشان می دهد. در این حالت به

سادگی می توانید فایلها را برای خود کپی کرده و یـا فایلـهایی را روی سـایت (در صـورت اجـازه) کپـی (Upload) کنید.

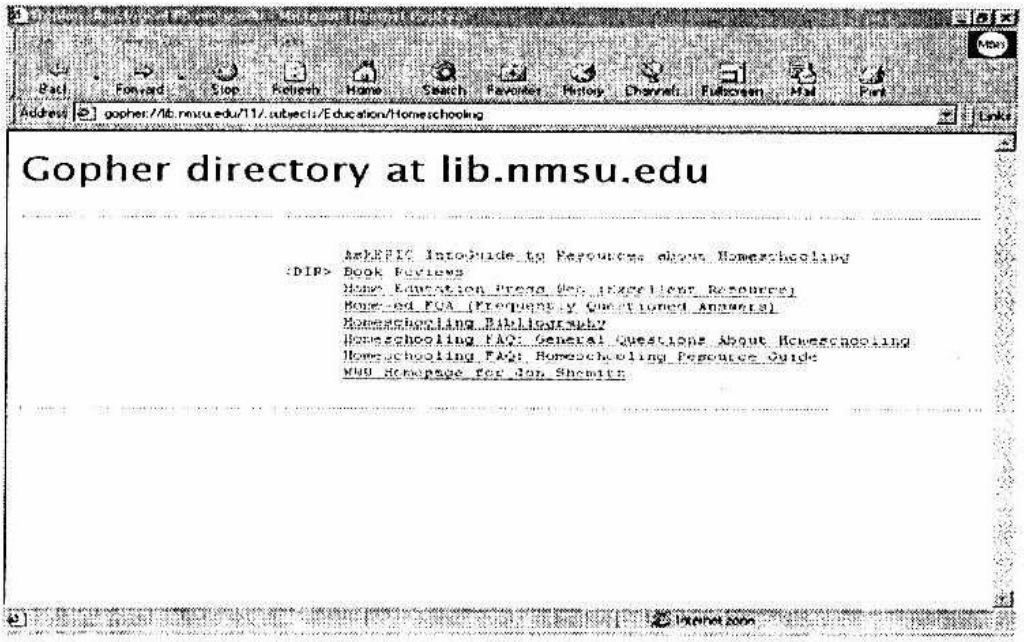
گوفر (Gopher)

گوفر به معنی سنجاب می باشد این پروتکل نیز همانند ftp یک پروتکل بسیار قدیمی روی اینترنت می باشد کـه از همان ابتدا روی اینترنت متنی (Unix) وجود داشت و مانند ftp دارای شناسه کاربری می باشد.

ازطریق سایتهای گوفر امکان جستجو روی اینـترنت (فقـط در همـان سـایت) وجـود داشـت. طریقـهٔ جسـتجو در سایت گوفر بدین صورت است که در سایت تعدادی عناوین اصلی وجود دارد کـه بسـته بـه موضـوع مـورد جسـتجو یکی را انتخاب کرده و بصورت سلسله مراتبی آنقدر پیش می رویم تا به مستندات مورد نظر خود برسـیم مثـلاً بـرای یافتن مقالات در مورد مدیریت حافظه باید بصورت زیر به موضوع مورد نظر دسترسی پیدا کنیم.

مثلاً:

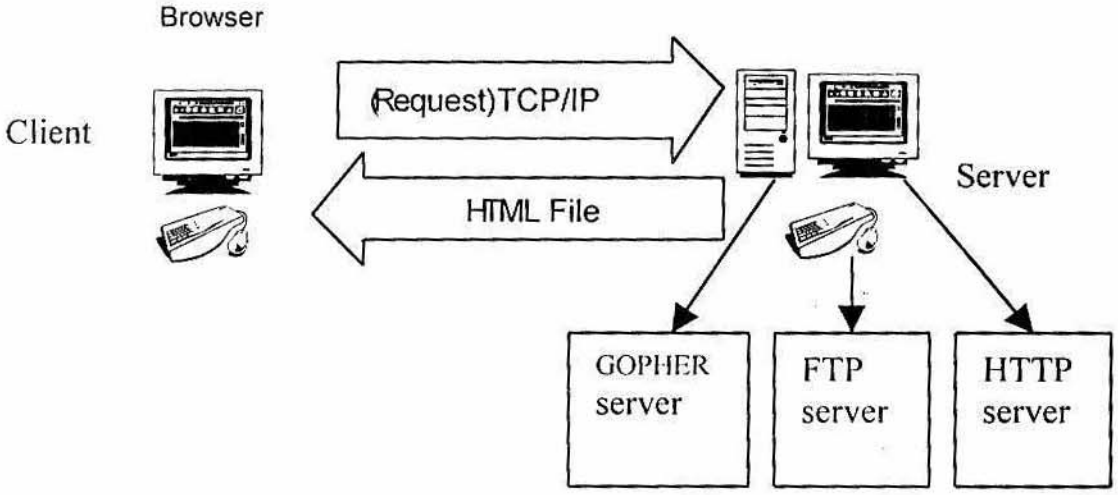
Engineering
↓
Computer
↓
Software
↓
Operating system
↓
Memory management



شکل۹-۱ سایت Gopher

پروتکل فرامتن[1] HTTP

HTTP یک پروتکل انتقال فرامتن می باشد پروتکلی است که اساس وب را شامل می شود. HTTP یک پروتکل غیر وابسته به حالت[2] است. به این معنی که هیچگونه اطلاعاتی از چگونگی انتقال و مسیریابی و ... روی سیستم سرویس گیرنده نگهداری نمی کند.



شکل ۱۰-۱ عملکرد پروتکل TCP/IP

طریقهٔ عملکرد پروتکل HTTP

در بخش سرویس گیرنده روی مرورگر وقتی یک درخواست با پروتکل HTTP ایجاد می کنید مثلاً روی یک پیوند ، کلید موس را می زنید یا آدرسی بصورت زیر روی مرورگر می نویسید.

http :// www.altavista.com/indext.htm

شما درخواستی به سرویس دهندهٔ HTTP سایت وبی با آدرس www.altavista.com ارسال کرده اید.

عملیاتی که سرویس دهندهٔ HTTP باید انجام دهد بصورت زیر است.

۱- فرض می کنیم بسته ای با آدرس بالا به دست سرویس دهندهٔ HTTP رسیده باشد در این صورت سرویس دهنده به دنبال فایل موجود در URL می گردد.

۲- اگر فایل یافت نشود یک سند خطا ، سرویس دهندهٔ HTTP ساخته و ارسال می کند.

۳- اگر فایل یافت شود ابتدا سرویس دهندهٔ HTTP نوع آنرا تشخیص داده (از روی پسوند آن) و یک سرآیند با نام (Content-Type) ساخته و به مرورگر ارسال می کند.

٤- اگر سرویس گیرنده بعد از دریافت فایل قادر به نمایش آن نباشد آنرا روی رسانه ذخیره و یا از یک برنامهٔ کمکی دیگر استفاده می کند.

توجه :

اگر در قسمت URL کاربر نام فایل HTML را ذکر نکند صفحه پیش فرض سرویس دهندهٔ HTTP که معمولاً default.htm یا index.htm می باشد در زیر شاخه زیر نمایش داده خواهد شد.

C : \Inetpub \ wwwroot\

مراحل و نحوهٔ کار کردن پروتکلهای اینترنت بطور کلی

هر سرویس دهندهٔ وب دارای تعدادی درگاه می باشد که آنها واسط بین لایهٔ TCP/IP و برنامهٔ سرویس دهندهٔ آن پروتکل مثل سرویس دهنده Http ، Ftp یا Gopher می باشد. باید توجه کرد که همین درگاه راه نفوذ برنامه های مخرب مثل Hacker ها یا Craker ها به سیستم می باشند

مراحلی که هنگام نوشتن یک آدرس اینترنتی یا زدن کلید موس روی یک پیوند ایجاد می شود بصورت زیر است.

۱- اگر روی پیوند ، کلیک شود دراین صورت اطلاعات مربوط به آدرس صفحه و حوزه در یک بسته قرار گرفته و به برنامه مدیریت پروتکل TCP / IP که روی سیستم نصب است ارسال می گردد.

۲- در لایهٔ TCP برنامه مدیریت پروتکل TCP / IP یک فیلد نوع پروتکل به بسته اضافه می کند که نام پروتکل در آنجا نوشته شده است. مثلاً http , ftp ...

۳- در لایهٔ IP بسته بصورت استاندارد با ساختار TCP / IP در آمده و روی خط فیزیکی انتقال می یابد.

٤- اگر خط شما IP باشد بسته از DNS ها و مسیر یاب ها عبور کرده تا به مقصد برود و اگر خط IP نباشد حتماً از یک Gateway عبور کرده تا به پروتکل مربوطه تبدیل شود.

٥- در مقصد در لایهٔ TCP سرویس دهنده پروتکل بیرون کشیده شده و بسته براساس نوع پروتکل درخواستی به یکی از گره های سرویس دهندهٔ وب ارسال می شود. برای هر گره روی سرویس دهندهٔ وب یک برنامهٔ مقیم در حافظه وجود دارد که سرویس دهندهٔ آن پروتکل می باشد مثلاً سرویس دهنده HTTP (که قبلاً عملیات آن گفته شد.) این برنامهٔ مقیم در حافظه پاسخ را دوباره به لایهٔ TCP برمی گرداند.

٦- پاسخ که ممکن است یک سند HTML یا هر برنامهٔ دیگری باشد به لایهٔ TCP برگردانده می شود. لایهٔ TCP روی سرویس دهنده وظیفه بسته بندی[1] کردن اطلاعات و چسباندن سرآیند پروتکل برای هر سند را برعهده دارد.

۷- در لایهٔ IP هر بسته به قالب استاندارد TCP / IP در آمده و روی خط فیزیکی ارسال می گردد.

۸- در سرویس گیرنده عمل جمع آوری اطلاعات در لایهٔ TCP انجام شده و یک صفحهٔ کامل به مرورگر جهت نمایش ارسال خواهد شد.

۹- اگر یک تصویر یا صفحهٔ دیگری (Frame ها) همراه صفحهٔ ما باشد دوباره یک درخواست جدید ساخته شده و همهٔ این مراحل تکرار خواهند شد (State less).

۱-۱۰ امکانات مورد نیاز جهت راه اندازی یک ISP

جهت راه اندازی یک ISP نیاز به یک سری امکانات نرم افزاری و سخت افزاری بصورت زیر است.

۱- راه اندازی یک شبکهٔ TCP / IP از طریق سیستم عامل WinNT یا Unix

۲- فراهم کردن تجهیزاتی برای کاربران جهت ارتباط راه دور.

۱- استفاده از Multiport

۲- استفاده از RAC (Remote Access Controller)

الف) Multiport

Multiport یک کارت خاص می باشد که روی یکی از Slot های سرویس دهنده نصب شده و بسته به نوع Multiport امکان ایجاد صدا کند تا 128 درگاه را فراهم می کند که از طریق یک کابل به تعدادی Box متصل می گردد. به هر درگاه یک مودم خارجی متصل خواهد شد.

ب) استفاده از RAC (Remote Access Controller)

RAC یک سیستم مجزا است که خود به تنهایی دارای خیلی از امکانات یک سیستم کامل می باشد که امکان برقراری ارتباط را فراهم می کند. جهت ارتباط تنها کافی است کاربران روی RAC تعریف شده باشند. خود RAC هنگام برقراری ارتباط تلفنی با آن ارتباط را برقرار کرده و از کاربر شناسهٔ کاربری و رمز عبور سؤال می کند این شناسه از طریق یک نرم افزار مخصوص توسط مدیر سیستم روی خود RAC تعریف شده است. باید توجه کرد که بعضی از این مسیر یابها نیز امکان ارتباط راه دور را فراهم می آورند.

توجه :

امنیت در RAC بسیار بیشتر از Multiport است. امکان از کار افتادن نرم افزار کنترل کنندهٔ Multiport (RAS)و قطع شدن ارتباط همهٔ کاربران در آن وجود دارد.

۳- گرفتن خط اینترنت

در این حالت می توان از طریق خطوط اجاره ای یا ماهواره ای سایت خود را از طریق یک مسیر یاب به اینترنت متصل کرد.

٤- گرفتن کلاس IP

به مجموعه IP منحصر بفردی که سایت شما می تواند استفاده کندِ کلاس IP آن سایت گویند.

مثلاً" IP های بین 194.224.1.21تا 194.224.1.0

این مجموعه IP یک کلاس C می باشد (در مورد کلاسها بعداً" توضیح داده خواهد شد) این مجموعه IP روی اینترنت معتبر[1] و منحصر بفرد[2] می باشد. باید توجه کرد که به ازای هر IP باید هزینه پرداخت شود. روشهایی وجود دارند که از طریق این روشها می توان IP های مجازی روی سایت استفاده کرد و یکی از این روشها استفاده از نرم

افزار Proxy است. این نرم افزار سه وظیفهٔ اصلی دارد یکی ایجاد IPهای مجازی دیگری ایجاد یک سطح امنیت و Caching.

امکانات نرم افزاری

بعد از فراهم کردن امکانات سخت افزاری نیاز به سرویسهایی می باشند که کاربران بتوانند به سادگی از اینترنت استفاده کنند.

۱- نصب حوزه

۲- بسته به نیازنصب سرویسهای DNS و DHCP و RAS و .....

یک زیرشبکهٔ TCP / IP که طریقهٔ ارتباط و تمامی پروتکلهای آن همانند اینترنت باشد ولی بـه اینـترنت متصـل نباشد را اینترانت گویند و از اتصال چندین اینترانت یک شبکهٔ بزرگتر ایجاد می شود که به آن Extranet گویند.

حال هر کدام از آنها که به اینترنت وصل شوند دیگر زیرشبکه ای از اینترنت خواهند شد.

۱-۱۱ کلاسهای IP

به مجموعه IP هایی که به هر ISP یا زیرشبکهٔ اینترنت اختصاص می یابد کلاس IP آن زیر شبکه گویند.

مثلاً مجموعه 194.224.1.32 تا 194.224.1.63که مشخص کنندهٔ ۳۲ تا IP از کلاس Cمی باشند.

اصلی ترین کلاسهای IP عبارتند از:

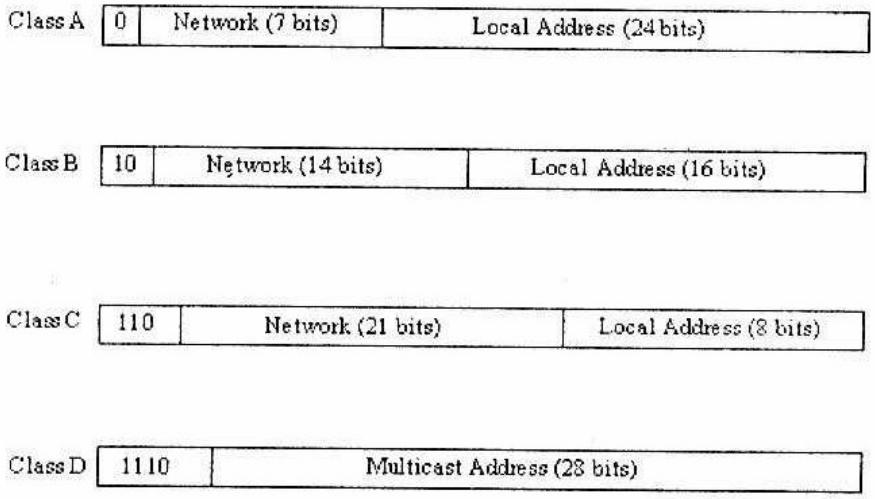
۱- کلاس A

۲- کلاس B

۳- کلاس C

البته کلاسهای D,E نیز وجود دارند که کلاسهای اصلی نیستند.

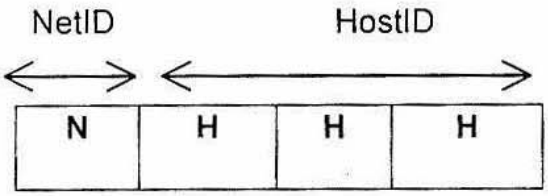
* محدودهٔ عدد اول در آدرس IP مشخص کنندهٔ نوع کلاس است.



شکل ۱۱-۱ کلاسهای IP

کلاس A

بزرگترین کلاس از نظر تعداد سرویس گیرنده ها در اینترنت می باشد قالب آدرسها در کلاس A بصورت زیر است. عدد اول در آدرس IP،NetID بوده و سه عدد آخر HostID است. با ارزشترین بیت در این کلاس (MSB) 0 است.



مثال:

34.23.2.76

تعداد زیرشبکه کلاس A

تعداد ۱۲۶ زیرشبکه کلاس A در اینترنت وجود دارد.

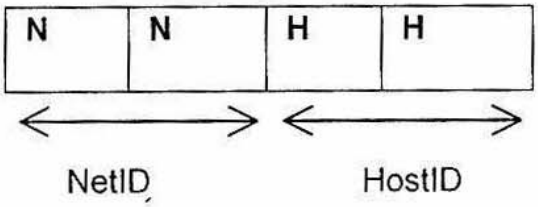
تعداد Host ها در کلاس A

به جهت غیر قابل استفاده بودن بعضی از IP های خاص برای عملیات Loopback , Multicast , Broadcast و تشخیص خطا تعداد Host کلاس A عبارت خواهند بود از:

$\text{تعداد Host} = 2^{24} - 2$

کلاس B

قالب آدرسها در کلاس B بصورت زیر است. در این کلاس دو بیت با ارزش (10) می باشد. عدد اول همواره بین 128 تا 191 می باشد.



مثال :

140 . 2 . 0 . 3

NetID HostID

توجه: باید توجه کرد که در کل شبکه NetID ثابت است بنابراین همهٔ کامپیوترهـای یـک زیرشـبکه دارای Net ID ثابتی هستند و HostID تنها می تواند متفاوت باشد

Class	IP Address	Default Subnet Mask
A	001.y.z.w to 126.y.z.w	255.0.0.0
B	128.y.z.w to 191.y.z.w	255.255.0.0
C	192.y.z.w to 223.y.z.w	255.255.255.0

جدول ۲-۱ کلاسها در شبکه TCP/IP

تعداد زیرشبکه های کلاس B

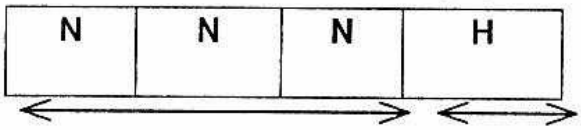
$(192-128)\ 2^{8}=2^{14}$

تعداد Host در کلاس B

در این کلاس تعداد Host ها $2-2^{16}$ می باشد.

کلاس C

کوچکترین کلاس از نظر تعداد کاربر می باشد قالب آدرسها بصورت زیر است :



NetID HostID

در این کلاس سه بیت باارزش عبارتست از (100) . عدد اول در آدرس بین 192 تا 223 می تواند باشد.

مثال :

194 . 224 . 1 . 2

NetID HostID

IP Address Class	First Octal Minimum	Maximum	Start in Binary	Number of Networks	Hsots
Class A	1	126	1	128	16777214
Class B	128	191	10	16384	65534
Class C	192	223	110	2097152	254
Class D	224	239	1110		
Class E	204	247	11110		

جدول ۳-۱ کلاسها در شبکهٔ TCP/IP

تعداد زیرشبکه ها در کلاس C

تعداد زیرشبکه ها در اینترنت برای این کلاس بسیار زیاد می باشد و برابر است با $(224-192)2^{16}=2^{21}$

تعداد Host ها در کلاس C

در این کلاس تعداد Host ها از همه کمتر است و برابر (2-256) می باشد.

تقسیم کلاسها

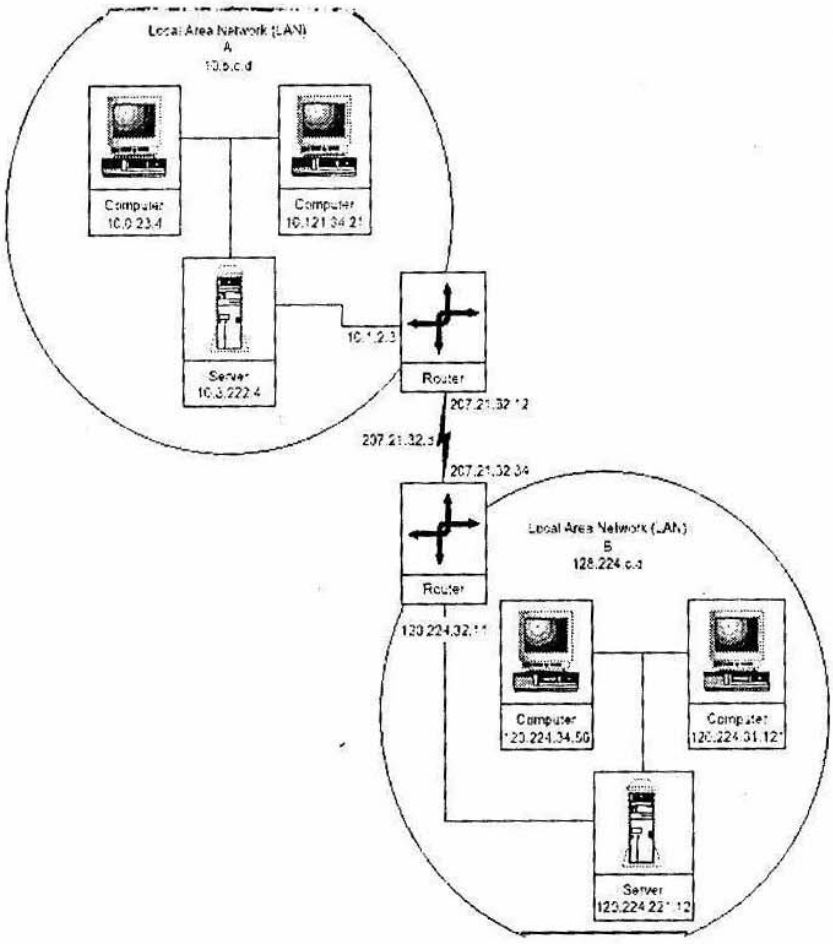
بسته به تعداد Host های یک زیرشبکه می توان یک کلاس را بصورت کامل استفاده کرد یا تقسیم نمود. این تقسیم می تواند توانی از 2 باشد.(مثلاً $\frac{1}{2}$، $\frac{1}{4}$، $\frac{1}{8}$ ....) طوری که تعدادHost بسته به کلاس به 2، 4 ،... 8 تقسیم می شود.

مثال : فرض کنید یک زیرشبکهٔ اینترنت۱/٤ کلاس C می باشد (۱/٤ اول) در این صورت آدرس بین 194.224. 1.0تا 194.224.1.63 خواهد بود.

یعنی ٦٤ تا Host می تواند داشته باشد.

Subnet mask

تعدادی Bit می باشد که با آدرس IP ،AND شده تا آدرس Subnet یا NetID بدست آید باید توجه کرد این آدرس هنگام نصب سرویس دهنده ، باید وارد گردد. بسته به نوع کلاس این آدرس فرق می کند.



شکل ۱۲-۱ اختصاص IP بین دو زیرشبکهٔ TCP/IP

برای کلاس کامل A : 255.0.0.0

برای کلاس کامل B : 255.255.0.0

برای کلاس کامل C : 255.255.255.0

مثال : اگر آدرس 194.224.1.2را با 0 .255.255.255 AND کنیم آدرس 194.224.1 بدست می آید که آدرس Subent است.

IP های خاص

این گونه IP ها که گروهی آنها را کلاسهای D و E معرفی می کند شامل IP های رزرو شده و Broadcast و Multicast و Loopback و غیره می باشند هیچ کامپیوتری حق استفاده از این IP ها را جهت ارتباط با اینترنت ندارد.

Broadcast

اگر بخواهید یک بسته به آدرس255.255.255.255 ارسال شود به کل کاربران اینترنت خواهد رسید (البته Router ها اجازهٔ عبور این بسته را به سادگی نمی دهند)

Multicast

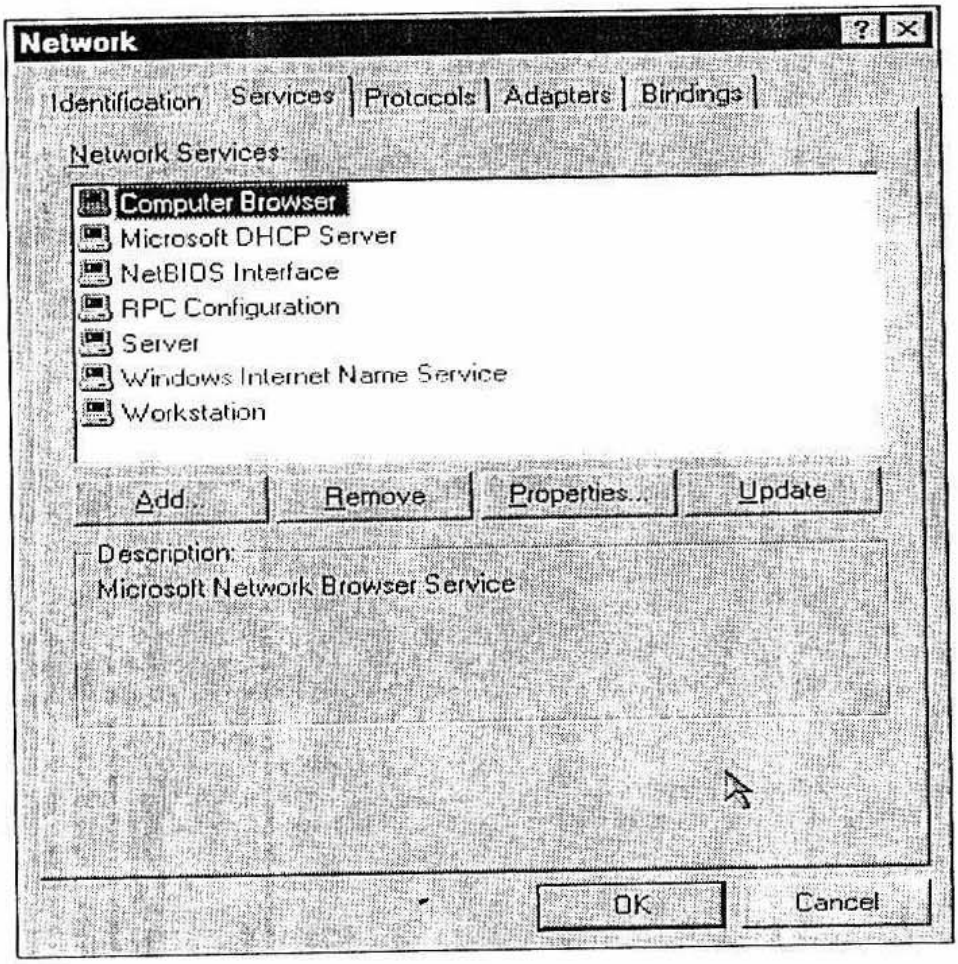
اگر بخواهید یک بسته را به تمام کاربران یک زیرشبکه ارسال کنید باید آنرا Multicast کنیم مثلاً برای زیرشبکه ای با آدرس Subent 194.224.1 مثلاً مربوط به زیرشبکهٔ دانشگاه آزاد باشدآدرس Multicast برابر 194.224.1.255خواهد بود بنابراین تمام کاربران زیرشبکه ای با آدرس NetID (194.224.1) این بسته یا پیغام را دریافت خواهند کرد.

۱-۱۲ سرویس دهندهٔ وب

در این بخش طریقهٔ ایجاد سرویس دهندهٔ وب در Win NT را شرح خواهیم داد، بصورت پیش فرض هنگامی که Win NT را نصب می کنیم سرویس دهندهٔ وب نمی باشد، ولی به کمک تعدادی سرویس در Control panel و Network می توان آنرا به یک سرویس دهندهٔ وب تبدیل کرد طوری که اگر کاربر از روی همان سرویس دهنده و یا یک سرویس گیرندهٔ محلی روی Bus آدرس IP یا حوزه را بنویسید می تواند سرویس دهندهٔ وب را ببیند.

مهمترین سرویسهای مورد نیاز جهت ایجاد یک سرویس دهندهٔ وب عبارتند از :

- IIS
- DNS
- DHCP
- RAS



شکل ۱۳ -۱ سرویسهای Win NT

IIS (Internet Information service)

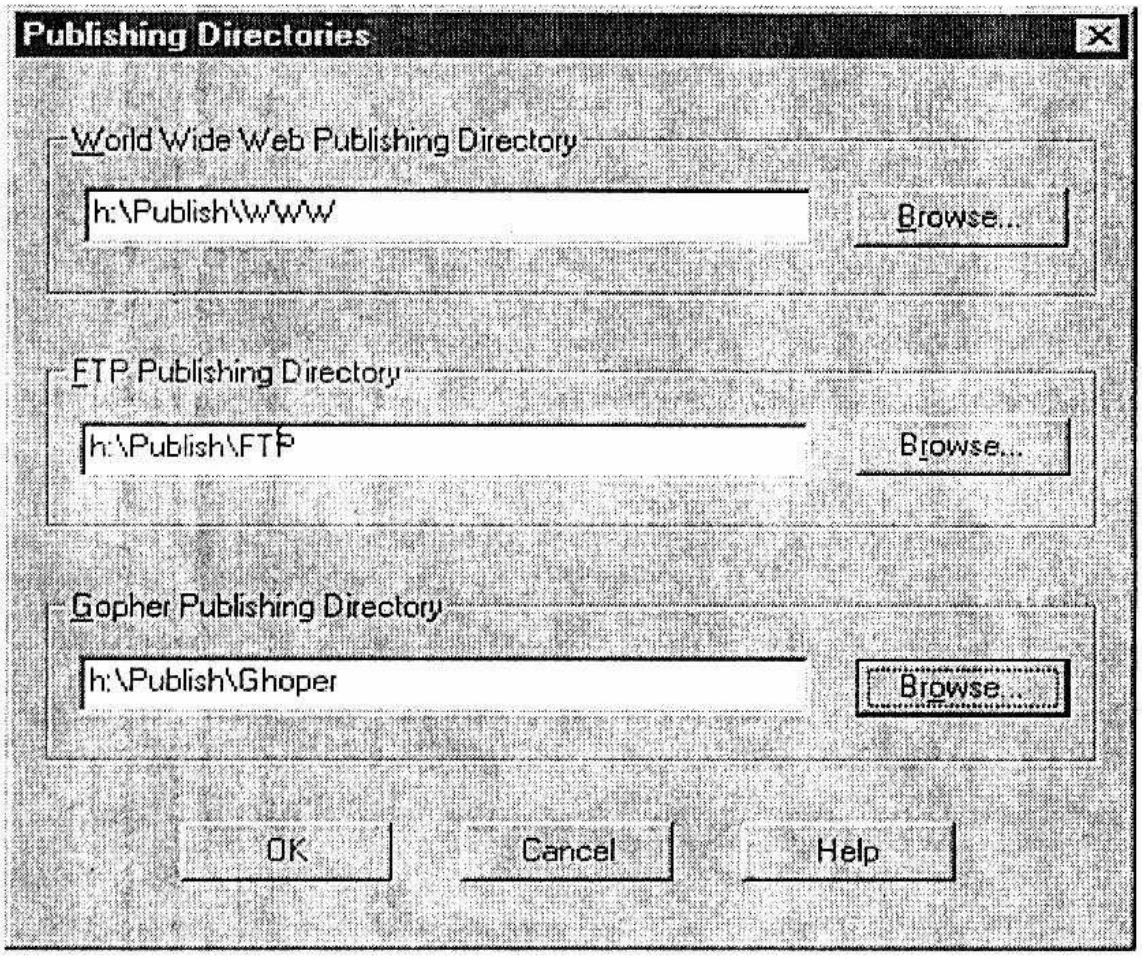
یک محصول مربوط به سرویس دهنده است که به عنوان بخشی از[1](MCIS) System ارائه گشته است. IIS در Windows NT Option Pack نیز قرار داره شده است. در کل IIS یک محیط برنامه نویسی و ارائه خدمات را برای نوشتن کاربردهای وب اینترنتی ارائه می نماید.

از آنجا که در بین ویرایش های مختلف IIS ، از ویرایش 4 به بعد تغییرات محسوسی در فرایند کاری آن پدید می آید توضیحات ما در حقیقت بر مبنای نسخه های 4 و IIS5 می باشد. موارد زیر به همراه IIS4 ارائه شده اند:

۱. سرویس دهندهٔ WWW
۲. سرویس دهندهٔ FTP
۳. سرویس دهندهٔ تراکنش Microsoft (MTS)
٤. سرویس دهندهٔ Microsoft SMTP
٥. سرویس دهندهٔ Microsoft MNTP
٦. سرویس دهندهٔ Microsoft NNTP
۷. سرویس دهندهٔ Index Microsoft

۸. سرویس دهندهٔ Microsoft Certificate

۹. Microsoft Server Express



شکل ۱۴-۱ نصب IIS و مشخص کردن فهرستهای ریشه

با استفاده از موارد فوق می توان سایتهای وب، سایتهای Ftp ، گروههای خبری، سرویس دهندهٔ خدمات پستی، دسترسی امن کاربر و ویژگی های مفیدی در محیطهای برنامه نویسی را ایجاد نمود.

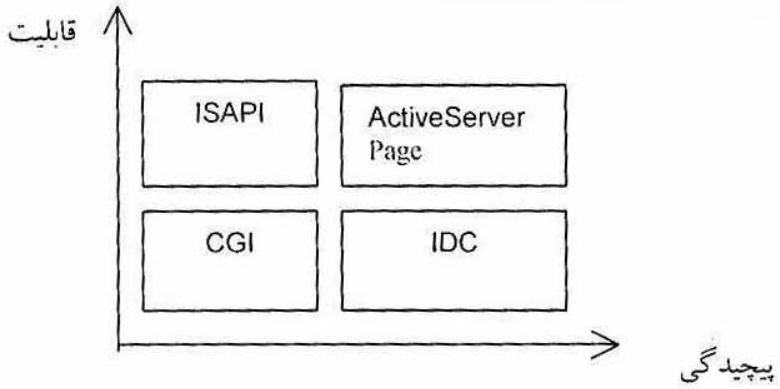
در IIS موارد زیر نیز وجود دارند:

۱. سرویس ارتباط با اینترنت برای RAS (Internet Connection Service for RAS)

۲. سرویس دهندهٔ صف پیغام Microsoft (Microsoft Message Queue Server)

۳. میزبان اسکریپت نویسی Windows (Windows Scripting Host)

امکانات برنامه نویسی IIS را با یکدیگر مقایسه نموده است. در بخشهای بعد به CGI اشاره شده است در اینجا صرفاً به ضعفهای CGI و بررسی اجمالی سایر تکنولوژیها پرداخته می شود.

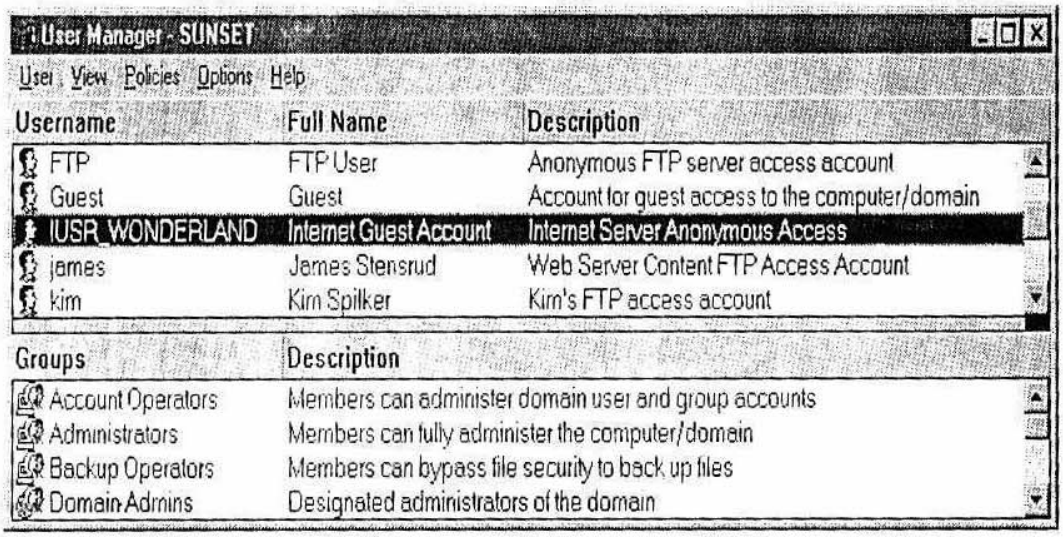


شکل ۱۵-۱ مقایسهٔ روشهای برنامه نویسی تحت وب از طریق IIS

از طریق این سرویس ، سرویس دهندهٔ اینترنت به یک Web Server تبدیل می شود با نصب آن سرویسهای وب ، Ftp و Gopher به NT اضافه می گردد. جهت نصب IIS بصورت زیر عمل می کنیم:

۱ نصب IIS از طریق Controlpanel و Network و Services است.

۲ با برنامه Internet service Manager در منوی Microsoft Internet ، Program می توان آنرا اجرا کرد.



شکل ۱۶-۱ شناسهٔ کاربردی اینترنت در IIS

هنگامی که IIS نصب شود یک شناسهٔ کاربری به نام IUSR-SERVE NAME می سازد.

↓

نام سرویس دهندهٔ NT

برای اینکه همهٔ افراد بتوانند صفحات موجود در فهرست ریشهٔ سرویس دهندهٔ وب wwwRoot را ببینند باید به این شناسه کاربری حقوق دستیابی به این فهرست را بدهید با نصب IIS از زیر شاخه های مربوط به ریشهٔ وب ، Ftp ، Gopher را سؤال می کند که بصورت پیش فرض عبارتند از :

c:\Inet pub\wwwroot\
c:\Inet pub\ ftproot\

c:\Inet pub\GopherRoot\

البته می توان آنها را تغییر داد.

(Domain Name Service) DNS

در اکثر شبکه های بزرگ یک سرویس دهنده به نام Name server داریم که وظیفهٔ انجام عملیات Namming را برعهده دارد.

DNS در یک شبکهٔ TCP/IP وظیفهٔ تبدیل نامهای اینترنتی مثل www.cnn.com به آدرسهای IP مثل 107.2 31.150را بر عهده دارد. بستهٔ انتقالی جهت رسیدن به مقصد نیاز به عدد IP دارد و از طریق حوزه مسیر یابی ممکن نیست. با نصب DNS در ویندوز NT این سرویس دهنده به سرویس دهندهٔ نام حوزه[1] تبدیل خواهد شد. برای اجرای آن بعد از نصب سرویس بصورت زیر عمل می کنیم :

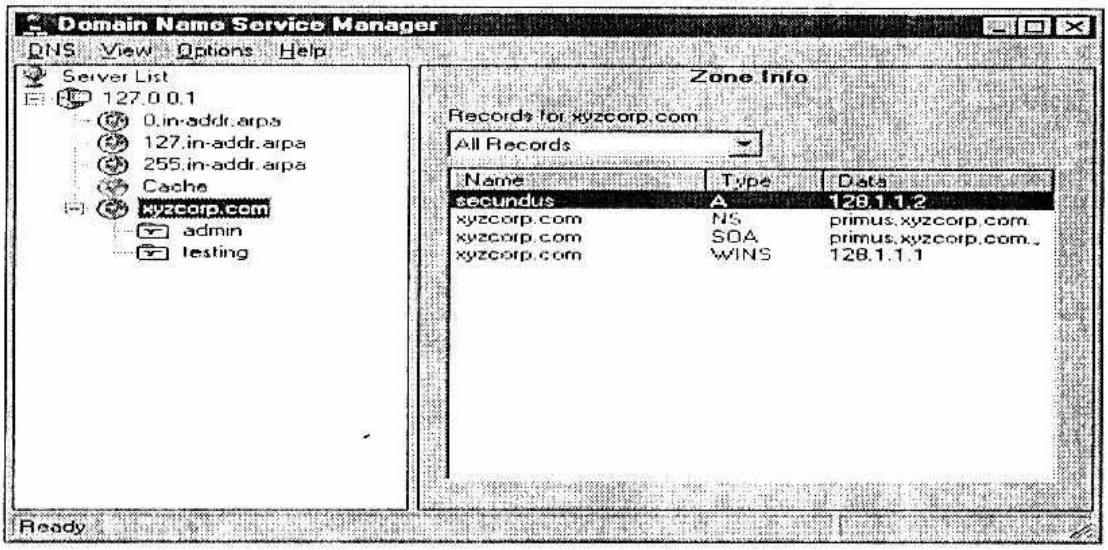
Programs → Administrative → DNS Manager

عملیات لازم جهت راه اندازی DNS

۱- از منوی DNS.... NEW Server برای اضافه کردن آدرس IP سرویس دهندهٔ NT به فهرست سرویس دهندهٔ DNS استفاده می کنیم.

۲- از DNS....NEW Zone برای مشخص کردن Zone و نوع آن استفاده می شود.

۳- در نهایت با بکار بردن DNS....NEW Domain حوزهٔ اولیه یا ثانویهٔ سرویس دهندهٔ DNS را مشخص می کنیم.



شکل ۱۷-۱ نصب DNS روی Win NT

حال بعد از این که بانک اطلاعاتی ایجاد می گردد می توان رکوردهای مجزایی بـه آن اضافه کـرد. ایـن بـانک اطلاعاتی یک فایل متنی است در زیر شاخهٔ زیر :

Server Root \ System 32 \ Dns \ Name .Dns

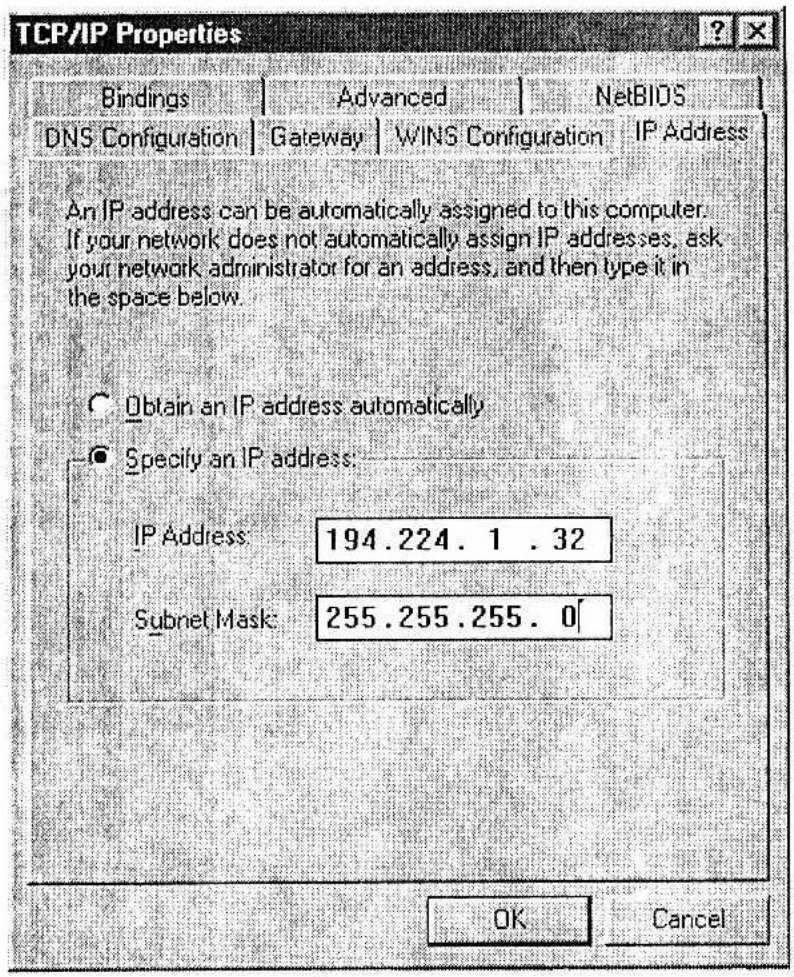
```
@    IN   SOA      randall.telemark.net.
postmaster.randall.telemark.net. (
                                1995101001       ; serial
number
                                10800    ; refresh [3h]
                                3600     ; retry   [1h]
                                691200   ; expire  [8d]
                                86400 )  ; minimum [1d]
        IN      NS      randall.telemark.net.
        IN      NS      www2.canada-stockwatch.com.
        IN      A       204.191.227.65
        IN      MX      10      randall

localhost       IN      A       127.0.0.1

randall         IN      A       204.191.227.65
                IN      MX      10      randall

randallg        IN      A       204.191.227.66
                IN      MX      10      randall

ras             IN      A       204.191.227.130

mail    IN      CNAME   randall
smtp    IN      CNAME   randall
pop     IN      CNAME   randall
www     IN      CNAME   randall.telemark.net.

pam     IN      CNAME   randallg
```

شکل ۱۸-۱ محتویات فایل DNS روی Win NT

(Dynamic Host Configurate Protocol) DHCP

طریقهٔ انتساب IP به کاربران به دو صورت ایستا، پویا انجام پذیر است در حالت ایستا کاربر یـک آدرس IP دارد که مخصوص خود او می باشد و باید آنرا در سیستم خود نصب نمایند.

Controlpanel → Network → Tcp / IP → Properties → IP Address



شکل ۱۹-۱ اختصاص ایستای IP به سرویس گیرنده در Win98

با نصب DHCP امکان انتساب پویای IP از سوی سرویس دهنده به سروپس گیرنده اضافه می شود. با این کار دیگر کامپیوترهای سرویس گیرنده به سادگی به شبکه متصل خواهند شد و Over head سرپرستی کاهش خواهد یافت.

توجه :

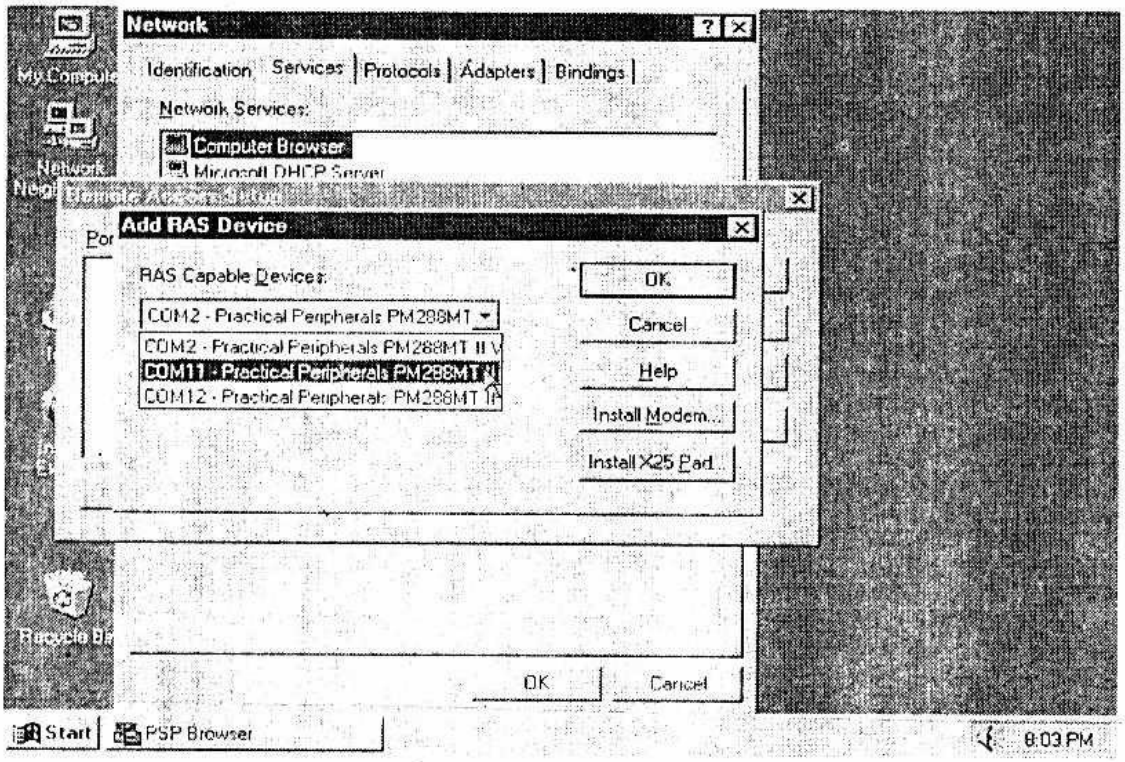
اگر نیاز به تغییری در شماره های IP وجود داشته باشد تنها DHCP تغییر می کند و نیازی بـه تغییـر در سـرویس گیرنده ها نیست.

با نصـب DHCP سـرویس دهنـده بـه یـک سـرویس دهنـدهDHCP تبدیـل خواهـد شـد در بخـش DHCP Management در منوی Scope می توان محدودهٔ IP که باید بصورت پویا به کاربران داده شود (طبق کـلاس IP) مشخص می گردد.

(Remote Access Service) RAS

RAS نـیز یک سرویس می باشد که با نصب آن سرویس دهنده به یک سرویس دهنـدهٔ راه دور تبدیـل خواهـد شد، وظیفهٔ RAS مدیریت و کنترل پورتها و مودمهای متصل به آنها همچنیـن کـاربران راه دوری کـه از طریـق خـط تلفن با سرویس دهنده ارتباط برقرار نموده است.

هنگامی که از طریق کارت Mutiport می خواهیم امکان دسترسی از راه دور را فراهم کنیم بعـد از نصـب RAS تمامی درگاهها و مودمهای تشخیص داده شده و مدیریت می شوند. در صـورت تمـاس تلفنـی کـاربر RAS خـودش گوشی را برداشته و شناسه کاربر و رمز عبور را سؤال می کند و ارتباط pppرا برقرار می نماید.



شکل ۲۰ -۱ تنظیمات RAS روی Win Nt

عملیات لازم قبل از ایجاد سایت وب

بعد از فراهم کردن تجهیزات سخت افزاری قبل ازشروع به برنامه نویسی بد نیست که عملیات زیر را انجام دهید:

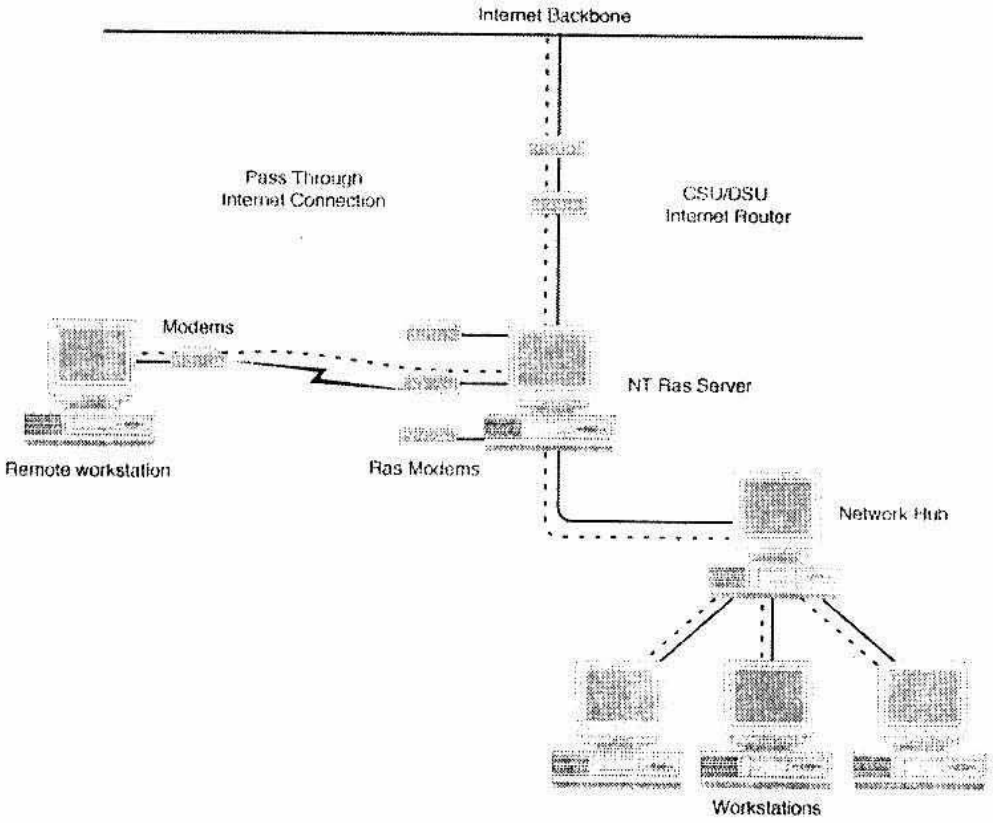
۱- صفحه وب خود را روی چندین سکوهای سخت افزاری[1] تست کنید.

۲ - وقتی شروع به برنامه نویسی می کنید مطمئن شوید که سرویس دهنده در امن ترین حالت خود قرار دارد.

۳ - بطور غیر مجاز به سایت خود وارد شوید این کار را با یک Pc روی شبکه و از طریق Telnet و Ftp یا ... انجام دهید.

٤ - از طریق یک شناسهٔ کاربری Dial-up از راه دور نیز این کار را تکرار کنید.

٥ - شروع به جستجو روی گزارش اشکالات و خطاها روی سرویس دهنده ها و مرورگرهای اینترنت نمایید.



شکل ۲۱ -۱ ایجاد شبکهٔ راه دور از طریق RAS

۱۳-۱ زبان HTML (Hyper text Markup language)

مهمترین زبان نشانه ای می باشد که بیشتر صفحات وب با این زبان نوشته شده یا از آن استفاده می کنند. HTML بر مبنای SGML [۱] که یک استاندارد مدیریت اطلاعات است ایجاد گشته است. این استاندارد توسط سازمان بین المللی استانداردسازی ISO [۲] در سال ۱۹۸۶ به قصد مهیا نمودن اسنادی که اطلاعات قالب بندی شده و دارای پیوند را مستقل از کاربرد و سکوی سخت افزار ارائه نماید، معرفی شد. SGML یک مکانیزم مشابه گرامر (Grammar-Like) را فراهم می آورد که در آن کاربران می توانند ساختار اسناد و Tag هایی را که بر ساختار یک سند مستقل دلالت می کنند را تعریف نمایند.

HTML دارای برچسب هایی می باشد که این برچسبها به سند اضافه می شوند و طریقهٔ نمایش سند را از طریق مرورگر مشخص می کنند. در این بخش در مورد زبان HTML چگونگی برنامه نویسی و برچسبهای آن بحث خواهیم کرد. فهرست مطالبی که در این بخش خواهیم گفت عبارتند از:

۱- ساختار برنامه های HTML

۲- برچسبهای Text

۳- تصاویر و صدا .(Picture & Sound)

٤- جداول[1] در HTML

٥- پیوندها (link)

٦- نقشه (Map)

٧- Frome ها در HTML

٨- Form ها و عناصر آنها مثل : Key, Check box , Radio botton ,Editor, Inputline ....

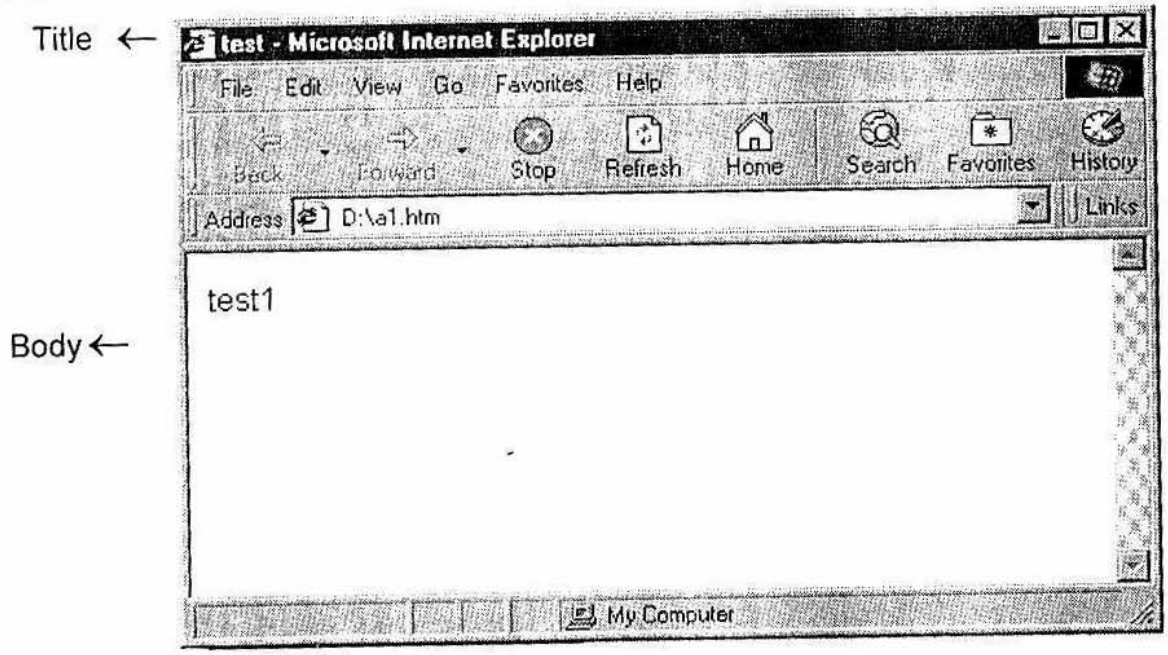
٩- Script ها در HTML (Client Side)

می توان برنامه های HTML را در هر ویرایشگر متنی مثل Edit در Dos و یا Notepad در Windows بنویسید و آنها را توسط مرورگرهایی چون Internet explorer یا Netscape اجرا و مشاهده نمائید. البته HTML دارای Generator هایی مثل Frontpage نیز می باشد که می توان از طریق آنها صفحهٔ HTML را تولید کرد.

ساختار برنامه های HTML

اکثر Tag های HTML بصورت دوگان بکار می روند طوری که با یک <Name> شروع و با یک </Name> خاتمه پیدا می کند. اگر یک Tag دارای آرگمان باشد این آرگمانها در Tag اول بکار می روند.

```
<HTML>
<HEAD>
<TITLE > Test </TITLE>
</HEAD>
<BODY>
 Test 1
</BODY>

</HTML>
```



شکل ۲۲-۱ خروجی برنامهٔ HTML

ساختار Body

Body دارای آرگمانهایی می باشد که بصورت زیر استفاده می شود:

```
<Body    BGColor ="          "
         Text = "            "
         LINK ="             "
         VLINK ="            "
         Background =  "     " >
         ———————
         ———————
         </Body>
```

BGColer رنگ زمینه را مشخص می کند که هم می تواند بصورت نام رنگ (RED) و یا کد رنگ #FF0000 بکار رود Text و Link نیز رنگ متن و پیوندها را در صفحه مشخص می کند VLINK رنگ پیوندها بعد از دیده شدن (Visit) می باشد. Background نیز آدرس یا URL عکسی است که می خواهید به عنوان زمینهٔ صفحه قرار داده شود که می تواند هر فایل Gif یا Jpg باشد.

رنگها در HTML

در HTML رنگها را هم می توان بصورت کد و هم نام بکار برد مثلاً:

```
# 000000    Black
# FFFFFF    White
#  ff0000    Red
#  00ff00    Green
# 0000ff     Blue
#  ffff00    Yellow
```

هر رنگ ترکیبی از RGB می باشد که بصورت زیر ساخته می شود :

بدین صورت می توان 2^{24} رنگ ایجاد کرد:

```
  R   G   B
# XX XX XX
```

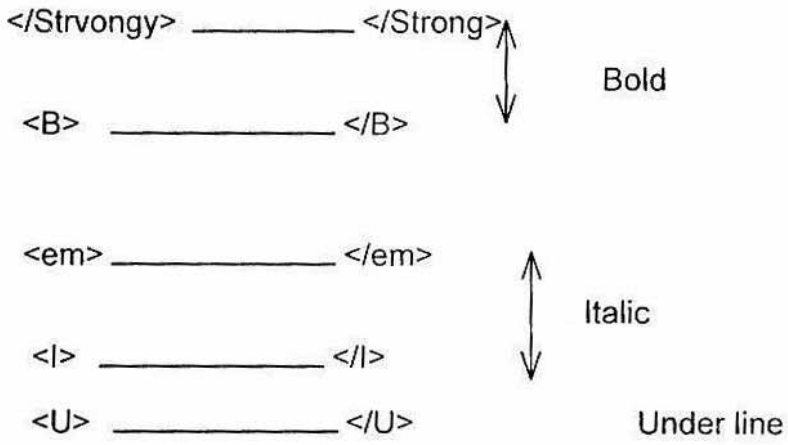
مثال :

این برنامه یک صفحه ایجاد خواهد کرد که در آن متنها بصورت سیاه رنگ و پیوندها به رنگ آبی و پیوندها بعد از دیده شدن زرد خواهند شدزیر صفحه نیز با تصویر Map.gif پر خواهند شد.

```
<HTML>
<Body    BGCOLOR = "White"
         TEXT = "Black"
         LINK = " # 0000ff"
         VLINK = " # ffff00"
         Background= " ../ Image/ Map . gif " >Hello
</Body>
```

</HTML>

Tag های متن



توجه :

در HTML بعضی برچسبها دوتایی و بعضی تنها بکار می روند.

پاراگرافها در متن

پاراگرافها با <P> مشخص می شوند. از پاراگرافها جهت راست چین یا چپ چین کردن متن می توان استفاده کرد.

```
<P  Align = "  " >
_____
_____
_____
</P>
```

طوری که در آن Align می تواند Center- Left -Right باشد. همچنین می توان از برچسب < Center > برای بکار بردن متن در وسط صفحه استفاده کرد. <Center> _____</Center>

توجه :

در حالت معمولی اگر متن را زیر هم بنویسیم مرورگر آنها را به هم می چ باند به همین جهت برای فرستادن متن به خط بعد از
 استفاده می شود.

توضیحات در HTML

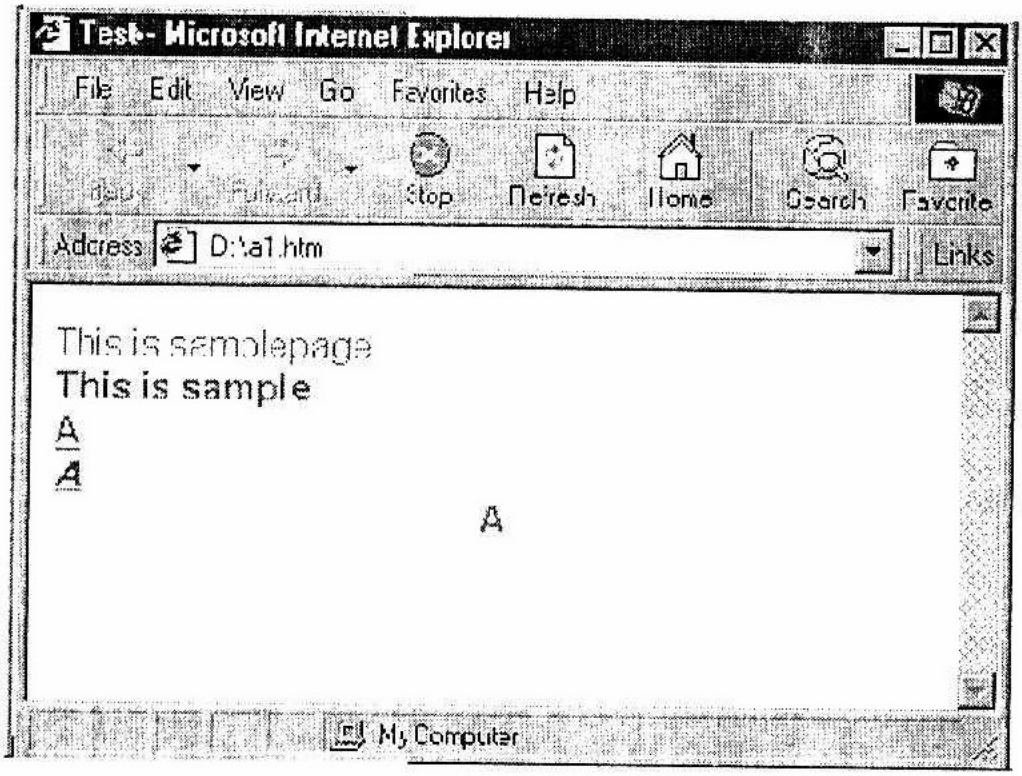
از طریق برچسب زیرتوضیحات یا Comment ها می توانند بکار روند :

```
< !_____ >
```

مثال :

```
<HTML>
<Head >
    <Title> Test  </Title>
</ Head>
<Body>
This is samplepage <Br>
<B> This is sample </B> <Br>
<U> A </U> <Br>
```

```
<U> <B><I> A </B> </U></I> <Br>
<Center> A  <Center>
</Body>
 <HTML>
```



شکل ۲۳ - ۱ اجرای برنامه HTML بالا

چشمک زن شدن متن

جهت اینکار از برچسب <Blink> استفاده می شود البته این برچسب درIE کار نمـی کنـد ولـی در Netscape می توان از آن استفاده کرد.

<Blink> _____ </Blink>

سرآیند ها در متن (عنوان)

برای این کار از برچسب های زیر استفاده می شود

<H1> ___ </H1>
<H2> ___ </H2>
</H6> ___ <H6>

اندازه فونتها افزایش می یابد

استفاده از اندیس و توان

جهت ایجاد چند جمله ای های ریاضی می توان از آنـها اسـتفاده کـرد جـهت اینکـار از بـرچسـبهای <SUB> و <SUP> استفاده می شود.

_{___} اندیس

^{___} توان

مثال : $2x_2^3$

2 × ₂ ³

مشخص کردن Font متن

جهت اینکار از برچسب <Font> استفاده می شود. در این برچسب آرگمان Size اندازه را مشخص می کند که می تواند اعداد 1 تا 7 باشد هرچه این عدد کوچکتر باشد Font بزرگتر است Color نیز رنگ و FACE نام فونت را مشخص می کند مثلاً Arial یا Sepehr برای فونت فارسی

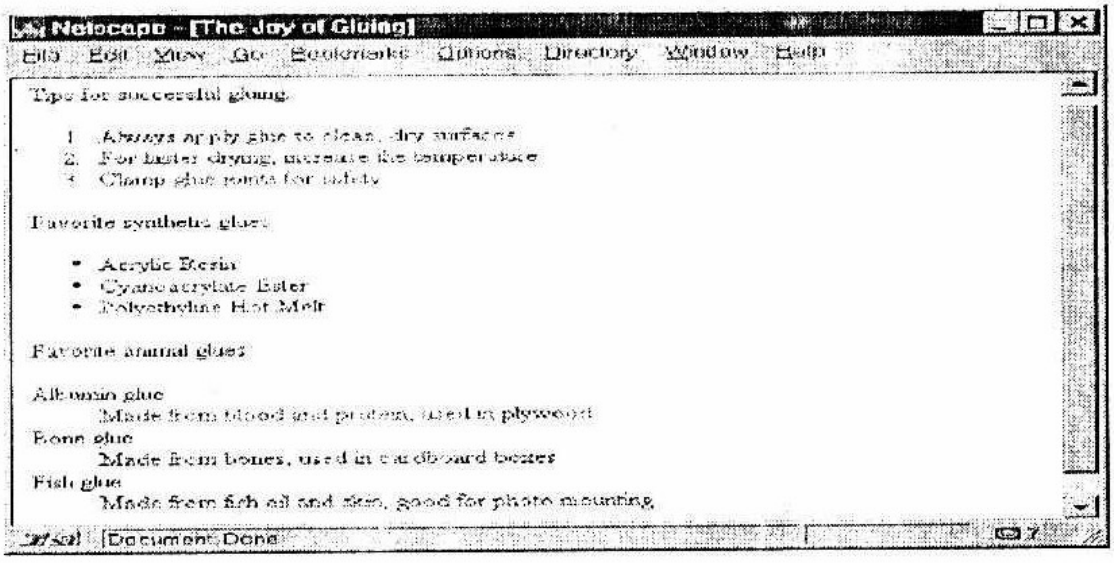
```
<Font    SIZE = "     "
         COLOR = "    "
         FACE = "     " >
         _____
         _____
         _____
         </Font >
```

*برای تعمیم دادن Font اصلی از نشانه زیر استفاده می شود

```
<BASE FONT   SIZE = "   " >
```

Bulleted list , Order list

برای ایجاد یک لیست ترتیبی و یا یک لیست گلوله دار از برچسب های زیر استفاده می کنیم. در این صورت لیست را خودش شماره گذاری می کند و یا با علامت گلوله آنرا مشخص می کند.



Bulleted list , Order list شکل ۲٤-۱

```
</OL>
        <LI>  _____ </LI>
        <LI>  _____ </LI>
    </OL>
    <UL>
        <LI>  _____ </LI>
        <LI>  _____ </LI>
    </UL>
```

استفاده از تصویر در HTML

برای نمایش تصاویر با قالب Gif و Jpg می توان از برچسب زیر استفاده کرد.

```
<Img  SRC = "  "   URL = "    "   ALT = "    "   WIDTH = "   "   HIGHT = "    "
                BORDER = "  "      ALIGN = "    " >
```

این برچسب که بصورت تنها بکار می رود دارای آرگمانهای SRC یا URL جهت مشخص کردن آدرس عکس Gif ، Jpg یا Bmp و Alt مشخص کنندهٔ متنی است که به جای عکس نمایش داده می شود به آن Alternative Text گویند، Width و Hiaht نیز ابعاد عکس و Border ضخامت کادر عکس را مشخص می کند Align نیز می تواند (Top یا Middle یا Buttom باشد) وضعیت عکس نسبت به خط را نشان می دهد. باید توجه کرد که از برچسب IMG می توان جهت فراخوانی برنامه های دروازه ای روی سرویس دهنده نیز استفاده کرد

جدول (Table)

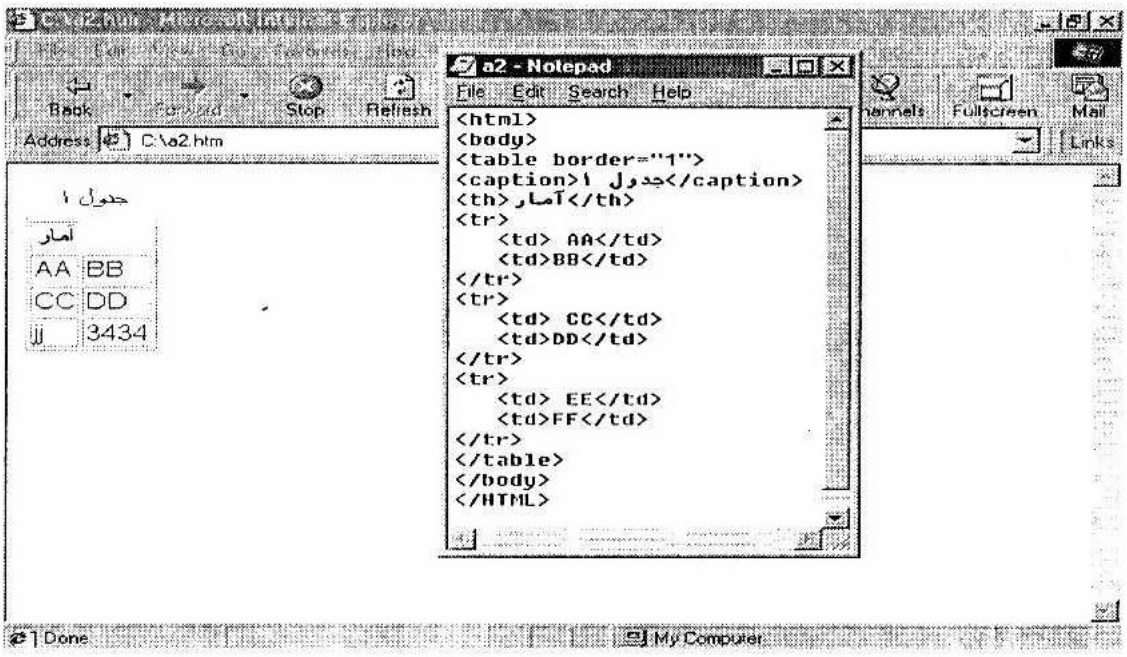
جداول HTML دو کاربرد دارند:

۱ جدول کشیدن

۲ تقسیم کردن صفحهٔ مرورگر

برچسب آن <Table> می باشد که بصورت زیر استفاده می شود.

```
<TABLE  Border = "1"   Color = "White">
<CAPTION>  جدول۱  <CAPTION>

  <TH>  آمار     < / TH>

  <TR >
  <TD>  _______  </TD>

  <TD>  _______  </TD>

  < /TR>
  </ TABLE>
```



شکل ۲۵-۱ مثالی از یک جدول در HTML

خط کشی در صفحه

جهت کشیدن خط در صفحه از برچسب <HR> استفاده می شود توسط آن می توان اندازه و رنگ خط را نیز مشخص کرد:

```
<HR Width="     " Color = "  "
Size="  ">
```

صدا در HTML

برای این کار از برچسب BGSOUND استفاده می شود. به کمک این برچسب می توان فایلهای WAV , MIDI را نمایش داد همچنین در صورت وجود داشتن Plugin مربوطه فایلهای RA و AU و MP2 و MP3 را نیز می توان بکار برد. (توجه کنید که با استفاده از برچسب <A> نیز می توان این کار را انجام داد:

```
<BGSOUND        SRC= "   "   URL = "    "      LOOP = "   ">
```

Loop تعداد دفعات پخش صدای مربوطه است. می توان به جای " "Loop= که تعداد دفعات تکرارموزیک است از Infinite استفاده کرد. در این صورت آهنگ بطور دائم نواخته خواهد شد.

پیوندها در HTML

برچسب مربوطه <A> (anchors) است که دو کاربرد می تواند داشته باشد.

۱ - برای email حالت کلی آن بصورت زیر است :

```
< A HREF = " Mailto : your email Address" >
_____
_____
</A>
```

مثال :

```
<A HREF = " Mailto : Akbari-behzad @ hotmail.Com "> A  </A>
```

۲- برای ایجاد پیوند های وب

```
<A HREF = "  " Name = "   " >
_____
_____
< /A>
```

مثال :

```
<A HREF = "http://www.sample.com/default.htm"> Sample  </A>
```

Map در HTML

در HTML این امکان وجود دارد که یک عکس را (gif یا jpa) به چندین ناحیه تقسیم نمود طوری که کاربر با کلیک کردن در آن ناحیه بتواند به صفحهٔ دیگری متصل گردد.

برچسب مربوط به آن <MAP> می باشد.

هر Map به یک تصویر مربوط می شود که اینکار از طریق برچسب <IMG> و آرگمان USEMAP انجام می شود.

```
<MAP   NAME = " Iran ">
      <AREA   SHAPE = " rect " CORDS = " 100,100,150,150 " HREF="
       <AREA    Shap = " default "     nohref >
</MAP>
<IMG  SRC = "    "  USEMAP=" #  Iran ">
```

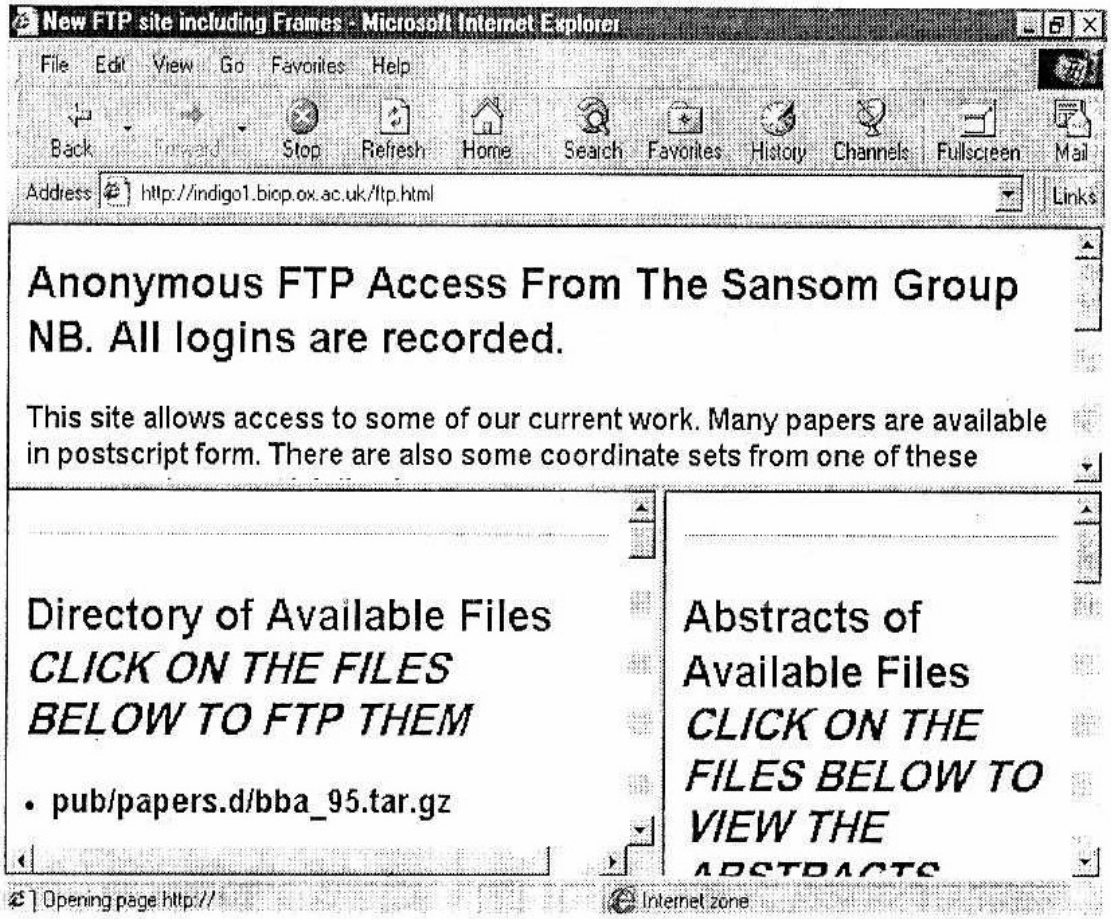
Frame ها در HTML

از طریق فرمها می توان چندین HTML را در یک صفحهٔ مرورگر نمایش داد در این حالت از برچسب frameset استفاده می کنیم مانند زیر :

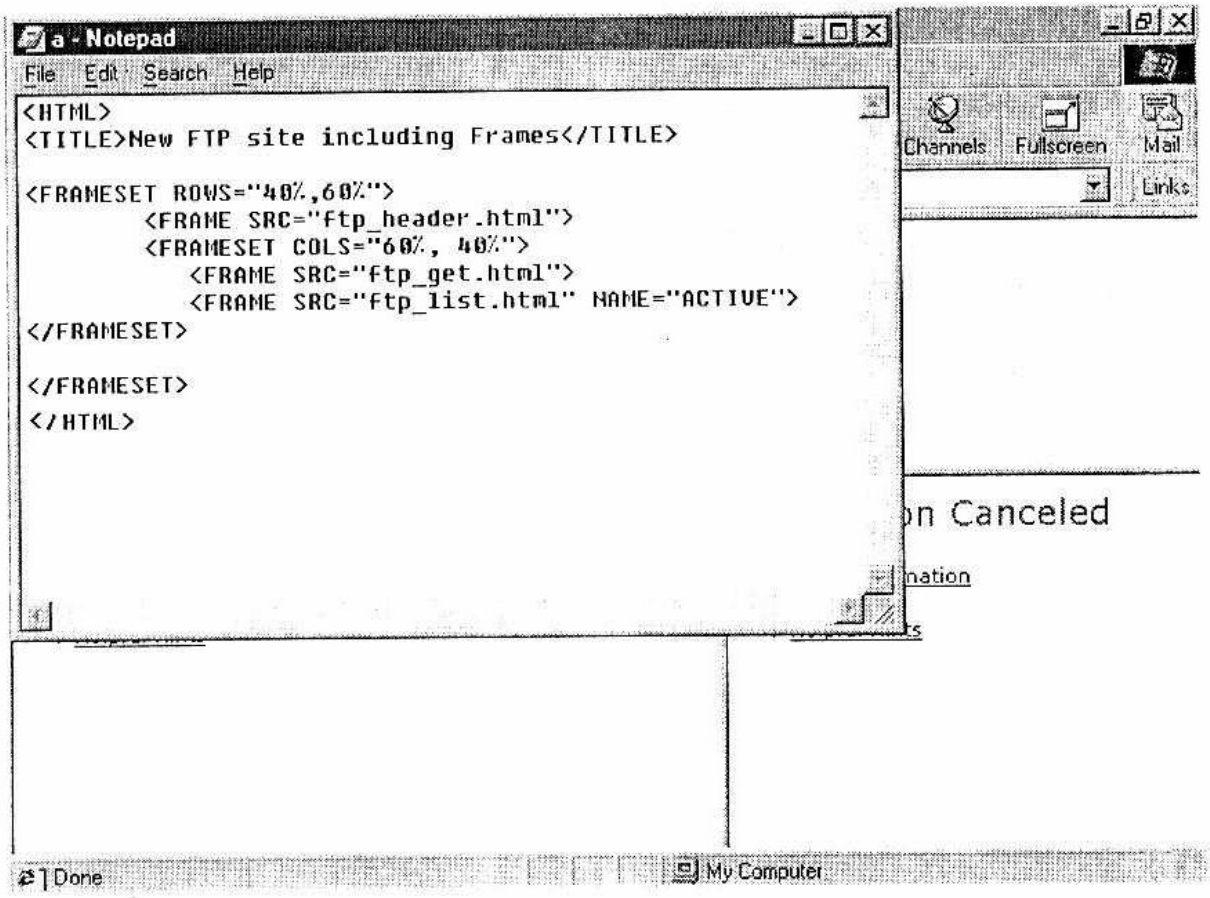
هر frameset می تواند عمودی یا افقی باشد که خودش می تواند تعدادی frame یا frameset داشته باشد.

برای استفاده از frame ها همواره باید یک برنامه hit.main تعریف کرد که در آن بدنهٔ frame را تعریف نمود سپس بسته به تعداد frame ها برای هر کدام یک html نوشت مثلاً" برای صفحهٔ زیر نیاز به چهار برنامه می باشد :

main.htm
titte.hit
menu.htm
about.htm



شکل ۲۶ –۱ مثالی از یک form در اینترنت



شکل ۲۷-۱ متن برنامهٔ HTML برای برنامهٔ اصلی شکل (۲۹-۱)

فرمت کلی :

هر مجموعه Frame یا افقی (Rows) است یا عمودی(Cols) به همین جهت باید در Frameset نوع را مشخص کرد.

```
<Frameset  Rows = "   ">
<Frame       SR = "    " Scrolling = NO NOResize >
 _______
 _______
</frameset >
```

مثال :

```
 <HTML>
<framset       ROWS = " 10% ; * ">
          <FRAME      SRC = " TLTLE.htm " Scrolling = No   NORESIZE>
           <Frameset   Cols=  "20%; *">
           <FRAME    SRC = " MENU.htm"  SCRO LING =NO  NORESIZE>
           <FRAME              SRC = " About.htm" Resize>
           </FRAMESET>
 </FRAMESET>
 </HTML>
```

*باید توجه کرد که Frame ها را نباید در Body تعریف کرد.

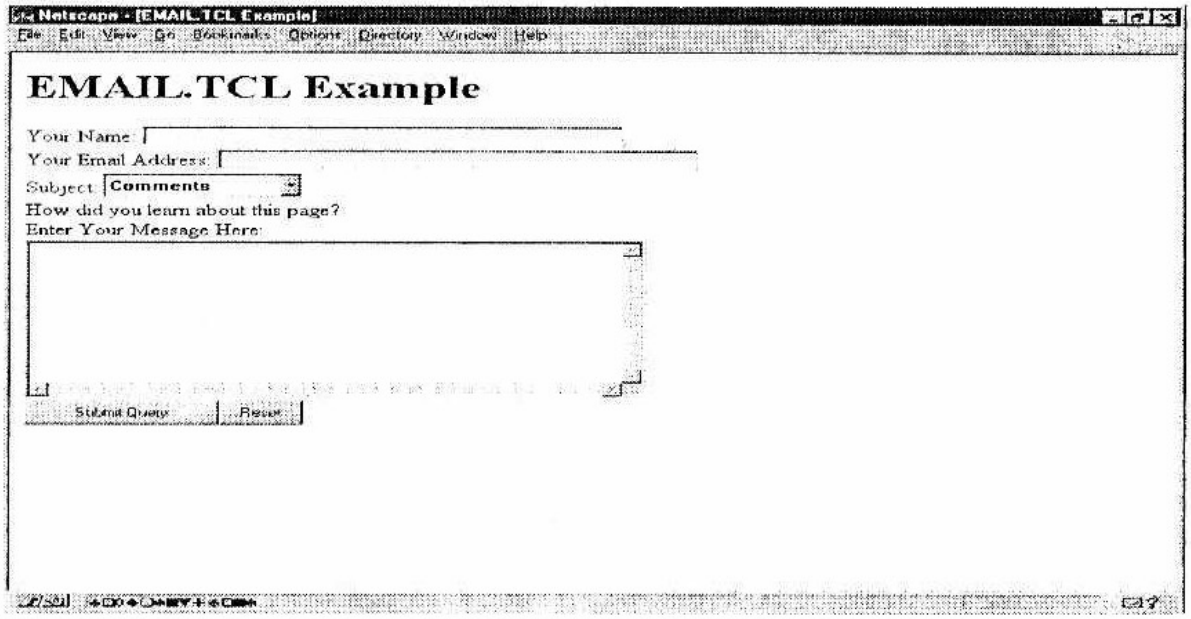
FORM ها در HTML

همانند سایر زبانهای برنامه نویسی در HTML نیز مـی تـوان فـرم ورود اطلاعـات ایجاد نمـود بـرای ایـن کـار از برچسب form استفاده می شود. form ها اجزای زیر را می توانند داشته باشند :

TEXT (Inputline)

Editor
Radiobotton
Check box
Key

هر کدام از شیء های بالا را می توان در فرم بکار برد مثلاً فرم زیر را داریم.



شکل ۲۸-۱ یک فرم ورود اطلاعات در HTML

هر کدام از شیء ها باید در داخل برچسب form بکار رود. برچسب form خودش دارای تعدادی آرگمان مـی باشد که بصورت زیر بکار می رود.

< form Method = " " Action= " " >

Method روش ارسال اطلاعات به برنامهٔ دروازه ای[1] روی سرویس دهنده است . در مـورد ایـن آرگمانـها در بخـش برنامه نویسی Cgi توضیح خواهیم داد.

اکثر این اشیاء را از طریق برچسب Input که داخل form بکار می رود می توان ساخت قالب آن بصورت زیر است:

باید توجه کرد نوع پیش فرض برای Input ، Text می باشد

<Input type = " text" Name = " I1 Size = " " Value = " ">

اندازه اگر گفته نشود بطور پیش فرض ۵۰ کاراکتر است و Name نام شیء است کـه در برنامهٔ دروازه ای یـا برنامـه نویسی Script بکار می رود.

این برچسب هم می تواند تنها بکار رود و هم دو تایی که در این صورت با </ Input> خاتمه می یابد.

کلیدها در HTML

سه نوع کلید در HTML می توان ساخت :

۱- Submit کنترل و مدیریت آن توسط مرورگر است.

۲- Reset کنترل و مدیریت آن توسط مرورگر است.

۳- کلیدهای خاص (Button) کنترل و مدیریت آن توسط کاربر است.

کلید Submit کلیدی است کـه باانتخـاب آن اطلاعـات موجـود در form بـه برنامهٔ دروازه ای کـه در Action مشخص شده اند ارسال می گردند. باید توجه کرد که این اطلاعات ابتدا کد شده و بعد از URL چسـبیده شـده و بـه برنامهٔ دروازه ای ارسال می شوند.

```
<Input    type = " Submit "Name = "OK" Value = " OK">
```

کلید Rest باعث خالی شدن Object های داخل form خواهند شد.

```
<Input   type = " Reset"   Name =" Res"    Value = " Rest">
```

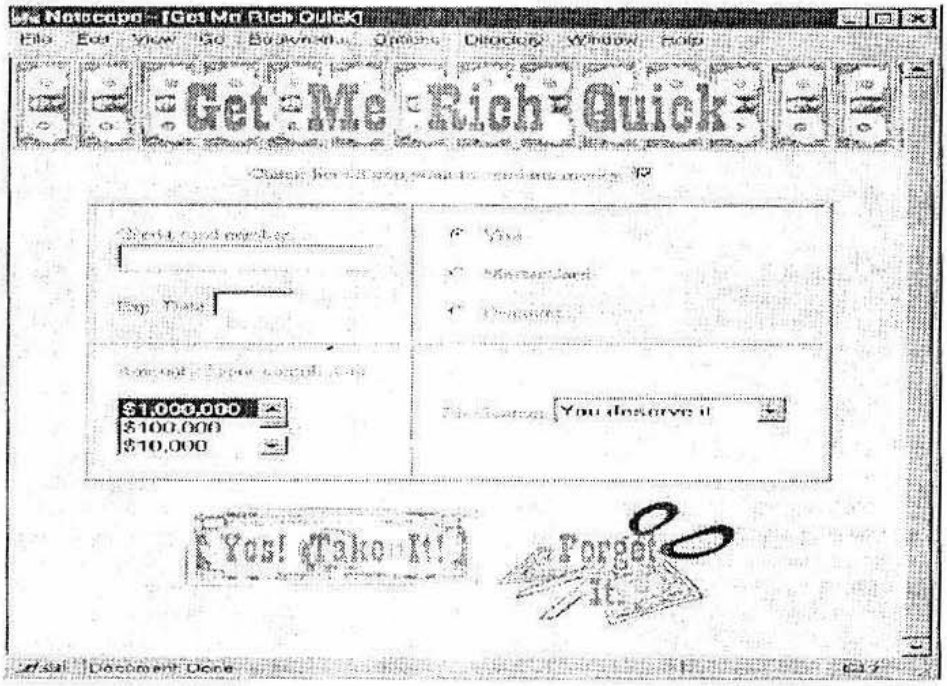
کلید Button برای برنامه نویسی خصوصا" اسکریپت ها بکار می رود.

```
< Input   type = " Button"   Name =" Bu1"    Value = " OK"    On click ="Win open ( )">
```

بافشار کلید OK زیرروال( ) Win open فراخوانی می شود.

Radio Button

Radio Button برای ایجاد لیستی از گزینه ها که بتوان هر کدام را انتخاب کرداز Radio استفاده می شود.مثـل لیست زیر :



شکل ۲۹-۱ Radio Button و Check Box در HTML

```
< Input   Type = "Radio"   Name ="P1">Black <Br>
< Input   Type = "Radio"   Name ="P1" >Red <Br>
< Input   Type = "Radio"   Name ="P1" >Green <Br>
```

CHECKBOX

برای ایجاد لیستی که بتوان هر کدام از گزینه ها را انتخاب نمود طوری کـه بتـوان بیشـتر از یکـی را نـیز انتخـاب نمود :

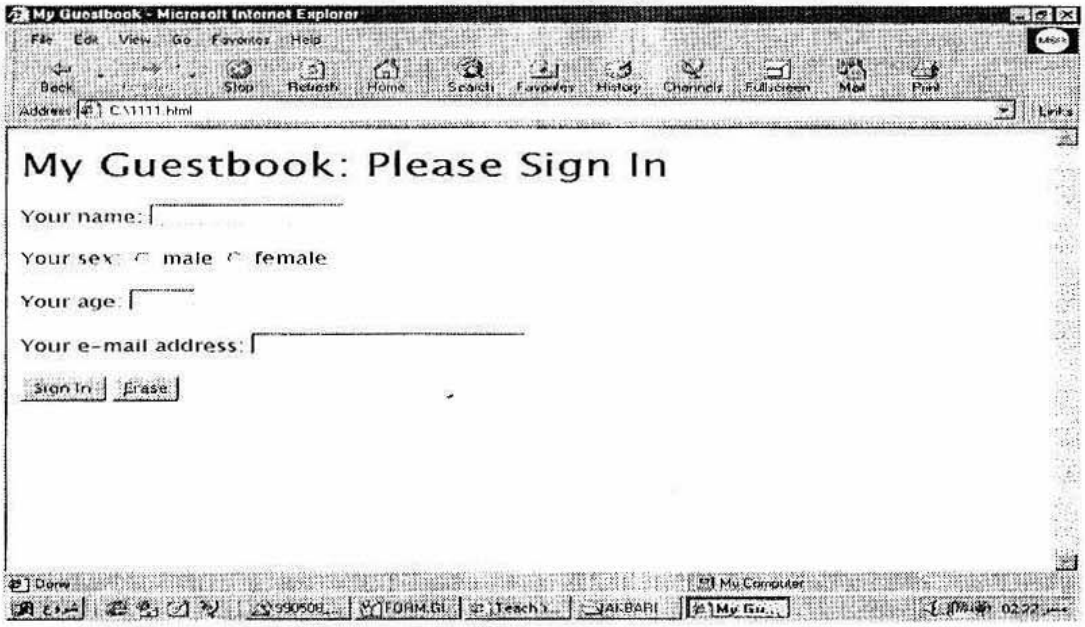
```
<Input   Type = "CheckBox"   Name ="P1"> Black <Br>
< Input   Type = " CheckBox "   Name ="P2"> Red <Br>
< Input   Type = " CheckBox "   Name ="P3">Green <Br>
```

ویرایشگر در HTML

برای ایجاد ویرایشگر از برچسب Textarea استفاده می شود قالب آن بصورت زیر است:

```
<Textarea Name = “Tname”    Rows =”5”   Cols =”6” >
_____
_____
_____
</Textarea>
```

در این Tag ابتدا باید سطر و ستون ویرایشگر و همچنین نام آنرا وارد کنیم. اطلاعـاتی کـه بیـن <Textarea> نوشـته شود داخل Editor قرار خواهند گرفت:



شکل ۳۰ –۱ مثالی از یک form در HTML

برنامهٔ زیر مربوط به شکل بالا می باشد

```
<HTML>
<HEAD>
    <TITLE>My Guestbook</TITLE>
</HEAD>
<BODY>
    <H1>My Guestbook: Please Sign In</H1>
    <FORM METHOD="post" ACTION=http://www.cgi_free.com/cgi/genevic.exe">
        Your name: <INPUT TYPE="text" NAME="name" SIZE=20>
        <P>
         Your sex: <INPUT TYPE="radio" NAME="sex" VALUE="male">
         male <INPUT TYPE="radio" NAME="sex" VALUE="female">
        female<P> Your age: <INPUT TYPE="text" NAME="age" SIZE=4><P>
        Your e-mail address: <INPUT TYPE="text" NAME="email" SIZE=30><P>
         <INPUT TYPE="submit" VALUE="Sign In">
        <INPUT TYPE="reset" VALUE="Erase">
    </FORM>
</BODY>
</HTML>
```

توجه کنید که در ین مثال به جای
 از <B> استفاده شده است.

HTML پویا[1]

HTML پویا DHTML ویژگی است که امکانئ خلق صفحات وبی تعاملی[2] با قابلیتهای چند رسانه ای را ایجاد می کند، برای خلق HTML پویا می توان از زبانهای اسکریپتی مختلفی استفاده نمود که با استفاده از قابلیتهای آنها می توان عناصر صفحه وب اعم از Tag ها، تصاویر، اشیاء، متون، خصایص و شیوه نامه ها (Style Sheet) را تغییر داد.

HTML پویا همچنین از رویدادهای صفحه کلید، ماوس و فوکوس بر روی صفحه حمایت می کند. HTML پویا دستاورد نسبتا جدیدی در عرصهٔ وب است. دشواری کار HTML پویا علاوه بر ضعف قوت مرورگرها، (در حمایت از این تکنولوژی) در عدم انطباق تکنولوژیهای مختلف ارائه شده در این زمینه نهفته است.

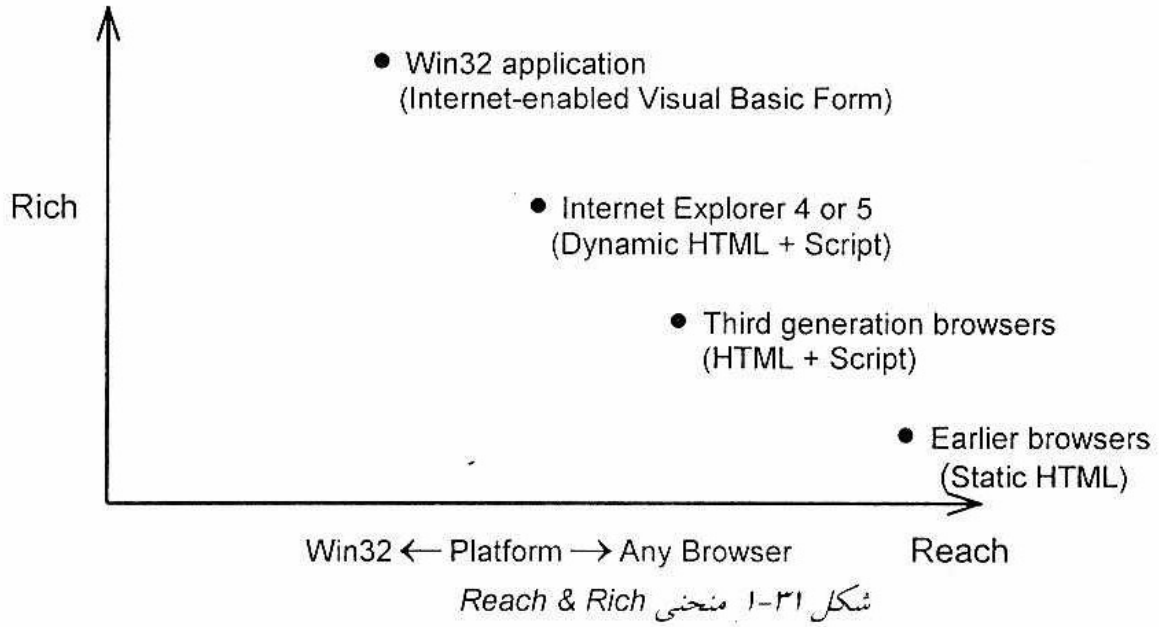
همانطور که در بخش قبل نیز به آن اشاره شد HTML پویا در واقع در جستجوی راهی برای مدیریت تک تک اجزای موجود در صفحه به منظور اعمال پویایی به آنها است. از آنجا که در عرضهٔ تکنولوژی اطلاعات رقابت بسیار شدید است؛ لذا تعریف جامعی از آنچه که DHTML بر آن دلالت می کند وجود ندارد.

بر مطالب فوق سنتی و دیرینه دو غول عرصهٔ مرورگرهای اینترنت یعنی Microsoft و Netscape را نیز باید افزود. رقابت بین این دو بازیگر اصلی، عملا چنان است که این دو شرکت تعمدا روشهای متفاوتی را در تعریف HTML پویا پیش گرفته اند. این اختلاف به حدی شدید است که شرکت Microsoft از نماد DHTML و شرکت Netscape از نماد (dHTML) برای معرفی HTML پویا سود می جویند. دو شرکت فوق با افزودن قابلیتهای خاصی بر مرورگرهای خود بر پیچیدگی مسئله انطباق می افزایند.

از آنجا که در عرصهٔ جهانی 60% از استفاده کنندگان اینترنت از Internet Explorer و 40% از Netscape Navigator استفاده می نمایند (البته این درصد تقریبی است و تعداد استفاده کنندگان از IE در حال افزایش است) لذا اگر کاربردی برای عرصه بر روی اینترنت ایجاد شود به هیچ عنوان نمی توان از مسئله انطباق چشم پوشی نمود. کوشش هایی که برای دستیابی به یک استاندارد معین صورت گرفته است به مدلی به نام DOM منجر گشته است. DOM مخفف (Document Object Model) است که می توان از آن با عنوان (مدل شی ء ای سند) یاد کرد. در واقع نسخهٔ HTML4 مدل DOM و زبانهای اسکریپتی را به عنوان راهکار برای بهبود عملکرد پویای صفحه ارائه نموده است. DOM در واقع طریقه ای است که به کمک آن می توان به تک تک اعضای صفحه به مثابه یک شیء نگریست. سپس می توان به اجزای صفحه با کمک زبانهای اسکریپتی دست یافت و در نهایت با انجام کدنویس های کوتاه در چنین زبانهای پویایی را به صفحه اعمال نمود. ولی از آنجا که حتی در عرصهٔ زبانهای اسکریپتی موجود توسط شرکتهای مختلف (از جمله Microsoft و Netscape) ارائه شده است.

در این بخش صرفا به این مطلب اشاره می گردد که اختلاف دیدگاه بین دو شرکت بزرگ عرصهٔ مرورگر معمولا به این صورت است که شرکت Netscape با افزودن Tag هایی قابلیت HTML استاندارد را افزایش می دهد، در حالی است که Microsoft با افزودن قابلیتهای جدید به Tag های موجود امکانات مرورگر را توسعه می دهد.

نمودار زیر بعد منحنی را معرفی می کند که به منحنی Reach and Rich مشهور است که کاربران و سطوح قدرت دسترسی آنها را مقایسه می کند. این منحنی می تواند به عنوان مبنایی برای مقایسه بین سطوح دسترسی یک کاربرد مفروض و قابلیتهای آن مدنظر قرار گیرد.



شکل ۳۱-۱ منحنی *Reach & Rich*

زبانهای سمت- سرویس گیرنده و سمت- سرویس دهنده[1]

زبانهای سمت- سرویس گیرنده اصطلاحا به زبانهایی گفته می شود که می تواند توسط مرورگر تفسیر و اجرا گردد، وقتی یک برنامه توسط یکی از زبانهای سمت سرویس گیرنده نوشته می شود هنگامی که این برنامه ها به درون مرورگر بار می شوند مرورگر به صورت خودکار برنامه را اجرا می کند.

همچنین زبانهای سمت- سرویس دهنده به زبانهایی گفته می شود که بر روی سرویس دهنده اجرا می گردد. یک زبان سمت- سرویس دهنده تمامی اعمال خود را بر روی سرویس دهنده صورت می دهد. به عنوان مثال Vbscript ، Javascript هر دو می توانند نقش زبانهای سمت سرویس دهنده و سرویس گیرنده را بازی نمایند اما از آنجا که هر دو شرکت Microsoft و Netscape ملزم به حمایت از Java در مرورگرهای خود هستند لذا از Javascript ، Jscript (ویرایش ارائه شده توسط Microsoft از زبان Javascript) غالبا برای برنامه نویسی سمت سرویس گیرنده استفاده می گردد و Vbscript برای برنامه نویسی سمت- سرویس دهنده بکار می رود.

این امر به دلیل آن است که اسکریپتهای نوشته شده توسط زبانهای سمت- سرویس دهنده پیش از ارسال بر روی اینترنت مورد پردازش قرار می گیرند و مرورگر کاربران در عمل چیزی غیر از یک صفحه HTML استاندارد دریافت نمی دارد لذا می توان از آن به عنوان زبان سمت- سرویس دهنده سود جست.

اسکریپتهای سمت سرویس گیرنده

اگر برنامه های اسکریپت را داخل HTML بنویسیم طوری که روی سرویس گیرنده انتقال پیدا کند و توسط مرورگر اجرا شود آنرا اسکریپتهای سمت سرویس گیرنده گوییم که بحث این بخش در این رابطه است.

با زبانهای برنامه نویسی مثال VBScript و Javascript و غیره می توان در داخل HTML برنامه نویسی نمود برچسب مربوطه عبارتند از :

```
< ! -
< Script  language = "     " >

    _____

    _____

    _____

</ Script >
 ->
```

طوری در قسمت Language می تواند زبانهای JavaScript,Vbscript,Jscript ,... استفاده کرد.

توجه :

از آنجا که ممکن است بعضی از مرورگر ها قدیمی <Script> راپشتیبانی نکنند از اینرو در Comment قرار می گیرد.

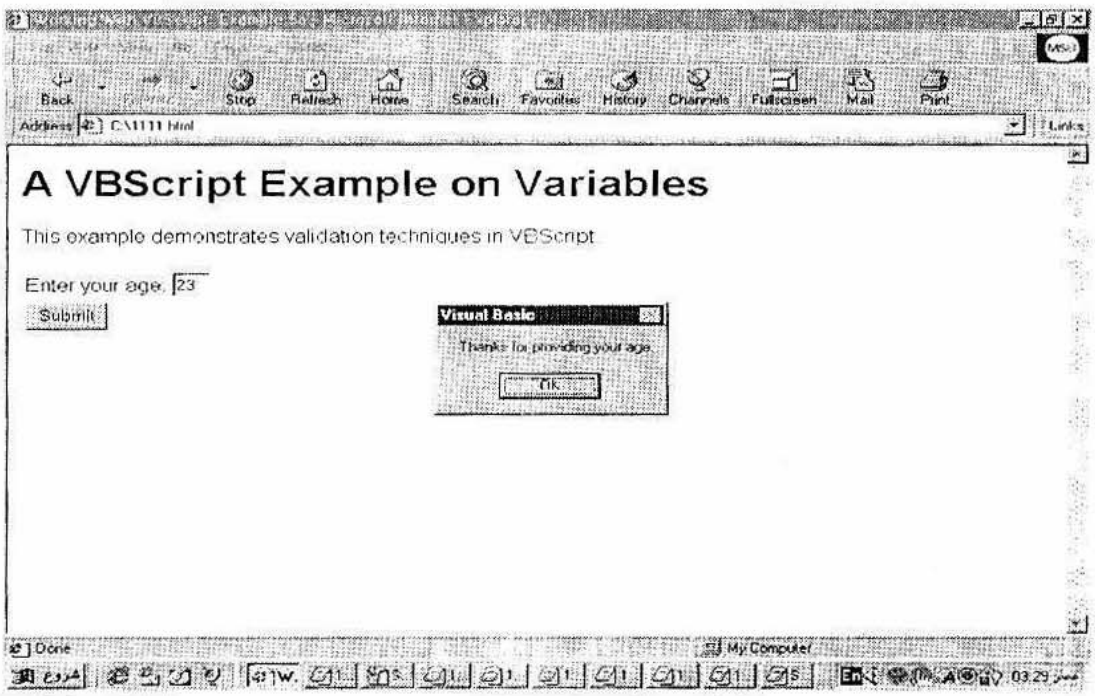
مثال:

```
<HTML>
<head>
      <Title>adder </title>

      <Script  language ="vbscript">
            SUB  add()
            x=2
            y=5
            msgbox  X+Y
            END SUB
      </Script>
</head>
 <body>
     <input  type="button"   value="compute" Onclick="add()" >
 </body>
</HTML >
```

مثال: زیر مربوط به برنامهٔ Vbscript می باشد که به ازای ورود سن مشخص اگر اطلاعات درسی وارد نشده باشد پیغام خطا خواهد داد.

```
<HTML>
<HEAD>
<TITLE>Working With VBScript:</TITLE>
<SCRIPT LANGUAGE="VBScript">
Sub cmdSubmit_OnClick
  If (Len(document.frmExample5a.txtAge.value) = 0) Then
    MsgBox "You must enter your age before submitting."
    Exit Sub
  End If
' Check to see if the user entered a number.
  If (Not(IsNumeric(document.frmExample5a.txtAge.value))) Then
    MsgBox "You must enter a number for your age."
    Exit Sub
  End If
' Check to see if the age entered is valid.
  If (document.frmExample5a.txtAge.value < 0) Or (document.frmExample5a.txtAge.value >
100) Then
    MsgBox "The age you entered is invalid."
    Exit Sub
  End If
  MsgBox "Thanks for providing your age."
End Sub
-->
</SCRIPT>
</HEAD>
<BODY>
<H1>A VBScript Example on Variables</H1>
<P> This example demonstrates validation techniques in VBScript. </P>
<FORM NAME="frmExample5a">
  <TABLE>
    <TR>
      <TD>Enter your age:</TD>
      <TD><INPUT TYPE="Text" NAME="txtAge" SIZE="2">
    <TR>
```

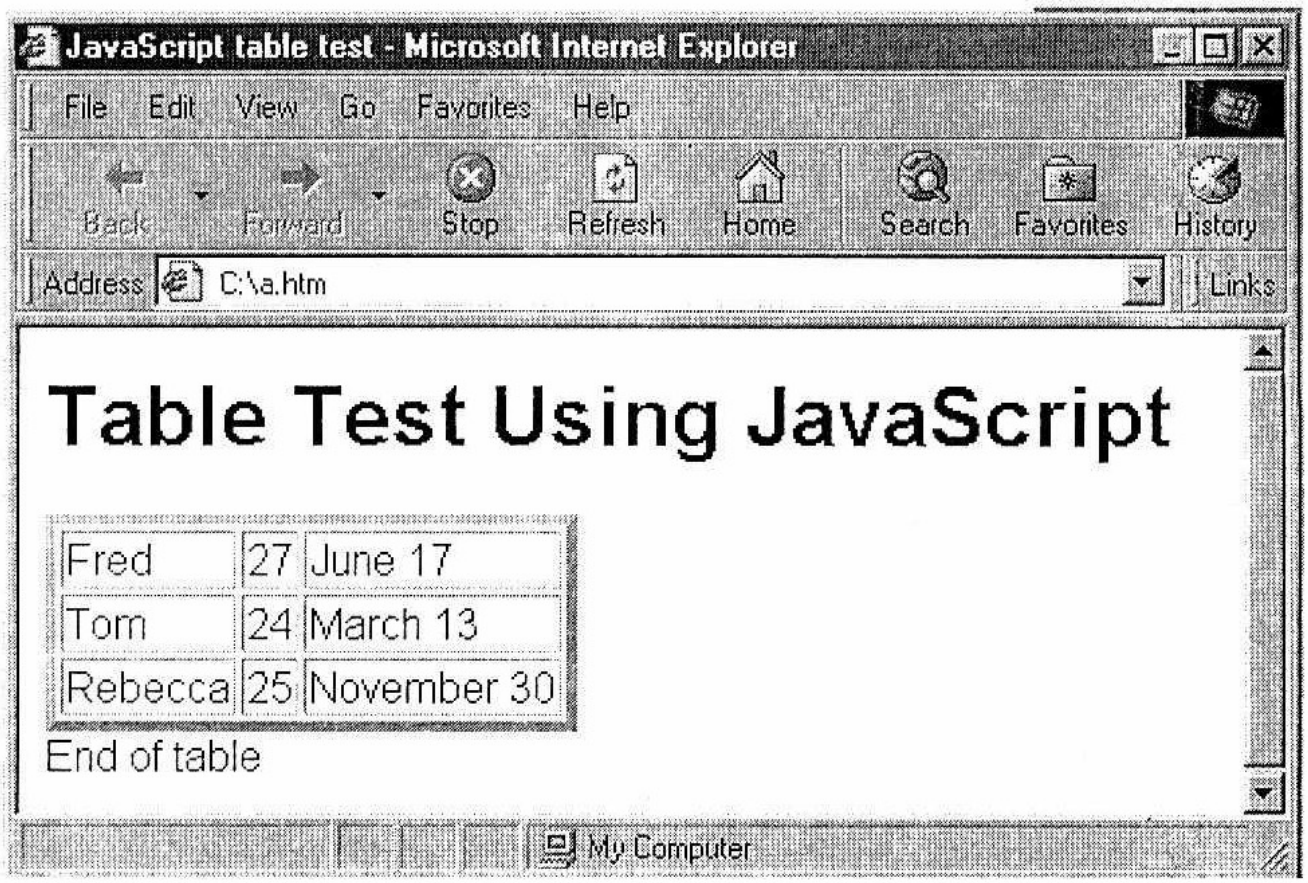
```
    <TD><INPUT TYPE="Button" NAME="cmdSubmit" VALUE="Submit"></TD>
    <TD></TD>
</TR>
</TABLE>
</FORM>
</BODY>
</HTML>
```



شکل ۳۲-۱ اجرای برنامهٔ VBSCRIPTقبل

مثالی از یک برنامه Javascript

```
<HTML>
<HEAD>
  <TITLE>JavaScript table test </TITLE>
  <SCRIPT Language = "JAVASCRIPT">
     function printRow(name,age,birthday) {
     document.write("<TR><TD>", name,"</TD><TD>",age,"</TD><TD>",
      birthday,"</TD></TR>\n");
     }
  </SCRIPT>
  </HEAD >
  <BODY>
    <H1> Table Test Using JavaScript </H1>
    <TABLE border=4>
    <SCRIPT LANGUAGE ="JAVASCRIPT">
    printRow ( "Fred" , 27 , "June 17" );
    printRow ( "Tom" , 24 , "March 13");
    printRow ( "Rebecca" , 25 , "November 30 " );
   </SCRIPT>
   </TABLE>
    End of table
  </Body>
</HTML>
```



شکل ۳۳-۱ اجرای برنامهٔ Javascript

در این برنامه یک جدول رسم خواهد شد طوری که برای رسم هر خط از این جدول یک فراخوانی به تابع PrintRow که به زبان جاوا نوشته انجام می شود.

متد Document . Write همانند( ) Printf در C می باشد و از طریق آن یک عبارت درخروجی چاپ می شود.

۱۴-۱ Cgi (Common Gateway Interface)

Cgi یک استاندارد برای برقراری ارتباط بین سند وب و برنامهٔ دروازه ای[1] روی سرویس دهنده است. Cgi روشی است که برنامه های خارجی می توانند با سرویس دهنده ارتباط برقرار کنند.

کارهایی که Cgi می تواند انجام دهد:

HTML	CGI+HTML	Task
No	Yes	Handle form
No	Yes	ایجاد هر چیز غیرStatic
No	Yes	Searching
No	Yes	ایجاد برنامه های کاربردی

برای استفاده از Cgi ابتدا باید سرویس دهنده وب را آماده نمود اگر سرویس دهندهٔ وب درست نصب شده باشد یک زیر شاخه با نام

Inetpub\Cgi-bin \ ...

ایجاد نموده است این زیر شاخه مربوط به فایلهای Cgi می باشد.

اگر از روش Cgi جهت برنامه نویسی تحت Web استفاده کنید برنامهٔ شما دو بخش دارد :

HTML
Gateway Program

اجزای یک برنامهٔ CGI

۱- برنامهٔ HTML

این بخش اکثر اوقات یک form بوده و یا از روش SSI استفاده می کند (این برنامه روی سرویس گیرنده توسط مرورگر اجرا می شود)

۲- برنامهٔ دروازه ای

این برنامه یک برنامهٔ اجرایی (.EXE) است که می تواند با هر زبانی نوشته شود و در زیر شاخهٔ Cgi-bin یا Scripts قرار دارد. (روی سرویس دهنده اجرا می شود)

عملکرد Cgi روی Server

برنامه های CGI از نوع Event oriented (واقعه گرا) هستند به این معنی که هنگامی که یک سند به آنها دسترسی پیدا می کند شروع می شوند.

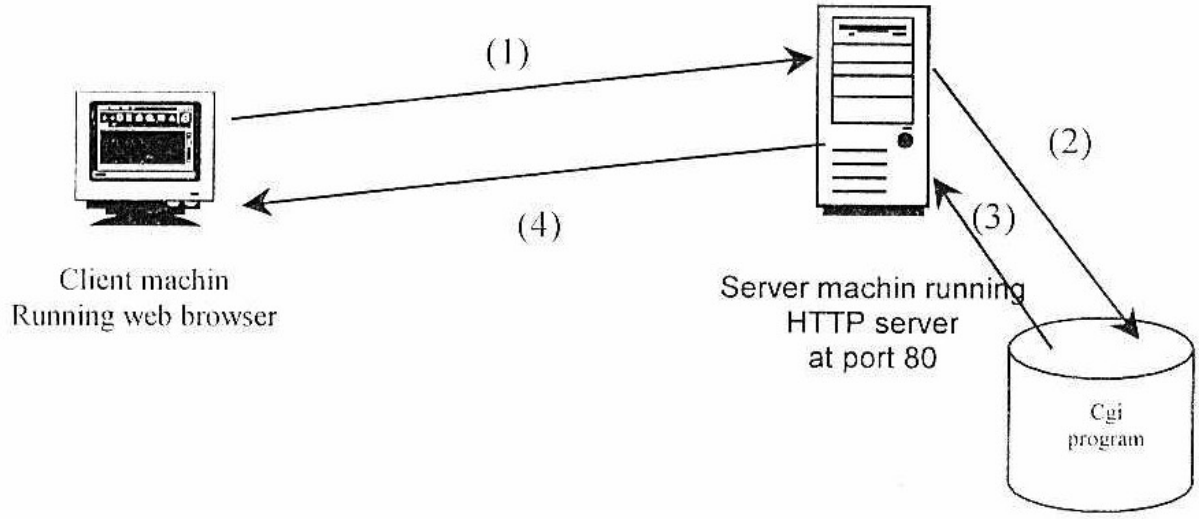
هنگامی که یک درخواست به برنامهٔ دروازه از طریق SSI یا زدن کلید Submit در یک form ارسال می شود. سروس دهندهٔ وب بلافاصله به زیر شاخهٔ موجود در سیستم خود (Cgi-bin) مراجعه کرده و به دنبال برنامهٔ دروازه ای که آدرس آن در " "= Action مشخص شده است می گردد و اگر یافت شود یک Process ایجاد شده و یک ناحیهٔ کاری و حافظه به آن اختصاص یافته و فعال می گردد و پارامترها و متغیرهای محیطی آن Set می شوند. برنامه اجرا شده و نتیجهٔ آن که یک HTML است به مرورگر بر می گردد.

توجه :

برنامه های CGI نیاز به فراخوانی های خاصی به سیستم عامل سرویس دهنده دارند از این نظر باید روی سیستم عامل خاصی که بکار می برید تبحر حاصل کنید. اولین برنامه های Cgi با Perl و C روی Unix نوشته شدند.

کاربردهای برنامه های CGI

- *گرفتن اطلاعات از افراد ملاقات کنندهٔ سایت و ذخیرهٔ داده ها در یک فایل*
- *فرستادن اطلاعات از طریق فرمها و ارسال آنها به یک آدرس خاص از طریق e-mail*
- *بروز در آوردن فایلهای محلی روی سرویس دهنده*
- *تولید اطلاعات خودکار بصورت گرافیکی یا متنی*
- *ایجاد سطح امنیت*
- *تولید گرافیک یا تغییر فایلهای گرافیکی بطور خودکار*



شکل ۳٤-۱ عملکرد برنامه های Cgi

ارتباط برنامه CGI با HTML از طریق form

این کار از طریق form قابل انجام است در این صورت حتما باید یک از کلید Submit وجود داشته باشد که با زدن آن اطلاعات به برنامه دروازه ای ارسال گردد.

```
<HTML>
<BODY>
< form action =" http://www.cgi-free.com/cgi-bin/cgiprogram.cgi "    method =" post " >
What is your Message: <Textarea name =" name">     < / Textarea >
< Input type ="submit"  Value = "send"> < / Input >
< Input type ="Reset"   Value = "Reset"> </ Input >
</Form >
</ Body>
</ HTML>
```

روش SSI (Server side include)

یکی از روشهای فراخوانی برنامه های دروازه ای همانطور که گفتیم از طریق form بود طوری که در form حتماً باید کلید Submit باشد که با زدن آن اطلاعات به برنامهٔ دروازه ای ارسال شود بعضی اوقات لازم است بدون زدن هیچ کلیدی با دیده شدن صفحه در مرورگر درخواستی به برنامهٔ دروازه ای ارسال گردد. این روش SSI می باشد.

در SSI از دستور EXEC #در HTML استفاده می شود که بصورت زیر استفاده می شود. البته می توان در مرورگرهایی که EXEC#را پشتیبانی نمی کند. از دستور IMG نیز استفاده کرد و طوری که در آرگومان SRC در IMG نام برنامه دروازه ای را می نویسیم:

```
<HTML>
<HEAD>
<BODY>
<TITLE> Counter program page </ TITLE>
</ HEAD>
```

```
See How Many hits this page has taken :
<!# EXEC Cgi="Test.Cgi" >
<HR >
</Body>
</ HTML>
```

بلافاصله با View شدن این صفحه توسط مرورگر یک درخواست ساخته شده و به برنامهٔ دروازه ای که آدرس آن مشخص شده است ارسال خواهد شد.

CGI Methods

متدهای ارسال اطلاعات به برنامهٔ دروازه ای روی سرویس دهنده چندین روش دارد که دو تا از مهمترین این روشها عبارتند از:

Get

در این حالت برنامهٔ CGI داده ها را از متغیر محیطی[1] QUERY-STRING دریافت خواهد کرد. حال برنامهٔ دروازه ای باید رشته را پویش[2] کرده و داده ها را بیرون بکشد.

در این روش محدودیت ۱۰۲٤ بایت (محدودیت URL) داریم زیرا اطلاعات ارسالی نمی تواند بیشتر از ۱۰۲٤ بایت باشد. (می توان این متغیر محیطی را باتابع ( ) getenv در C گرفت و پویش کرد تا داده ها بدست آیند)

Post

در این حالت سرویس دهنده وب داده ها را از طریق ورودی استاندارد یا stdin[3] به برنامه CGI انتقال می دهد. سرویس دهنده می تواند پایان داده ها را با یک کاراکتر EOF علامتگذاری کند بنابراین برنامه باید از یک مقدار CONTENT-LENGTE برای خواندن ورودی استاندارد بصورت درست استفاده کند. اگر شما بخواهید داده های بیشتر از ۱۰۲٤ بایت را ارسال کنید باید ازاین روش استفاده کنید.

متغیرهای محیطی

هر سرویس دهندهٔ وب دارای تعدادی متغیر خاص می باشد که هنگامی که برنامه کاربردی CGI توسط سرویس دهندهٔ وب فعال می گردد این متغیرها مقداردهی می شوند بعضی از این متغیرها عبارتند از :

QUERY - STRING

اطلاعات ارسالی از مرورگر به برنامهٔ دروازه ای اگر از روش Getاستفاده شده باشد در این متغیر قرار خواهد گرفت. این متغیردر روش Post خالی می باشد اطلاعات با علامت ؟ شروع می شوند که بعد از URL قرار می گیرد.

CONTENT-LENGTH

طول اطلاعات ارسالی به برنامهٔ CGI می باشد (برحسب بایت) ، که در حالت Post مقدا ردهی مـی شـود اگر از روش Get استفاده شده باشد این متغیر خالی است.

CONTENT-TYPE

نوع یا Type اطلاعات ارسالی از سرویس دهندهٔ وب به مرورگر است که مربـوط بـه MIME اسـت (هـر گاه در برنامهٔ دروازه ای بخواهید اطلاعاتی را به مرورگر ارسال نمایید ابتـدا بـاید CONTENT-TYPE را ارسـال کنیـد مـی تواند از نوع زیر باشد.)

Msword Zip pdf	application
Midi wav	audio
Aiff Gif Jpeg bmp	image
Plain Html	text
Mpeg Quick time	video

مثال :

```
Print f( "CONTENT-TYPE :    Text / plain)
```

این اطلاعات به مرورگر ارسال می شود، مرورگر خود را آمادهٔ دریافت این فرمت فایل خواهد کرد.

* این سرآیند همواره از سرویس دهندهٔ وب به مرورگر ارسال می شود.

PATH-INFO

اطلاعات اضافی Path شامل نام برنامه CGI و URL آن است که برنامهٔ دروازه ای می تواند از آن استفاده کند.

REMOTE-HOST، REMOTE–ADDR

آدرس IP کـامپیوتری کـه درخواسـت ارسـال کـرده اسـت در REMOTE-ADDR قـرار دارد و آدرس IP کامپیوتری که به آن درخواست ارسال شده در REMOTE-HOST قرار دارد.

REMOTE-IDENT

نام کاربر (User name)را با استاندارد RFC931 می دهد.

REQUEST–METHOD

روش ارسال اطلاعات را می دهد که یا Get است یا Post از طریـق ایـن متغیر محیطـی مـی تـوان برنامـه هـایی نوشت که با هر دو روش کارکنند.

SERVER-NAME

نام حوزه یا آدرس IP سرویس دهنده است.

SERVER-PORT

شمارهٔ پورت مربوط به سرویس دهندهٔ وب که معمولاً ۸۰ است.

SERVER-PROTOCOL

نام و نسخهٔ پروتکل سرویس دهندهٔ وب می باشد مثلاً (http / 1.0).

SERVER-SOFTWARE

نام سرویس دهندهٔ وب که برنامه CGI را اجرا می کند مثلاً (NCSA / 1.5bs).

همچنین کامپیوترهای سرویس گیرنده ممکن است سرآیندهای HTTP را به برنامه CGI ارسال کنند این متغیرها با <HTTP> شروع می شوند که گروهی از آنها در زیر بیان شده است.

HTTP - ACCEPT

شامل لیستی از انواع فایلهایی که مرورگر می تواند از سرویس دهندهٔ وب دریافت کند :

/audio / basic , image / gif , text / ** /"

HTTP-USER–AGENT

نام مرورگری است که به سند دسترسی پیدا کرده است مثلاً Netscape 2.0 یا Internet Explorer 5

توجه: در زبان C ازدستور ( )getenv جهت گرفتن متغیرهای محیطی از سیستم عامل استفاده می شود.

```
Strcpy (str , getenv ("HTTP-USER-AGENT "));
```

زبانهای برنامه نویسی سرویس دهنده

زبانی که انتخاب می کنید دارای محدودیتهایی می باشد که به سرویس دهنده بستگی دارد (انتخاب سرویس دهنده مهم تر است) اغلب برنامه های کاربردی سرویس دهندهٔ وب در اینترنت با یکی از سه زبان Perl ، ویژوال بیسیک و C نوشته می شوند.

(Practical Extraction language and Reporty) Perl

اگر از سیستم عامل دیگری غیر از Unix بخواهید استفاده کنید می توانید برنامه را روی سیستم دیگر نوشت و با FTP آنرا به Unix ارسال کرد (همچنین از Telnet نیزمی توان استفاده کرد)Perl زبان تفسیری است (کامپایلر ندارد) حلقهٔ ویرایش، کامپایل و آزمایش سریعتر می شود.اولین خط اشاره گر به مفسر خودش است و ابتدا مفسر را به حافظه آورده و سپس برنامه را اجرا می کند.

Perl برای کارهای کوچک بهترین است ولی برای برنامه های بزرگتر چندان خوب عمل نمی کند.

C و C++

C++ از C متولد شده و یک مرحله بالاتر از C است C و C++ رویَ خیلی از سیستم ها از Pc گرفته تا Mac و Unix قابل انتقال است و نیاز به کامپایل دارند. وحافظه کمتری را نسبت به Perl مصرف می کند.

برنامه نوشتن با C دشوار است زیرا کنترل همهٔ جنبه های برنامه نویسی و سیستم برعهدهٔ برنامه نویس است. از نظر اجرا نیز یک برنامه کامپایل شده مطمئنتر از Perl است.

ویژوال بیسیک (Visual Basic)

بیسیک در اواسط سال ۱۹۶۰ در کالج Portmouth جهت تعلیم و آموزش تولید شد، ولی ویـژال بیسـیک چنـد سال بعد در سال ۱۹۹۱ ایجاد شد که اساس کار آن بر پایه[1] است. از طریق OLE[2] ، VB می تواند با برنامه هـای دیگـر مثل EXEL یا ACCESS ارتباط داشته باشند.

البته برنامه نویسی VB برای اینترنت تقریبا[3] غیر ویژوال است ولی VB توانایی آنرا دارد.

جهت مقایسهٔ این سه زبان می توان گفت: Perl بسیار سازگار با Unix است ودر بکار گیری حافظه کـارایی نـدارد و صلاحیت کمی در پشتیبانی از سیستمهای مختلف دارد.

C++زبان قابل حمل روی سیستمهای مختلف بوده ولی نوشتن اشتباهات کوچک در برنامه نویسـی بـاعث خطاهـای زیاد می شود.

VB نیزاز OLE استفاده می کند ولی مربوط به محصولات مایکروسافت است.

مثال :

برنامه ای با Cgi بنویسید که تعداد دفعات ملاقات یک صفحه وب را بشمارد (متد get باشد) بـه ایـن برنامـه Hit counter گویند.

الف) بخش HTML

```
<HTML>
<BODY>
  HitNumber of : <BR>
  <HR>
  < IMG SRL = "http://www.sample. com/cgi _bin / sample.cgi ">
</BODY>
</HTML >
```

ب) بخش Serverبرنامهٔ دروازه ای

```
#include <stdio.h >
#include <stdlib.h >
 main( )
{
    int  k ;
    fILE  * F ;
    f = Fopen ( " hit . dat " , "a" );
    fscanf ( f , "%d" , &k );
    printf ( " Content _type : Text / htm" );
    printf ( " <B> %d </ B> " , K+1);
    fseek ( f , 0); // به اول فایل پرش می کند
    fprintf ( f , " %d" , k+1 );
    fclose ( f );
}
```

توجه: خروجی برنامه های دروازه ای روی مانیتور سرویس دهنده نمایش داده نمی شود بلکه از طریق سرویس دهندهٔ HTTP به مرورگر ارسال می شوند.

برنامهٔ Chat با Cgi

از Chat برای صحبت کاربران روی اینترنت استفاده می شود. کافی است یک فایل داشته باشیم که کاربران اطلاعات به آن اضافه کنند.

ابتدا هنگام نمایش این HTML یک درخواست به برنامهٔ دروازه ای ارسال شده تا فایل اطلاعات وارد شده توسط کاربران روی مرورگر نشان داده شود و به ازای زدن کلید Send نیز دوباره اطلاعات به این فایل اضافه خواهد شد.

بخش HTML

```
<HTML>
<BODY>
<IMG SRC ="http:\\www.sample.com\chat1.cgi">
<Form method ="get"Action="http:\\www.sample.com\Cgi-bin\Chat2.cgi">
<Input Type ="Text "NAME="T1">
<Input Type ="Submit "NAME="S1"Value="send">
</ Form>
</BODY>
</ HTML >
```

برنامهٔ دروازه ای

بخش برنامهٔ دروازه ای Chat دارای دو بخش است Chat1 که وظیفهٔ خواندن اطلاعات از فایل را برعهده دارد Chat2 که وظیفهٔ اضافه کردن یک خط به فایل را انجام می دهد.

متن این دو برنامه بصورت زیر است:

```
//CHAT 1
#include <stdlio.h>
#include <Stdio.h>
Main ( )
{
char c , Str[100];
FILE *f;
f=Fopen ("Chat.Text", "r");
fseek (f,hile size(f)-100);
fread(f,100 Str);
print f ("Content-Type: Text/Html");
print f ("%S",Str);
fclose (f);
}
//chat 2
#include<stdlio.h>
#include<stdlib.h>
main ( ){
FILE  *F;
char str [100];
strcpy (str,getenv ("QUERY-STRING"));
f=fopen("chat.txt","a");
```

```
f printf(f,"%s",str);
fclose(f);
}
```

مثال: برنامه ای بنویسید که اطلاعات دانشجو را در یک فایل ثبت کند (هر دو متد را پشتیبانی کند)

- بخش HTML

```
<HTML>
<BODY>
< Form  method = " post " Action ="http://wwwsample.com/cgi-bin/student.cgi">
Name : <Input  Name = "N">  <BR>
Family : <Input  Name = "F ">  <BR>
Stdn : <Input Name = " STU ">  <BR>
<Center>
< Input Type = " Submit " Value = " send " Name = " Bt1" >
</Center>
</Form>
</ Body>
</Html >
```

- بخش برنامهٔ دروازه ای

```
#include <stdio . h >

#include <stdlib . h>

Main( )

{ int  n,i;

Char   str[100] , Method[10] , Name=[15] , Family[20] , Number[11];
 FILE *f ;
Strcpy ( method , getenv ( "REQUEST-METHOD"));
If ! strcmp ( method , " post ")
{
strcpy (num , getenv ("CONTENT-LENGTH"));
n= atoi (num);
for (i=0;i<n;i++)
str=getche ( );
str= " \0";
}
else
   Strcpy ( Str , getenv ( "QUERY-STRING" ));
Parser ( Str , & Name , & Family , & Number ");
f= fopen (" Student.dat" ، "a" );

fprintf (f ، "%s%s%s " ، Name ، Family ،Number" );
fclose ( );
}
```

فرض می شود تابع Parser اطلاعات داخل Str را پویش کرده و نام و نام خانوادگی و شماره را بیرون کشیده در متغیرهای بالاتر قرار می دهد.

۱-۱۵ برنامه نویسی به زبان جاوا

جاوا زبان برنامه نویسی است که توسط شرکت Sun Microsystem ارائه شده و شباهتهای بسیاری به زبان C++ دارد. این زبان برنامه نویسی را می توان در بازه متنوعی از کاربردهای وب و مهمتر از همه در کاربردهای توزیع شده بکار گرفت.

برنامه های کوچک نوشته شده به زبان جاوا، که اصطلاحاً اپلت[۳] نامیده می شوند. را می توان درون صفحات HTML قرار داد. بدین ترتیب این برنامه ها همراه صفحات HTML به ماشین کاربر منتقل شده به اجرا در می آیند.

شرکت SUN جهت عمومیت دادن جاوا Source کامپایلر آنرا بصورت رایگان در اختیار شرکتهای مختلف خصوصاً شرکتهای نویسندهٔ مرورگر قرار داد این باعث شد که بسیار عمومیت پیدا کند طوری که به عنوان یک زبان برنامه نویسی باز (Open) معرفی شود.

با استفاده از جاوا می توان کاربردهای توزیع شده وب بوجود آورده، چرا که منطق تجاری را می توان در سمت کاربر به اجرا در آورد.همچنین دسترسی به پایگاههای داده به طور مستقیم توسط مرورگر امکان پذیر است. (از طریق روشهایی چون RMI).

استانداردی به نام JDBC ایجاد شده است که امکان دسترسی اپلتهای جاوا به پایگاههای داده رابطه ای را فراهم می سازد. با این وجود جاوا برای کاربردهای بزرگ مناسب نیست ولی می توان حداقل بعضی از قسمتهای کاربر، را با استفاده از جاوا پیاده سازی کرد (مثلاً کاربردهای موجود را به وب متصل ساخت).

برنامه نویسی با جاوا به دو صورت انجام می گیرد :

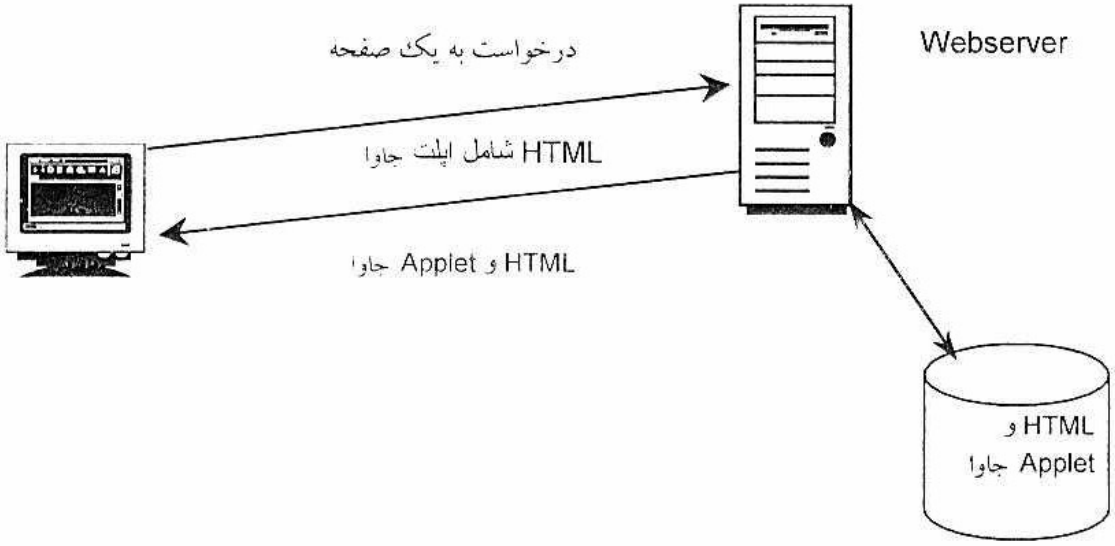
۱) برنامه های کاربردی (Application)

۲) اپلتها

برنامه های کاربردی که بصورت EXE هستند بعد از ترجمهٔ برنامهٔ جاوا ایجاد می شوند جهت ایجاد آنها از کامپایلر جاوا استفاده می شود شکل ترجمهٔ یک برنامهٔ جاوا در سیستمهای Unix بصورت زیر است:

```
$ Javac hello.java
```

ولی خروجی یک اپلت بعد از کامپایل توسط کامپایلر یک Bytecode می شود که این کد روی هر ماشینی که در آن ماشین مجازی جاوا[۴] موجود باشد قابل اجرا است این برنامه زبان ماشین نیست این خاصیت باعث قابل حمل شدن اپلتهای جاوا می شود.



شکل ۳۵-۱ ارتباط اپلتهای جاوا با سرویس دهندهٔ وب

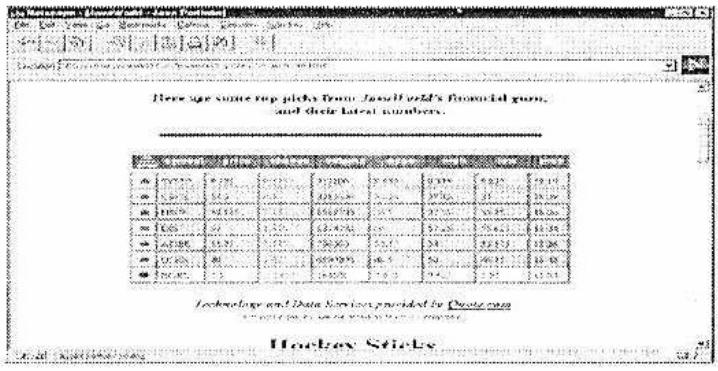
استفاده از اپلتها در HTML

اپلتها نیز مانند کنترلهای Activex می توانند هم روی سرویس دهنده اجرا شوند و هم روی سرویس گیرنده . ولی بحث ما در مورد اپلت سمت سرویس گیرنده است.

```
< HTML >
< Body >
<  Applet        Codebase="Http://www.Applet.Com/Hello.Cass " Width = " 500 " Height  =
" 500" >
< Param        Name = " Ver "          Value = " 3.1 " >

< / Applet >< / Body >< / HTML >
```

خصوصیتی که اپلتها دارند این می باشد که برای دیدن یک صفحه شامل اپلت برنامه آن روی سرویس گیرنده انتقال می یابد حتی اگر قبلاً انتقال یافته باشد.

در Activex های سمت سرویس گیرنده که بعداً توضیح خواهد شد نیز بدین صورت است با این تفاوت که فقط یکبار عمل کپی انجام می گیرد. در مورد کنترولهای Activex بعداً شرح داده خواهد شد.



شکل۳۶-۱ اجرای یک اپلت جاوا

طریقهٔ نوشتن اپلتهای جاوا

جهت نوشتن یک زیربرنامه باید از کلاس اپلت در جاوا یک زیر کلاس به ارث ببرید.

```
Public  Class  Myclass  Extends  Java . applet . Applet
{
   _____ Properties
   _____ Methods
   _____ Program
}
```

توجه :

نام زیرکلاس (Myclass) باید با نام برنامه (Myclass.Java) یکی باشد. که بعد از کامپایل فایل Myclass.class ایجاد شود.

کلاس اپلت در جاوا دارای تعدادی متد می باشد که این متدها به ترتیب اجرا می شوند حال اگر شما متدها را دوباره تعریف کنید از متدهای شما استفاده خواهد شد. در غیر این صورت از همان متدهای پیش فرض استفاده می گردد.

این متدها عبارتند از :

۱- init ( )

جهت مقدار دهی اولیه می باشد که دارای خصوصیات زیر است.

- فقط یکبار اجرا می شود.
- متغیرها را در این قسمت تعریف و مقداردهی اولیه می کنیم.
- برای گرفتن پارامترها از برنامه HTML از این قسمت استفاده می شود این کار با استفاده از دستور( )Get parameter انجام می شود.

۲- Start ( )

این متد بصورت خودکار بعد از init اجرا می شود و بدنهٔ اصلی در این قسمت نوشته می شود همچنین هنگامی که اپلت متوقف شده باشد و دوباره فعال شود این متد اجرا خواهد شد.

۳- Paint ( )

این متد جهت قرار دادن اطلاعات روی ناحیه ای از مرورگر که مشخص شده است بکار می رود. ورودی این متد یک نمونه از شیء Graphicsمی باشد.

```
Paint ( Graphics g )
```

حال می توان از متدهای شیء Graphics برای قرار دادن اطلاعات روی صفحه استفاده کرد.

٤- Stop ( )

این متد اجرای اپلت را متوقف می کند ولی منابع آنرا آزاد نمی کند قبل از Destray کردن یک اپلت همواره باید آنرا متوقف کرد (Stop).

۵-( ) Destory

این تابع با بستن پنجرهٔ مرورگری که اپلت،را اجرا می کند رخ خواهـد داد، طـوری کـه حافظـهٔ اختصـاص داده شده، ProcessTime و Swap Disk Space و سایر منابع آزاد می گردد.

مثال :

```
1: import java.awt.*;
2:
3: public class ColorCycle extends java.applet.Applet {
4:     float hue = (float).5;
5:     float saturation = (float)1;
6:     float brightness = (float)0;
7:     Button b;
8:
9:     public void init() {
10:        b = new Button("Next Color");
11:        add(b);
12:    }
13:
14:    public void start() {
15:        setBackground(Color.black);
16:        repaint();
17:    }
18:
19:    public boolean action(Event evt, Object o) {
20:        if (brightness < 1)
21:            brightness += .25;
22:        else
23:            brightness = 0;
24:        Color c = new Color(Color.HSBtoRGB(hue, saturation, brightness));
25:        setBackground(c);
26:        repaint();
27:        return true;
28:    }
29: }
```

کد مربوط به بخش HTML

```
1: <html>
2: <body>
3: <applet code=ColorCycle.class height=250 width=250>
4: </applet>
5: </body>
6: </html>
```

۱-۱۶ ISAPI (Internet Server Application Programming)

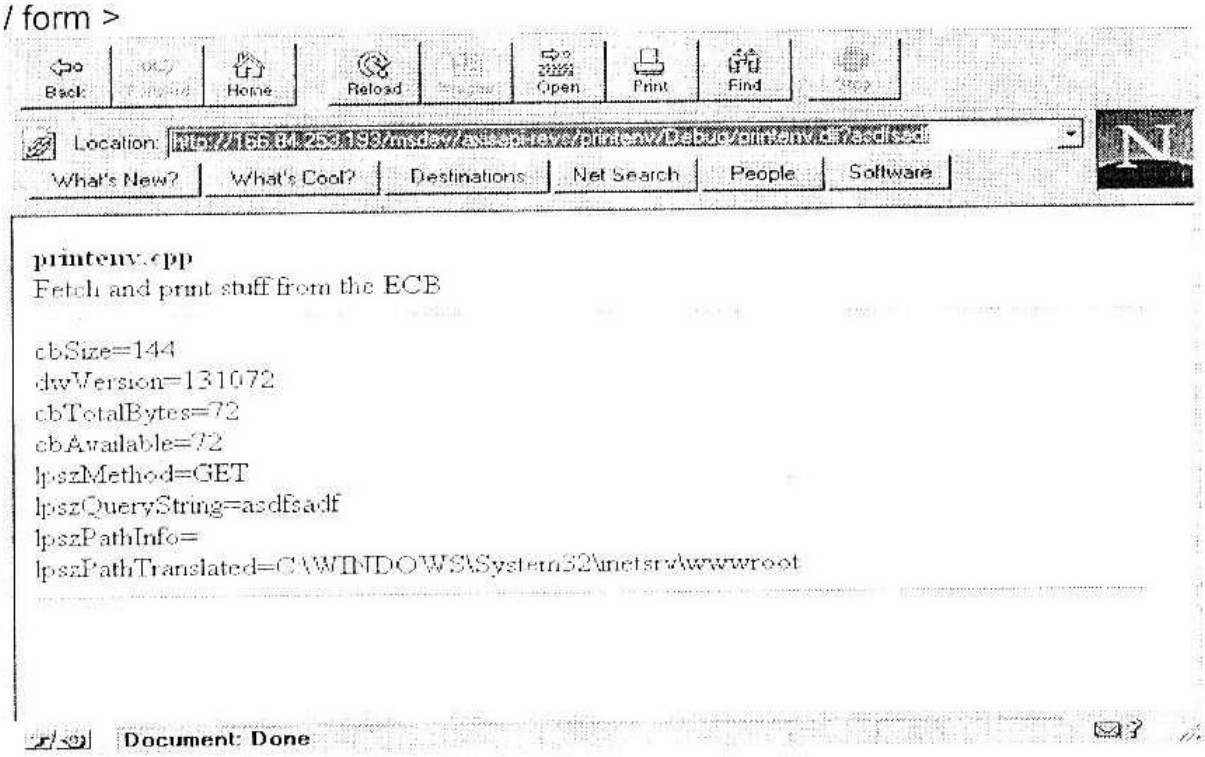
یک روش برنامه نویسی برای سرویس دهنده مایکروسافت می باشد دقیقاً مانند Cgi می باشد. یعنی شما یک HTML و یک برنامه دروازه ای دارید. که در HTML از طریق یک Form یا SSI آنرا فراخوانی می کنید. ISAPI هم مثل Cgi است با این تفاوت که در Cgi برنامه های دروازه ای ExE هستند. ولی در ISAPI ، DLL می باشند.

- به ازای یک درخواست از سرویس گیرنده به یک برنامه دروازه ای Cgi یک پردازش روی سرویس دهنده با یک ناحیهٔ کاری جدا و حافظهٔ جدا فعال می گردد و این کار با درخواستهای بعدی مرتب تکرار می شود.

- در برنامه های دروازه ای ISAPI که از نوع Dll هستند فقط یکبار برنامه در حافظه بار می شود. به ازای هر درخواست فقط یک نخ[۵] از آن در حافظه فعال خواهد شد.

مثال :

```
< Form  Method = " Post " Action = " http://www.sample.com/cgi-bin/hello.dll">
    .
    .
    .
< / form >
```



شکل ۱-۳۷ مثالی از یک فراخوانی ISAPI

۱-۱۷ Activex

برنامه نویسی ماژولار سالها برنامه نویسان را به خود مشغول داشته بود لذا محرک اصلی که به توسعه سیستم عامل ویندوز منجر شده نیز کدهای قابل اشتراک و قابل استفاده مجدد بود. اولین گام در راه پیاده سازی عناصر ماژولار تکنولوژی OLE[۶] بود. هدف اولیه OLE ایجاد مستندات مرکب[۷] با استفاده از برنامه های مختلف بود. به عنوان مثال

سندی که مقداری متن و نمودار دارد که هر کدام با نرم افزارهای خاص خود ایجاد شده اند، نمونه ای از یک سند مرکب است. در چنین مثالی هنگامی که با متن کار می شود نرم افزار واژه پرداز کنترل را به دست می گیرد و هنگامی که با نمودارها کار صورت می گیرد نرم افزار ترسیمی مسئولیت را بعهده می گیرد. ایراد اصلی OLE سرعت کم آن بود.

تکنولوژی OLE بر مبنای استاندارد کلی تری به نام COM (Component Object Model) بنا شده بود. با پیشرفت تکنولوژی، تکنولوژی COM نیز توسعه یافت و از سطح سندهای مرکب فراتر رفت. و DCOM پدید آمد در آن هنگام OLE به اصطلاحی برای هر آنچه از تکنولوژی COM استفاده می کرد بدل شد. ویرایش های قدیمی Office ، Visual basic نیز دربردارندهٔ کنترلهایی بودند که کنترل های OLE نامیده می شدند.

امروزه به بخشی از تکنولوژیهای COM که در آنها یک قطعه نرم افزاری امکانات خود را در اختیار برنامه های دیگر گذارد و یا توانایی پشتیبانی از نرم افزارهای توزیع شده را داشته باشد Activex گفته می شود.

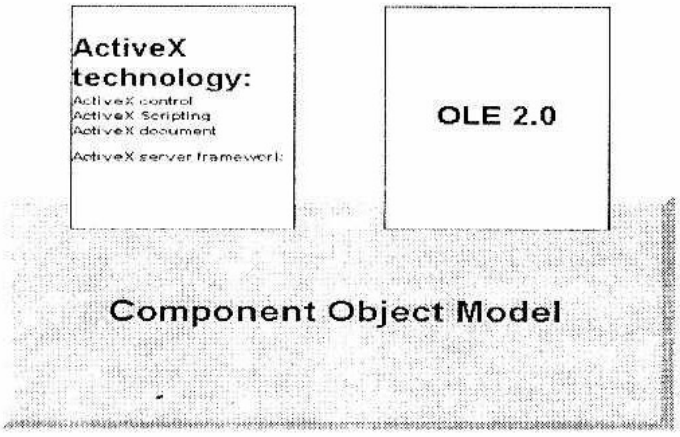
Activex در سال ۱۹۹۶ بعنوان استراتژی اصلی مایکروسافت برای اشیاء توزیع شده و وب ارائه شد.

خصوصیات اصلی Activex عبارتند از:

مؤلفه های Activex از DOM و DCOM برای ارتباط با یکدیگر استفاده می کنند. بعبارت دیگر DCOM نقش ORB را در محیطهای مبتنی بر Activex بازی می کند.

مرورگر می تواند همچون یک Container عمل کند. برای مثال Internet Explorer می تواند محتوای مؤلفه هایی همچون اسناد Word ، اپلتهای جاوا و کدهای HTML باشد.

در کاربردهای توزیع شده می توان تکنولوژی Web (مرورگر، صفحات HTML ، اپلتهای جاوا) را با ابزارهای رومیزی (صفحه گسترده ها، پردازشگرهای متن) در هم آمیخت.



شکل ۳۸-۱ ارتباط Activex, OLE, Com

کاربران Activex می توانند مؤلفه هایی همچون کارگزار SQL و درگاههای وب را فراخوانی کنند.

انواع Activex

نکته ای که باید به آن دقت بسیار نمود آن است که Activex امروزه صورتهای گوناگونی به خود گرفته است که اگر چه با یک نام خوانده می شوند ولی کاربردهای متفاوتی دارند ریشهٔ آن را باید در تعریف بسیار کلی مایکروسافت از Activex جستجو کرد:

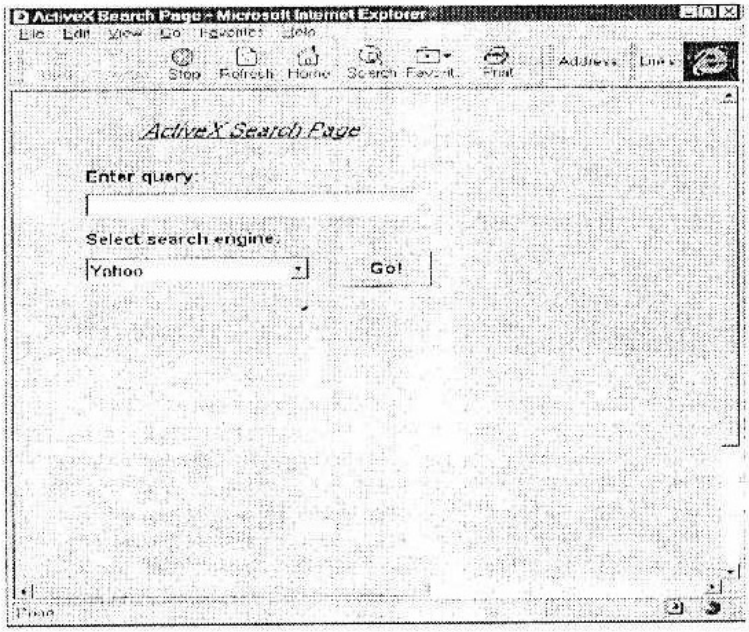
Activex نام تجاری ارائه شده توسط مایکروسافت برای تکنولوژیهایی است که ارتباط داخلی بین کاربردها را با استفاده از مدل COM فراهم می آورند. تکنولوژی Activex دربردارندهٔ یک یا بیشتر واسط است که هر یک از آنها دارای یک یا بیشتر Property و Method هستند. تمامی واسطه های COM از کلاس پایه lunknown منشعب شده اند. تکنولوژی هایی که بر مبنای مدل COM بنا شده اند عبارتند از: Activex و MAPI و OLE .

در وب امکان استفاده از اشیای طراحی شده با زبانهای ویژوال وجود دارد می توان از طریق زبانهای ویژوال اشیاء مورد نظر را طراحی کرد و بصورت فایلهای OCX درآورده و در HTML استفاده نماییم.

کنترلهای Activex نیز همانند اسکریپتها می توانند سمت سرویس گیرنده یا سرویس دهنده باشند در حالت سمت سرویس گیرنده خود برنامه OCX به سیستم سرویس گیرنده منتقل شده و روی Registery سیستم عامل خود را ثبت می کند حال اگر دوباره درخواستی به آن ارسال شود فایل OCX نیازی به بار شدن دوباره ندارد درصورتی که برای برنامه های جاوا این مسئله وجود ندارد.

برای استفاده از کنترولهای Activex ، سمت سرویس گیرنده از برچسب زیر استفاده می شود.

```
< Object    ID = " XYZ " WIDTH = 100 HELGHT = 100
        ClassID = " ClsID = 593#1A b3DDK3097 "
        Code Base = "Http://www.Sample.Com/Sample.ocx ">
 <Param     Name = " p1 "      Value = "AA">
 <Param     Name = " p2 "      Value = "2.1">
 </Object >
```



شکل ۳۹ -۱ مثالی از یک برنامهٔ Activex

۱-۱۸ برنامه های Plug-Ins

برنامه هایی هستند که توسط شرکتهای مختلف نوشته می شوند و این امکان را به مرورگر می دهد که بتواند فایلهایی با پسوند خاص را نمایش دهد مثلاً" فایلهایی بصورت زیر:

Adobe Acrobat (PDF) Portable Document Format
Macromedia Director
Quick Time
VebFX VRML

یک Tag به نام < EMBED > در HTML برای کنترل Plug In ها وجود دارد.

< EMBED SRC ="http:www.server.com/amovie avi">

مرورگر از روی پسوند فایل avi به کمک MIME^ برنامه اجرایی یا Plug-In آنرا پیدا کرده و اجرا می کند.

۱-۱۹ Asp (Active Server Page)

شرکت مایکروسافت از طریق این روش بهترین ویژگیهای روشهای قبلی مثل دسترسی آسان به بانکهای اطلاعاتی دسترسی به سیستم عامل مثل Cgi یا ISAPI و سرعت بالا را دارا می باشد همانطور که می دانید در برنامه نویسی وب دو نوع اسکریپت داریم:

اسکریپت سمت سرویس دهنده (روی سرویس دهنده اجرا می شوند)

اسکریپت سمت سرویس گیرنده (روی سرویس گیرنده اجرا می شوند)

Asp محیطی مبتنی بر اسکریپت نویسی سمت سرویس دهنده می باشد که موجب ساخت برنامه های پویا و محاوره ای می شود. این برنامه ها زمان زیادی را برای پردازش تلف نمی کنند. محیط Asp هم برای طراحان حرفه ای وب و هم مبتدیان مطلوب است

Asp وابسته به IIS می باشد و IIS نسخهٔ ۳ به بالا آنرا پشتیبانی می کند. Asp همان فایل HTML است و می تواند هر چه که HTML پشتیبانی می کند پشتیبانی نماید. مثل اپلت های جاوا – متن چشمک زن – اسکریپتهای سمت سرویس گیرنده و Activex های سمت سرویس گیرنده و... ولی دستوراتی در آن وجود دارد که مربوط به اسکریپتهای سمت سرویس دهنده است که روی سرویس دهنده اجرا می شود. دستورات Asp در داخل برچسبی بصورت <%___%> قرار می گیرند

خاصیتهای خاص Asp

Asp دارای خاصیتهای زیر می باشد.

۱- اسکریپت های سمت سرویس گیرنده را نیز می تواند داشته باشد (برای ایجاد صفحات پویا)

۲- Asp دارای تعدادی شیء داخلی است که امکانات بسیار زیادی را به اسکریپت های شما می دهد حتی اجازه ارسال یا دریافت اطلاعات از یک مرورگر را نیز می دهند.

۳- Asp می تواند تعدادی از اجزای Activex سمت سرویس دهنده را استفاده کند.

٤- Asp با بانکلهای اطلاعاتی سرویس دهند ه از طریق SQL می تواند کار کند.

زبانهای اسکریپت نویسی Asp

توجه به این نکته ضروری است که Asp یک تکنولوژی است و نه یک زبان لذا اسکریپت صفحهٔ Asp می تواند با هر زبانی (c, Perl,Jscript,Javascript,Vbscript....) نوشته شده باشد و تنها کافی است که کارگزار ابزار لازم برای اجرای این اسکریپت را داشته باشد. از آنجا که Vزبان بسیار ساده ای است و از سوی دیگر script زبان پیشفرض IIS برای اسکریپت نویسی سمت سرویس دهنده می باشد لذا می توان Vbscript را به عنوان زبان تولید اسکریپتهای سمت سرویس دهنده در نظر گرفت. جهت تغییر دادن زبان از Vbscript به Jscript کافی است دستورات زیر را بنویسید:

مثال :

<%@Longuage=Jscript%>

همچنین می توانید هر جا که می خواهید از جاوا استفاده کنید:

<Script Language ="Jscript" RunAt ="Server">

برنامه نویسی Asp با توجه به مقدمه بالا هیچ ارتباط مستقیمی با محیط برنامه نویسی Visual basic ندارد اما چون از Vbscript به عنوان زبان سمت سرویس دهنده استفاده می شود و این دو محیط به نحوی با یکدیگر ارتباط برقرار می نمایند. در قیاس با Visual basic نسخه اسکریپتی (Vbscript) چند ویژگی را دربرندارد:

۱- دستورات واسط کاربر از قبیل Input Box,Msgbox.

۲- توابع مربوط به ایجاد اشیاء از قبیل Getobject و Createobject.

نبود این دو ویژگی از دو جهت قابل توجیه است:

۱- از آنجا که اسکریپت Asp در سمت کاربر اجرا نمی شود نیازی به وجود واسط کاربر نمی باشد.

۲- Asp از توابع سرویس دهنده برای تعامل با اشیاء استفاده می نماید و در چنین وضعیتی نیازی به توابع Visual basic وجود ندارد.

تجمع اشیاء و اجزا در درون Active Server Page

Asp دربردارندهٔ چندین شیء درون سازه است که می تواند با اجزای Activex قابل نصب دیگری توسعه یابند. اشیاء درونی Asp شباهت بسیاری با اجزای Activex (Activex Components) دارند ولی دو تفاوت عمدهٔ آنها به قرار زیر است:

۱- یک جزء می تواند دارای چندین شیء باشد.

۲- قبل از استفاده از یک جزء باید حتماً نمونه ای از آن ایجاد شده باشد. (برای اشیاء درونی نیاز به نمونه سازی نیست)

اشیاء درونی Asp

جدول زیر اشیاء درونی Asp را نشان می دهد. با استفاده از این اشیاء می توان کنترل پاسخهایی که سرویس دهنده به درخواستهای مرورگر می دهد را بدست گرفت و همچنین بر درخواستهای مرورگر نیز نظارت داشت.

هر یک از اشیاء درونی به همراه Collection ها، وقایع و Method هایشان کاربردهای ویژه ای دارند که در فصلهای بعدی صرفاً به برخی از موارد مهم کاربرد در عمل اشاره می شود.

جدول اشیاء درونی ASP

Application Collectipn: StaticObjects Contents Contents Collection Methods: Remove Remove All Methods: Lock Unlock Event: Application-OnEnd Application-OnStart Object Context Object Methods: Set Abort Set Complete Events: OnTransaction Abort OnTransaction Commit Request Object Collection: Client Certificate Cookies Form Query String Server Variables Properties: Total Bytes Methods: Binary Read Asp Error Object Properties: Asp Code Number Source Category File Line Column Description Asp Description	Response Object Collectipn: Cookies Properties: Buffer Cache Control Charset Content Type Expires Expires Absolute PICS Status Methods: Addheader Appen To Log Binary Write Clear End Flush Redirect Write Server Object Properties: Script Timeout Methods: Create Object Execute Get Last Error HTML Encode Map Path Transfer URL Encode Session Object Collectipn: Static Object Contents Contents Collection Methods: Remove Remove All Properties: Code Page LCID Session ID Timeout Methods: Abandon Events: Session-OnEnd Session-OnStart

بخش دوم

برنامه نویسی از طریق Active Server Page

فصل دوم :
ساختن Active Server Page

این فصل رسماً شما را با ASP که موضوع اصلی این کتاب نیز می باشد، آشنا می کند. و شما فرا می گیرید که ASP چیست ، چه کاربردهایی دارد و چگونه عمل می کند. این فصل یک تعبیر کلی از چگونگی بکارگیری اسکریپت های ASP در صفحات HTML و برآوردی از ظرفیتهای اشیاء و اجزاء ASP را پیش روی شما می گذارد. سرانجام شما چگونگی پیکره بندی سرویس دهندهٔ وب خود را برای استفاده از ASP و رفع عیب مسائل احتمالی فرا می گیرید.

۲-۱ ASP چیست؟

یک ASP فایل استاندارد HTML ای است که با یک سری ترکیبات الحاقی توسعه یافته است. همانند فایل استاندارد HTML ، یک ASP می تواند شامل برچسب های HTML ای که توسط مرورگر وب تفسیر و نمایش داد می شود ، باشد . هرچیزی مثل اپلت های جاوا ، متن چشمک زن ، اسکریپت ها و کنترلهای اکتیوایکس روی سرویس گیرنده که بتوانید در یک فایل HTML جای دهید، در یک ASP نیز قابل جای دادن است. البته ASP پنج ویژگی منحصربفرد نیز دارد که بصورت زیر است :

یک ASP می تواند شامل اسکریپت های سمت سرویس دهنده باشد.

در فصلهای بعدی کتاب به شما یاد داده می شود که چگونه اسکریپت های ASP توسط جاوا اسکریپت و VBScript ایجاد می شود. با بکارگیری اسکریپتهای روی سرویس دهندهٔ یک ASP شما می توانید صفحات وبی با اجزاء پویا ایجاد کنید. به عنوان یک نمونهٔ بسیار ساده شما می توانید صفحهٔ وبی را ایجاد کنید که هریک پیغام جدید یا تاریخ آن روز را نمایش دهد.

ASP شماری از اشیاء تعبیه شده را فراهم کرده است.

با استفاده از اشیاء تعبیه شده قابل دسترس در یک ASP شما می توانید اسکریپت های خود را بسیار قوی تر کنید در بین چیزهای دیگر این اشیاء به شما اجازهٔ دریافت و ارسال اطلاعات به مرورگر یا از آن را می دهد. برای مثال با استفاده از شیء Request می توانید اطلاعاتی را که یک کاربر با فرم HTML ارسال کرده دریافت و به آن اطلاعات توسط یک اسکریپت پاسخ دهید.

- اطلاعات را از فرمها دریافت و در یک پایگاه داده ذخیره کنید.
- یک صفحه وب شخصی بسازید که شامل مشخصات متفاوت کاربران مختلف باشد.
- یک شمارنده صفحه یا وب ایجاد کنید.
- با توجه به مشخصات مرورگرهای مختلف صفحات وب مختلفی را نمایش دهید .
- صفحات مختلف را به هم متصل کنید.

فعالیتهای کاربران سایت وب را دنبال واطلاعات آنها را بدست آورده و این اطلاعات در یک Logfile ذخیره کنید.

۲-۳ چگونه ASP کار می‌کند ؟

بهترین راه برای فهمیدن اینکه ASP به چه صورت کار می‌کند ، مقایسه سرویس دهنده‌های وبی است که ASP را حمایت می‌کنند با آنهایی که حمایت نمی‌کنند. ماکروسافت ، ASP را با سومین نسخه IIS معرفی کرد. با معرفی ASP IIS، سرویس دهنده ای با محتوای ثابت و ایستا به یک سرویس دهنده با محتوای پویا و تأثیر پذیر تبدیل شد. این کار به چه صورت انجام می شود؟ مهمترین کار IIS سرویس دادن به صفحات HTML استاندارد بود . هنگامی که کسی نیاز به یک صفحه وب داشت از سرویس دهنده IIS استفاده می‌کرد و آن فایل ثابت HTML را از دیسک یا حافظه بدست می‌آورد وبه مرورگر شخص می‌فرستد.

IIS با بقیه سرویس دهنده های دیگر وب متفاوت است . مهمترین هدف هرسرویس دهنده وب ، سرویس دادن به فایلهای HTML است . این مهم است که بدانیم سرویس دادن به فایلهای HTML به چه صورت انجام می‌شود مراحل زیر را درنظر بگیرید :

۱ - یک کاربر ، آدرس اینتر نتی فایل HTML ای را که می‌خواهد درون خط آدرس قرار می‌دهد ، وبا فشار دادن کلید Enter این درخواست را می‌فرستد .

(به عنوان مثال http://.www.aspsite.com/hello.htm)

۲ - مرورگر این درخواست را برای یک سرویس دهنده وب مثلا IIS می‌فرستد .

۳ - سرویس دهنده وب درخواست را دریافت می‌کند و تشخیص می‌دهد که یک فایل HTML درخواست شده است به این دلیل که دارای پسوند htm یا html است .

٤ - سرویس دهنده وب آن فایل را از دیسک یا حافظه دریافت و آن را برای مرورگر ارسال می کند .

٥ - فایل HTML توسط مرورگر شخصی ترجمه و تفسیر می‌شود و نتیجه در پنجره مرورگر نمایش داده می‌شود .

البته این فرآیند اندکی پیچیده تر می باشد . اما این مراحل فعالیت لحظه به لحظه یک سرویس دهنده وب را نشان می‌دهد . یک سرویس دهنده در خواست ها را از فایلهای مخصوصی دریافت می‌کند وبا بدست آوردن آن فایل از یک سخت افزار یا حافظه و فرستادن آن فایل پاسخ می‌دهد .

ASP همه چیز را تغییر داد. دیگر علاوه براینکه IIS می تواند برای سرویس HTML ثابت مورد استفاده قرار گیرد ، با ASP ، IIS قادر به انجام خدمات HTML پویا و تأثیر پذیر به همان خوبی قبل شد. با بکار بردن ASP می‌تواند صفحاتی با محتوای جدید برای پاسخگویی به درخواستهای کاربربه وجود آورد. یک سرویس دهنده وب

در این فرآیند برای به وجود آوردن صفحه وب خود به خود فعال می‌شود . مهم است بدانید که سرویس دهنده ASP و HTML چه تفاوتهایی دارند. به مراحل زیر توجه کنید :

۱ - یک کاربر آدرس اینترنتی فایل ASP را درون خط آدرس قرار می‌دهدوبا زدن کلید Enter درخواست ASP را می‌فرستد . (به عنوان مثال (http://www.aspsite.com/hello.asp)

۲ - مرورگراین درخواست را برای یک سرویس دهنده وب مثلا IIS می‌فرستد .

۳ - سرویس دهنده وب درخواست را دریافت می کند به این دلیل که درخواست پسوند asp دارد. تشخیص می‌دهد که یک فایل ASP در خواست شده است .

٤ - سرویس دهنده وب فایل ASP را از دیسک یا حافظه باز می کند.

٥ - سرویس دهنده وب فایل را به برنامه مخصوصی به نام ASP.dll می‌فرستد.

٦ - فایل ASP از بالا تا پایین پردازش می‌شود و تمام دستورات اجرا می‌شود ، نتیجه این پردازش یک فایل HTML استاندارد است.

۷ - فایل HTML برای مرورگر فرستاده می‌شود .

۸ - فایل HTML توسط یک مرورگر شخصی ترجمه و تفسیر می‌شود و نتیجه در پنجره مرورگر نمایش داده می‌شود

یک ASP با یک فایل HTML عادی خیلی تفاوت دارد . یک فایل HTML عادی بدون پردازش به مرورگر فرستاده می‌شود . تمام دستورات در یک فایل HTML در ابتدا باید اجرا شود تا یک صفحه HTML را به وجودآورد. ASP در اغلب موارد به صورت کامل مانند یک HTML نرمال است ، تنها فرقی که دارد این است که پسوند آن به جای htm ، asp است . وقتی درخواستی برای به دست آوردن یک ASP به وجود می آید ، مرورگر یک صفحه HTML عادی را دریافت می کند، همین امر باعث می شود که یک ASP با تمام مرورگرها سازگار شود.

۲-۴ بکار بردن اسکریپت ها درون ASP

یک ASP ابتدائی ترین محیط اسکریپت نویسی است . می‌توان درون ASP هردو نوع اسکریپت (Jscript VBScript) استفاده کرد . از بقیه زبانهای اسکریپت نیز می‌توان به همان خوبی استفاده کرد. تمام زبانهای اسکریپت سازگار با استاندارد اسکریپت نویسی اکتیوایکس می‌توانند در یک صفحه ASP استفاده شوند. راحت‌ترین راه برای استفاده از اسکریپت درون ASP استفاده از کاراکترهای <%-%> می‌باشد . هر متنی که درون این کاراکترها قرار بگیرد به عنوان اسکریپت سمت سرویس دهنده پردازش می‌شود .

به عنوان مثال:

```
< HTML >
< HEAD > < TITLE > ASP Script < /TITLE > < /HEAD >
<BODY>
This is a
<% FOR i=1 TO 10 %>
very
```

<% NEXT %>

.very long sentence

</BODY >

</HTML>

هنگامی که این ASP توسط مرورگر دریافت می شود جمله زیر نمایش داده خواهد شد:

This is a very very very very very very very very very very very very long sentence.

این اسکریپت کلمه very را ۱۱ بار تکرار می کند و این کار را با استفاده حلقه FOR...NEXT در VBscript انجام می‌دهد . به هر حال ، سه روش برای استفاده از اسکریپت ها درون ASP وجود دارد.

می‌توان با استفاده از Internet Service Manager یک زبان اسکریپت را به عنوان پیش فرض برای تمام ASP ها در نظر گرفت . بعد از اینکه یک زبان اسکریپت را به عنوان پیش فرض انتخاب کرده اید می‌توانید آن زبان را درون Asp به همان راحتی وبا استفاده از کاراکترهای <%- %> استفاده کنید. به عنوان مثال اگر می‌خواهید از جاوا اسکریپت استفاده کنید باید این زبان را به عنوان پیش فرض انتخاب کنید.

همچنین می‌توانیم یک اسکریپت خاص را فقط دریک صفحه بکار ببریم . برای انجام این کار زبان مورد نظر را در اولین خط از فایل ASP تعریف می کنیم مانند زیر :

<%@LANGAGE=JScript%>

<HTML>

<HEAD><TITLE>ASP Script</TITLE></HEAD>

<BODY>

This is a

<{%FOR (i=1 ; i<11 ; i++)%>

very

<% } %>

.very long sentence

</BODY>

</HTML>

راهنمایی که در خط اول قرار داده شده است نشان می‌دهد که می‌توان فایلی را به جای بقیه زبانهای اسکریپتی با جاوا اسکریپت به وجود آورد . هنگامی که از این راهنما استفاده می‌شود. باید توجه داشت که این دستور باید قبل از هر دستوری در صفحه ASP قرار بگیرد.

سومین روش برای استفاده از اسکریپت ها درون ASP ،استفاده از برچسب <Script> در HTML است.

به مثال زیر توجه کنید :

<HTML>

<HEAD><TITLE>ASP Script</TITLE>>/HEAD>

```
<BODY>
<SCRIPT LANGUAGE=" JScript" RUNAT=" Server" >
function sayhello( )
{
response.write(" Hello!")
{
</SCRIPT>
<%
sayhello( )
%>
</BODY>
</HTML>
```

در اینجا یک تابع جاوا اسکریپت استفاده شده است . LANGUAGE نشان دهنده نوع زبان اسکریپت ای است که استفاده می‌شود . RUNAT نشان می‌دهد که این اسکریپت باید در سرویس دهنده اجرا شود . در مثال بالا اگر عبارت "RUNAT=" Serverفراموش شود ، دستورات در سرویس گیرنده اجرا می‌شوند . در این حالت سرویس دهنده دستورات را نادیده می گیرد و سرویس گیرنده تلاش در اجرای دستورات می‌نماید .

چرا اکثراً ازبرچسب های %>و<% به جای کاراکترهای %> و <% استفاده می کنیم ؟ باید گفت که به صورت عادی از برچسب های <SCRIPT> استفاده می‌شود ولی دو تفاوت اصلی بین این دو روش وجود دارد : اول اینکه اسکریپت هایی که شامل برچسب<SCRIPT> می باشند به سرعت در ابتدا اجرا شده و هیچ مشکلی با ASP ندارند به عنوان مثال صفحه زیر را در نظر بگیرد:

```
<HTML>
<HEAD><TITLE>ASP Script </TITLE></HEAD>
<BODY>
This is the first sentence.
< SCRIPT  LANGUAGE= "JScnpt " RUNAT = "server">
    response. write("This is the second sentence")
</SCRIPT>
</BODY>
</HTML>
```

وقتی به این اسکریپت نگاه می کنید ممکن است تصور کنید که جملهٔ "This is the first sentence" وجملهThis is the second sentence به همان ترتیب در صفحه چاپ شدند . به هر حال وقتی ASP در صفحه

نمایش داده می‌شود ترتیب دوجمله دقیقاً به صورت وارونه و اشتباه می‌شود و یاچیزی نمایش داده نمی‌شود ، چرا که صفحه HTMLنادرستی به وجود آمده است . این اتفاق به این دلیل می‌افتدکه هر چیزی که داخل برچسب<SCRIPT> باشد قبل از هر چیز دیگری در صفحه اجرا می‌شود . اگر از این روش استفاده کنید نتیجه زیر را روی مرورگر خواهید دید :

```
<This is the Second sentence. <HTML>
<HEAD><TITLE>ASP Script</TITLE></HEAD>
<BODY>
This is the first sentence.
</BODY>
</HTML>
```

این عمل دو دلیل دارد ; اول این بودکه شما می توانستید از دستورات <SCRIPT> درهر جا استفاده کنید و دوم اینکه برچسب<SCRIPT> برای بیشتر اهداف ، محدود به محتوای توابع است . خروجی اسکریپتی که شامل توابع نمی‌باشد فوراً در صفحه نمایش داده می‌شود و نتیجه یک HTML نادرست خواهد بود.

برچسب <SCRIPT> یک مزیت اصلی نسبت به کاراکتر های <٪ و ٪> برای تعریف اسکریپت دارد . بطوریکه شما می‌توانید از چندین زبان اسکریپتی مختلف درون یک ASP استفاده کنید مثال زیر را در نظر بگیرید :

```
<٪@ LANGVAGE= "Vbscript" ٪>
<HTML>
<HEAD><TITLE>ASP Script</TITLE></HEAD>
<BODY>
<SCRIPT  LANGVAGE=" JScript" RUNAT= "Server ">
function sayhello( )
{
response.write ( Hello!)
</SCRIPT>
٪>
FOR i=I TO 10
Sayhello( )
NEXT
<٪
</BODY>
</HTML>
```

این اسکریپت کلمه !Hello را ده بار در یک سطر تکرار می کند . اما باید توجه داشت که این کار را به چه صورت انجام می‌دهد .اسکریپتی که درون کاراکترهای تعریف %> و <% قرار دارد اسکریپت ویژوال بیسیک می‌باشد در اینجایک تابع را از JavaScript صدا می‌زنیم . که درون برچسب <SCRIPT> تعریف شده است. هنگامی که از یک اسکریپت به عنوان زبان اصلی استفاده می کنید و همچنین نیاز به زبان اسکریپت از نوع دیگری در این صفحه دارید از این روش استفاده می کنید.

در اینجا بطور خلاصه سه روش استفاده ازاسکریپت ها درونASP بیان شده است :

- یک زبان اسکریپت با استفاده از Internet Service Manager به عنوان زبان اصلی تعریف می کنیم .
- یک زبان اسکریپت را برای یک صفحه با استفاده از راهنمایASP به صورت زیر تعریف می کنیم :

<%@ LANGUAGE=" Scripting language "%>

- با استفاده از <SCRIPT> می‌توان چندین زبان اسکریپت را در یک ASP استفاده نمائیم .
- با استفاده از خروجی ها درASP می‌توانید مقدار عبارات را در پشت صفحه نمایش دهید .

به عنوان مثال :

```
<HTML>
<HEAD><TITLE>ASP Example</TITLE></HEAD>
<BODY>
At the tone , the time will be : <%=TIME%>
</BODY>
</HTML>
```

با استفاده از کاراکترهای تعریف <% و %> می‌توانید مقدار متغیرها ، توابع و متدها را چاپ نمائید . در مثال بالا با استفاده از تابع TIME درVBScript می‌توانیم زمان را در صفحه نمایش چاپ کنیم .یک راه دیگر برای انجام این کار وجود دارد مثال زیر را در نظر بگیرید :

```
<HTML>
<HEAD><TITLE>ASP Example </TITLE></HEAD>
<BODY>
At the tone , the time will be= <% Response.write(TIME) %>
</BODY>
</HTML>
```

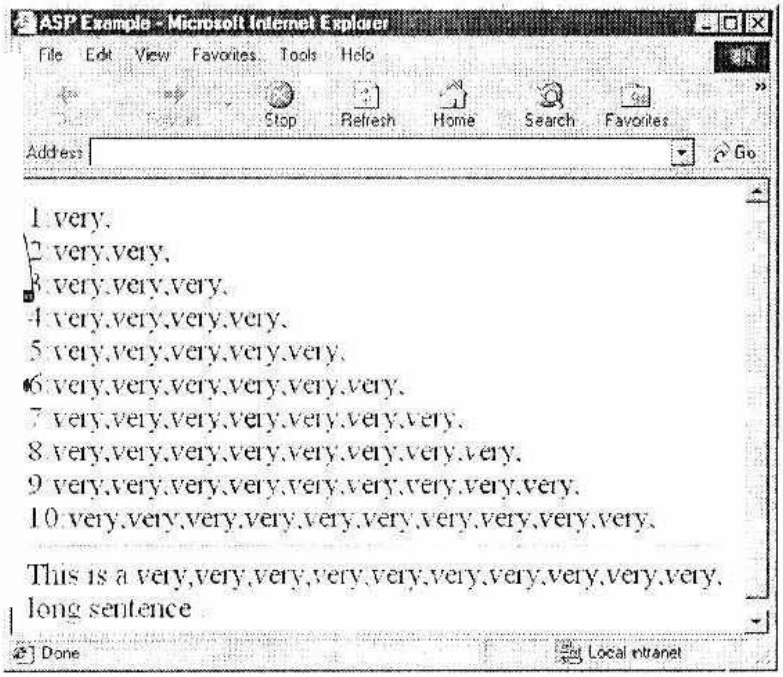
در این مثال مقدار تابع TIME در VBScript بوسیله یک شئ از ASP نمایش داده می‌شود . متد write از شیء Response مقدار متغیر خروجی را روی صفحه نمایش می‌دهد .چه وقت باید از متد ( ) write به جای <% و %> به عنوان خروجی استفاده کنیم؟ این موضوع زیاد مهم نیست .ASP به صورت داخلی به متد صدا شدهٔ

Response.write( ) پاسخ می‌دهد. این دو متد برای نمایش دادن مقدار خروجی ، به صورت داخلی بوده و قابل تبدیل به یکدیگر می باشند .

در مواقعی که یکی از روشهای خروجی مناسب‌تر از دیگری است ؛ به عنوان مثال وقتی شما یک عبارت خروجی را می‌خواهید داخل یک اسکریپت تعریف کنید راحتتر است که از متد Response.write() استفاده کنید، به عبارت دیگر وقتی شما می‌خواهید مقدار خروجی یک عبارت را به وسیله یک قسمت از کد HTML نشان دهید راحت تر است از <=و%> استفاده کنید در مثال زیر هر دو روش نشان داده شده است :

```
<HTML>

<HEAD><TITLE>ASP Example </TITLE></HEAD>

<BODY>

<%
FOR i=1 TO  10
myvar = myvar &"very"
Response.write(i&" : "& myvar& "<BR>")
NEXT
%>
<HR>

This is a <%=myvar %> long sentence.
</BODY>

</HTML>
```

در این مثال متد Response.write( )درون یک حلقه برای نشان دادن مقدار متغیر myvar وبرای کم کردن اندازه برنامه استفاده می‌شود و از علامت <=و%> به صورت عادی در یک کد ساده HTML استفاده می‌شود .



شکل ۲-۲ دو متد از عبارتهای خروجی

۲-۵ به وجود آوردن اشیاء و اجزاء در ASP

ASP شامل تعدادی از اشیاء تعبیه شده و اجزاء قابل نصب Activex است . این اشیاء و اجزاء می‌توانند ASP های قدرتمند تری بسازند . اما واقعا اشیاء واجزاء چه می‌باشند؟ یک شیء دارای متد[۱]، خواص[۲] و مجموعه هایی[۳] دارد . یک متد کاری را که با یک شیء می‌توان انجام داد ، تعریف می کند . خواص شیءها به منظور خواندن و تنظیم کردن وضعیت اشیاء بکار می‌رود و مجموعه های یک شیء شامل دسته های مختلفی از کلید ها و مقادیر اشیاء هستند.

یک مثال روزمره را درنظر بگیریم . کتاب تام سایر [4]مثالی از یک شیءاست . یک شیء دارای متدهایی است و کاری را که آن شیء انجام می‌دهد تعیین می کند؛در این مثال می‌توانید کتاب را بخوانید .یک شیء دارای یک سری خواص است ؛ در این مثال صفحات دارای شماره های مخصوص به خود هستند ودر آخر شامل مجموعه هایی از کلیدها ، شماره هرصفحه (کلید) و متن (محتوای) مربوط به خود می باشد.

یک جزء ActiveX خیلی شبیه به اشیاء تعبیه شده در ASP است. به هر حال هنگام استفاده از ASP ها دو تفاوت مهم بین اشیاء و اجزاء وجود دارد. اولا" اینکه یک جزء مشتمل بر بیش از یک شیء می باشد. ثانیا یک جزء قبل از اینکه بتوان از آن استفاده کرد باید به صورت کاملا آشکار به وجود آید .

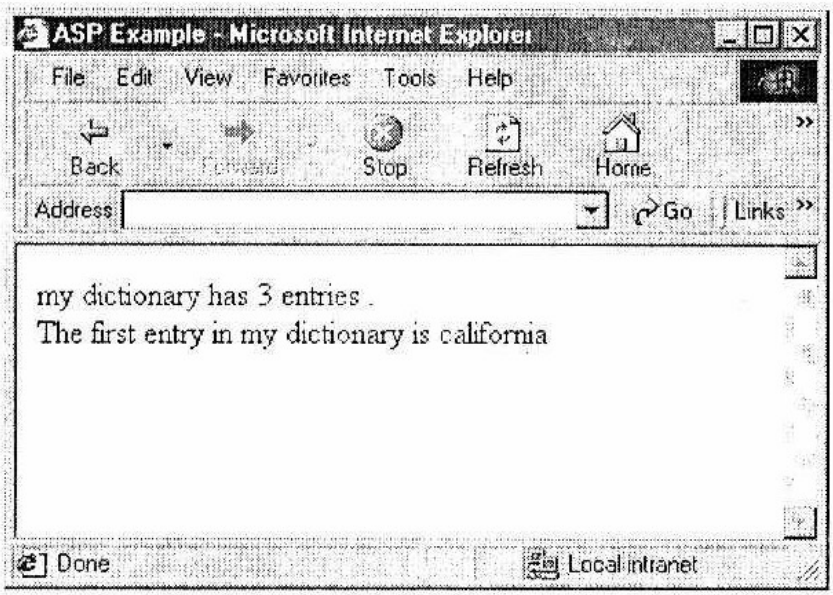
هم زبان VBscript وهم جاوا اسکریپت دارای تعداد کمی شیء می‌باشند به عنوان مثال هم در VBScript وهم در JavaScript می توانید شیء دیکشنری را داشته و آن راتغییر دهید شیء دیکشنری جزء ASP نمی‌باشد ولی آن را می‌توان در ASP استفاده کرد . به عنوان مثال :

```
<%
Set MyDict=Server.CreateObject("Scripting.Dictionary")
MyDict.add  "CA" , " california "
MyDict.add " MA" , "Massachusetts "
MyDict.add "MI" , "Missouri "
<%
 my dictionary has  <%=MyDict. count %> entries .
<BR>

 The first entry in my dictionary is <٪  <%= MyDict.item("CA " )%>
```

وقتی این اسکریپت اجرا می‌شود یک نمونه از شیء دیکشنری به وجود می آید بعد سه جفت کلید ومقدارآنها به دیکشنری ا ضافه می‌شود در آخردوتا از خواص دیکشنری نمایش داده می‌شود. (مراجعه شود به شکل ۲-۳)

خط اول این مثال نشان می‌دهد که متد عمومی یک نمونه جدید از شیء اکتیوایکس را به وسیله ASP بوجود می آورد. متد Server . Create Object یک نمونه ازشیء را ایجاد می کند. در این مثال متغیر، MyDict نامیده

می‌شود که آن را به یک نمونه از شیء دیکشنری نسبت می‌دهیم. همینکه نمونه ای از شیء ساخته شد ، متد آن قابل صدا زدن است. در این مثال متد Add که متعلق به شیء دیکشنری می باشد برای اضافه کردن ورودی ها به دیکشنری ، صدا زده می شود. اولین متد Add ای که صدا زده می‌شود برای افزودن کلید و مقدار "CA" و"california" استفاده می‌شود .



شکل ۲-۳ مثالی از شیء دیکشنری

بعد از اینکه یک نمونه شیء به وجود آمد می‌توانیم به خواص آن دسترسی داشته باشیم در این مثال خاصیت Count برای تعیین تعداد ورودی ها خوانده می شود. همچنین خاصیت Item برای برگرداندن مقدار کلیدی خاص ، خوانده می شود.در آخر که شما از شیء بوجود آمده استفاده کردید، می توانید وقتی که سرویس دهنده پردازشهای ASP خود را پایان داد آنرا از بین ببرید. البته لزومی ندارد که خودمان این شیء را از بین ببریم زیرا سرویس دهنده خود این کار را انجام می دهد. ولی اگر خواستیم شیء را خودمان از بین ببریم به صورت زیر عمل می کنیم :

```
<% Set MyDict =Noting %>
```

اگر چه شیء دیکشنری جزء ASPنمی‌باشد ولی مثال بالا نشان می‌دهد که اشیاء به چه صورت درون ASP استفاده می شود این شیء را می‌توان با صدا کردن متد ()Server.Createobjectبه وجود آورد .بعد از به وجود آوردن یک شیء جدید می‌توان متدهای آن را صدا زد و خواص آنرا خواند و تنظیم نمو د.

۲-۵-۱ اشیاء ASP

ASP شامل تعدادی از اشیاء تعبیه شده است این اشیاء باعث افزایش قدرت اسکریپت های ما می‌شود .با استفاده از این اشیاء می‌توانید تعداد زیادتری از در خواست های مرورگر را دریافت کنید و چگونگی پاسخ سرویس دهنده به این در خواست ها را کنترل نمائید. این اشیاء تعبیه شده امکان کنترل Session ها و برنامه های سرویس دهنده وب را به وجود می‌آورد . با توجه به شیء تعبیه شدهٔ Response در مثال قبل از این شیء می‌توانید برای فرستادن خروجی هایتان به مرورگر استفاده کنید این شیء دارای شماری از خاصیتها ، متدها ومجموعه های مهم می‌باشد .

در زیر مرور مختصری بر اشیاء Asp تعبیه شده می‌باشد:

شیء Application : این شیء برای ذخیره وبازیافت اطلاعاتی بکار می‌روند که می توانند بین درخواستهای کاربران به اشتراک گذاشته شوند به عنوان مثال می توان ازشیءApplication برای فرستادن اطلاعات بین کاربران سایت وب خود استفاده کرد .

شیء Request: این شیء برای دستیابی به تمام اطلاعات فرستاده شده با درخواستی که از مرورگر برای سرویس دهنده شما فرستاده می شود ،قابل استفاده است. می‌توان ازاین شیء برای بدست آوردن اطلاعاتی که کاربر درون فایل HTML قرار می دهد استفاده کرد .

شیءResponse: این شیء برای فرستادن اطلاعا ت به مرورگر استفاده می شود این شیء می‌تواند خروجی را توسط اسکریپت به مرورگر بفرستد .

شیء Server: این شیء اجازه استفاده از توابع گوناگون و مفیدی را از سرویس دهنده می‌دهد، به عنوان مثال با استفاده ازآن می توان مدت زمان اجرای دستورات اسکریپت را قبل از آن که فرصت زمانی آنها پایان یابد کنترل نمود . می‌توان با استفاده از این شیء،اشیاء دیگری را به وجود آورد .

شیء Session: این شیء برای ذخیره وبازیابی اطلاعاتی درمورد جلسات کاربران خاص، استفاده می شود. از این شیء برای ذخیره اطلاعا ت کاربری که دارد ازسایت وب شما دیدن می کند می توان استفاده کرد.

شیء Context: این شیء برای کنترل نقل و انتقالاتASP استفاده می‌شود این نقل و انتقالات توسط نرم افزار Microsoft Transaction Server(MTS)مدیریت می‌شود.

اشیاء تعبیه شده ای که در بالا ذکر شده با اشیاء معمولی تفاوت دارند. با استفاده از آنها دیگر نیاز به ساخت شیءنداریم مگر اینکه بخواهیم درون یک اسکریپت از شیء جدیدی استفاده کنیم. باید اضافه کرد که متدها ، خواص ومجموعه های هر شیء بطور خودکار در دسترس قرار می گیرند .

۲-۵-۲ اجزاءASP

اجزاء ASP مانند اشیاء تعبیه شده در آن برای قدرتمند کردن اسکریپت ها استفاده می‌شوند. اجزاء با اشیاء تعبیه شده تفاوت دارند . زیرا آنها برای کار های خاص تری استفاده می‌شدند در زیر لیستی از اجزاء ASP بیان شده است :

جزء Ad Rotator : این جزء برای نمایش اعلان آگهی در صفحات سایت وب استفاده می شود می‌توان از این جزء برای تعیین چگونگی تکرار اعلان آگهی که باید نمایش داده شوند استفاده کرد.

جزء Browser Capabilities: این جزء برای نمایش محتوای HTML های مختلف مطابق با قابلیتهای مرورگر ها استفاده می‌شود.

جزء Cortent Linking: با استفاده از این جزء می‌توانید تعدادی از صفحات HTML را به هم متصل کنید که باعث می‌شود هدایت کاربر آسانتر شود به عنوان مثال با استفاده از این جزء می‌توانید صفحات یک کتاب را نمایش دهید .

جزء Counters: این جزء می‌تواند کاربران مختلف را در سایت وب ردیابی کند . با استفاده از این جزء می‌توانید یک شمارنده برای صفحه وب ایجاد کنید.

جزء Counters Rotator : این جزء شما را قادر به حرکت درون محتوی یک HTML می کند . به عنوان مثال می‌توان به صورت تصادفی آگهی های مختلفی را درصفحهٔ خانگی سایت وب نمایش داد .

جزء Page Counter: دقیقا مانندجزء Countersمی‌باشد . این شمارنده می تواند برای پیگیری تعداد ملاقات کننده های یک صفحه وب استفاده شود . با استفاده از این جزء می‌توان یک شمارنده برای یک صفحه خاص وب گذاشت .

جزء Permission Checker : این جزء برای نمایش اتصال های مختلف به صفحات وب به شرط اینکه کاربر اجازه دیدن آنها را داشته باشد استفاده می شود. با استفاده از این جزء می‌توان صفحات وبی را به وجود آورد که فقط مدیر سایت یا افراد خاصی می‌توانند آنها را ببینند .

ActiveX Data object (ADO) : این شیء به شما اجازه می‌دهد تا داده را درون یک بانک اطلاعاتی سرویس دهندهٔ SQL ماکروسافت ذخیره وبازیابی نمائیم .

۲-۶ تنظیم و عیب یابی ASP

قبل از استفاده از ASP باید کاملاً مطمئن شد که ASP روی سیستم نصب شده است یا نه؟. برای این کارباید IIS را نصب کنید . هنگامی که سرویس دهندهٔ NT را نصب کردید ممکن است ASP نصب نشده باشد ASP از IIS نسخه ۳ به بعد وجود دارد و برای گرفتن آخرین نسخه ASP از IIS از سایت وب ماکروسافت در آدرس http://www.microsoft.com/iis دیدن کنید .

بعد از اینکه ASP را نصب کردید برای اینکه بتوانید از ASP استفاده کنید باید IIS را تنظیم کنید شما حداقل یک فهرست در سایت وب خود برای اجرای فایلهای ASP نیاز دارید مراحل زیر را در نظر بگیرید .

۱- برنامه Internet Service Manager را از برنامه Microsoft Internet Information server از منوی Start بیاورید .

۲- در سمت چپ پنجره Internet Service Manager شما را به سوی پیش فرض سایت وب راهنمایی می کند . (اگر نخواهید چیزی راتغییر دهید سایت وب به نام سایت وب پیش فرض که ممکن است همان نام کامپیوتر شما نیز باشد، نامیده می‌شود.)

۳- با استفاده از هدایت کننده ای که درون Internet Service Manager وجود دارد یک فهرست برای ذخیره صفحات فایلهای ASP بوجود آورید .

٤- با زدن کلیک سمت راست Mouse روی فهرست ، نامی را انتخاب و خواص آن را تعیین کنید.

٥- روی فهرست Labeled یا فهرست Virtualکلیک کنید (این بستگی به نوع فهرست دارد).

٦- سپس قسمت Execute ;Permission یا Script را انتخاب کنید.

اکنون که یک فهرست برای اجرای فایلهای ASP به وجود آورده اید برای اینکه بتوانید از ASP خود استفاده کنید باید آن را درون این فهرست ذخیره کنید .

اگر فهرست شما فیزیکی باشد صفحات را می‌توان به راحتی توسط نام آنها ذخیره نمود ولی اگر یک فهرست مجازی است باید مسیر محلی آن، روی درایو مشخص شود .

برای بار کردن یک ASP درون مرورگری که روی همان شیء IIS قرار دارد باید در خط آدرس ، آدرس مرورگر وب خود را تایپ نمائید . به عنوان مثال اگر شیء شما Mymachine نامیده می شود و ASP درون فهرست ریشه از سایت وب شما قرار دارد این صفحه را می توانید با آدرس http://Mymachin/mypage.ASP دسترسی داشته باشید.

۲-۶-۱ تست کردن این ترکیب

می توانید توسط ایجاد یک ASP ساده از صحیح بودن پیکره بندی ASP خود اطمینان حاصل کنید اگر این صفحه با موفقیت روی مرورگر نمایش داده شود بدانید که همه چیز هماهنگ است .

برای بوجود آوردن یک ASP نیاز به یک ویرایشگر متنی می‌باشد به عنوان مثال می‌توان از Notepad که در accessory در سرویس دهنده ویندوز NT قرار دارد استفاده کرد Notepad را اجرا کرده و متن زیر را درون آن تایپ نمائید:

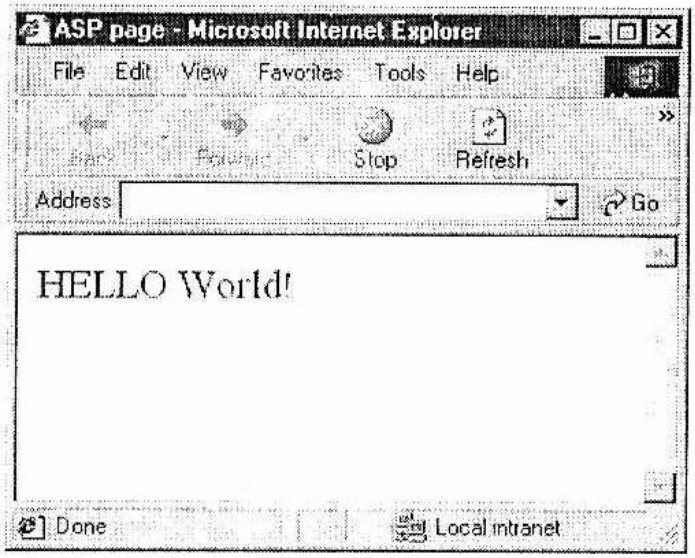
```
<HTML>
<HEAD><TITLE>ASP page</TITLE></HEAD>
<BODY>
<%
Respone.Write("HELLO World!")
%>
</BODY>
</HTML>
```

فایل ASP را با نام Test.asp ذخیره کنید و مطمئن شوید که ویرایشگر عبارت دیگری به نام فایل اضافه نمی کند . و همچنین مطمئن شوید در آن فهرست خاص با Premission ای که شما تعیین کرده اید که می توانید

Script یا Execute با هر، ذخیره شده است. حال سعی کنید این ASP را در مرورگر خود نمایش دهید. اگـر آن را درون یک فهرست ریشه از سایت وب خود ذخیره کرده باشید با تایپ http://mymachin/test.asp می توانیـد بـه آن دسترسی داشته باشید، اگر آن را درون یک زیرشاخه از ریشه اصلی ذخیـره کـرده باشـید بـاید مسیر کـامل آن را بصورت http://mymachin/test.asp بدهید. اگر آنرا درون یک فهرست مجازی با نام myvirtualdir ذخیره کرده باشید باید آدرس را به صورت زیر بنویسید .

http://mymachin/myvirtualdir/ test.asp

اگر همه این کارها انجام شودASP باید نمایش داده شود و متن Hello world! روی صفحه مرورگر قرار گیرد.



شکل ۲-٤ اجرای برنامهٔ ASP

۲-٦-۲ عیب یابی ASP هایی که تنظیم شده اند

البته امیدواریم که شما هیچگـاه نیـازمند مطالعهٔ ایـن قسـمت نشـوید ولـی اگـر در نمـایش ASP دچـار مسـائل و مشکلاتی شدید این قسمت به شما کمک خواهد کرد. در زیر لیستی از مشکلاتی که هنگام کار باASP ممکن اسـت به آن بربخورید آمده است :

مشکل : هنگامی که سعی می کـنید یکASP را روی صفـحه بیاورید پیغامی دریافت می کـنید که می گوید مرورگر نمی تواند به سرویس دهنده وصل شود .به عنوان مثال وقتی ازNetscape Navigator استفاده می کنید، این پیغام را دریافت می کنید :

There was response. The server could be down or is not responding

یا وقتی ازInternet Explorer استفاده می کنید ممکن است این پیغام را دریافت کنید

Internet Explorer cannot open The Internet site http://mymachine/test.ASP

یعنی اتصال با سرویس دهنده برقرار نشده است .

علت : اگرسرویس دهنده خود را بهInternet وصل نکرده باشیم این مشکل اغلب به وجود می‌آید پس می توان نتیجه گرفت که سرویس دهنده وب خاموش است . باید مطمئن شوید که سرویس دهنده وب در حـال اجـرا اسـت . برای انجـام ایـن کـارInternet Server Manager را از مجموعـه برنامـه Microsaft Internet Information Server از منوی start انتخاب کنید و نام پیش فرض سایت وب را انتخاب کنید و((VCR control)) را در بـالای

صفحه چک کنید . اگر IIS در حال اجرا نباشد Runbutton غیر فعال است برای اینکه سرویس دهنده راه اندازی شود روی این دکمه کلید Mouse را فشار دهید.

علت دیگر : هنگامی که شما سعی می کنید مرورگر خودرا به اینترنت وصل کنید ممکن است پیغامی مبتنی بر اینکه ترافیک اینترنت سنگین است دریافت کنید . مدتی صبر کنید و دوباره سعی کنید اگر هنوز نتوانسته اید به سرویس دهنده وصل شوید ، با ISP خود تماس بگیرید و مشکل را با آنها مطرح کنید.

مشکل : هنگامی که سعی می کنید ASP را روی صفحه بیاورید پیغام زیر را دریافت می کنید:

ttp/1.0 404 object Not found

علت : آدرس غلطی است که برای ASP خود درون مرورگر قرار بکاربرده اند. اگر ASP خود را در ریشه سایت وب ذخیره کرده اید می‌توانید این صفحه را با نوشتن این آدرس بدست آورید : http://mymachin/test.ASP اگر ASP خود را در یک فهرست مجازی ذخیره کرده‌اید باید نام آنرا در آدرس خود بنویسید به صورت زیر:

http://mymachin/myvirtualdir/ test.asp

علت دیگر: ممکن است ویرایشگر متنی عبارت اضافی به فایل چسبانده باشد به عنوان مثال test.txt ذخیره کرده باشد.

مشکل : هنگامی که سعی می کنید ASP را نمایش دهید ممکن است این پیغام را دریافت کنید

Http/1.1 403 Access Forbidden.Execte Access Denied

علت : اجازه تنظیم کردن فهرست هایی که فایل ASP در آن ذخیره می‌شود داده نشده است .

مشکل : هنگامی که می‌خواهید یک ASP را نمایش دهد متن اسکریپت آن روی صفحه نمایش داده می‌شود.

علت : فایل ASP خود را با پسوند htm. یا htm1. به جای .asp ذخیره کرده‌اید . یک فایل ASP باید پسوند .ASP داشته باشد .

علت : آدرس آن را درون خط آدرس قرار نداده اید و فقط آن را صدا زده‌اید.

فصل سوم :
کار کردن با ASP تکی

در این فصل به تفضیل در مورد چگونگی کار با یک ASP صحبت خواهد شد. اولین بخش نگاهی اجمالی بر اشیاء Response و Request دارد. قسمت بعدی طریقه بافر کردن خروجی ASP رانشان می دهد. در قسمت سوم برخی متدها ی کار با اسکریپتها با اجرای طولانی[1] و صفحات HTML طولانی را توضیح می دهد.قسمت انتهایی فصل نیز شما را با سرآیند های HTTP و متغیرهای سرویس دهنده آشنا می کند.

۳-۱ اشیاء Response , Requst

این دو شیء را اغلب در برنامه های ASP استفاده می کنید .قبل از آشنایی با این دو شیء باید پیش زمینه ای در رابطه با پروتکل HTTP داشته باشید.

پروتکل HTTP

واقعاً تارجهان گستر[2] به چه صورت کار می کند؟ وقتی آدرس یک صفحه را در مرورگر تایپ می کنید ، اگر همه چیز فراهم باشد ، آن صفحه نمایش داده می شود. به عنوان مثال اگر آدرس سایت Hotwired را در مرورگر خود تایپ کنید صفحه خانگی[3] سایت Hotwired روی صفحه مرورگر شما ظاهر می شود. برای این کار چه اتفاقی می افتد؟ هنگامی که شما از یک مرورگر برای بدست آوردن یک صفحه HTML استفاده می کنید از پروتکل HTTP استفاده می کنید. پروتکل HTTP مشخص می کند که پیغام ها به چه صورت روی اینترنت انتقال داده شدند. این پروتکل بهترین راه برای ارتباط مرورگر و سرویس دهنده وب می باشد.

وقتی یک صفحه را از سایت وب دریافت می کنید، مرورگر با سرویس دهنده وب ارتباط برقرار کرده و درخواست فرستاده می شود. سرویس دهنده وب درخواست را دریافت و به آن پاسخ می دهد. به همین دلیل پروتکل HTTP پروتکل درخواست و پاسخ نیزنامیده شده.

تمام ارتباط بین مرورگر و سرویس دهنده وب به وسیلهٔ این درخواست و پاسخ ها انجام می شود. یک مرورگر اغلب با فرستادن یک درخواست این ارتباط را آغاز می کند. سرویس دهندهٔ وب نیزبطور کامل غیرفعال است و با درخواست فعال می شود.

درخواست مرورگر دارای ساختار خاصی است. یک پیغام درخواست شامل خط درخواست ،فیلدهای سرآیند و احتمالاً بدنهٔ پیغام است. متداول ترین نوع درخواست ، درخواستی ساده برای صفحهٔ وب همانند مثال زیر می باشد :

```
GET/Hello.HtmHTTP/1.1
Host =www.Aspsite.Com
www.Aspsite.Com
```

این پیغام درخواست ، درخواستی برای صفحه وب Hello.Html از سایت وب www.Aspsite.Com است. اولین خط، خط درخواست است. خط درخواست ، متد درخواست و منابع درخواستی و نسخه پروتکل HTTP ای که بکار برده می شود را مشخص می کند.

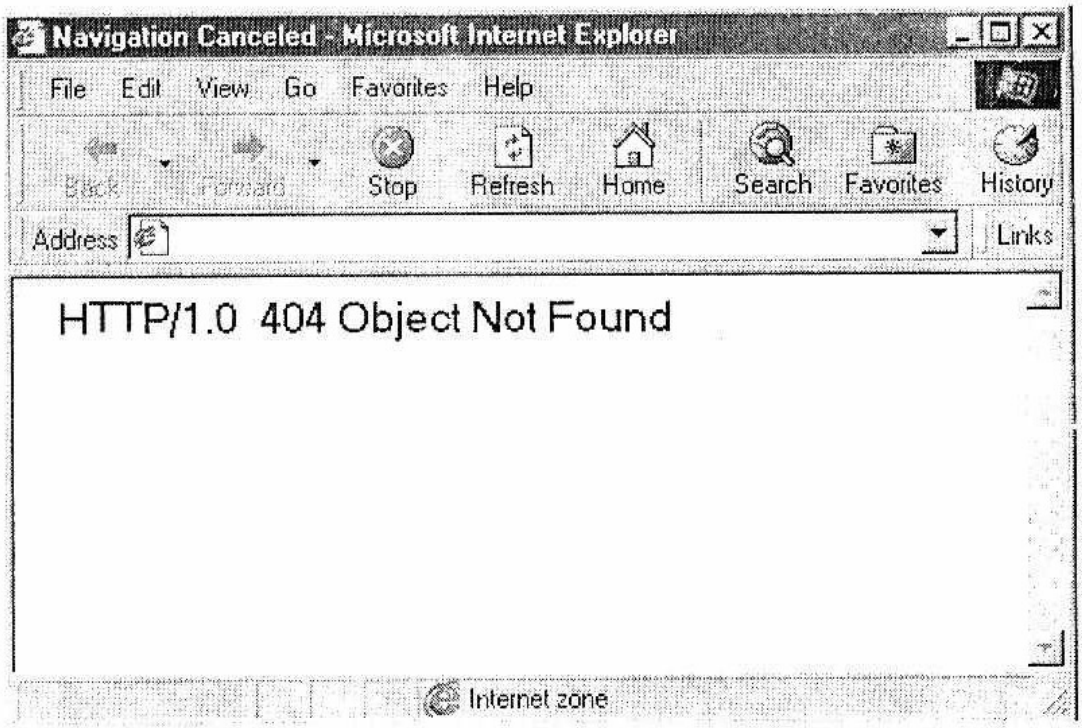
در این مثال متد درخواست GET است. متد GET منابع خاصی را دریافت می کند. در این جا ، متد GET برای بدست آوردن صفحهٔ وب Hello.htm استفاده می شود. دیگر متدهای درخواست ، DELETE,OPTIONS HEAD,POST,PUT,TRACE می باشند. که بطور متداول تنها از متدهای Get و Post استفاده می شود. متد Post هم برای ارسال محتویات یک فرم HTML بکار می رود.

خط دوم این مثال همان سرآیند است. سرآیند میزبان برای مشخص کردن آدرس سایت وبی می باشد که .Hello htm در آن قرار دارد که در اینجا میزبان WWW. Aspsite.Com است.

به طور نوعی یک در خواست ممکن است شامل چندین سرآیند باشد. سرآیندها اطلاعاتی اضافی دربارهٔ محتوای پیام یا منشاء درخواست را مهیا می کنند. بعضی از این سرآیندها استاندارد می باشند و بقیه توسط مرورگر تعریف می شوند. یک درخواست ممکن است حاوی بدنه پیغام نیز باشد. به عنوان مثال ، اگر درخواست متد Post را بجای Get بکار برده باشد ممکن است بدنهٔ پیام شامل محتویات یک فرم HTML باشد.

هنگامی که در فرم از آرگومان Action="Post" استفاده می کنید ، اگر کلید Submit را در فرم HTML بزنید، داده های درون فرم برای سرویس دهنده ارسال می شوند. با استفاده ازمتد Post ، محتویات فرم ارسال می گردد. وقتی سرویس دهندهٔ وب یک درخواست را دریافت می کند پاسخی را برمی گرداند. یک پاسخ هم دارای ساختار خاصی است هر پاسخ با خط وضعیت[1] که مشتمل بر تعدادی سرآیند (بصورت اختیاری) و بدنه پیغام می باشد ، شروع می شود. احتمالاًشما می دانید خط وضعیت چیست. خط وضعیت همان پیغامی است که مثلاً وقتی آدرسی را اشتباه وارد می کنید روی صفحهٔ مرورگر می بینید (مراجعه شود به شکل ۳-۱)

یک خط وضعیت شامل پروتکل مورد استفاده، کد وضعیت و (به صورت اختیاری) بدنهٔ پیام می باشد. به عنوان مثال ، اگر سرویس دهندهٔ وب مشکلی با درخواست داشته باشد یک پیغام خطا به همراه توضیحاتی در مورد آن پیغام خطا را به خط وضعیت برمی گرداند. اگر سرویس دهندهٔ وب با موفقیت به یک درخواست صفحه وب پاسخ دهد خط وضعیتی حاوی 200 ok را برمی گرداند.



شکل ۳-۱ مثالی از خط وضعیت

سرآیندهای پاسخ شامل اطلاعاتی دربارهٔ محتوای پاسخ و اطلاعاتی دربارهٔ سرویس دهنده ای که این پاسخ رافرستاده است می باشند. بعضی از این سرآیندها استاندارد هستند و بقیه به سرویس دهنده وب بستگی دارند.

بدنه پیغام پاسخ شامل محتویات صفحه وب است. به عنوان مثال ، اگر درخواست برای صفحه وب Hello.htm باشد، بدنه پیغام پاسخ باید حاوی Hello.htm باشد. گر چه بدنه پیغام می تواند حاوی سندهای دیگری (همانند Text , Document و غیره) نیز باشد ولی همهٔ این فایلها به خوبی HTML منتقل می شوند.

Response , Request

ASP حاوی دو شیء تعبیه شده هستند که مشابه پیغام درخواست و پاسخ در پروتکل HTTP عمل می کند. شیء Request مشابه درخواست در HTTP و شیء Response مشابه پاسخ در HTTP است. مانند تمام اشیاء ASP دو شیء Response , Requeat دارای مجموعه ها ، خواص و متدهایی می باشند.

با استفاده از مجموعه ها ، خواص و متدهای شیء Request می توانید تمام اطلاعات دررابطه با درخواست مرورگررا بدست آورید و به سرویس دهنده وب بفرستید و با استفاده از مجموعه ها ، خواص و متدهای شیء Responseمی توانید تمام اطلاعات مربوط به پاسخ سرویس دهنده وب رابه مرورگر بفرستید و می توانید تمام مدلهایی را که سرویس دهنده وب پاسخ می دهد کنترل کنید. به طور مثال، شیء Response حاوی تعدادی متد است که می توانید سرآیندهای پاسخ HTTP را تغییر دهد.

۳-۲ بافر کردن خروجی

معمولاً وقتی یک ASP در سرویس دهنده اجرا می شود صفحه خروجی بعد از اجرای هر کدام از دستورات در سرویس دهنده به روی مرورگر نمایش داده می شود. به عنوان مثال ASP زیر را در نظر بگیرید.

```
<HTML>
<HEAD> <TITLE> Buffer Example </ TITLE> </ HEAD>
<Body>
```

```
For i =1To 500
 NEXT
 %>
</ BODY>
</ HTML>
```

این اسکریپت اعداد 1 تا 500 را از بالا به پایین در مرورگر نمایش می دهد. صفحهٔ خروجی دقیقاً بعد از اجـرای هر دستور به مرورگر فرستاده می شود و می توانید هر عددی که ظاهر می شود را در همان لحظه می بینید. در بعضـی از مواقع شاید بخواهید خروجی ASP را بافر کنید. وقتی خروجی ASP را بافر می کنید تـا وقتـی پـردازش صفحـات درسرویس دهنده به صورت کامل پایان نیافته هیچ صفحه ای به مرورگر فرستاده نمی شود.

مثال زیر همان اسکریپت بالا است که براساس مطالب گفته شده تغییر یافته است.

```
<%Response . Buffer = Ture%>
<HTML>
<HEAD>
<BODY>
%>
FOR i =1 To 500
Response. Write (I &"<BR>")
NEXT
%>
</ BODY>
</ HTML>
```

بین این اسکریپت وقبلی تنها یک تفاوت وجود دارد اینکه در خط اول به خاصیت Buffer از شیء Response مقدار True داده شده است. وقتی این صفحه در مرورگر نمایش داده می شود کل محتوای صفحه بـه صـورت یـک جا به مرورگر فرستاده می شود و صفحه تا وقتی پردازش اسکریپت پایان یابد ، بافر می شود.

هر گاه بخواهید بافر را فعال کنید این کار را باید قبل از هر نوع خروجی اسـکریپت یـا HTML ، انجـام دهـد. در غیر اینصورت، پیغام خطا دریافت می کنید.

با بافر کردن یک صفحه می توانید صفحات مختلفی را با توجه به شرایط مختلف بدست آورید به عنـوان مثـال ، در زیر می توان دو صفحه مختلف به صورت تصادفی ایجاد کرد.

```
<% response . Buffer = True%>
<HTML>
<HEAD><TITLE> First Page </ TITLE> </ HEAD>
<BODY>
This is the first page
</ BODY>
</ HTML>
<% Randomize
 If int(2*RND)=1 Then Response.End
 Else
 Response.Clear
 End If
<HTML>
<HEAD> <TITLE> Second page </ TITLE> </ HEAD>
<BODY>
This is the second page.
</ BODY>
```

```
</ HTML>
```

در این مثال ، دو متد از شیء Response بکار می رود که عبارتند از متد Endو متد Clear. متد End فوراً پردازش یک صفحه ASP را متوقف و نتیجهٔ پردازش را ارسال می کند. باید توجه کرد که متد End را بدون توجه به اینکه از بافر استفاده کرده ایم، می توان بکار برد. در این مثال متد End تا وقتی صفحهٔ اول نمایش داده نشده، مانع نمایش صفحهٔ دوم می شود. متد Clear بافر صفحه جاری را بدون استخراج محتوای بافر خالی می کند. از متد Clear وقتی که خروجی ASP بافر شده باشد ، استفاده می کنیم. در مثال بالا متد Clear مانع از نمایش صفحه اول تا زمانی که صفحه دوم در حال نمایش است ، می شود وسپس صفحهٔ اول را از بافر پاک می کند. عموماً یکی دیگر از متدهای شیء Response نیز هنگام بافر کردن یک ASP بکار می رود. متد Flash محتویات بافر صفحه را مستقیماً خارج می کند. همانند متد Clear اگر شما مبادرت به استفاده از این متد با صفحه ای که بافر نیست بکنید ، پیغام خطا دریافت می کنید و برخلاف متد End پس از فراخوانده شدن متد Flash پردازش صفحه ادامه می یابد.

همیشه لازم نیست برای خروجی ASP از بافر استفاده کنیم . مثال زیر را درنظر بگیرید: در این مثال برنامهٔ قبل بدون استفاده از بافر نوشته شده است.

```
<%
Randomize
IF INT(2*RND) =1 Then
%>
<HTML>
<HEAD> <TITLE> First page</TITLE></HEAD>
</BODY>
</HTML>
<%ELSE%>
<HTML>
<HEAD> <TITLE> Second page </ TITLE> </ HEAD>
<BODY>
This is the second page
</ Body>
</ HTML>
<%END IF %>
```

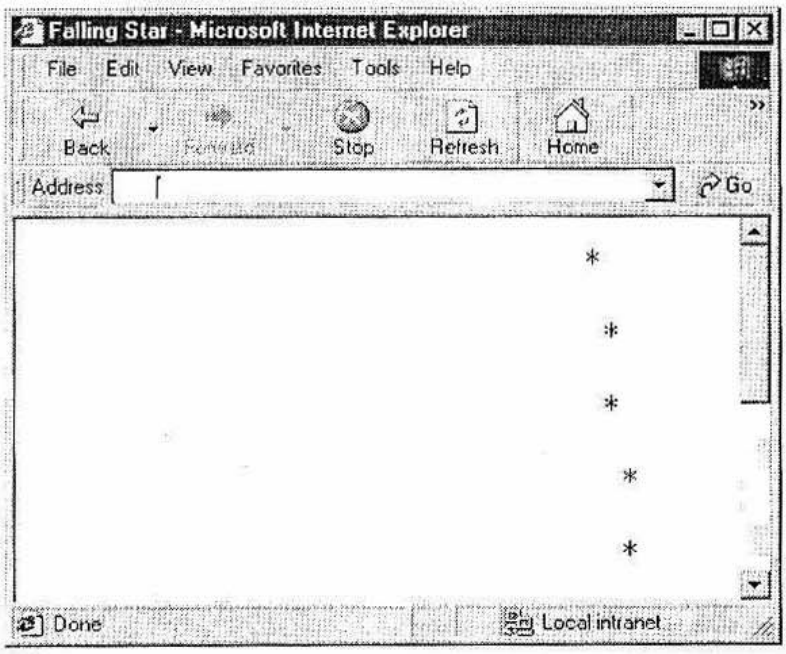
۳-۳ کاربا اسکریپت هایی با اجرای طولانی و صفحات طولانی HTML

بنا به تعریف ، بیشرین زمانی که به اجرای یک اسکریپت درون ASP اختصاص می یابد 90 ثانیه می باشد. که این باعث جلوگیری از حلقه های بینهایت می شود. در بعضی موارد نیاز است که به دستورات اسکریپت زمانی بیشتر از 90 ثانیه اختصاص دهیم. به عنوان مثال وقتی که می خواهیم از اسکریپت برای خروجی یک HTML طولانی استفاده کنیم و نمی خواهیم تا قبل از آنکه پردازش به صورت کامل تمام شود، وقت به پایان برسد. می توانیم با استفاده از خاصیت Script Timeout متعلق به شیء Server حد اکثر زمان مورد نیاز برای اجراء را کنترل کنیم.

```
<HTML>
<HEAD> <TITLE> Falling Star </ TITLE> </ HEAD>
<BODY>
<% Server.Scripttimeout=150
RANDOMIZE
Starx = 60
FOR K =1 To 10
```

```
NextScond = DATEADD ( " S " , 10 , Time )
Do WHILE TIME<NextSecond
Starx = Starx + 3 * RND ( ) –1
FOR i =1 To Starx
Response.Write ( "   ;" )
NEXT
Response . Write ( " *<p>" )
NEXT
%>
</ BODY>
</ HTML>
```

در این برنامه زمان اجرا اسکریپت بطور معمول قبل از اینکه افتادن ستاره ها پایان یابد ، تمام شود. به هر حال خط اول این صفحه از وقوع این حالت جلوگیری می کند ، چرا که خاصیت ScriptTimeout از شیء Server روی زمان انقضای[1] 150 ثانیه تنظیم شده است.



شکل ۳-۲ برنامهٔ نمایش ستاره ها

باید توجه کرد که نمی توان از خاصیت Server.scriptTimeout برای کاهش زمان انقضای اسکریپتها کمتر از 90 ثانیه استفاده کرد. برای اینکه اسکریپت خود را وادار کنید که قبل از 90 ثانیه خاتمه یابد لازم است که خاصیت ScriptTimeout را توسط Internet Service Manager تغییر دهید. این خاصیت در گزینه های ASP در منوی Application Configuration قرار دارد. اگر شما خاصیت ScriptTimeout را روی 1 تنظیم کرده باشید زمان اسکریپت شما هرگز خاتمه نخواهد یافت.

اگر به اسکریپت ها اجازهٔ اجرا برای دوره های زمان طولانی را بدهید ، این امکان وجود دارد که سرویس دهندهٔ وب شما دچار کمبود منابع شود. درواقع ممکن است یک اسکریپت درون یک ASP حتی بعد از اینکه شخص درخواست کنندهٔ ASP از آن بیرون رفت ، به اجرای خود ادامه بدهد. در این صورت اسکریپتی که به اجرای خود ادامه می دهد برای هیچ کس فایده ای نخواهد داشت. خوشبختانه خاصیتی از شیء Response وجود دارد که می

تواند به ما در این مورد کمک کند. خاصیت Is Client Conneted می تواند بررسی کند که آیا هنوز ارتباطی بین مرورگر و سرویس دهنده برقرار است یانه. برای مثال اسکریپت مثال بعد اجرا را تا هنگامی که زمان اسکریپت منقضی شود و یا اینکه ارتباط مرورگر از سرویس دهنده قطع شود ، ادامه می دهد.

```
<HTML>
<HEAD><TITLE> Obnoxious page </ TITLE> </ HEAD>
<BODY>
<%
WHILE  1 =1
Response.Write ( "Hello . How Are you?" )
If not Response.IsClientConnected THEN Response.End
WEND
%>
</ BODY>
</ HTML>
```

نکتهٔ قابل ذکر این است که خاصیت IsClientConnected تنها وقتی عکس العمل نشان می دهد که مرورگر تا زمان فراخوانی متد Response.Write آخر وصل باشد. اگر شما یک اسکریپت با زمان طولانی که هیچ چیزی را به مرورگر برنگرداند داشته باشید، آن موقع خاصیت مزبور بی فایده خواهد بود.

۳-۴ کار با سرآیندها و متغیرهای محیطی

همانطور که قبلاً گفتیم هم درخواست های مرورگر و هم پاسخ های سرویس دهنده حاوی سرآیندهایی است که در فصل اول در بخش Cgi توضیح داده شدند سرآیندها اطلاعاتی اضافی در رابطه با محتوی درخواست یا پاسخ فراهم می کنند. همچنین می توانند حاوی اطلاعاتی در رابطه با مرورگر یا سرویس دهنده باشند.

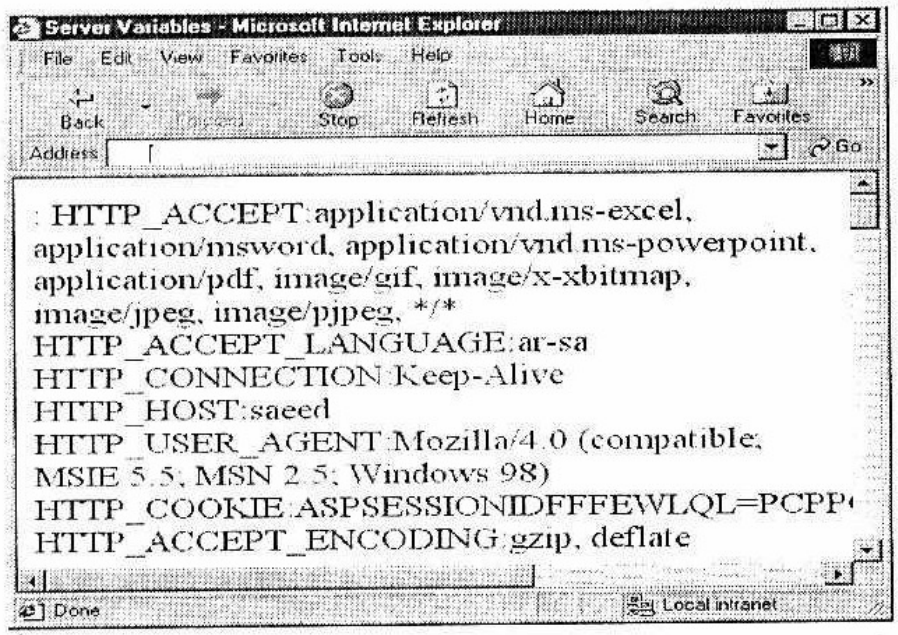
ASP دارای یک سری مجموعه ها و متدهایی است که با استفاده از آنها می توانیم سرآیندها و متغیرهای محیطی کار کنیم.

۳-٤-۱ دریافت سرآیندها

وقتی مرورگر درخواست یک صفحه وب از سرویس دهنده می کند این درخواست شامل تعدادی از سرآیندها می باشد. با استفاده از یکی از مجموعه های شیء Request به نام ServerVariables می توان این سرآیندها را بدست آورد. مجموعه ServerVariables حاوی سرآیندها و آیتمهای اضافی از اطلاعات اضافی در رابطه با Server است.

برنامهٔ زیر از محتویات مجموعهٔ ServerVariables برای مرورگر یک کپی می فرستد. (رجوع شود به شکل ۳-۳)

```
<HTML>
<HEAD> <TITLE> Server Variables </ TITLE> </HEAD>
<BODY>
<%
FOR EACH Name IN Request.ServerVariables
Response.Write( "<P><B>"&"</B>: ")
Response.Write (Request.ServerVariables (Name))
NEXT
%>
</BODY>
</HTML
```



شکل ۳-۳ مجموعهٔ ServerVariables

سرآیندها ومتغیرهای سرویس دهنده در مجموعه Server Variables دسته زیادی از اطلاعات را نگه می دارند در لیست زیر بعضی از مهمترین های آنها آمده است.

- HTTP_REFERER : هر گاه کسی با کلیک روی لینکی از صفحهٔ جاری خود به صفحهٔ موردنظر بیاید این سرآیند حاوی اطلاعات صفحهٔ جاری خواهد بود. با این سرآیند می توان فهمید که بینندگان صفحات شما از کجا به سایت شما وارد شده اند. یعنی مثلاً اگر کاربر با زدن لینکی از سایت YAHOO به سایت شما رسیده اند ، HTTP _ RE FERER حاوی آدرس YAHOO خواهد بود.
- HTTP_USER_AGENT : نوع مرورگری را که کاربر ملاقات کننده سایت وب استفاده کرده است مشخص می کند.
- REMOT_ADDR : این سرآیند آدرس IP کسی است که از سایت وب شما دیدن می کندرا نگه می دارد.
- QUERY_STRING: همانطور که در بخشهای قبلی نیز گفته شده بود این متغیر محیطی که شامل قسمتی از URL است که بعد از علامت سؤال قرار دارد. هنگامی که از متد Get استفاده شده باشد از اطلاعات فشرده شدهٔ Form در آن قرار دارد.
- SCRIPT_NAME : این متغیر سرویس دهنده شامل مسیر مجازی ASP است. از این متغیر برای بدست آوردن ASP ها استفاده می کنند.
- SERVER_NAME : این متغیر سرویس دهنده شامل آدرس اینترنتی سرویس دهنده است.
- PATH_TRANSLATED :این سرآیند تعدادی از رنگهایی که توسط مرورگر می تواند نمایش داده شود را نشان می دهد.

- HTTP_UA _CPU :این سرآیند نوع ماشینی را که مرورگر روی آن نصب شده است را نشان می دهد.
- HTTP_UA_OS : این سرآیند نوع سیستم عاملی را که مرورگر روی آن نصب شده است را نشان می دهد.
- HTTP _ UA _ PIXELS : این سرآیند نوع صفحه ای را که مرورگر روی آن اجرا می شود نشان می دهد.

می توان محتوای این متغیرها را با قرار دادن اسم آنها درون مجموعه ServerVariables بدست آورد . به عنوان مثال ، در اسکریپت زیر کسانی می توانند به این صفحه دسترسی داشته باشند که از اسکریپت Origin.ASP به این صفحه بیایند.

```
<HTML>
<HEAD> <TITLE> Server Variable </ TITLE> </ HEAD>
<BODY>
<%
Where Form = Request.ServerVariable ( "HTTP_REFERER" )
IF Where Form = "Http:// WWW.Mysite.Com / Origin. Asp" THEN
%>
Welcome to This page !
<%
ELST
%>
You are not authorized to viem this page !
<%
END IF
%>
</ BODY>
</ HTML>
```

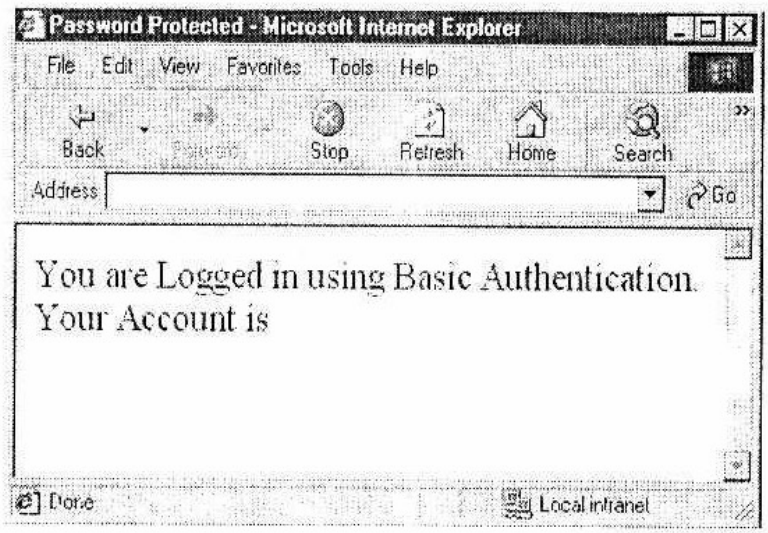
در این اسکریپت سرآیندHTTP_REFERER مسیر صفحه ای را که کاربر از آن به صفحهٔ موجود وصل شده است ، نشان می دهد. اگر کاربر از مسیر Http:// Mysite/Origin.Asp استفاده نکند اجازه دیدن صفحه به آن داده نمی شود.

۳-٤-۲ بکار بردن سرآیندها برای کنترل چگونگی Cache شدن صفحات

سرویس دهندهٔ پروکسی[1] باعث بالا رفتن سرعت دسترسی به صفحات وب می شوند. این سرویس دهنده هایک کپی محلی از صفحات وب را درون Cache نگهداری می کنند وقتی کسی صفحه وبی را مجدداً درخواست می کند می تواند آنرا به جای سرویس دهنده اصلی از پروکسی بدست آورد.

برای ملاحظهٔ ASP ها ، سرویس دهندهٔ پروکسی مایه دردسر هستند چرا که تمام نکتهٔ یک ASP نمایش محتویات پویا و اثرپذیر (دینامیک) می باشد. به عبارت دیگر ، یک ASP می تواند صفحاتی را که در هر لحظه عوض می شوند ، نشان دهد. ولی در سرویس دهندهٔ پروکسی نمی توان کپی تمام این صفحات را داشته باشیم. سرآیندی متعلق به ASP با نام CACHE_CONTROLGeneral دستورالعملی را برای چگونگی

```
You are Logged in using Basic Authentication.
You are Logged in Using NT challenge and Response. Server Variables ("Logon _
user" )%>
<%ELSE%>
You are Logged in Using NT Challenge and Response You Account is
<% =Response.ServerVariables ( "LOGON _USER" )% >
<%END IF%>
</ BODY>
</ HTML>
```



شکل ۳-٤خروجی برنامهٔ بالا

۳-٤-۳ تغییر دادن سرآیند Content –Type

سرآیند Content -Type نشان دهندهٔ نوع رسانه های بدنه پاسخ است (نوع MIME) .به عنوان چند مثال معمولی می توان به "Application / Msword" , Image / Gif" , "Text / HTML" اشاره کرد. با استفاده از خاصیت Content - Type از شیء Response می توانید این سرآیند را تنظیم نمائید.

از خاصیت Content - Type به طور عمومی برای نمایش منابع اسناد HTML استفاده می شود. اگر خاصیت Type–Content را به صورت Text / Plain تنظیم کنیم ، بدنه پاسخ به جای HTML به صورت Text فرستاده می شود. مثال زیر را درنظر بگیرید.

```
<%
Response . Content Type = Text / Plain
%>
<HTML>
<HEAD> <TITLE> HTML Document </ TITLE> </ HEAD>
<BODY>
<H1> This is an HTML Document ! </H1>
</ BODY>
</ HTML>
```

وقتی این فایل نمایش داده می شود تمام متنی که در زیر خط اسکریپت قرار دارد به همان صورت به روی مرورگر نمایش داده می شود. با این کار آن را از حالت HTML درآورده ایم.

۳-۵ کد وضعیت

خاصیت Status برای تعریف کد وضعیتی است که پاسخ HTTP باز می گرداند. وقتی که سرویس دهنده ای به یک درخواست پاسخ می دهد ، اولین خطی که فرستاده می شود خط وضعیت[1] است. خط وضعیت شامل سه خانهٔ کد وضعیت و توضیحی درباره کد وضعیت (که Response نامیده می شود) می باشد. در زیر ۵ کلاس از کدهای وضعیت نمایش داده شده است.

- Information 1xx ، این کد وضعیت در این کلاس آزمایشی است.
- Success 2xx ، کد وضعیت در این کلاس برای نشان دادن موفقیت درخواست بکار می رود. به عنوان مثال کد 200 نشان می دهد که صفحه وب مورد درخواست با موفقیت دریافت شده است.
- Redirection 3xx کد وضعیت در این کلاس برای نشان دادن این است که قبل از اینکه درخواست بتواند اجرا شود باید عمل دیگری انجام شود. به عنوان مثال کد وضعیت 301 نشان می دهد که صفحه وب برای همیشه به آدرس دیگری منتقل شده است ، در این حالت مرورگر به صورت خودکار به آدرس جدید برمی گردد.
- Cline Error 4xx این کد وضعیت هنگامی که مرورگر درخواستی را تولید کرده باشد که نتواند انجام شود، بازگردانده می شود. به عنوان مثال کد 404 موقعی که صفحه وب وجود نداشته باشد بازگردانده می شود.
- Server Error 5xx این کد وضعیت نشان دهنده مشکل در سرویس دهنده است. برای مثال کد وضعیت 503 نشانگر این است که سرویس دهنده خراب شده است.

می توان از خاصیت Status در شیء Response برای تعریف کد وضعیتی که باید در یک پاسخ بازگردانده شود استفاده کرد. به عنوان مثال اگر کسی در دریافت ASP زیر در روز چهارشنبه مشکل داشته باشد کد 407 Not Authorized برگردانده می شود. (این نتایج در جعبهٔ محاورهٔ کلمه عبور ظاهر می شود)

```
<%
IF WEEKDAYNAME (WEEKDAY (DATE)) = "Wednesday" THEN
Response.Status = "401 Not Authorized."
Response.End
ELSE
%>
<HTML>
<HEAD> <TITLE> Not Wednesday </ TITLE> </ HEAD>
<BODY>
 Welcome! Today is Not Wednesday.
</ BODY>
</ HTML>
<%END IF %>
```

فصل چهارم :
کارکردن با بیش از یک ASP

در این فصل شما یاد می گیرید که چگونه با چندین ASP کار کنید. در بخش اول این فصل شما چگونگی دریافت اطلاعات وارد شده در فرم های HTML را می فهمید. در بخش دوم برخی متدهای دریافت رشته های درخواست مطرح می شوند. در بخش سوم شما چگونگی ارجاع یک کاربر به صفحهٔ جدید را فرا می گیرید و در نهایت شما فرامی گیرید که چگونه یک ASP را در دیگری بگنجانید.

۴-۱ دریافت محتویات یک فرم HTML

بهترین راه برای گرفتن اطلاعات ازکسی که وارد سایت وب می شود یک فرم HTML است (به شکل ۱-٤ مراجعه شود). فرم HTML تنها فرمی است که با مرورگر کار می کند . فرض کنید می خواهید قبل از اینکه کسی از سایت شما دیدن کند از آن شخص ثبت نام کنید. برای این ثبت نام نیز می توانید از فرم HTML استفاده کنید . این فرم نام ونام خانوادگی ملاقات کننده را می پرسد . بعداز اینکه کاربر نام ونام خانوادگی خودرا وارد کرد کاربر می تواند دکمهٔ submit یا move را برای رفتن به صفحهٔ بعدی بزند. باید این اطلاعات از فرم دریافت شود. حال نیاز به متدی برای دریافت این اطلاعات می باشد. هنگامی که شما فرم HTML را می فرستید آن فرم همانند یک درخواست پروتکل HTTP ارسال می شود. در اینجا با استفاده از متد POST اطلاعات را به سرویس دهنده می فرستیم . شما اطلاعات را می توانید از شی ء Request از ASP دریافت کنید.

```
<HTML>
<HEAD> <TITLE>  Register  < /TITLE> </HEAD>
<BODY>
<H4> Registration :</H4>
<FORM  METHOD = "POST"  Action = "/regresults .asp">
<P> Please  Enter  Your  First  Name:
<BR> <INPUT  NAME="FirstName"  Type = "TEXT">
< p> Please  Enter Your  Last   Name:
<BR> <INPUT   NAME = "LastName"  Type  = "TEXT">
<INPUT    TYPE = " SUBMIT " valve = " continue">
</FORM>
</BODY>
</HTML>
</HTML>
```

شکل ٤-١ یک فرم ثبت نام ساده

شیء Request دارای مجموعهٔ Form می باشد که از طریق آن می توان که تمامی اطلاعات وارد شده در فرم HTML را دسترسی پیداکرد. هرکلیدی روی Form به منزله یک ورودی در فرم HTML می باشد . بطور نمونه در مثال بالا اگر کاربر فرمی را پر کرده و سپس ارسال کرده باشد مجموعهٔ Form شامل دو خط ورودی خواهد بود و اولین خط ورودی FirstName دومی LastName است.

به اسکریپت زیر توجه کنید :

```
<HTML>
<HEAD> <TITLE> Registration Results < /TITLE> </HEAD>
<BODY>
Thank you <%= Request.Form ("first Name ")%> for registering!
</BODY>
</HTML>
```

وقتی که این صفحه روی مرورگر نمایش داده می شود اسمی که کاربر درون فرم HTML وارد کرده است نمایش داده می شود. اطلاعات وارد شده در این Form به برنامهٔ Asp مشخص شده در Action ارسال می شود. باید بدانید که وقتی فرم انتشار می یابد شما قادر نیستید که به محتوای یک فرم HTML در صفحهٔ مشابه ای دسترسی پیداکنید. فرم HTML باید اول قبل از اینکه شما بتوانید مقادیر و اجزاء آنرا ارسال کنید با زدن دکمهٔ مربوطه ارسال شود. وقتی یک صفحه فرم HTML دارید باید یک صفحه اضافه نیز برای پردازش فرم داشته باشید. همین امر نشان می دهد که سایتهای وب اینترنت محدودیت زیادی در دادن سرویس دارند.هنگامی که فرم ثبت نام پردازش می شود یک صفحه شامل نتیجهٔ کار ساخته شده وبرای کاربر ارسال می شود.

٤-٢ کپی کردن محتوای مجموعهٔ Form

راههای زیادی برای دریافت محتوای داخل Form وجود دارد . اگر شما بخواهید مروری بر محتوای داخل Form داشته باشید و هر کدام را که لازم بود نمایش دهید باید از اسکریپتی مشابه اسکریپت زیر استفاده کنید :

```
<%
FOR EACH   name IN  Request.Form
 Response.Write("<BR>"& name&"=")
Response.write(Request.Form (name))
NEXT
%>
```

این اسکریپت محتویات داخل Form را نمایش می دهد.به عنوان مثال اگر بیل گیتز از سایت دیدن کند وفرم ثبت نام را پر کند خروجی این اسکریپت به صورت زیراست :

```
LASTNAME = Gates
FIRSTNAME =Bill
```

توجه کنید که در مثالهای بالا اسامی چطور به حروف بزرگ تبدیل شده اند . هنگامی که می خواهید یک آیتم را در فرم وارد کنید نمی خواهد نگران کوچک و بزرگ بودن آنچه که وارد می کنید باشید به عبارتی تفاوتی بین First Name و First name وجود ندارد. بجای استفاده از حلقه FOR....EACH می توان ازحلقه FOR....NEXT برای کپی کردن محتویات Form استفاده کرد. اسکریپت زیر مقادیر هر جزء داخل Form را بر می گرداند.

```
<%
FOR i = I TO  Request.Form.Count
Response.write ("<BR>"& Request . Form (i))
NEXT
%>
```

در این اسکریپت خاصیت Count مربوط به مجموعهٔ Form برای تعیین تعداد اجزاء Form مجموعه استفاده می شود . با استفاده از خاصیتCount هنگامی که فرم ارسال می شود می توان تعداد، فیلدهای فرم HTML ای که توسط کاربرپرشده است را بدست آورد .

در آخر اگر می خواهید عناصر موجود در مجموعه Form رابا یک رشتهٔ کدشدهٔ URL تکی برگردانید می توانید از اسکریپت زیر استفاده کنید :

```
<%= Request.Form %>
```

خروجی این اسکریپت به صورت زیراست:

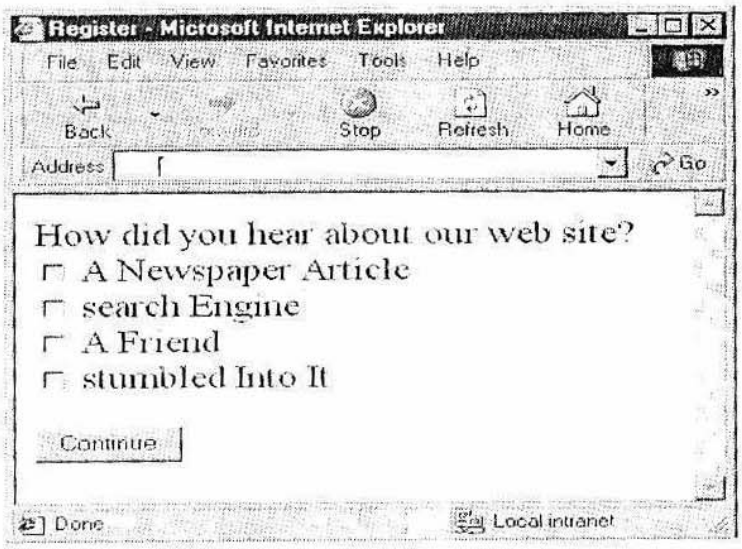
```
FirstName = Bill & LastName  = Gates
```

۴-۳ عناصر فرم با چندین مقدار

عناصر موجود در یک فرم HTML می توانند چندین مقدار داشته باشند . به عنوان مثال هم check Box وهم list Box ها می توانند چندین مقدار مختلف داشته باشند :

```
<FORM  METHOD = "POST "   ACTION = " /regresults.asp">
How did you hear  about  our  web site?
```

```
<BR> <INPUT NAME="Howhear" Type ="CheckBox " VLUE =" A Newspaper">
A Newspaper Article
< BR><INPUT NAME =" How Hear "  Type =" checkBox " value ="A Friend">
search Engine
< BR> <INPUT  NAME= "HowHtear"  Type =" checkBox" VALUE=" Afriend">
A Friend
 <BR> <INPUT  NAME = "HowHear"  Type ="checkBox "  VALUE=" stumbled Into It">

stumbled Into It
< P> <INPUT  TYPE =" SUBMIT"  VALUE = "Continue">
</FORM>
```



شکل ٤-٢ فرمی با چندین مقدار

این فرم را می توانید برای اینکه بفهمید کاربران سایت وب شما آنرا چطور پیدا کرده اند مورد استفاده قرار دهید. ولی ممکن است کسی از طرق مختلفی به سایت شما رسیده باشد. به عنوان مثال هم فرم friend وهم فرم newspaper را خواسته باشید ، این فرم به کاربر اجازه می دهد که بیش از یک مقدار را دریک زمان انتخاب کند. (مراجعه شود به شکل ٢-٤)

به چه صورت می توان مقادیر عناصری را که بیش از یک مقدار دارند بدست آورد؟ پاسخ این است که شما می توانید از یک پارامتر اضافی مجموعهٔ Form استفاده کنید . به مثال زیر توجه کنید:

```
<HTML>
<HEAD> <TITLE> Your Response < /TITLE> </HEAD>
<BODY>
Accding  to your respone ,  you heard about this web site in
Request.form ("How Hear"). count %> ways =/.>
<P>  you heard about this form
<%
```

```
FOR  EACH  way IN Request.farm("How Hear")
Response.write ("<P>" & way")
NEXT
%>
</BODY>

</HTML>
```

در این اسکریپت خاصیت Count برای بازگرداندن شماره جعبه های انتخاب شده به کار می رود . در این مثال، خاصیت Count تعداد کلی عناصر مجموعهٔ Form را باز نمی گرداند. به جای آن فقط شماره کلی مقادیر عنصر فرم کهHowHear نامیده می شود را با ز می گرداند. حلقه تکرار FOR....EACH به تعداد مقادیر تکرار می شود. در این مثال اگر کسی هم friend وهم Newspaper را انتخاب کند هردومقدار نمایش داده می شود .

۴-۴ text Area ها و مجموعهٔ Form

شما می توانید متنی را که درون Text Area قرار دارد، به همانند سایر عناصر فرم دریافت نمائید. متغیرهای رشته درون VBScript می توانند کاملاً طولانی باشند. در این جا محدودیت ۲۵۵ کاراکتری که در زبانهای دیگر هست وجود ندارد.مثال زیر یک مجموعهٔ Form به همراه یک Text Area است :

```
<"FORM  METHOD ="POST" ACTION ="/Respoose .asp>
Please enter any feedback on this web site below:
<P>
<TEXTAREA  NAME = "Feedback" cols =30  ROWS=10></TEXTAREA>
< P> <INPUT  TYPE = " SUBMIT" VALUE =" Submit feedback">
</FORM>
```

این فرم HTML یک Text Area را که می تواند به عنوان خروجی کاربر روی سایت وب مورد استفاده قرار گیرد، نشان می دهد. هنگامی که کاربر روی کلید feedback کلیک می کند به روی صفحه Response.asp می آید. اگر می خواهید متنی را که درون Text Area نوشته شده است نمایش دهید مانند زیر عمل کنید :

```
<HTML>
<HEAD> <TITLE> Feedback Response </TITLE > </HEAD>
<BODY>
Thank you for submiting feedback   you wrote:
<P>
<%=Request.Form ("feedback")%>
</BODY>

</HTML>
```

۴-۵ برچسب ها و فرمهای HTML :

یک کاربر می تواند هر متنی را در یک خط ورودی HTML وارد کند. هیچ چیز کاربر را نمی تواند از وارد کردن برچسب های HTML باز دارد و این می تواند معایب و مزایایی داشته باشد.

در برخی حالتها شما می خواهید که یک کاربر با استفاده از برچسب ها قادر به قالب بندی متن فرمی که وارد کرده است باشد. مثلاً تصور کنید که می خواهید در سایت وب خود با استفاده از ASP ها و فرم های HTML،یک تابلوی بولتن[1] ایجاد کنید. کار با این تابلوی بولتن اگر متن را به رنگها و فونتهای مختلف نمایش دهید، برای کاربر جالبتر می شود و او رغبت بیشتری برای آن نشان می دهد. این کار می تواند به راحتی توسط برچسب های HTML صورت بگیرد.

در دیگر حالتها شاید شما مایل نباشید که قالب بندی HTML بکار برود. مثلاً بخواهید سایت وبی را طراحی کنید که برنامه نویسی را با استفاده از ASP به کاربر یاد بدهد. فرض کنید کسی یک کد نمونه را با استفاده از HTML ارسال کرد و شما می خواهید که برچسب های واقعی HTML نمایش داده شود و مرورگر آنرا تفسیر[2] نکند در این صورت به چه روشی باید عمل کنید ؟

خوشبختانه ASP دارای متدهای مخصوصی برای این منظور خاص می باشد. متد ( )Server.HTMLEncode برچسب های HTML را به کدهای کاراکتر HTML ترجمه می کند. مثال زیر چگونگی استفاده از این متد را نشان می دهد :

```
< %=Server.HTMLEncode  "<B> This is Bold </B>"
```

در حالت عادی اگر رشتهٔ "< B > This is bold </B>" توسط مرورگر دریافت شود متنی نمایش داده می شودکه برجسته و ضخیم خواهد بود ولی اگر قبل از آن با HTML Encode آنرا کد گذاری کنیم چیزی که توسط مرورگر نمایش داده خواهد شد همان رشته به همراه برچسب های مزبور خواهد بود.

۴-۶ بررسی وجود اجزاء فرم

بعضی وقتها باید بررسی کرد که شخص فرم را به صورت کامل پر کرده است یاخیر. به عنوان مثال یک فرم ثبت نام ساده یک سری فیلدهای خاص را لازم دارد . شما نمی خواهید اجازه بدهید که کاربر بعضی ازاین فیلد ها راخالی بگذارد.با اسکریپت زیر می توانید این مسئله رابررسی کنید .

```
<%
IF Request.Form ("firstname") =" " THEN
Response.write("you must enter your first  name.")
ELSE
Response.write("Thank you for regstering.")
END IF
%>
```

در این اسکریپت بررسی می شود که فیلد firstname دارای مقدار باشد .عنصر firstname با یک رشتهٔ تهی مقایسه می شود. اگر کاربر در وارد کردن نام خود کوتاهی کرده باشد از این مسئله آگاه خواهد شد. هنگامی که کاربر فرم اطلاعات را ناقص پر کرده باشد شما باید او را دوباره به آن فرم برگردانید. شما می توانید این کار را با گذشتن یک پیوند مثلاً با نام Back یا هر چیز دیگر انجام دهید. البته بهتر خواهد بود اگر بتوان این مسئله راخودکار کرد یعنی اگر فرم ناقص بود دوباره فرم قبلی بیاید.

۴-۷ دریافت یک رشته درخواست[1]

یک رشتهٔ درخواست قسمت اعظم URL را تشکیل داده و بعد از علامت سؤال نمایش داده می شود. اگر از موتورهای جستجوگر اینترنت مثل AltaVista استفاده کرده باشید با رشتهٔ درخواست آشنا شده اید .این کاراکترها را شما می توانید در خط آدرس مرورگرتان موقع جستجو ببینید. شما می توانید از رشتهٔ درخواست توسط یک پیوند برای فرستادن اطلاعات از یک صفحه به صفحه دیگر استفاده کنید به مثال زیر توجه کنید :

```
<HTML>
<HEAD> <TITLE> Query String Example  </TITLLE > </HEAD>
<BODY>
<A HREF = "http : //www.aspsite.com/newpag.asp?clink=yes ">Click    Me!</A>
</BODY>
</HTML>
```

دراین مثال پیوند به صفحه newpage.asp بر می گردد. هر چند که پیوند می تواند شامل یک رشتهٔ درخواست هم باشد ولی به هر صورت وقتی کسی روی کلمه click me! کلیک می کند . عبارت "click = "yes به صورت یک درخواست برای صفحه جدید فرستاده می شود .

می توانید رشتهٔ درخواست را مستقیماً با نوشتن آن درون خط آدرس مرورگر نیز بفرستید. این مسئله برای سرویس دهنده فرقی نمی کند. به عنوان مثال وارد کردن درخواست زیر در خط آدرس اثر کلیک کردن روی لینک را دارد.

HTTP://www.aspsite.com /newpage.asp? click =yes

رشتهٔ درخواست برای زمانی که شما می خواهید منویی از گزینه ها بسازید مفیدتر است. اگرچندین لینک دارید که به یک صفحه اتصال می یابند ، می توانید از رشتهٔ درخواست برای تعریف لینک مخصوصی که روی آن کلیک می کنید، استفاده کنید . مثال زیر را در نظر بگیرید :

```
<HTML>
<HEAD> <TITLE>product List </TITLE > </HEAD>
<BODY>
<H3> Welcom  To Our Store:   </H3>
```

```
Please select the item you want to purchase form the list belon:
<P><A HREF = " /Purcnas.ase ? ITEM=1" > Used Book </A>
<P><A HREF = " /Purchas.asp ? ITEM=2"> Broken Typewriter </A>
<P> <A HREF = "/Purchase.asp? ITEM=3"> Horseshoe </A>
</BODY>
</HTML>
```

دراین صفحه چندین عنصر که کاربر برای خرید می تواند آنها را انتخاب کند وجود دارد. درهرنمونه یک پیوند تعریف شده است. وقتی کاربری روی هر عضو کلیک می کند ، آن شخص به صفحه purchase.asp وصل می شود. (مراجعه شود به شکل ٤-٣)

شکل ٤-٣ صفحه ای با لیست عناصر فروش

برای روشن شدن انتخاب خرید شما ، باید هرپیوندی که روی آن برای اتصال کلیک می کنید، مشخص شود. تعریف کنید. هر پیوند به صورت منحصر به فردی توسط رشتهpurchase.aspشما باید آن را درون صفحه درخواست تعریف شده است . چگونگی دریافت این اطلاعات به این صورت است که رشته های درخواست به دارای مجموعه ای بنامASP درRequest فرستاده می شوند . جالب است بدانید که شیء HTTPوسیله درخواست بدست می آیند . برای بدست آوردنFormمجموعه است . عناصر در این مجموعه به همان صورتQueryString استفاده کنید .به مثال زیر توجه کنید:QueryStringیک رشتهٔ درخواست خاص شما به آسانی می توانید از اسم

```
<HTML>

<HEAD> <TITLE> Purchase  </TITLLE > </HEAD>

<BODY>

< %

 SELECT CASE Request.QuerySting ("item")
CASE"1"
Response.Write ("Thank you for purchasing a used book.")
CASE "2"
Response.Write ("Thank you for purchasing a broken typewriter")
    CASE "3"
Response.write ("Thank you for purchasing  a  horseshoe.")
END SELECT
% >

</BODY>

</HTML>
```

مجموعه QuerySting در مثال بالا هر پیوندی که روی آن کلیک شده است رامشخص می کند. عبارت SELECTCASE بسته به نوع رشتهٔ درخواست، پاسخی را اختصاص داده است. به عنوان مثال اگر کسی horseshoe را انتخاب کند از کاربر برای این انتخاب تشکر می شود .

٤-٧-١کد کردن یک رشتهٔ درخواست

یک رشتهٔ درخواست قبل از اینکه به صفحه دیگر فرستاده شود باید به صورت یک کد URL تبدیل شود . به عنوان مثال تمام جا های خالی به علامت جمع تبدیل می شود. اگر فراموش شود که رشتهٔ درخواست را به صورت کد URL تبدیل کنیم ممکن است نتیجه غلطی را بدست آوریم .

اتفاقاً در ASP تبدیل رشتهٔ درخواست به کد URL بسیار آسان است .متد ( )Server.URLEncode هررشته را به کد URL تبدیل می کند . به مثال زیر توجه کنید:

```
<A HREF = "/response .asp? Message = <%= server.URLEncode

("This query   string  hasbeen  URL   encoder . ")%>"> Click  Here  </A>
```

توجه داشته باشید که نام رشتهٔ درخواست وعلامت مساوی را تبدیل به کد نکنید با انجام این کار دچار مشکل می شوید . شما فقط باید مقدار رشته درخواست را تبدیل به کد کنید. رشته درخواستی که در مثال قبلی تبدیل به کد شده است به صورت زیر نوشته می شود .

```
Message = This + query+string+has+been +URL+encoded %2E
```

نباید نگران کد کردن رشته های خود باشید زیرا این کار را ASP به صورت خودکار انجام می دهد.به عنوان مثال صفحه response.asp که شامل خط زیر است را درنظر بگیرید .

```
<% =Request .QuerySting ("message")%>
```

خروجی این پیغام به صورت کد URL نمی باشد وشما می توانید آن را به صورت زیر ببینید .

```
This Query string  has  been  URL  encoded
```

٤-٧-٢ رشته های درخواست با پارامترها ومقادیر چندتایی

شما می توانید بیش از یک نام ومقدار را در رشتهٔ درخواست بفرستید . به عبارت دیگر شما می توانید رشته های درخواستی با چند پارامتر به وجود آورید . برای فرستادن چند پارامتر باید از کارکتر (&) استفاده کنید. مثلاً در رشتهٔ درخواست زیر دو پارامتر فرستاده شده است.

```
<A  HREF ="/Response .asp ?Firstparam= <%=Server .URLEncode ("this is the first
parameter .")%>&secondparam=<%server.URLEncode(this is the second parameter .
")>"%> Click Here </A>
```

این رشتهٔ درخواست شامل دو پارامتر است که Firstparam و Secondparam نامیده می شود . پارامتر اول مقدارش برابر با this is the first parameter وپارامتر دوم مقدارش برابر با This is the seconde paramter می باشد. متد ( )server.URLEncode مقدار این پارامتر را عوض می کند تا آنها بتوانند فرستاده شوند .در صفحه response.asp شما می توانید مقدار دو پارامتر رابه صورت زیر خارج کنید:

```
<P> <% = Request .QuerySting ("Firstparam")%>

<P><%= Request .QueryString ("second param")% >
```

با فرستادن نام هر رشتهٔ درخواست در مجموعه QueryString می توانید مقدار هر پارامتر را بدست آورید. خروجی دو جملهٔ قبلی به صورت خواهدبود.

```
This is the first parameter.
This is the second  parameter.
```

همچنین می توانید به یک پارامتر چندین مقدار را نسبت دهید، برای این منظور به راحتی می توانید یک نام را دوبار در رشته درخواست همانند مثال زیر بیاورید :

```
<A HREF ="/Response.asp?only param= <%=Server.URLEncode ("I am the   first value of
The only parameter.")%>& Only param = <%= Server .URLEncode  ("I am The second value
of The only parameter. ") %>"> Click Here </A>
```

در این مثال به پارامتر Onlyparam دو مقدار نسبت داده شده است . با استفاده از خاصیت count می توانید تعداد مقادیری که به هر پارامتر نسبت می دهید را مشخص کنید. مثال زیر تعداد مقادیری را که به یک پارامتر نسبت داده اید و در ادامه مقادیر آنرا نمایش می دهد.

```
The  Onlyparam parameter  has
<% Request .QueryString ("onlyparam").Count%> values .
<P> Ther are
<%.

  FOR EACH  pvalue IN Request .QueryString ("Onlyparam ")
Response.write ("<BR>"& pvalue")
NEXT
%>
```

حلقهٔ FOR....EACH به تعداد مقادیر پارامتر Onlyparam تکرار می شود. اگر پارامتر Onlyparam دارای مقدار صفر باشد ، خاصیت Count شماره صفر را با ز می گرداند و هیچ مقداری را نمایش نمی دهد .

٤-٧-٣ کپی کردن مجموعه Query String

اگر می خواهید تمام پارامترهای مجموعه رشته درخواست را بدست آورید باید از حلقه FOR...EACH درون مجموعه استفاده کنید به عنوان مثال اسکریپت زیر را در نظر بگیرید :

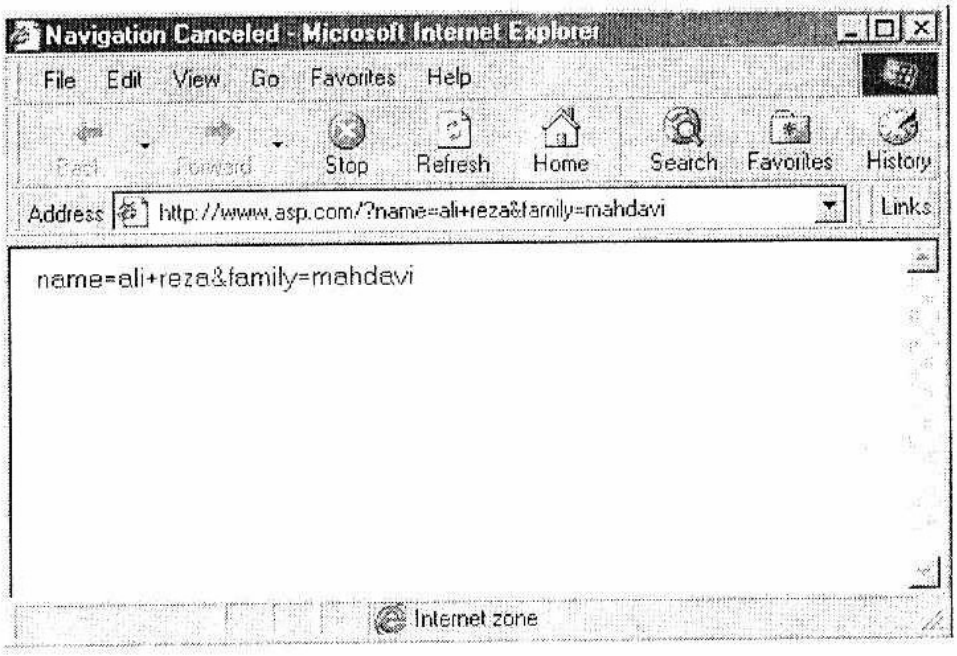
```
<%
FOR EACH  QSParam IN  Request .QueryString
  Response.Write ("<BR>"&QSParam&"=")
Response .Write (Requrest.QueryString (QSParam)
NEXT
%>
```

به جای حلقه FOR...EACH می توانید از حلقه FOR...NEXT نیزاستفاده کنید . برای این کار باید شماره هر عنصر درون مجموعه QueryString را بدست آورید . این کار را می توانید باخاصیت Count انجام دهید . همانند مثال زیر:

```
<%
FOR i=1  TO    Request.QueryString .Count
 Response.Write ("<BR>"&Request.QueryString (i))
NEXT
%>
```

اگر شما ترجیح می دهید که رشته درخواست را به صورت غیر مجزا بدست آورید، می توانید از مجموعه Request.Query String بدون پارامتر استفاده کنید .که به صورت زیر می نویسید: (رجوع شود به شکل ٤-٣)

```
<% =Request.Query String %>
```



شکل ٤-٤ یک رشتهٔ پویش نشده

٤-٧-٣ زمان هایی که از رشتهٔ درخواست استفاده نمی کنید

رشته درخواست برای وقتی مفید است که لازم است چند بیت یا اطلاعات محدود را از یک صفحه به دیگری بفرستید و در دو وضعیت از رشته درخواست استفاده نمی کنیم :یکی وقتی که بخواهیم اطلاعات مخفی را بفرستیم

و دیگری وقتی که بخواهیم اطلاعاتی با حجم زیاد را بفرستیم.یک رشته درخواست به هیچ وجه نمی تواند مخفی باشدو معمولاً روی خط آدرس مرورگر ظاهر می شود. بنابراین فرستادن کلمهٔ عبور از یک صفحه به صفحه دیگر کار درستی نمی باشد.

رشته درخواست برای اطلاعات باحجم بالا مفید نمی باشد. تعداد کاراکترهایی راکه یک رشته درخواست می تواند داشته باشد به چند پارامتر بستگی دارد ؛یکی از آنها مرورگری که استفاده می شود،می باشد. مرورگرهای مختلف محدودیتهای در اندازه های رشته درخواست دارند . به عنوان مثال Microsoft Internet Explorer نسخهٔ 4.0 نمی تواند رشته درخواست با بیشتر از ۲۰۰۰ کاراکتر را پشتیبانی کند . بنابراین برای نگه داشتن رشته کوتاه رشته درخواست ایده خوبی است ولی برای رشته های طولانی نه. اگر شما نیاز به فرستادن میزان زیادی اطلاعات از یک صفحه به صفحه دیگری دارید می توانید از یک فیلد مخفی فرم استفاده کنید بدین ترتیب دیگر محدودیتی وجود نخواهد داشت. پرتکل HTTP فیلدهای یک فرم را به صورت های بهتر ی نسبت به رشته درخواست می فرستد.

۸-۴ هدایت یک کاربر به صفحه دیگر

در بعضی مواقع نیاز است که یک کاربر را به صفحه دیگری هدایت کنیم به عنوان مثال اگر یک کاربر سعی کند فرم ثبت نام را بدست آورد . باید به صورت خودکار به صفحه ثبت نام هدایت شود یا اگر کاربری فرم اطلاعات را تکمیل کرد باید به صورت خودکار به صفحه دیگر راهنمایی شود . با استفاده از ASP هدایت یک کاربر به صفحه دیگر بسیار ساده انجام می شود. متد Redirect از شی ء Request به شما اجازه می دهد تا کاربر را به صفحه جدید هدایت کنید . به مثال زیر توجه کنید :

```
<%
IF Request.Form ("First Name ")= ""  THEN Reponse.Redirect "/register.asp"
%>
<HTML>
< HEAD> <TITLE> Registration Results </TITLE> </ HEAD>
<BODY>
Tank you <%= Request .Form("Fistrname ")%> For registering!
</BODY>
</HTML>
```

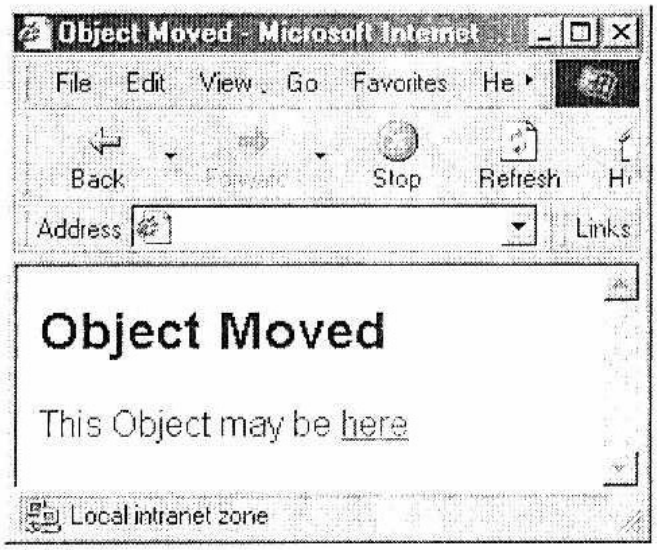
تصور کنید که کاربر فرم ثبت نام را کامل کرده و این صفحه آمده است. متد Response.Redirect در این مثال برای هدایت کاربر به فرم ثبت نام در مواقعی که کاربر نام خود را وارد نکرده بکار می رود. باید از متد Response.Redirect قبل از اینکه متنی برای مرورگر کاربر فرستاده شود استفاده کنید، از اینرو استفاده از این متد در اسکریپتی که بالای آن برچسب <HTML > وجود دارد ایدهٔ خوبی است. شما می توانید با استفاده از متد Response.Redirect کاربر را به هر URL قابل دسترسی راهنمایی کنید ؛ که می تواند صفحه دیگری روی سایت شما یا روی یک سایت دیگر در اینترنت باشد. البته تنها مشکل این است که متد Response.Redirect با برگرداندن یک کد وضعیتی خاص کار می کند . وقتی که سرویس دهنده به درخواستی پاسخ می دهد، کد

وضعیتی را دراولین خط پاسخ قرار می دهد. به طور مثال وقتی که متد Response.Redirect صدا زده می شود کد وضعیتی 302 که کد مربوط به Object Moved می باشد بازگردانده می شود.یک سرآیند موقعیت نیز به پاسخ اضافه می شود تا مکان صفحه جدید را بدهد. بدین ترتیب کد وضعیتی و سرآیند موقعیت به صورت خودکار مرورگر را به صفحه جدید هدایت می کنند .

توجه : متد Response.Redirect کاملاً هم ارز کدهای دوخط زیر است:

```
<%
Response.Status = "302 Object Moved"
Response.AddHeader  "Location " ,  "URL"
%>
```

در حقیقت همیشه اتفاق بالا نمی افتد و مرورگرهای قدیمی در حالتهای خاصی با این نوع هدایت دچار مشکل می شوند. هنگامی که مرورگر نتواند بطور خودکار به کد وضعیت پاسخ دهد، پیغامی مشابه پیغام شکل ٤-٥ را دریافت می کنید.بنابراین برای جلوگیری از این مشکل از این متد اجتناب کرده و از متد هدایت شبیه سازی شده که در بخش بعد آمده استفاده کنید.



شکل ٤-٥ نتیجهٔ راهنمایی سرویس دهنده

۴-۹ همراه کردن فایل ها

با استفاده از هدایت کننده INCLUDE به راحتی می توانید روی سرویس دهنده ، یک فایل دیگر رابا ASP همراه نمایید. هدایت کننده INCLUDE روی سرویس دهنده نباید درون اسکریپت قرار گیرد. بلکه آن راخارج از اسکریپت ومانند جزئی از کد HTML می آوریم؛همانند مثال زیر :

```
<HTML>
< HEAD> <TITLE>Welcome </TITLE> </ HEAD>
<BODY>
<!--#INCLUDE  VIRTUAL ="mybanner.inc"-- >
```

```
Welcom To Our Web Site!
</BODY>

</HTML>
```

در این مثال فایل mybanner.inc را در یک ASP و زیر برچسب <BODY>وارد می کنیم.وقتی این ASP اجرا می شود هر اسکریپت یا کد HTML ای که درون فایل mybanner.inc قرار دارد شامل ASP بالا نیز می شود . یک فایل را می توایند به دو صورت همراه کنید ؛یا با به وجود آوردن یک مسیر مجازی برای فایل همانند مثال قبل یا بااستفاده از مسیر فیزیکی فایل که مثال زیر روش دوم را نشان می دهد.

```
<HTML>

< HEAD> <TITLE>Welcome </TITLE> </ HEAD>

<BODY>

<!--#INCLUDE  VIRTUAL ="mybanner.inc"-->

Welcom to our web site !
</BODY>

</HTML>
```

اگر برای همراه کردن فایل از مسیر فیزیکی استفاده می کنید باید ازمشخصهٔ FILE استفاده کنید، فایل باید درون فهرست جاری یا زیر فهرست آن قرار گیرد . مسیر فایل به فهرست جاری بستگی دارد ، این یک محدودیت مهم است . بنابراین شما معمولاً از مشخصهٔ VIRTUAL استفاده خواهید کرد .

فایل همراه هر نام یک پسوند می تواند داشته باشد. معمولاً فایل هایی را که همراه می کنیم دارای پسوند inc است اما می توان از فایلهای htm. ،html. ،asp.وهرپسوند دیگری استفاده کرد .

همراه کردن فایل ها به دو دلیل مناسب است . اول آنکه شما می خواهید همان محتوا یا اجرای همان صفحهٔ اسکریپت را به صورت صفحات پی درپی نمایش دهید . به عنوان مثال غیرعادی نیست که هر صفحه درون سایت وب bannerوfooter یکسان داشته باشد . به جای اینکه کدهای HTML را مرتباً تکرار کنید می توانید به راحتی فایل Footer و Banner را مانند قسمتی از هر صفحه همراه آن کنید. شما می توانید یک اسکریپت ASP را در چندین صفحه با استفاده از هدایت کننده INCLUDE همراه کنید. به هرحال چون هدایت کننده INCLUDE باید خارج از اسکریپت باشد اسکریپت شما باید کاملا در علامتهای جداکننده قرار بگیرد. همچنین آنها را به صورت قطعه قطعه نمی توان همراه کرد .

دلیل دوم برای اینکه یک فایل راهمراه دیگری می کنیم این است که می خواهیم هدایت سرویس دهنده را شبیه سازی کنیم . برای انجام این کار تمام ASP های خودرا همراه صفحه دیگر می کنیم برای انجام این کار باید یک Save As خالی در ASP دوم همراه کنیم. مثال زیرا رادرنظر بگیرید.

```
<%
IF Request .Form ("FirstName ")=" "  THEN
%>
```

```
<!--#INCLUDE VIRTUAL = "Register. Asp"-->
%>
Response .End
END  IF
<%
<HTML>
< HEAD> <TITLE>Registration  Results </TITLE> </ HEAD>
<BODY>
Thank you <%= Request.Form("FirstName ")%>For    registering
</BODY>
</HTML>
```

این مثال کاملا همان اثر متد Response. Redirect را دارد. اگر کاربری فراموش کند فیلد FirstName از فرم ثبت نام را پر کند آن شخص به صفحه ثبت نام بازگردانده می شود . به هر حال ، چون همراه کردن فایل به صورت کامل درون سرویس دهنده هانجام می شود ، این هدایت شبیه سازی شده فایل قابل اعتمادتر از هدایت کامل است .

به متد Reposone.End در این مثال توجه کنید . متد Reposone.End تا زمانی که صفحه ثبت نام نمایش داده می شود از نمایش صفحات دیگر ASP وماندن آنها روی صفحه جلوگیری می کند .

دانستن اینکه IIS هدایت کننده INCLUDE را قبل از هر اسکریپت دیگری پردازش می کند حائز اهمیت است. این بدان معناست که شما نمی توانید از دستور اسکریپت زیردر آن استفاده کنید و در صورت نوشتن کار نخواهد کرد.

```
<%
IF Request . Form  ("First Name ")=" "  THEN
MYInclude = "register.asp"
ELSE
myInclude ="Homepape .asp"
END IF
%>
<!--# NCLUDE   VIRTUAL = <%=" myInclude%>" -->
```

این اسکریپت به این دلیل کار نخواهد کرد که سرویس دهنده همراه کردن فایل را قبل از اجرای هر اسکریپت دیگر انجام می دهد . یعنی سرویس دهنده اول ،خط <%=myinclude%> را اجرا می کند که دراین صورت چیزی برای اجرا وجود ندارد.

فصل پنجم :
کاربا Session های Asp

در این فصل شما چگونگی کار با Session ها و طریقهٔ استفاده از مجموعه ها ، صفتها و متدها و رخدادهای Session را می آموزید. همچنین یاد می گیرید که چگونه Cookieها را بخوانید و ایجاد کنید.

۵-۱ مقدمه ای بر Session

براستی Session چه می باشد؟ در پاسخ باید گفت Session همان چیزی است که وقتی کاربری درخواست صفحه ای را از سایت وب شما می‌کند شروع و با ترک کردن سایت شما توسط کاربر پایان می‌یابد. به هر بازدید کننده از سایت وب شما یک SessionID منحصر به فرد داده می شود.

Session ها می‌توانند سلیقه‌های بازدیدکنندگان را ذخیره کنند. به عنوان مثال اینکه آیا ملاقات کننده ترجیح می‌دهد صفحات وب دارای پس زمینه آبی باشد یا سبز؟ آیا ملاقات کننده ترجیح می دهد که نسخه غیر گرافیکی و متنی از سایت شما را ببینید؟ تمام این انتخابها می‌تواند توسط Sessionها پی گیری شود.

همچنین Session ها می‌توانند برای تهیه کارتهای خرید مجازی بکار بروند. هر وقت بازدید کننده ای، اقلامی را برای خرید از سایت وب شما انتخاب می‌کند، این نمونه ها به کارت خرید او اضافه می شود وقتی کاربر می خواهد سایت شما را ترک کند او تمام چیزی های را که درکارت خریدش وجود دارد به یکباره خریداری کرده است. تمام اطلاعات درباره این نمونه ها درون کارت خرید توسط Session ها ذخیره می‌شوند. در آخر Session ها می‌توانند اعمال بازدیدکنندگان سایت شما را تعقیب کنند.

۵-۲ خط توضیحات در Session

Session برای محدوده پروتکل HTTP ابداع شده بود. بیاددارید که پرتکل HTTP به چه صورت کار می‌کند. وقتی کاربر درخواستی را مطرح می‌کند سرویس دهنده پاسخ آنرامی فرستد. تمام این فعل و انفعالات بین مرورگر و سرویس دهنده وب به صورت جفتهای درخواست و پاسخ مطرح می‌شود.

هیچ چیزی در پرتکل HTTP به سرویس دهنده اجازه نمی دهد کاربری که کد درخواستی را به وجود آورده است را تعقیب کند. پس از اینکه سرویس دهنده به درخواست فرستاده شده پاسخ داد،دیگر برای سرویس دهنده مرورگری که این درخواست را فرستاده قابل شناسایی نیست.

سرویس دهنده وب هر درخواست جدید را به عنوان شخص جدیدی در نظر می گیرد. این مسأله همان حالت State Less بودن پروتکل HTTP است که قبلاً شرح دادیم. پروتکل HTTP وضعیت کاربر را نمی تواند نگه دارد. این یک محدودیت جدی است به این معنی که شما نمی‌توانید کاربر را روی صفحات مختلف سایت وب تشخیص بدهید.

Session ها برای بر طرف کردن این مشکل به وجود آمده اند. با استفاده Session ها می‌توانید اطلاعاتی درباره کاربرانی که روی صفحات مختلف وب حرکت می‌کنند بدست آورید. Session ها توانایی کارهای بسیار مشکل و یا غیر ممکن را دارند.

۵-۳ ذخیره کردن اطلاعات Session

کار با Sessionدر ASP بسیار آسان است. توسط شیء Session از ASP می‌توان تمام کنترلهای لازم را روی Session انجام داد.اگر لازم است اطلاعات کاربر را درون یک Session ذخیره کنید. می‌توانید این داده ها را درون یک مجموعه از شی Session ذخیره کنید. به مثال زیر توجه کنید:

```
<HTML>
<HCAD> <TITLE> Session Example </TITLE> </HEAD>
<BODY>
<%
 Session (" Greeting") = " Welcome"
Response. Write (Session("  Greeting"))
%>
</BODY >
</ HTML >
```

با نمایش این ASP روی مرورگر جمله Welcome نمایش داده می‌شود. جمله اول درون این اسکریپت متن Welcome را به متغیر Greeting از شیء Session نسبت می‌دهد .خط دوم متغیر Greeting را روی صفحه نمایش می‌دهد. شما این کارها را می‌توانید با VBScript انجام دهید .به هر حال تصور کنید همان کاربر صفحه دیگری را درخواست کند. به عنوان مثال صفحه زیر را درخواست کند.

```
<HTML>
<HEAD > <TITLE > Another page < / TITLE > < / HEAD>
<BODY>
<%=Session (" Greeting") %>
< /BODY>
</HTML>
```

وقتی کاربر این صفحه را می‌بیند همان Welcome دوباره روی صفحه نمایش داده می‌شود به هر حال متغیر Session مقداری به این صفحه نسبت نمی‌دهد.این متغیر Greeting ازشیء Session است که مقداری را که به صفحه قبلی نسبت داده شده است، نگه می دارد.

شما نمی‌توانید این کار را با متغیرهای معمولی اسکریپت انجام دهید. یک متغیر معمولی فقط درون یک صفحه دارای اعتبار است. بر خلاف آن متغیر Session تا زمانی که کاربر سایت وب را ترک نکند باقی می‌ماند.

باید بدانید متغیر Sessionای را که بوجود آورده اید ، فقط در رابطه با کاربری خاص است متغیر Session ای که به یک کاربر نسبت داده اید نمی‌تواند روی متغیر Session کاربران دیگر اثر بگذارد. به عبارت دیگر یک داده که درون متغیر Session ذخیره می‌شود میان کاربران دیگر به اشتراک گذاشته نمی‌شود. به عنوان مثال،اسکریپت زیر که در یک ASP آمده است را در نظر بگیرید.

```
<%
Randomize
If  INT (2 *RND) = 1 THEN
Session ( "Favoritecolor") =" Blue"
ELSE
Session  (" Favoritecolor")= " Red"
END IF
%>
```

این اسکریپت به صورت اتفاقی مقدار Blue یا Red را به Favoritecolor که یک متغیر شیء Session است نسبت می‌دهد. این متغیر با توجه به کاربران مختلف می‌تواند مقدارهای مختلفی را داشته باشد . مقدار متغیر Favoritecolor فقط به Session یک کاربر خاص مربوط می شود.

۴-۵ محتوای یک Session

بیشتر متغیرهای Session درون مجموعه ای از شیء Session به نام Contents ذخیره می‌شود. در مثال زیر دو عبارت هم ارز اند.

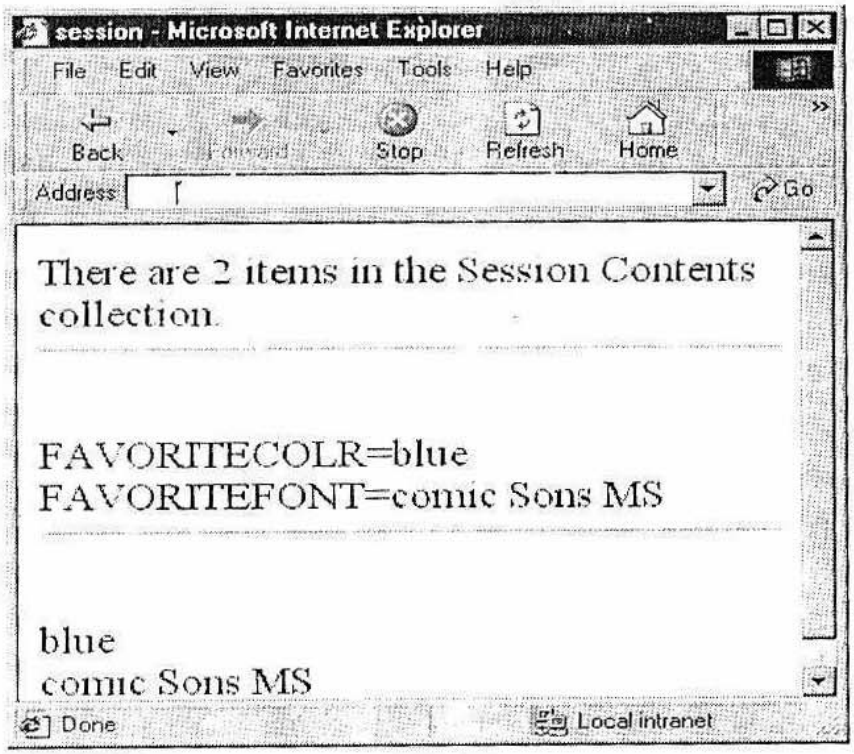
```
< % Session  (" MyVar")= " Some data" %>
< % Session.Contents ("  myvar")= " some data" % >
```

همانطوری که قبلاً در مجموعه ها بحث شد می‌توانید از صفت count برای تعیین تعداد عناصر مجموعه Contents استفاده کنید. همچنین با استفاده از حلقه FOR...NEXT , FOR... EACH می‌توانید تمام عناصر مجموعه contents را نمایش دهید. مثال زیر از این متدها استفاده می کند.

```
<%
Session  (" Favoritecolr") =" blue"
Session  (" FavoriteFont")="  comic Sons"
% >

There are
<% =Session.Contents.Count %>

items in the Session  Contents collection
<HR>

<%
FOR EACH thiny IN Session.Contents
Response.write (" <BR>"  & thiny & " ="  & Session.contents (thiny))
NEXT
Response.write (" <HR >")
```

```
FOR i=I TO Session.Contents.count
Response.write(" <BR>" & Session.contents( i ))
NEXT
%>
```

در این اسکریپت دو متغیر Session به نامهای FavoriteFont , Favoritecolor به وجود آمده است.



شکل ۵-۱محتویات مجموعهٔ contents

۵-۵ شناسایی یک Session

ASP به هر کاربر یک شناسه Session منحصر به فرد نسبت می‌دهد. وقتی اولین بار یک کاربر وارد می شود ویک شناسه به آن اختصاص داده می‌شود ، این شناسه تا زمانی که او در سایت است می‌ماندیک ISessionID وقتی به وجود می‌آید که کاربر جدیدیی به وجود آید و بخواهد مدتی را در سایت بماند. برای بدست آوردن این ID Session می‌توان از صفتSession ID از شی Session استفاده کنید. مانند مثال زیر:

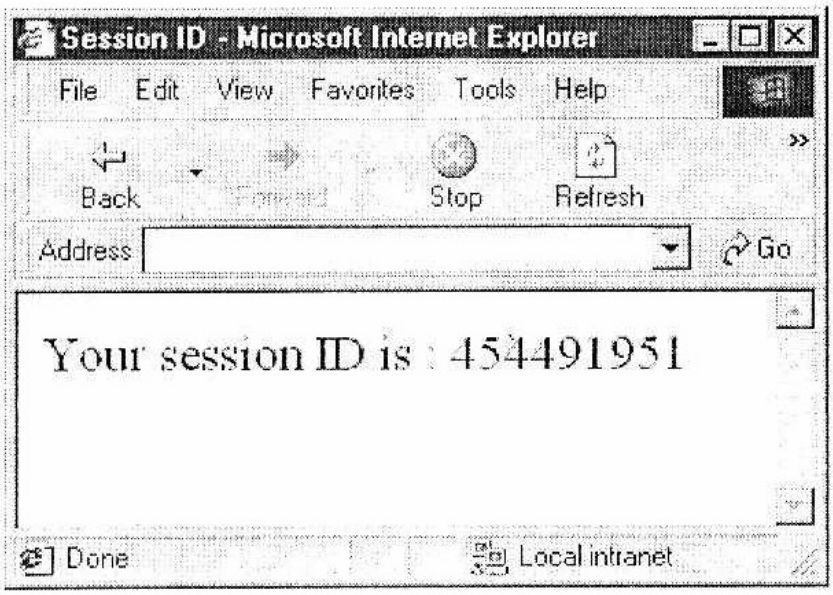
```
<HTML>
<HEAD > <TITLE > Session ID < / TITLE > < / HEAD>
<BODY>
Your session ID is :  < %= session.sessionID%>
</ BODY>
</HTML>
```

ASP بالا به راحتی مقدار صفت SessionID را استخراج می کند. (رجوع به شکل ۵-۱). وقتی کاربران صفحه را دریافت می کنند، برای هر کدام SessionID متفاوتی نمایش داده خواهد شد. یکی از کابردهای صفت

SessionID،دنبال کردن حرکت یک ملاقات کننده است. به عنوان مثال می‌توانید صفحاتی را که یک کاربر می بیند درونlogfile سایت وب خود ذخیره کنید. برای این کار می‌توانید فایل زیر را بسازید و آن را درون صفحات خود استفاده کنید.

```
<%
who =Session.SessionID
Currentpage = Request.ServerVaribales (" SCRIPT_ NAME")
Response.AppendToLog (who&":"&Currentpage)
%>
```

در این اسکریپت با استفاده از متدAppendToLog از شیء Response می‌توانید ورودی را به logfile سرویس دهنده اضافه کنید. در این مثال رشته ای که به فایل اضافه شده، یک SessionID می‌باشد که از صفت SessionID دریافت شده است. همچنین رشته شامل مسیر صفحه جاری می‌باشد که از متغیر محیطی SCRIPT-NAME بدست می آید.



شکل ۵-۱مقدار sessionID

۵-۶ کنترل هنگام پایان یافتن Session ها

چگونه یک سرویس دهنده تشخیص می‌دهد که یک Session پایان یافته است؟ به عبارت دیگر چگونه یک سرویس دهنده تشخیص می‌دهد که یک کاربر، سایت وب شما را ترک کرده یا دستگاه خود را خاموش کرده و رفته است یک سرویس دهنده فرض را بر این می گذارد که اگر کاربری بیش از ۲۰ دقیقه درخواستی نفرستاد و یا صفحه خود را refresh نکرد، یعنی از سایت خارج شده است و زمان استفاده از Session برای آن تمام شده است .برای درخواست های سایت های وب خاص این مدت زمانی بسیار کوتاه است. به عنوان مثال فرض کنید شما یک سایت بازی دارید که شامل تعدادی سرگرمی است که کاربر برای حل هرکدام نیاز به کاغذ و مداد دارد. بنابراین به مدت زمان بیشتری برای از دست دادن Session نیازدارد.

برای درخواست های بعضی دیگراز سایت های وب ، مدت زمان ۲۰ دقیقه بسیار زیاد است . اگر شما یک سایت بسیار بزرگ داشته باشید و بخواهید بار روی سرویس دهنده را تا حد امکان کم کنید. مدت زمان کمتری برای خروج هر Session لازم دارید.

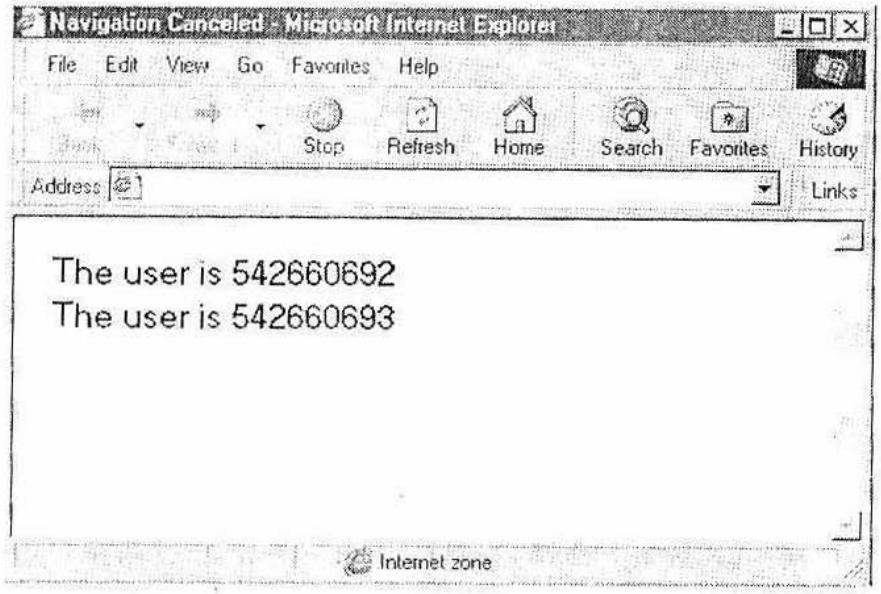
بنابراین باید بیشترین زمان اختصاصی به هر Session را کنترل کنید. شی Session دارای صفتی برای این منظور است. شما می‌توانید میزان اختصاص زمان را قبل از اینکه زمان پایان یابد، با استفاده از صفت Timeout تنظیم کنید به عنوان مثال اسکریپت زیر صفت Timeout را روی ٦٠ دقیقه تنظیم کرده است:

```
<%Session .Timeout =60% >
```

هنگامی که زمان یک Session پایان می‌یابد و کاربر تقاضایی می کند ، سرویس دهنده مانند یک کاربر جدید با او رفتار می کند. سرویس دهنده یک Session جدید به وجود می‌آورد و تمام اطلاعات Session قبلی از دست می‌رود. شما می‌توانید با استفاده از متد Abandon از شی Session این کار را انجام دهید. مثال زیر را در نظر بگیرید.

```
<HTML>
<HEAD > <TITLE >Abandon Session  </TITLE > </HEAD>
<BODY>
< BR> The  user  is  <%= session.sessionID%>
<% session.Abandon %>
< BR> The  user  is  <%= session.sessionID%>
</ BODY>
</HTML >
```

در این مثال Session ID روی صفحه نمایش داده می‌شود. بعد از آن Session.Adandon صدا زده می‌شود. وقتی کاربرSessionID دوباره روی خروجی نمایش داده می‌شود این ID شمارهٔ دیگری است. بعد از این که متد Abandon صدا زده می‌شود، سرویس دهنده با کاربر همانند یک کاربر جدید رفتار می کند. (رجوع به شکل ۵-۳)



شکل ۵-۳ واگذاری یک session

۵-۷ رخدادهای Session

برخلاف دیگر اشیا مطرح شده، شیء Session چند رخدادنیزدارد. دو تای آنها عبارتند از Session_OnStart که هنگام شروع Session رخ می دهد و Session_OneEND که هنگام پایان Session رخ می دهد. شما فقط یک اسکریپت را می‌توانید با هر یک از این رخدادها تلفیق کنید.

جملات درون اسکریپت وقتی که رخداد شروع شد، اجرا می شود. هر دوی این اسکریپت ها درون یک فایل مخصوص به نام Global.asa قرار می گیرند. هر تقاضا برای سایت وب می‌تواند یک فایل Global.asa داشته باشد. این فایل درون فهرست ریشه سایت شما قرار دارد. فایل Global.asa دارای ساختار زیر است.

```
<SCRIPT  LANGUAGE =VBScrtpt    RUNAT= Server>

SUB  Application_Onstart
END SUB
</SCRIPT>

<SCRIPT   LANGUAGE = VBScript   RUNAT = Server>

SUB APPlication _onEnd
End SUB
</SCRIPT>

<SCRIPT  LANGUAGE = VBScript   RUNAT = Server>

SUB Session_Onstart
END SUB
</SCRIPT>

<SCRIPT  LANGUAGE = VBScript   RUNAT = Server>

SUB Session_On End
END SUB
</SCRIPT>
```

فایل Global.asa می‌تواند شامل چهار اسکریپت باشد. یکی از اسکریپت ها با رخداد Session-Onstart راه اندازی شده و یکی دیگر بارخداد Session_OnEnd .

توجه کنید در این فایل از برچسب <SCRIPT> به جای روش معمول یعنی استفاده از کاراکترهای <٪و٪> استفاده می‌شود. دراین فایل از VBScript استفاده می کنیم البته می توان از زبانهای دیگر نیز استفاده کرد. دراین فایل نمی توانید هیچ خروجی دیگری داشته باشیم. به عنوان مثال نمی توانید از برچسب HTML یا متد Response.write() استفاده کنید. فایل Global.asaخودش هیچ وقت نمایش داده نمی‌شود این فایل فقط برای اشیاء و اسکریپتها استفاده می شود. برای ایجاد اسکریپتی که هر وقت یک Session شروع می شود، اجرا شود می توانید به سادگی با اضافه کردن آن به قسمت Session_Start از فایل Global.asa آنرا انجام داد.همانند مثال زیر:

```
<SCRIPT  LANGUAGE = VBScript  RUNAT= Server>

SUB Session _Onstart
Session (" UserName") = " UnKnown"
Session (" UserPassword") = " UnKnown"
```

```
END SUB
</SCRIPT>
```

این اسکریپت مقدار "Un known" را بـه دومتغیر UserPasswordوUserName که متعلق بـه Session می‌باشند نسبت می‌دهد. دراین مثال یکی از توابع اصلی اسکریپت یعنی Session_Onstart برای مقدار دهـی اولیـه به متغیرهای Session ، تشریح شده است.

از Session_ Onstart می‌توانیـم بـرای منظورهـای دیگـری نـیز اسـتفاده کنیـم . یکـی از کاربردهـای جـالب Session_Onstart هدایت کاربر به صفحه ای جدید است. به عنوان مثال اگر بخواهید هنگامی کـه کـاربری وارد سـایت شـما می‌شـود ابتـدا بـه صفحـه خـانگی راهنمـایی شـود و نـه صفحـه دیگـری؛ می‌توانیـد بـا اسـتفاده از متـد Response.Redirect این کار را انجام دهید. همانند مثال زیر :

```
< SCRIPT  LANGUAGE= VBScript  RUNAT = Server>

SUB  Session_Onstart
MyHomepage=" /homepage .asp"
Requestpage  = Request.ServerVariables (" SCRIPT_ NAME")
IF NOT (STRCOMP (MyHomepage,Requestpage ,VbTextCompare=0 )) THEN
Response.Redirect (MyHomepage)
END IF
END SUB
</SCRIPT>
```

در این اسکریپت مسیر صفحه در خواست شده با مسیر صفحهٔ خانگی مقایسه می‌شود اگر همان صفحه نبـود بـه طور خودکار به صفحهٔ خانگی راهنمایی می‌شودبه یک مثال دیگر توجه کنید:

```
< SCRIPT  LANGUAGE=VBscript  RUNAT=Server>

SUB  Session_Onstart
Response.AppendTolog (Session.Session ID&" starting" )
END   SUB
<SCRIPT/>

< SCRIPT  LANGUAGE= VBscript  RUNAT = Server>

SUB  Session _ OnEnd
Reponse .AppendTolog (Session . Session ID &"  ending")
END   SUB
</SCRIPT>
```

در این مثال از Session_Onstartو session_OnEnd بـرای ثبـت sessionID یـک کـاربر درونlogfile استفاده می‌شود.بخاطر اینکه Session_Onstart هنگامی اجرا می شود که یک کاربر برای اولین بار وارد شود؛ ایـن اسکریپت زمانی را که کاربر یک Session جدید را شروع کرد ، ثبت می کند. Session_Onend نـیز زمـانی را که کاربر خارج می‌شود، ثبت می کند. شما می‌توانید این اطلاعات را برای مشخص کردن صفحاتی که از آنها بیشـتر به سایت شما وارد یا از آن خارج می‌شوند استفاده کنید.

۵-۸ Session ها به چه صورت کار می کنند

Session ها از cookie ها استفاده می کنند. وقتی یک کاربر برای اولین بار تقاضای صفحه ای را از سایت شما می کند، سرویس دهنده روی مرورگر کاربر یک cookie برای دنبال کردن session به وجود می آورد. وقتی session پایان می‌یابد cookie نیز به خوبی پایان می یابد. cookieها برای هر کاربری که ASP ESSIONID نامیده می‌شود، به وجود می آید. منظور از cookie به وجود آوردن یک شناسه منحصر به فرد برای هر کاربر می‌باشد.

اگر شما بخواهید ASP خود را refresh یا reload کنید حداقل یک بار باید cookie را نمایش دهید. متغیرهای session خود شان به تنهایی روی مرورگر کاربر ذخیره نمی‌شوند. به هر حال ASPESSIONID cookie نیاز دارد که از متغیرهای session استفاده کند. یک سرویس دهنده از ASPESSIONID cookie برای وصل کردن یک متغیر session خاص به یک کاربر خاص استفاده می کند. بدون cookie یک سرویس دهنده راهی برای شناسایی یک کاربر که روی صفحات سایت وب حرکت می کند ندارد. SessionID ذخیره شده درون ASPSESIONID cookie به همان صورت صفت sessionID نمی‌باشد. ماکروسافت از یک الگوریتم پیچیده برای تولید مقدار ASPESSIONID cookie استفاده می کند.

چون session از cookie استفاده می کند، ممکن است با مرورگرها ی قدیمی ناسازگار باشد . مرورگرهای قدیمی نمی‌توانند از cookie استفاده کنند.چون cookie با همه مرورگرها سازگار نمی‌باشند وشما باید در استفاده از شیء session درون سایت وب خود محتاط باشید.البته بدو ن استفاده از session شاید خیلی از کارها را نتوانید انجام دهید ولی می‌توان session را به روش دیگری شبیه سازی کرد.

۵-۹cookie

مرورگرهایی که از cookie پشتیبانی می کنند فایلهای مخصوصی دارند. این فایلها Cookie Files نامیده می‌شوند و برای ذخیره داده درون سایت های وب استفاده می‌شوند. یک سرویس دهنده وب می‌تواند مقداری از اطلاعات را درون فایلهای cookie قرار دهد بعضی از کاربران وب عکس العمل شدیدی نسبت cookie ها نشان می‌دهند. بعضی از کاربران تصور می کنند cookie حریم خلوت آنها را تهدید می کند یا به حریم خصوصی آنها تجاوز می کند.

cookie های خاصی موقتی و بقیه ماندگار هستند. به عنوان مثال cookieهایی که در ASP برای دنبال کردن session های کاربر بکار می‌روند پس از اینکه کاربری سایت وب را ترک می کند ، تمام می شوند. بقیه cookie ها درون فایل cookie می مانند تا هنگامی که کاربر برگرددو توسط سرویس دهنده دوبارخوانده شوند. Cookie هایی که در فایل cookie می مانند آنهایی هستند که بیشتر دخالت دارند. بیم آن می رود که cookie ها بتوانند برای اهداف سودجویانه شخصی مورد استفاده قرار می گیرند. البته کاربر می تواند از نوشته شدن cookie درون شاخه فایل های cookie خود با فقط خواندنی کردن آن جلوگیری کند ولی واضح است که خواندن فایلهای شما انکار ناپذیر خواهد بود.

۵-۹-۱ cookie ها به چه صورت کار می کنند

cookie ها توسط سرآیندها HTTP بین سرویس دهنده و مرور گر ، جابجا می شوند. ابتـدا سـرویس دهنـده در پاسخ خود به وسیله سرآیند Set cookie یـک cookie بوجـود مـی‌آورد. و بعـد توسـط درخواسـت مـرورگـر ایـن cookie درون سرآیند cookie بازگردانده می‌شود.

فرض کنید می‌خواهید یک cookie بنام UserName به وجود آورید که شامل نـام ملاقـات کننـدگـان سـایت شما باشد. برای به وجود آوردن این cookie ، سرویس دهنده باید سرآیند زیر را بفرستد :

```
set_cookie : UserName =BILL+Gates ; Path=/;domain =aspsite.com;
Expires= Tuesday , 01-Jan-99  00:00:01  GMT
```

این سرآیند به مرورگر دستور می‌دهـد کـه یـک cookie بـا نـام UserName و بـا مقـدار Bill Gates درون فـایل cookie اضافه کند.همچنین سرآیند ،مرورگر را آگاه می کند که این cookie باید بدون توجه به مسیر بکار رفتـه در درخواست ، به سرویس دهنده بازگردد. اگر صفت مسیر به روی مقادیر دیگر همچون Prirate / تنظیـم شـده بـود ، cookie می بایست تنها در این مسیر برگردانده می شد. به عنوان مثـال درخواسـت بـرا ی فـایل /privatepage.htm باید شامل سرآیند cookie و نه درخواست /mypage.htm باشد.

صفت domain برای این است که نشان دهد مرورگـر cookie هـا را بـه کجـا مـی توانـد بفرسـتد. در ایـن مثـال cookie می‌تواند فقط به سایت www.aspsite.com فرستاده شود و در این صورت cookie به هیچ سایت دیگـری فرستاده نمی‌شود.

در آخر صفت Expries تعیین می کند که cookie باید تا چه زمانی باقی بماند. سرآیند ایـن مثـال بـه مـرورگـر می‌گوید که cookie را تا نیمه اول ژانویهٔ 2003ذخیره کند، یک cookie ممکن است زودتر از ایـن از بیـن بـرود بـه این صورت که اگر یک cookie بیش از حد بزرگ بشود مرورگر به طور خودکار شروع به برداشتن آنها می کند.

اولین با ر که مرورگر یک cookie را به وجود می‌آورد آن را درون تمام درخواست هایی که به وجود می‌آورد قرار می‌دهد و به سایت های وبی که مورد درخواست واقع شده اند می‌فرستد.البته مـرورگـر cookieهـا را بـه سـایت وبی که نامهای حوزه[1] متفاوت دارندنمی‌فرستد. مرورگرفرستادن cookie را ادامه می‌دهد تا آن cookie از بین برود . یک سرآیند cookie شبیه زیر است.

```
cookie : User NAME: Bill +Gates
```

۵-۹-۲ به وجود آوردن و خواندن cookie ها باASP

به منظور به وجود آوردن یک cookie باASP باید از مجموعه cookie از شیء Response استفاده کنید.شـما می‌توانید دو نوع از cookie ها را به وجود آورید که اولی یک cookie با یک مقدار تکی ، و دومـی یـک cookie dictionary شامل نام و مقدار آن می باشد.

برای به وجود آوردن یک cookie با مقدار تکی شما می‌توانید از اسکریپت زیر استفاده کنید.

```
<%
Response.cookies (" UserName ") =" Bill Gates"
Response.cookies (" UserName "). Expires =" jan 1,2003"
%>
```

این اسکریپت یک cookie با نام UserName با مقدار کاراکتری " Bill Gates " به وجود می‌آورد. این cookie توسط مرورگر تا ژانویهٔ 2003 یا تا وقتی که مرورگر آنرا پاک نکرده باشد برگردانده می شود. اگر زمان انقضا برای cookie در نظر گرفته نشود هنگامی که کاربر سایت وب را ترک کند cookie نیز ازبین می رود.چون این اسکریپت یک سرآیند تولید می کند بنابراین باید قبل از هر عبارت خروجی درون فایل ASP قرار بگیرد. در نهایت شما می توانید صفحه را بافر کنید. اسکریپت زیر مثال ساده ای از چگونگی ایجاد یک Cookie می باشد.

در این مثال صفت Expires استفاده شده است، ولی مجموعه cookie هادارای خواص دیگری نیز است ، به مثال های زیر توجه کنید.

```
<%
Response . Cookies (" UserName ")=" Steve Jobs"
Response . Cookies (" UserName ").Expires="  Jan 1,2003"
"Response . Cookies (" UserName ") .path= " /examples"
Response . Cookies (" UserName "). Domain = "aspsite.com"
Response . cookies ("UserName "). secure=True"
%>
```

این اسکریپت یک cookie با نام UserName به وجود می‌آورد که دارای صفات زیر است.

صفت path برای تعریف مسیر دقیقتر جایی که مرورگر باید cookie را بفرستد ، استفاده می‌شود. در این مثال cookie فقط به مسیرهای مورد درخواستی که با examples /شروع می‌شوند ،فرستاده می شوند. به عنوان نمونه درخواست های زیر فرستاده می‌شوندexamples/chapater/cookie.asp / examples/hello.asp/ ولی برای "/hello.asp" فرستاده نمی شوند. صفت Domain تعیین می کند که یک cookie چه وقت باید فرستاده شود در مثال قبل cookie تنها با درخواستهایی که به حوزه aspsite.com می رود ارسال می شوداین بدان معنی است که cookie به مقاصد www. aspsite یا crieket.aspsite .comفرستاده می‌شود. اگر این خاصیت تعریف نشود از نام حوزه سرویس دهنده وب استفاده می شود.

نهایتا" صفت Secure مشخص می کند که Cookie باید بصورت رمز شده انتقال یابد. برای خواندن یک cookie درون یک ASP باید از مجموعه cookie از شیء Request استفاده کنید. به عنوان مثال، برای استخراج مقدار cookie می‌توانید ازاسکریپت زیر استفاده کنید:

```
<%=Request .cookies ("UserName")%>
```

این اسکریپت مقدار یک cookie با نام UserName را استخراج می کند. در اینجا نیز می‌توانید با استفاده از صفت count تعداد عناصر مجموعه در cookie هارا بدست آورید. همچنین می‌توانید با حلقه های FOR...NEXT , FOR...EACH بین عناصر مجموعه cookie حرکت کنید. به مثال زیر توجه کنید :

```
<%
```

```
FOR  EACH  thing IN  Request.cookies
Response.Write ("<BR>"&thing&(Request.cookies (thing))
NEXT
%>
```

۵-۹-۳ به وجود آوردن بیش از یک cookie

می‌توانید با استفاده از مجموعه Response.cookie(همانند مثال قبل) بیش از یک cookie به وجود آورید. هر چند که خیلی از مرورگرها فقط سه یا چهار cookie را از یک سایت وب خاص پشتیبانی می کنند.

یک روش جایگزین برای به وجود آوردن چند cookie وجود دارد: شما می‌توانید یک cookie dictionary به وجود آورید. یک cookie dictionary یک cookie تکی با چند مقدار ونام می باشد. مثالی برای به وجود آوردن cookie dictionary در زیر آمده است.

```
<%
Response.cookies ("User")("Name")="Bill  Gates"
Response.cookies ("User")( "password")= "Bill ons"
%>
```

این اسکریپت cookie dictionary با نام user و کلیدهای Name و password به وجود می‌آورد. وقتی یک cookie dictionary به وجود آید. سرآیند زیر به مرورگر فرستاده می‌شود.

```
set . cookie : User = Name =Bill +Gates &password =billions
```

یک cookie با نام user به وجود می‌آید. نام و مقدار هر کلید از cookie dictiovary دریک cookie بزرگ قرار می گیرند.

برای بدست آوردن یک cookie dictionary می توانید همانند مثال قبل از مجموعه cookie dictionary استفاده کنید برای بدست آوردن کلیدهای خاص از cookie dictionary نام هر کلید را درون مجموعه می‌آوریم. مثال زیر را در نظر بگیرید.

```
<%= Request. cookies ("User")%>
<%=Request. cookies ("User)( "Name")%>
<%=Request. cookies ("User ")("password")%>
```

برای تعیین اینکه آیا یک cookie از نوع cookie dictionary است یا نه از صفت Haskeys استفاده می کنیم به عنوان مثال اسکریپت زیر اگر یک cookie جدیداز cookie dictionary باشد. مقدار True ودر غیر این صورت مقدار false را باز می گرداند.

```
<%=Request.cookies ("User").Haskeys%>
```

۵-۱۰ نگه داشتن موقعیت بدون cookie

بکار بردن session و cookie خالی از دردسر نیست زیرا همه مرورگرها از آنها پشتیبانی نمی کنند. وقتی از cookie استفاده می کنید از طرف مرورگرهای متفرقه پیغامی مبتنی بر اینکه نمی‌تواندا Cookie ها استفاده کند را دریافت می کنید. در ادامه شما یاد می گیرید که از متدهایی بدون نیاز به Cookie برای دریافت وضعیت استفاده کنید.

۵-۱۱ بدست آوردن موقعیت با استفاده از Query string

شما می توانید یک Query string به هر پیوندی در Asp ها اضافه کنید. با استفاده از Query string می‌توانید اطلاعات را از یک صفحه به صفحه دیگر بفرستید. همانند مثال زیر:

```
<HTML>
<HEAD> <TITEL> Query State <TITLE> </HEAD>
<BODY>
<%
UserName =Server .URLEncode("Bill  Gates")
%>
<A HREF ="/nextpage .asp ? <% = UserName %> ">Click Here </A>
</BODY>
</HTML>
```

این اسکریپت نام Bill Gates را به متغیری با نام UserName نسبت می‌دهد. مقدار این متغیر به صورت Query string در صورتی که کاربر روی لینک کلیک کند. به صفحه Nextpage.asp ، فرستاده می‌شودبا بدست آوردن UserName از مجموعه Query string می‌توانید آن را از صفحه ای به صفحه دیگر بفرستید.به عنوان مثال صفحه Nextpage.asp به صورت زیر است:

```
<HTML>
<HEAD> <TITEL>Next page <TITLE> </HEAD>
<BODY>
<%
UserName =Server .URLEncode(Request .Querystring( "UserName")
%>
<A  HREF= "/nextpage .asp ? <%= User Name %>">Click Here</A>
</BODY>
</HTML>
```

مزیت این روش این است که با تمام مرورگرها کار می کند. اما خیلی پرزحمت است اگر می‌خواهید یک کاربر را روی صفحات سایت وب خود دنبال کنید.باید با هر پیوند در سایت وب یک Query string همراه کنید و هر Query string باید شامل نام کاربر باشد.

ضرر این روش این است که نمی‌توانیم اطلاعات با حجم بالارا بفرستیم.زیرا یک Query string نمی‌تواند حجیم باشد. اگر یک Query string بیشتر از 1024 کاراکتر باشد تولید اشکال می کند.

۵-۱۲ بدست آوردن وضعیت با استفاده از فیلدهای فرم مخفی :

اگر شما می‌خواهید بدون استفاده از متغیرهای session مقدار زیادی اطلاعات را از یک صفحه به صفحه دیگری بفرستید، چاره ای جز استفاده از فرم HTML ندارید . با استفاده از یک فیلد فرم مخفی می‌توانید اطلاعات را همانند مثال زیر ارسال کنید.

```
<HTML>
<HEAD> <TITLE> Form State </TITLE ></HEAD>
<BODY>
<%
UserName ="Bill Gates"
%>
< FORM  METHOD = "POST" ACTION= "/nextpage .asp">
<% INPUT NAME ="UserName " Type ="HIDDEN "  VALUE =<%=UserName%>">
<INPUT   TYPE ="SUBMET "  VALUE ="Nextpage">
</FORM>
</BODY>
</HTML>
```

این صفحه شامل یک فرم HTML است. فرم یک فیلد مخفی به نام UserName دارد که حاوی مقدار متغیر Username است. فرم شامل یک دکمه نیز می‌باشد. هنگامی که روی دکمه کلیک می کنیم nextpage.asp فراخوانی می‌شود و داده ها در فیلد فرم مخفی به صفحه جدید فرستاده می‌شود. شما می‌توانید داده ها را به این صورت ارسال کنید. در هر صفحه، از مجموعه Form شیءRequest برای دریافت داده ای فیلد مخفی استفاده کنید بعد باید برای اینکه داده ها بتوانند به صفحه جدید فرستاده شوند ؛یک فیلد مخفی جدید به وجود آورید. به مثال زیر توجه کنید:

```
<HTML>
<HEAD> <TITLE> Next page </TITLE ></HEAD>
<BODY>
<%
UserName =Request .Form("UserName")
%>
< FORM  METHOD = "POST" ACTION = "/Nextpage .asp">
<INPUT NAME ="UserName " Type ="HIDDEN" VALUEE = " <%=User Name%>">
<INPUT  TYPE ="SUBMET " VALUE ="Nextpage">
```

```
</FORM>
</BODY>
</HTML>
```

۵-۱۳ روشهای ترکیبی

هیچ یک ازدو روش ذکر شده به تنهایی مناسب نمی‌باشند. هر چند که آنها تنها روش های جایگزین برای بدست آوردن موقعیت بدون استفاده از session و cookie می‌باشند. با استفاده از Query string، به همراه فیلدهای مخفی فرم، با مرورگرها مشکلی نخواهید داشت.

اگر شما نیاز دارید کاربر را روی تمام صفحات وب خود تعقیب کند باید Query string یا فیلد مخفی فرم را روی هر صفحه در سایت وب خود قرار دهید. شما می‌توانید این دو متد را برای بدست آوردن موقعیت این دو روش را با هم ترکیب کنید.می توانید دو روش نگهداری وضعیت را با هم ترکیب کنید. برای مثال در بعضی از صفحات می‌توانید از Query string ودر بعضی صفحات دیگر از فیلد مخفی فرم استفاده کنید اگر این کار را بکنید نیاز ندارید که هم مجموعه های form و هم Query string را در هر صفحه بررسی کنید اگر از متدRequest بدون تعریف مجموعه استفاده کنید.هر دو مجموعه به صورت خودکار بررسی می‌شود. به مثال زیر توجه کنید:

```
<HTML>
<HEAD> <TITLE> Next page </TITLE ></HEAD>
<BODY>
<%
UserName= Request .Form("User name")
%>
< FORM   METHOD = "POST" ACTION = "/Nextpage .asp">
INPUT   NAME ="UserName "Type ="HIDDEN">
< VALUEE ="<%=UserName"%>
<INPUT  TYPE ="SUBMET"   VALUE ="Nextpage">
</FORM>
< A  HREF ="/nextpage .asp ?<% =Server .URLEncode (UserNam e %> "<Click Here </A>
</BODY>
</HTML>
```

در این مثال متغیر UserName نام کاربر بدون در نظر گرفتن اینکه نام کاربر با Query string فرستاده شده است یا با فیلد مخفی فرم تخصیص می یابد. با صدا کردن Request ("UserNme") می‌توان مقدار UserName را از Querystring یا Form دریافت کرد.

فصل ششم :

کار با Application های ASP

در این فصل شما طرز کار با Application را می آموزید. در بخش اول فصل شما پیش زمینه ای دربارهٔ Application ها پیدا می کنید. در بخش دوم شما خواهید آموخت که از متدها ، مجموعه ها و رخدادهای یک Application چگونه استفاده کنید. و در آخر این فصل دو برنامه که در آنها از Application استفاده شده مورد بررسی قرار گرفته است که یکی به شما چگونگی ساخت یک Chatچند کاربرهٔ ساده و دیگری طریقهٔ ایجاد یک Asp که بصورت Real time آمار استفاده از سایت وب شما را نمایش دهد، را یاد می دهد.

۶-۱ یکApplication چیست ؟

مایکروسافت می خواهد که شما در مورد Asp در حیطهٔ اصطلاحات برنامه نویسی بیاندیشید و بدانید که وقتی یک ASP تکی بوجود می آورید یک روال ساخته اید ولی وقتی یک دستهASP مرتبط به هم می سازید یک Applicationبه وجود آورده اید.

هرچندکه یکApplication چیزی بیشتر از یک سری صفحه که روی دیسک سخت قرارگرفته است نمی باشد لیکن وقتیASP ها با هم به صورت یکApplication در می آیند دارای خصوصیاتی می شوندکه به تنهائی فاقد آن بودند. درزیر لیستی از ویژگیهای مشترک یک Application از ASP آمده:

- درApplication هاداده را می توانید بین صفحات و در نتیجه بین چندین کاربر در سایت وب به اشتراک بگذارید.
- یکApplication دارای رخدادهایی است که می تواند اسکریپت هایApplication ها را راه اندازی کند. به عنوان نمونه یک شیء می تواند بین تمام صفحات در برنامه های کاربردی به اشتراک گذاشته شود.
- Application های مجزا می توانند برای داشتن خصوصیات مختلف توسط Internet service Managerپیکره بندی شوند.
- Application مجزا می توانند بطور جداگانه روی قسمتی از حافظه که به خود آنها تعلق دارد اجرا شوند. این بدان معنی است که اگر یکی ازApplication ها از کار افتاد[1]بقیه نباید از کار بیافتند.
- شما می توانید یکApplication رابدون آنکه رویApplication دیگر اثربگذارد متوقف کنید.

هنگامی که برای اولین بار ASP ها را نصب می کنید، تعدادی از Application ها بطور پیش فرض ایجاد می شوند .بطور مثال یک Application برای سایت وب پیش فرض شما بوجود می آید. گرچه شما می توانید بعداً با توجه به

نیازتان بسیاری از Application های اضافی را ایجاد کنید، مراحل زیر را برای تعریف یک Asp Application دنبال کنید:

۱- Internet Service Manager را از گروه برنامه های Microsoft Internet Information Server باز کنید.

۲- روی سایت وب پیش فرض خود (که اگر نامش را تغییر نداده باشید Defult Web Site یا هم نام با سیستم خواهد بود) در درخت هدایت[1] کلیک کنید.

۳- شما می توانید فهرست موجود ، سایت وب پیش فرض یا فهرست جدیدی را برای Application خود انتخاب کنید. برای ایجاد یک فهرست مجازی جدید ، روی نام سایت وب پیش فرض خود کلیک راست کنید و سپس از قسمت New , Virual directory را انتخاب کنید.

٤- پس از اینکه یک فهرست برای Application خود انتخاب کردید ، شما باید صفحهٔ خصوصیات آنرا ببینید برای این کار می توانید یا روی نام فهرست کلیک راست کرده و سپس دکمهٔ Properties را بزنید و یا روی دکمهٔ Properties که در بالای صفحه وجود دارد کلیک کنید.

٥- در صفحهٔ خصوصیات روی دکمه ای که روی آن Virtual Directory یا Home Directory نوشته کلیک کنید.

٦- سپس در قسمت Application Setting روی دکمهٔ Create کلیک کنید.

اکنون شما یک Application جدید را با موفقیت ایجاد کرده اید. پس از اینکه Application با موفقیت ایجاد گردید می توانید شماری از خصوصیاتش را بوسیلهٔ انتخاب Configuration از صفحهٔ Application Settings تنظیم کنید. برای مثال می توانید تعیین کنید که برنامهٔ کاربردی از Session استفاده کند یا خیر.

یک سایت وب می تواند بیش از یکApplication داشته باشد.بطورنمونه هنگامی که شما مجموعه هایی از صفحات که دارای منظورهای مخصوص به خود هستند را دارید ، Application جداگانه ای را برای هر یک به وجود می آورید .به عنوان مثال ممکن است یک Application شامل تمام صفحات، برای مصرف عمومی بوجود آورید و یا یکApplication دیگر برای استفادهٔ مدیران شبکه به وجودآورید.

یکApplication توسط Internet Service Manager از فهرست ریشهٔApplication تعریف می شود. یک Application شامل یک فهرست خاص و همهٔ زیرشاخه هایش می شود. اگر یکی از این زیر شاخه های تعریف شده Application باشد ، در این صورت آن زیرشاخه نیز یک Application مجزا می سازد. به بیان دیگر هیچ Application ای که با دیگری اشتراک داشته باشد وجود ندارد.

۶-۲ بکاربردن شیءApplication

شیءApplication دارای تمام مجموعه ها، متدها و رخدادهای مربوطه به Application ها می باشد.

۳-۶ مقدمه ای برمتغیرهایApplication

یک متغیرApplication شامل داده ای است که تمام صفحات وتمام کاربرانی که از Application استفاده می کنند می توانند به آن دسترسی داشته باشند. متغیرهای Application می توانند شامل هر نوع داده ای باشند از جمله آرایه ها و اشیاء.یک متغیرApplication از دو جهت با متغیرهای session تفاوت دارد:

۱- برخلاف متغیرهای session متغیرهایApplication به cookieها وابسته نمی باشند و به عبارتی سرویس دهنده وب در استفاده از متغیرهایApplication با مرورگرمشکلی ندارد.

۲-برخلاف متغیرهای session یک متغیرApplication می تواند بین کاربران به اشتراک گذاشته شود . یک داده می تواند توسط یک کاربر درون متغیرApplication ها ذخیره شود وتوسط کاربر دیگرخوانده شود.

برای مثال فرض کنید شما از یک متغیر Application برای ثبت تعداددفعاتی که یک اعلان آگاهی کلیک شده استفاده می کنید. بدین ترتیب هر وقت این اعلان کلیک می شود اسکریپتی مشابه اسکریپت زیر اجرا می شود:

```
<%
NumClicks = Application ("BannerClicks")
NumClicks = Num Clicks + 1
Application ("BannerClicks") = NumClicks
%>
```

اسکریپت بالا به سادگی یکی یکی شمارهٔ ذخیره شده در BannerClicks را اضافه می کند. اما فرض کنید که دو کاربر هم زمان روی یک آگهی کلیک کنند یعنی یک اسکریپت همزمان برای دو کار باید اجرا شود. اگر چنین شود مقدار BannerClicks مقدار نادرستی خواهد بود که ارزش ندارد چرا که هر دو کاربر در آن واحد مقدار متغیر را اضافه کرده اند.

در زیر برخی از موارد استفاده از متغیرهای Application ،آورده شده است :

- یک متغیرApplication می تواند برای نمایش اطلاعات زود گذر در صفحهٔ وب بکار رود.به عنوان مثال می توانید از متغیرهای Application برای نمایش نوع روزها یا برای تغییر دادن اخبار روزانه در هر صفحه وب استفاده کنید.
- یک متغیرApplication برای ثبت تعداد دفعاتی که یک اعلان آگهی[1] در سایت وب شما کلیک شده ، استفاده شود.
- یک متغیرApplication برای نگه داشتن اطلاعات بدست آمده از یک پایگاه داده استفاده می شود . برای مثال می توان لیستی از آیتمهای فروش را از پایگاه داده دریافت کرد و این لیست را توسط متغیرهای Application در چند صفحه نمایش داد.
- یک متغیر می تواند شامل تعدادملاقات کنندگان سایت وب شما باشد.
- یک متغیرApplication می تواند برای برقراری ارتباط بین کاربران سایت وب استفاده شود به عنوان مثال می توانید با استفاده از متغیرهایApplication یک بازی چندنفره به وجود آورید.

۶-۴ به وجود آوردن وخواندن متغیرهایApplication

به وجود آوردن یک متغیرApplication بسیار ساده است . باقراردادن نام متغیر درون شی ءApplication می توانید یک متغیر جدید به وجود آورید مانند مثال زیر:

```
<HTML >
<HEAD> <TITLE> Application Example</TITLE> </HEAD>
<BODY>
<%
Application ("Greeting") ="Welcome! "
%>
<% =Application ("Greeting ") %>
</BODY>
</HTML>
```

در این مثال متغیر Greeting به وجود آمده ومقدار Welcome به آن نسبت داده شده است . ودر آخر محتـوای متغیر روی مرورگر نمایش داده می شود.یک متغیرApplication که به آن مقداری نسبت داده مـی شـود مـی توانـد روی تمام صفحاتApplication نمایش داده شود . به عنوان مثال صفحه زیر نیز می تواند متغیر Greeting رانمایش دهد:

```
<HTML >
<HEAD> <TITLE> Another page</TITLE> </HEAD>
<BODY>
<% Application ("Greeting ") %>
</BODY>
</HTML>
```

باید توجه داشت که برخلاف متغیرهای session ،متغیرهایApplication بعد از ترک کاربر از بین نمی روند. مقداری که به این متغیر نسبت داده شده است تا زمانی که سرویس دهنده وب shut down نشود یاApplication Unloade نشود باقی می ماند و اگر خوش شانس باشد روزها وحتی ماهها باقی می ماند.

چون متغیرهایApplication بطور خودکار بعد از ترک کاربر از بین نمـی رود. بـاید مراقـب باشـید کـه آنـها را بدون هدف ایجاد نکنید. البته از آنجائیکه متغیرهایApplication حافظه را اشغال می کنند باید از آنـها کـم اسـتفاده کرد.

بهتر است بدانید که متغیرApplication مربوط بـه یـک کاربرخـاص نیسـت . اگریـک کاربردرخواسـت یـک صفحه وب را بکندکه به متغیرApplication آن یک مقدار داده شده است و کاربردیگری درخواست همان صفحـه را با مقداری دیگری بکند مقدار این متغیر برای هر دو صفحه عوض می شود ,اسکریپت زیرا را در نظر بگیرید:

```
<%
Rondomize
If  INT(2* RND )=1        THEN
  Application  ("Favoritecolor")= "Blue"
ELSE
Application ("Favoritecolor") ="Red"
END IF
%>
```

این اسکریپت به صورت تصادفی به متغیر Application ای بنام Favoritecolor مقدار Blue یا Red را نسبت می دهد . به هر حال مقدار متغیر برای هر دو کاربر یکی می شود، این کار می تواند باعث مشکل شود. از آنجائیکه هردو کاربر در آن واحد می توانند به متغیر Application دسترسی داشته باشند ، باید تداخلها را از بین برد.

خوشبختانه شیء Application دارای دومتداست که می توانند در تصحیح این وضع به ما کمک کنند. دو متد Lockو Unlock وقتی کاربر متغیری را تغییر می دهد به طور موقت می تواند مانع شوند که کاربردیگر آن راتغییر دهد. به مثال زیر توجه کنید:

```
%>
 Application.Lock
NumClicks= Application ("BannerClick")
NumClicks =NumClicks +1

Application ("BannerClicks ")= NumClicks
Application.Unlock
%>
```

خط اول این اسکریپت تمام متغیرهای شیء Application را قفل می کند . وقتی متغیرها قفل می شوند تا وقتی باز نشود کاربران دیگر نمی توانند این متغیرها را تغییر دهند .متغیرهای Application ، تازمانی که متد Unlock صدا زده شودیا انتهای صفحه دریافت شود ، قفل می ماند.

توجه داشته باشید که شما نمی توانید متغیرهای Application رابه صورت انتخابی قفل نمائید یاهمه راانتخاب می کنید یا هیچ کدام را؛ مثلاً اسکریپت قبل به طور موقت کاربران را از تغییر همهٔ متغیرهای Application ای که ممکن است وجود داشته باشد ، باز می دارد. مهم است بدانید که قفل کردن متغیرهای Application مانع تغییر متغیرها توسط کاربران به صورت دائمی نمی شود. قفل کردن تغییرات را منظم می کند. متغیرهای Application بیشتر به صورت پشت سرهم تغییر پیدا می کند تا به صورت تصادفی تغییر کنند.

۶-۵ ذخیره کردن متغیرهای Application

بیشتر متغیرهای Application در مجموعه Contents ازشیء Application ذخیره می شوند. وقتی یک متغیر Applicationجدید به وجود می آورید یک عنصر جدید به مجموعه اضافه می شود در مثال زیر دوعبارت معادل هم هستند.

```
<% Application ("Favoritecolor") =" Blue"%>
<% Application.Contents ("Favoritecolor") ="Blue"%>
```

از آنجائیکه متغیرهای Application درون مجموعه ذخیره می شوند، می توانید هر نوع تغییری را با استفاده از متدها روی آنها انجام دهید . با استفاده از متد count می توانید تعداد متغیرهای Application را بدست آورید وبا استفاده از حلقه های FOR...EACH و FOR...NEXT می توانید تمام آیتمهای مجموعهٔ Contents را نمایش دهید که در مثال زیراز حلقه FOR...EACH استفاده شده است.

```
<%
FOR EACH  thing  IN Application.Contents
Response.Write("<BR> " &  Thing &"="& Application.Contents (thing ))
```

```
NEXT
%>
```

این اسکریپت با قرار دادن مجموعهContents Application درحلقه تمام متغیرهای Application را نمایش می دهد.

۶-۶ رخدادهایApplication

مانند رخداد های Session ، Application نیز دارای دو رخداد است Application_Onstart و Application OnEnd_اولین رخداد وقتی به وجود می آید که یکApplication ازASP شروع شود ودیگری وقتی اتفاق می افتد که آن پایان یابد.

چه وقت یک Application شروع می شود؟ باید گفت یکApplication شروع نمی شود مگر اینکه اولین صفحه آن در خواست شود . یک رخداد Application_Onstart همیشه قبل از session_Onstartبه وجود می آید.

برخلاف رخداد Session_Onstart رخداد Application_Onstart هنگامی که کاربری جدید از Application درخواست یک صفحه می کند ، شروع نمی شود .رخداد Application_Onstart وقتی که اولین بازدید کننده به سایت وصل می شود شروع می شود.

Application_OnEndوقتی رخ می دهد که سرویس دهنده وب shut down شود یا Application ، unloadeشود . یک رخداد Application_onEnd بعد از آخرین رخداد session_ OnEnd به وقوع می پیوندد. به طور مثال این رخداد پس از اینکه با ISM سرویس دهندهٔ وب را خاموش می کنید به وقوع می پیوندد.

Application_onstartو Application_onEnd فقط یک اسکریپت را راه اندازی می کنند. هر دو اسکریپت باید درون فایل Global.asa قرار بگیرند این اسکریپت های خاص نمی توانند توسط صفحات دیگر صدا زده شوند.

فایل Global.asa یک فایل مخصوص است که در فهرست ریشه Application قرار دارد . هر Application فقط یکی از این فایلها را دارد . فایل Global.asa شامل تمام اسکریپتها واشیائی است که به صورت سراسری در یکApplication قرار دارند . این فایل دارای ساختار زیراست :

```
<SCRIPT  LA NGUAGE= VBScript      RUNAT = Server>
SUB Application _Onstart
END   SUB
</SCRIPT>

<SCRIPT      LANGVAGE = VBScript      RUNAT= Server>
SUB Application _OnEnd
</SCRIPT>
<SCRIPT     LANGAGE =VBScript        RUNAT= Server>
SUB Session_Onstart
END SUB
</SCRTPT>

<SCRIPT     LANGAGE =VBScript        RUNAT= Server>
SUB Session_OnEnd
END SUB
</SCRIPT>
```

آنچه که می توانید در اسکریپتهای Application_Onstart و Application_OnEnd بگنجانید ، بسیار محدود است . در این اسکریپت ها هیچ عبارت دیگری که مقدار خروجی داشته باشد رانمی توانید قرار دهید. به عنوان مثال از HTML یا متد Response.Write نمی توانید استفاده کنید. علاوه براین باید در استفاده از اشیائی که به همراه اسکریپتهای Application_Onstart و Application-OnEndبکار برده اید، محتاط باشید.

اسکریپت Application_Onstart برای مقدار دهی اولیه متغیرهای محدودهٔ Application ، بکار می رود. به عنوان مثال یک کاربرد عمومی session_Onstart و Application_Onstart دنبال کردن تعداد کل ملاقات کنندگان از زمانی که یکApplication شروع شده است ، می باشد. مثال زیر چگونگی انجام این کار رانشان می دهد .

```
<SCRIPT  LA NGUAGE= VBScript      RUNAT = Server>
SUB Application  _Onstart
Applicationn ("TotalUsers")=0
END   SUB
</SCRIPT>
<SCRIPT      LANGVAGE = VBScript      RUNAT= Server>
SUB session _Onstart
Applicationn .lock
Applicationn ("totalUseres")= Applicationn ("totalUsers")+1
END   SUB
</SCRIPT>
```

عبارت تکی اضافه شده به اسکریپت Applicationn_Onstart متغییری به نام Total Users مقدار اولیهٔ صفر می دهد.این اسکریپت تنها در شروع کار سرویس دهندهٔ وب اجرا می شود. اسکریپ session _Onstart هنگامی که یک کاربر جدید وارد سایت می شود مقدار TotalUser را یکی اضافه می کند. توجه داشته باشید که یک متغییرApplication قبل از هر تغییری قفل می شود. این مسئله ، از ناسازگاری هایی که با رسیدن چند کاربر رخ می دهد جلوگیری می کند. بعداز اینکه فایل Global.asa راتغییر دادید می توانید تعداد کاربران را توسط یک صفحه ASPبا اضافه کردن خط زیر نمایش دهید.

```
<% = Applicationn ("Total Users")%>
```

۶-۷ صفحهChat

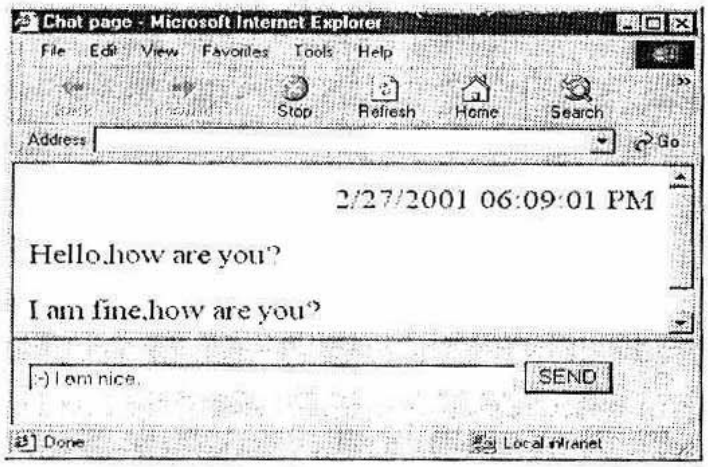
برای برقراری ارتباط بین دو کاربر برروی سایت وب می توانید از صفحه chat استفاده کنید . تمام کاربرانی که به صفحه chat احتیاج دارند می توانند پیغام های رسیده را ببینید . همچنین خودشان پیغام بفرستند. (رجوع شود به شکل٦-٢) برای به وجود آوردن صفحه chat باید سه فایل به وجود آیند یا تغییر کنند که این سه فایل عبارتند از:

۱- صفحهchat ، صفحه chat دوقاب دارد فریم بالایی برای پیغام های کاربران دیگر و فریم پایینی برای قرار دادن پیغام درون آن است.

۲- صفحه نمایش Display : این صفحه کل پیغام های رسیده توسط کاربران دیگر رانمایش می دهد با رسیدن هر پیغام جدید این صفحه به هنگام می شود.

۳- صفحه Message : این صفحه به کاربر اجازه خواهد داد پیغام جدید خود را قراردهد وشامل یک جعبهٔ ورودی متنی می باشد.

٤- فایل Global.asa که در آن یک اسکریپت در Onstart -Applicationn وجود دارد..



شکل ٦-١ صفحهٔ Chat

٦-٧-١ به وجود آوردن یک صفحهChat

اولین صفحه ای که لازم است به وجودآورید صفحه Chat است.این صفحه فقط وظیفهٔ جای دادن دو صفحهٔ دیگر را در خود دارد. از آنجائیکه این صفحه حاوی هیچ اسکریپتی نمی باشد شما باید آنرا بنام Chatpage.htm به طوری که در لیست زیر آمده ذخیره کنید:

```
<HTML>
<HEAD> <TITLE> Chat page  </TITLE> </HEAD>
<FRAMESET ROWS= "*;%10"
<FRAME SRC=" Display.asp">
<FRAME SRC ="Meseage.asp">
</FRAMESET>
</HTML>
```

فریم ها را به این دلیل که صفحهٔ نمایش هر پنج ثانیه یکبار تجدید[1] می شود ، بوجود می آوریم. این کار برای زمانهایی است که نیاز باشد یک پیغام جدید به یک صفحهٔ جدید وارد کرد و از طرفی پیغام کاربر هنوز نیمه کاره باشد و نباید تجدید بشود.

٦-٧-٢ تغییر فایلGlobal.asa

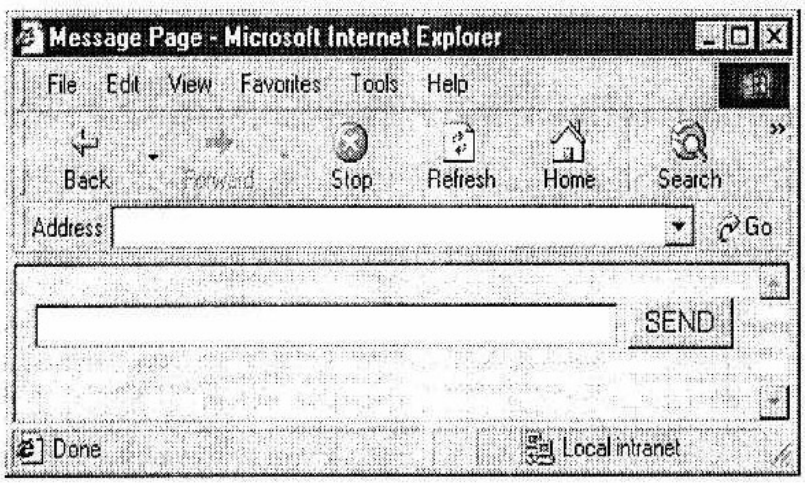
برای اینکه صفحهٔ Chat به کار بیافتد فایل Global.asa باید تغییر داده شود . اسکریپت های زیر متغیر های Application های مورد نیاز برای صفحه Chat را مقدار دهی می کند. متغیرها باید از نوع متغیرهای Application باشند تا تمام کاربران بتوانند به آنها دسترسی داشته باشند. اولین متغیر Talk می باشد. این متغییرآرایه ای است که تمام پیغام ها رانگه می دارد . آرایه Talk بوسیله تخصیص Temp Array به آن ایجاد می شود. دومین متغیر Tplace

(Talk place) نام دارد و برای اشاره کردن به آخرین پیغام در آرایه Talk بکار می رود .اسکریپت زیر به متغیر مقدار صفر می دهد :

```
<SCRIPT  LANGUAGE =VBScript     RUNAT= Server>
SUB     Application _Onstart
Dim     TempArray(5)
Application ("Talk  ") =TempArry
Application ("Tplace")=0
END   SUB
</SCRIPT>
```

٦-٧-٣ به وجود آوردن صفحه Message

منظور از به وجود آوردن صفحه Message ، دادن امکان به کاربران برای وارد کردن پیغام جدید است. (رجوع به شکل ٢-٦) این صفحه شامل یک فرم HTML است که درون آن نیز از جعبه ورودی متن[1] و , دکمهٔ submit استفاده شده است . هنگامی که روی کلید submit کلیک می شود صفحه خودش را reload می کند.



شکل٦-٢ صفحهٔ Message

درشکل ٦-٣ اسکریپت دوکار را انجام می دهد . اول چک می کند که آیا بیشتر از پنج پیغام وجود دارد یا نه، اگر بیشتر از پنج پیغام وجود داشته باشد دوباره به متغیر Tplace مقدار صفر داده می شود . این کار از سرریزشدن آرایه جلوگیری می کند ،سپس اسکریپت یک پیغام تازه را به آرایه Talk اضافه می کند ومقدار Tplace را یکی افزایش می دهد .Tplace معمولاً به مکان بعدی که پیغام باید درون آن قرار بگیرد اشاره می کند. لیست زیر محتوای صفحه Message.asp را نمایش داده شده است.

```
<%
IF   NOT   Request.Form ("message")= " "THEN
Application.Lock
IF     Appication  ("Tplace ")>4   THEN
       Application("Tplace ")=0
END   IF
 TempArray = Application("Talk")
```

```
 TempArray = Application("TPlace") = Request.Form("Message")
 Application("Talk ")=Temparray
 Application("TPlace")=Application("TPLACE")+1
     Application.Unlock
END IF
 %>
<HTML>
<HEAD > <TITLE>  Message   Page   </TITLE >   </HEAD>
<BODY   BGCOLOR ="LIGHTBLUE">
<FORM     METHOD ="POST"    ACTION= "message .asp">
<INPUT    NAME="MESSAGE"   TYPE="TEXT" SIZE=50>
<INPUT    TYPE = "SUBMIT"   VALUE= "SEND">
</FORM>
</BODY>
</HTML>
```

٥-٧-٤ به وجود آوردن Display

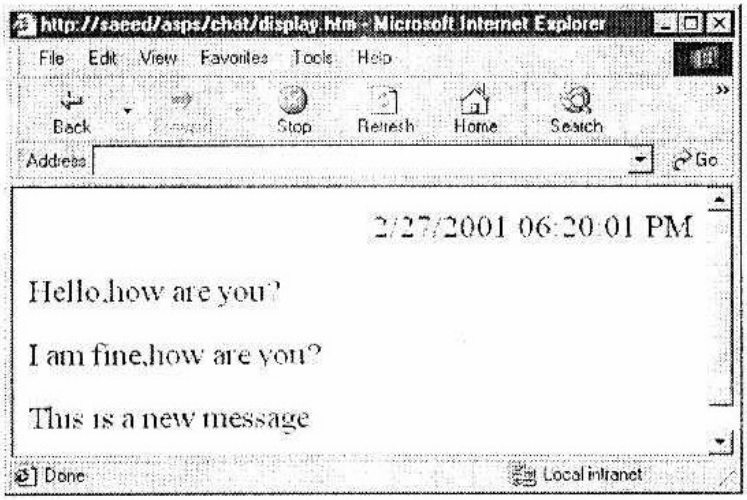
آخرین صفحه ای که باید به وجود آید Display است . این صفحه است که پیغام های همه کاربران رانمایش می دهد. (رجوع شود به شکل ٣-٦)

این صفحه به طور خودکار هر پنج دقیقه یکبار خودش را تجدید می کند . این کار را با استفاده ازclient-pull انجام می دهد .برچسب <META> شامل دستوراتی است که این کار را انجام می دهد . (سرآیند REFRESH را به ASP اضافه می کند)

ابتدای اسکریپت صفحه زیر برای مشخص کردن صفحه جاری بکار می رود . از مجموعهٔ ServerVariabels URL، کامل صفحه جاری را بدست آوریم وآن را به متغیر MySelf نسبت می دهیم . متغیر Myself نیزتوسط برچسب <META>برای تعیین صفحه ای که باید تجدید بشود بکار می رود.

اسکریپت اصلی برای نمایش محتوای آرایه Talk استفاده می شود . با استفاده از حلقه FOR...NEXT تمام پیغامهای جاری نمایش داده می شود. لیست زیر صفحهٔ Display.asp را نشان می دهد.

```
<%
 MyServer =Request.ServerVariables ("SERVER _ NAME ")
Mypath = Request.ServerVariables ("SCRIPT_ NAME ")
Myself = "HTTP:// "& Myserver&Mypath
%>
<HTML>
<HEAD >
<META   HTTP.EQUIV= REFRESH    CONTENT="5;<%=MySelf%>">
<TITLE > Display    page    </TITLE>
</HEAD>
<BODY>
<P  ALIGN = RIGHT><%= NOW%> </P>
<%
TempArray =Application ("Talk")
FOR    i=0   TO    Application("Tplace")-1
 Response.Write  ("<P>"& Temparray (1))
NEXT
%>
</BODY>
</HTML>
```



شکل ٦-٣ صفحهٔ Display

٦-٧-٥ طرح توسعه صفحهChat

راههایی برای تقویت صفحه chat وجود دارد . به عنوان مثال بیشترین پیغامی که صفحهٔ Chat می تواند در آن واحد نشان دهد پنج پیغام است که این برای استفاده های بالا کافی نمی باشد.

می توانید حداکثر تعداد پیغامها را با تغییر اندازه TempArray در فایل Global.asa وتغییر مقدار دهی مجدد TPlaceدر صفحه Message ، تغییر دهید.

صفحه chat به شما اجازه می دهد پیغام هایی که شامل فرمت HTML هستند را وارد کنید. به هرحال تغییر صفحه Message به منظور ساده تر کردن آن مشکل نخواهد بود . به عنوان مثال می توانید check Boxهایی برای قالب بندی پیغام داشته باشید که به شما اجازه تعریف فونت و رنگ پیغام را می دهد.

در آخر ، صفحه chat نمی تواند نام کاربر را همراه پیغام کندرفع این عیب کار مشکلی نمی باشد . می توانید یک صفحه logon را قبل از صفحه chat قراردهید تا نام کاربر را دریافت کند. سپس این نام می تواند در ابتدا همراه تمام پیغام هایی که توسط کاربر فرستاده می شود ، قرار گیرد.

٦-٧-٦ صفحهwhoson

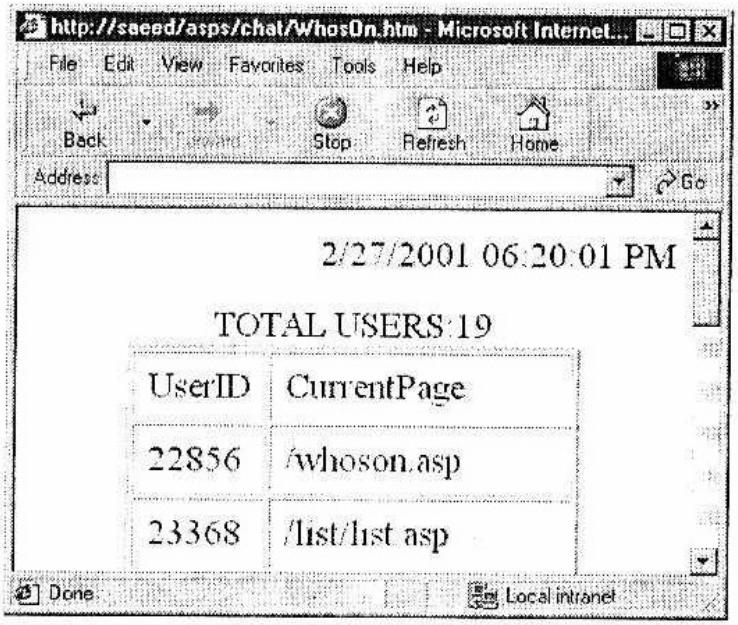
مدیران خوب وب یک هدف در زندگی دارند . آنها می خواهد کاربران سایت آنها بیشتر شود . بخشی از این وسواس مشخص کننده تعداد کاربرانی که روی خط می باشند است وتعیین می کند که آنها چه انجام می دهند . شما بلافاصله می توانید هر زمانی که بخواهید ،شمار بازدید کننده سایت وب خود را تعیین کنید. همچنین می توانید بفهمید آخرین صفحه ای راکه توسط کاربر درخواست شده است ، چه بوده. (رجوع به شکل ٦-٤)

این طرح توضیح می دهد که چگونه یک شیء را به یک متغیر Application نسبت دهیم . یک شیء Dictionary برای ذخیره اطلاعات کاربران مورد استفاده قرارگیرد. وقتی یک کاربر تقاضای یک صفحه را می کند ، اطلاعات درون شیء Dictionary بروز در آورده می شود . برای این طرح باید فایلهای زیر به وجود آمده یا تغییر داده شوند.

فایل Global.asa : برای این طرح هم اسکریپت Application_Onstart وهم اسکریپت Ses-sion_OnEndبایدتغییر داده شوند.

فایل Grabstats : این فایل شیء Dictionary را بروز می کند شما باید این فایل را همـراه همـه صفحـاتی کـه مـی خواهید آنهارا دنبال کنید بیاورید.

صفحه Whos On : این صفحه کاربران فعلی را روی سایت وب نشان می دهد.



شکل ٦-٤ صفحهٔ WhosOn

٦-٧-٧ تغییر فایل Global .asa

برای به وجود آوردن این طرح باید اسکریپتی که درون فایل Global.asa قـرار دارد را تغیـیر دهیـد. اول بـاید شیء Dictionary را به وجود آورید که اطلاعات کاربران را ذخیره می کند. از آنجاییکه این شیء باید فقـط یکبـار به وجود آید ، این کار را درون Application _ Onstart انجام می دهیم :

```
<SCRIPT LANGAGE=VBScript RUNAT= Server>
SUB Application_Onstart
Set Applicationn("stats")=Server.CreateObject("Scripting.Dictionary")
END SUB
<SCRIPT>
```

یک خـط بـه اسـکریپت قبلـی اضافـه شـده اسـت . عبـارتی کـه یـک نمونـه از شـی ء Dictionary رابـه متغیرApplication نسبت می دهد Stats نامیده می شود. این متغیر به محض ایجاد مـی توانـد در Application شـما مورد استفاده قرار گیرد. اسکریپت Session_OnEnd درون فـایل Global.asa بـایدتغییر داده شـود. هـدف از اسکریپت زیربرداشتن کاربر از شیء Dictionary هنگام خاتمه session کاربر، است.

```
<SCRIPT LANGUAGE = VbScript RUNAT = Server>
SUB Session _OnEnd
IF Application("stats").Exists(session.sessionID) THEN
Applicationn .Lock
Application("stats").Remove (session.sessionID)
Application.unlock
END IF
```

```
END SUB
</SCRIPT>
```

کاربران می توانند با sessionID هایشان دنبال شوند . کلیدهایی درون Dictionary با نام stats مطابق این sessionIDها می باشند . اسکریپت بالا بررسی می کند که آیا sessionID مربوط به کاربرفعلی Dictionary وجود دارد یا نه و درصورت وجودآن را حذف می کند..

٦-٧-٨ در وجود آوردن فایل Grabstats

برای تعیین صفحه ای که کاربر روی آن قرار دارد شما باید دستورات زیر را روی هر صفحه ای که می خواهید دنبال کنید، قراردهید . این دستورات شامل یک اسکریپت تک خطی می باشدکه در زیر نشان داده شده است.

```
<%
Application("stats") .item (session . session ID)=
Request .ServerVariables ("SCRIPT_NAME")
%>
```

این اسکریپت مسیرهر صفحه را به شیء Dictionary اضافه می کند. با استفاده از متغیر محیطی سرویس دهندهٔ SCRIPT_NAMEمسیر صفحه جاری را بدست می آوریم . بعد ، مقدار این متغیر را به یک کلید Dictionaryکه مطابق با کاربرفعلی است نسبت می دهیم. (اگر این کلید وجود نداشته باشد ،به صورت خودکار بوجود می آید).

این فایل را با نام Grabstats.asp ذخیره می کنیم و آنرا با هرصفحه ای که می خواهیم دنبال شود همراه می کنیم به راحتی می توانید این خط را در بالای ASP اضافه کنید.

```
<!-- # INCLUDE  VIRTUAL = "Grabstats.asp" -->
```

٦-٧-٨ به وجود آوردن صفحه whoson

صفحه whoson برای نمایش کاربران جاری استفاده می شود. sessionID هر کاربر پس آخرین صفحه ای که کاربر درخواست کرده ، نمایش داده می شود. در زیر صفحه whoson.asp نشان داده شده است.

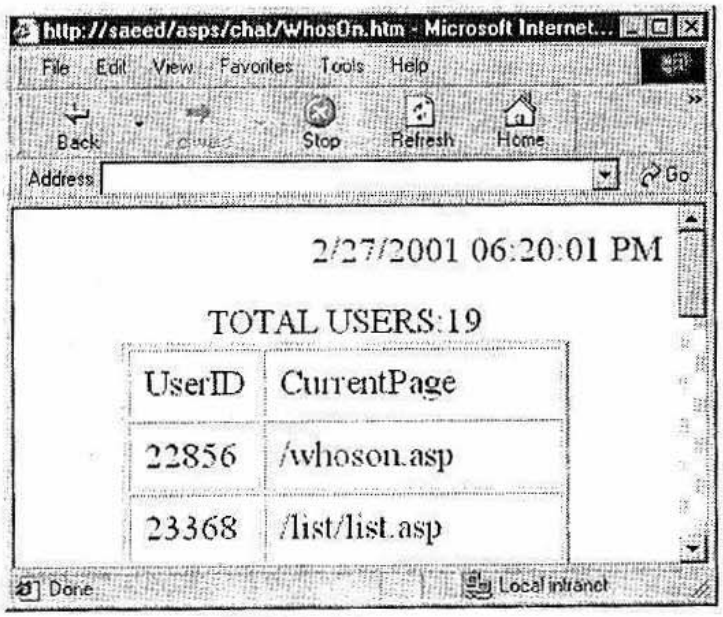
```
<!--# INCLUDE  VIRTUAL = "Grabstats.asp" -->
<%
MyServer =Request.ServerVariables ("SERVER _NAME")
Myself ="HTTP =//"&Myserver&Mypath
%>
<HTML>
<HEAD>
<META  HTTP.EQUIV="REFRESH" CONTENT="20;<%=Myself %>">
<TITLE >WHOSON</TITLE>
</HEAD>
</BODY>
<% Application .Lock
Set  TempStats = Application("stats")
Application.unlock
%>
<CENTER>
<B> TOTAL  USERS:</B>  <%=TempStats.Counts%>
<TABLE  BORDER =1  CELLPADDING=10>
<TR> <TH> User  ID  </TH><TH>  Current  Page</TH></TR>
```

```
<% TempItems=Tempststs.items
Tempkeys = TempStats.keys
For i=0 to UBOUND(TEMPKEYS)
%>
<TR><TD><%= Tempkeys ( i) %> </TD><TD><%=TempItems(i)%></TD>
   </TR>
<%
NEXT
%>
</CENTER>
</TABLE>
</BODY>
</HTML>
```

خط اول شامل فایل Grabststs.asp است. این کار به شماره اجاره می دهد که بدانید چه وقت این صفحه را می بینید . مسیر صفحه whoson بعداً در session ID شما ظاهر می شود.

اولین اسکریپت مسیر صفحه جاری را دریافت می کند . صفحه whoson با استفاده از client_pull هر ۲۰ ثانیه به صورت خودکار تجدید می شود . برای تجدید کردن این صفحه مسیر آنرا لازم دارید. اسکریپت دوم دیکشنری ذخیره شده در متغیر Application را در شیء موقتی Dictionary که Temp Stats نام دارد ذخیره می کند.Temp Statsبه طور خودکار در زمان خاتمه صفحه از بین می رود.

با استفاده از شمارش عناصر در dictionary می توانید تعداد جاری کاربران را بدست آورید. اسکریپت آخر با قرار دادن کلیدها و عناصرش در یک حلقه تمام مقادیر ورودی به dictionary را نمایش می دهد.



شکل ٦-٥ کاربران و آخرین صفحهٔ مورد دسترسی

توسعهٔ صفحهٔ WhosOn:

صفحهٔ WhosOn اطلاعات تغییر پذیری را دربارهٔ کاربران سایت وب شما را فراهم می کند. راههای زیادی وجود دارد که می توان توسط آن پروژهٔ قبل را توسعه داد. برای مثال اگر سایت وب شما نیاز به ثبت داشته باشد شما

می توانید اسامی کاربران را متناسب با Session ID ها نمایش دهید. برای این کار از اسامی کاربران همچون کلیدی در دیکشنری استفاده کنید.

به علاوه فهمیدن اینکه هر کاربرچه مدت Online بوده، مفید خواهد بود. شما می توانید پروژه را به روشهای مختلفی برای دنبال کردن این اطلاعات ، تغییر دهید. به طور نمونه شما می توانید به سادگی متغیر Application ثانویه ای ایجاد کنید که حاوی دیکشنری دیگری باشد. در این حالت شما می توانید طول مدت بازدید کاربر از سایت را با ذخیرهٔ این اطلاعات در این دیکشنری ثانویه بدست آورید.

فصل هفتم :

کاربا مرورگرها

این فصل اجزاء اکتیوایکسی که در Asp گنجانده شده اند را معرفی می کند. بخش اول به ما نمائی از متدهای جمع کردن اجزاء در Asp را می دهد. در قسمت بعدی فصل چگونگی استفاده از اجزء امکانات مرورگر[1] را فرامی گیرید . در انتهای فصل نیزیک برنامهٔ کاربردی ساده در رابطه با اجزء امکانات مرورگر آمده است.

۷-۱ استفاده از اجزاء ASP

در فصلهای پیشین چگونگی استفاده از اشیاء Asp مثل Request و Response شرح داده شده است. اجزاء ASP به این اشیاء بسیار شبیه می باشند. بااینکه یک جزء آنچنان با ASP پیوندی ندارند ، لیکن اجزاء یکی از بهترین گزینه ها برای توسعهٔ متدها و توابع فراهم آمده بوسیلهٔ اشیاء تعبیه شده می باشند. شما می توانید با استفاده از زبانهایی مثل C++ و جاوا و ویژوال بیسیک یا دلفی برای خودتان اجزائی را ایجاد کنید. همچنین می توانید اجزاء را از شرکتهای نرم افزاری خریداری کنید به غیر از اینها میکروسافت نیز تعدادی از اجزاء اکتیوایکس را با ASP همراه کرده است که در این فصل و فصلهای بعدی به آنها خواهیم پرداخت. قبل از آنکه شما بتوانید از یک جزء استفاده کنید. اول باید نمونه ای از آن را ایجاد کنید. با این کار شما می توانید به خصوصیات ، مجموعه ها ومتد های یک شی ء از آن جزء دسترسی داشته باشید .

۷-۲ به وجود آوردن یک جزء با محدودهٔ عمل صفحه

در بیشتر موارد شما یک نمونهٔ جزء را با محدودهٔ عمل صفحه به وجود خواهید آورد. یک جزء با محدودهٔ عمل صفحه روی یک صفحهٔ تکی به وجود می آید و وقتی که پردازش صفحه پایان می یابد آن نیز از بین می رود . شما نمی توانید یک جز ء با محدودهٔ عمل صفحه را روی صفحهٔ دیگری که به طور صریح ایجاد نشده استفاده کنید. برای ایجادنمونه ای از یک جزء با محدودهٔ عمل صفحه باید از متد ( ) Server.Createobject استفاده کنید.

اینجا یک مثال از ایجاد یک جزء با محدودهٔ عمل صفحه وجود دارد:

```
<%
Set MyBrow=Server.CreateObject ("MSWC.BrowserType")
%>
```

این اسکریپت یک نمونه ای از جزء امکانات مرورگر را ایجاد می کند. این اسکریپت متغیر MyBrow را به یک نمونهٔ این جزء نسبت می دهد . قابل توجه است که از عبارت set از VBScipt استفاده شده است. به علت اینکه شما یک نمونه از جزء رابه یک متغیر نسبت می دهید ، بایداز عبارت set استفاده کنید .متد ایجاد یک جزء با استفاده از

جاواسکریپت خیلی شبیه به این است تنهاباید به جای استفاده از set از عبارت Var استفاده کنید . مثال زیر از عبارت Var استفاده کرده است .

```
%>
Var MyBrow = Server.CreateObject ("MSWC.Browsertype")
<%
```

مایکروسافت توصیه می کند که عملکرد اجزاء را بوسیلهٔ محدودهٔ عمل صفحه بوجود آورید. با ایجاد اجزاءبا محدودهٔ عمل صفحه شما فضای کمتری از منابع سرویس دهندهٔ وب را اشغال می کنید. وقتی پردازش خاتمه می یابد یک جزء با محدودهٔ عمل صفحه هر حافظه ومنبعی را که احتیاج داشته است رها می کند.

۷-۳ ایجاد اجزاء با محدودهٔ عمل session

دومتد برای ایجاد اجزاء باsession وجود دارد . یکی این است که یک جزء را با استفاده از server.CreatecObject () به متغیری از نوع session به شکلی که در مثال زیر می بینید نسبت دهیم.

```
<%
Set session("MyBrow") = Server.CreateObject ("MSWC.BrowseType")
%>
```

این اسکریپت یک متغیر session به نام MyBrow را به یک نمونه ازاجزاء امکانات مرورگر نسبت می دهد . این متغییر session روی تمام صفحاتی که یک کاربرمشخص تقاضا می کند قابل استفاده است. شما می توانید این اسکریپت را درون Session_Onstart از فایل Global.asa یا هر ASP دیگری قرار دهید.

روش دومی برای ایجاد یک جزء با session محدودهٔ وجود دارد . با استفاده از برچسب <object>می توانید یک جزء را درون فایل global.asa ایجادکنید . مانند زیر:

```
<OBJECT RUNAL="Server" SCOPE= "session" ID="MyBrow"
PROGID= "MSWC.BrowserType"</OBJECT>
```

این مثال ، چگونگی ایجاد یک نمونه از جزء امکانات مرورگر با استفاده از برچسب <object> رانشان می دهد . ویژگی محدودهٔ عمل تعیین می کند که جزء ایجاد شده محدودهٔ عمل sessionدارد. ویژگی ID به جزء یک شناسه خاص نسبت می دهد. بطوریکه شما می توانید در اسکریپتهای ASP خود به آن رجوع کنید. PROGID برای تخصیص یک نام ثبت شده می باشد . این نامی است که سرویس دهنده برای شناسایی جزء وقتی که یک نمونه ازآن به وجود می آید، استفاده می کند و همان نامی است که از آن در متد Server.Createobjectاستفاده می کنید.

هنگامی که از برچسب <object>درون فایل Global.asa استفاده می کنید آن را باید خارج از هر اسکریپت دیگری قراردهید . از برچسب <object>درون Session , Session_OnEnd , Application_Onstart یا Application_OnEnd استفاده نمی کنیم .

وقتی یک جزء بامحدودهٔ عمل Sessionباهریک از دو روش شرح داده شده ایجاد شود ، هر یک از متدها ، مجموعه ها یاخواصش روی هر صفحه ای که یک کاربر درخواست می کند , در دسترس می باشد . به هرشکل یک

نمونه خاص از جزء باید برای هر کاربر ایجاد شود . شبیه یک متغیر Session، یک جزء نیز که بامحدودهٔ عمل Sessionایجاد شده باشد به یک کاربرخاص وابسته است.

۷-۴ به وجودآوردن اجزاء با استفاده ازمحدودهٔ عمل Application

هنگام ایجاد نمونهٔ یک جزء دارای محدودهٔ عمل Application شما می توانید با آن همانند یک شیء تعبیه شده رفتار کنید. به محض ایجاد، متدها ، مجموعه ها و خصوصیات جزء آنها برای کاربران در هر صفحه ای قابل دسترسی می باشند، وتازمانی که سرویس دهندهShut down نشده باشد یا فایل global.asa تغییر داده نشده باشد و یا Unload Application نشود ، در دسترس باقی می ماند.

می توانید جزء دارای محدودهٔ عمل Application را با همان متدی که در ایجاد جزء دارای محدودهٔ عمل session استفاده می کنید به وجود آورید. این کار را می توانید توسط متد Server.Createbject() همانند مثال زیر انجام دهید :

```
<%
Set Application("MyBrow") = Server.Ceateobject ("MSWC.BrowserType")
%>
```

در بالا جزء امکانات مرورگر به یک متغیر Application تخصیص داده شده است . شما می توانید این کار را در یک اسکریپت درون فایل global.asa مثل اسکریپت Application_Onstart انجام دهید . بااین روش می توانید یک جزء را درون هر صفحه ASP ای بگنجانید. بعد از اینکه یک جزء دارای امکانات مرورگر با محدودهٔ عمل Application به وجود آمد از خواص آن می توانید در تمام صفحات استفاده کنید .

شما می توانید یک جزء با محدودهٔ عمل application را با برچسب <object > به وجود آورید.

مانند زیر:

```
<OBJECT RUNAT= " server" SCOPE="Application" ID="MyBrow"
PROGID= "MSWC.BrowserType" </OBJECT>
```

در این مثال برچسب < object >برای ایجاد یک نمونه از جزء امکانات مرورگر با محدودهٔ عمل Application بکاررفته است . ویژگی محدودهٔ عمل مشخص کرده است که جزء بیشتر باید دارای محدودهٔ عمل Application باشد تا session.ویژگیID یک نام برای جزء فراهم می کند. ویژگی PROGID به سرویس دهنده اجازه می دهد تا یک جزء راشناسایی کند.شما می توانید برچسب <object> را درون فایل global.asa قرار دهید . البته این برچسب باید خارج از هر اسکریپت دیگری قرار گیرد. همچنین برچسب <object>درون ، اسکریپتهای Application_Onstart ، session_Onstart, Session_OnEnd و Application_OnEnd قرار نمی گیرد.

براستی چه وقت شما نیاز به ایجاد شیء با محدودهٔ عمل Application دارید؟ در برنامه نویسی WhosOn که در فصل قبل آمده شما چگونگی پیگیری درخواستهای کاربران را فراگرفتید. این اطلاعات در یک دیکشنری ایجاد

شده و دارای محدودهٔ عمل Application می شود. لازم است که این اجزاء با محدودهٔ عمل Application به وجود آیند در غیر اینصورت کاربر روی تمام صفحات نمی تواند به آنها دسترسی پیدا کند.

۷-۵ جزء امکانات مرورگر

یکی از دلایل رشد سریع اینترنت بازبودن استانداردهای آن است . HTML طراحی شده تا با هر سیستم عامل و مرورگری کار کند. در تئوری یک صفحهٔ وب باید صرفنظر از هر مرورگر و کامپیوتری نمایش داده شود . البته این اصل در عمل همیشه صادق نیست. نت اسکیپ از ابتدا شامل برچسب های HTML اختصاصی بود. مثلاً نت اسکیپ نسخهٔ 1.0 می توانست برچسب HTML ای را که متن را چشمک زن می کرد تفسیر کند که حتی تا مدتها مرورگرهای دیگر قادر به تفسیر آن نبودند. فریمها نیز یکی دیگر از موارد توسعهٔ HTML توسط نت اسکیپ بود. در همان موقع رقابت بین شرکت مایکروسافت و نت اسکیپ بالا گرفته بود ، رقابتی که هیچ ثمره ای نداشت. به طور مثال مایکروسافت نیز برچسب هایی را برای HTML ایجاد کرده بود همانند <BGSOUND> (که صدای پس زمینه را پخش می کرد) و یا <MARQUE> که تنها توسط مرورگر مایکروسافت قابل تفسیر بود. بدین ترتیب همواره این سؤال برای طراحان وب مطرح بود که با این تفاوتها و شکافها بین مرورگر ها چه باید بکنند؟ مایکروسافت راه چاره را در استفاده از جز امکانات مرورگر ، معرفی کرد.

۷-۵-۱ استفاده از جزء امکانات مرورگر

شما می توانید از جزء امکانات مرورگر برای نمایش صفحات مختلف وب بسته به امکانات مرورگر استفاده کنید. به عنوان مثال بعضی از مرورگرها از فریم ها پشتیبانی می کنند بعضی دیگر نمی کنند . با استفاده از جزء امکانات مرورگر می توانید تعیین کنید که آیا مرورگر از فریم ها حمایت می کند یا نه .

به طور پیش فرض ، جزء امکانات مرورگر می تواند خصایص ذیل را از مرورگر وب تشخیص بدهد:

- bowser: نوع مرورگر را تعیین می کند، به عنوان مثال Netscape یا Internet Explorer
- version: نسخه مرورگررا تعیین می کند.
- Majorver: نسخه اصلی مرور گر راتعیین می کند (شماره قبلی مرورگر).
- Minorver: نسخه بعدی مرورگر
- Frames: تعیین می کند که آیا مرورگراز فریم ها حمایت می کند ؟
- Table: تعیین می کند که آیا مرورگراز Table ها حمایت می کند؟
- Cookies: تعیین می کند که آیا مرورگراز cookie ها حمایت می کند؟
- Back ground sounds : تعیین می کند که آیا مرورگر از برچسب <BGSOUND> حمایت می کند؟
- vbscript : تعیین می کند که آیا مرورگراز VBscript ها حمایت می کند؟
- javascript: تعیین می کند که آیا مرورگراز javascript ها حمایت می کند؟
- Activexcontrol : تعیین می کند که آیا مرورگر از اکتیوایکس سمت سرویس دهنده حمایت می کند؟
- beta: تعیین می کند که آیا مرورگر حاضر نسخهٔ بتا است؟

- Platform : تعیین می کند که سیستم عامل مرورگر چیست مثلاً Win 98، WinNtیا Macpower Pc می باشد.
- Win16 : تعیین می کند که آیا مرورگرروی win3.x جای win95وwin NT اجرا می شود؟

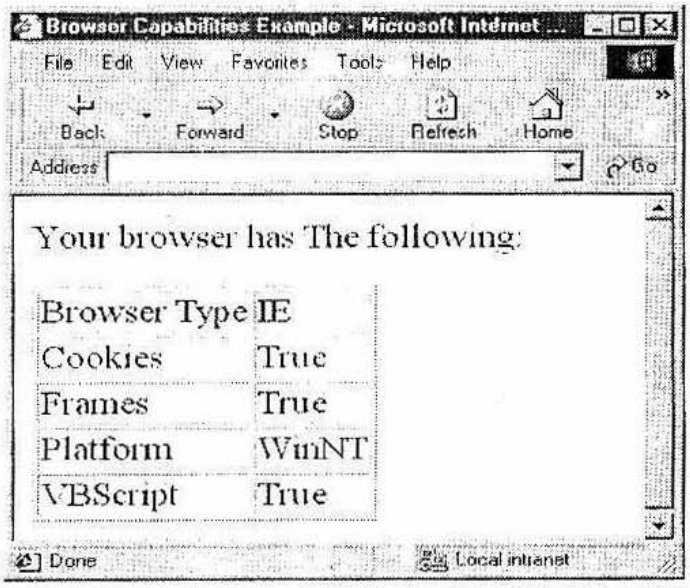
برای استفاده از جزء امکانات مرورگر شما احتیاج به ایجاد یک نمونه از آن دارید . بعدا شما می توانید ابزاری را که برای تعیین نام جزء نمونه می خواهید، ضمیمه آن کنید .در زیر مثالی وجود دارد که شماری از خصایص مرورگر را نشان می دهد:

```
<HTML>
<HEAD> <TITLE> Browser  Capabilities Example </TITLE></HEAD>
<BODY>
<%
Set MyBrow=Server.CreateObject ("MSWC.BrowserType")
%>
<P>Your browser has The following:

<TABLE  BORDER=1 CELLPADDINE =10>
<TR>
<TD> Browser Type  </TD> <TD>  <%=MyBrowser%></TD>
</TR>
<TR>
<TD> Cookies  </TD> <TD> <% =MyBorw.Cookies%></TD>
</TR>
<TR>
<TD>  Frames  </TD><TD>  <%=MyBrow.Frame %></TD>
</TR>
<TR>
<TD> Platform </TD> <TD> <%= MyBrow.platform %></TD>
</TR>
<TR>
<TD> VBScript </TD> <TD> <%=MyBrow.VBScript%> </TD>
</TR>
</TABLE>
```

```
</BODY>
</HTML>
```

این اسکریپت شماری از خصایص مرورگر را پیدا کرده و نمایش می دهد. برای مرورگر متفاوت نتایج متفـاوتی نمایش داده خواهد شد. شکل (۱-۷) یـک خروجـی نمونـه در رابطـه بـا NerscapeNavigator3.0 و شـکل (۲-۷) خروجی نمونه در رابطه با Internet Explorer را نشان می دهد.

شکل ۱-۷ نمونه ای خروجی جزء قابلیتهای مرورگر با استفاده از نت اسکپ ۳



شکل ۲-۷ نمونه ای خروجی جزء قابلیتهای مرورگر با استفاده از IE 4.0

۷-۵-۲ جزء امکانات مرور گر چگونه کارمی کند

فهم چگونگی کارکرد حقیقی جزء امکانات مرورگر مهم است چرا که شما به برخی از محدودیتها پـی خواهیـد برد. این جزء خصایص مرورگر را بااستفاده از یک سرآیند درخواست HTTP ویـک فـایل متنـی خـاص کـه شـامل اطلاعاتی درباره مرورگر می باشند ، پیدا می کند.

وقتی یک مرورگر درخواستی می فرستد ، آن درخواست شامل یک سرآیند USER-AGENT نیز می باشد. این سرآیند شامل اطلاعاتی درباره نوع مرورگر بکار رفته ونسخه آن می باشد . شما می توانید این سرآیند را مستقیماً با استفاده از مجموعه ServerVariables از شی ء Request همانند زیر بدست آورید.

<%=Request.ServerVariables ("Http_USER_AGENT")%>

هنگامی که به طور مثال از Netscape Nevegator 3.0 استفاده می کنید مقدار سرآیندUSER_AGENTرا به صورت زیر می بینید.

Mozilla /3.0 (win NT ;bl)

مهم است بدانید که این (درخواستها) تنها اطلاعاتی می باشند که بین مرورگروسرویس دهنده ردو بدل می شود. جزء امکانات مرورگربستگی به اطلاعات درون این سرآیند دارد . جزء هیچ یک از خصایص مرورگر را به صورت مستقیم پیدا نمی کند.

جزء امکانات مرور گر ، مقدار سرآیند USER_AGENT را دریافت می کند و سعی می کند آن را با تعریف یک مرورگر که در فایل خاصی به نام browscap.ini وجود دارد تطبیق دهد. فایل browscap.ini درون سرویس دهنده قرار دارد . وقتی ASPهارا نصب کردید این فایل به صورت خودکار به خوبی نصب می شود .

فایل browscap.ini چیزی بیشتر از یک فایل ساده متنی نیست که حاوی لیستی از مرورگرها وخصوصیات آنها است . به عنوان مثال مرورگر Netscape Navigtor نسخه ۲که در فایل browscap.iniتعریف شده به صورت زیراست.

```
[Netscape    2.0]
browser=Netscape
version=2.0
majorver=2
minorver=Q
frames= TRUE
tables=TRUE
cookies =TRUE
backgroundsounds=FALSE
vbscript=FALSE
javascript =TRUE
javaapplets = TRUE
beta = False
win16= False
```

جزء امکانات مرورگر این تعریف را هنگام گزارش خصوصیات Netscape Navigator نسخهٔ 2.0 بکار می برد. شما می توانید مستقیماً این فایل متنی را تغییر دهید . برای مثال می توانید با تغییر backgrounddsound از False به True بگوئید که Netscape Navigtor نسخهٔ 2.0 می تواند از برچسب <Bgsound> استفاده کند .

تعدادی ازخصوصیات مرورگر ها هستندکه جزء امکانات مرورگر بایدآنهارا تشخیص دهد ولی این کار را نمی کند. به عنوان مثال خیلی مفید بود اگر می توانستید ازجزء برای تشخیص اینکه آیا یک مرورگر معین می تواند از Secure Sockets Layer یا cascading style sheets استفاده کند بهره ببرید. از آنجائیکه این اطلاعات در

فایل browscap.ini وجود ندارد ، به هرحال شما نمی توانید از جزء امکانات مرورگربرای تشخیص این خصایص استفاده کنید . ولی می توانید این اطلاعات رابه فایل browscap.ini اضافه کنید . برای نمونه می توانید دو خط ورودی زیر رابه Netscape Navigator 2.0 اضافه کنید .

```
SSL=TRUE
CSS=FALSE
```

وقتی این دو خط را اضافه می کنیم جزء امکانات مرورگر این خصایص را از Netscape Navigator 2.0 گزارش می دهد . هر جا که جزء تعیین کند که مرورگر Netscape Navigator 2.0 است (توسط سرپیام USER-AGENT) . جزء فرض خواهد کرد که مرورگر این خصوصیت را دارد برای مثال ، اسکریپت زیر وقتی که داخل صفحه ای باشد که با Netscape Navigator 2.0 دریافت شده ، True را برمی گرداند :

```
<%=MyBrow.ssl%>
```

باید توجه داشته باشید که تعریف خیلی از مرورگرها درفایل browscap.ini مانند زیر وجود دارد:

```
[(Mozilla /2.0   (win 95 ;u]
parent=Netscape   2.f
platform= win95
```

وقتی که تعریف مرورگر یک پارامتر parent دارد . این تعریف تمام خصوصیات parent رابه ارث می برد. تعریفی که در قبل آمده تمام خصوصیات مرورگر Netscap Navigator 2.0 را به ارث می برد. به عنوان مثال ، اگر چه گفته نشود که آیا Netscape نسخه windows 95 می تواند از فریم ها استفاده کند ، جزء امکانات مرورگر گزارش خواهدداد که این نسخه از Netscape Navigator می تواند ازفریم استفاده کند زیرا Parent آن می تواند این کار را انجام دهد .با پارامتر parent ،اجباری در وارد کردن تکراری اطلاعات یکسان وجود ندارد. شما می توانید یک تعریف parent به وجود آورید و شماری از تعاریف Child کوچکترکه شامل اطلاعات خاصتری هستند را به وجود آورید . هر خصوصیت مرورگر تعریف شده در child که با تعریف parent مغایریت دارد دارای تقدم خواهد بود .

جزء امکانات مرورگر وابسته به درستی فایل browscap.ini می باشد. اگر کسی از یک نسخه از مرورگری استفاده کند که درون فایل browscap.ini نباشد ، جزء امکانات مرورگر گزارش درستی نمی تواند بدهد . وقتی یک جزء امکانات مرورگر نتواند مرورگر را شناسائی کند، خصوصیات تعیین شده برای مرورگر پیش فرض را گزارش می دهد. مثال زیر تعریف مرورگر پیش فرض می باشد:

```
[Defaulf   Browser   Capability   Settings]
browser =Default
version =0.0
majorver=#f
minorver=#f
frames = false
Tables = True
Cookies= false
backgroundsound =false
vbscript= false
```

```
javascript=False
Javaapplets =Fals
activexcontrols = Fals
AK=Fals
SK= Fals
AOL= Fals
beta =false
win16=False
crawler =False
CDF =False
```

اگر شما علاقه ای به خصوصیات پیش فرض تعریف شده در فایل Browscaap.ini ندارید می توانید مستقیماً آن راتغییر دهید. برای مثال شاید نخواهید که تمام مرورگرها به طور پیش فرض از جداول استفاده کنند. برای تغییر این فرض شما می توانید به سادگی مقدار خصوصیت Table را در تعریف پیش فرض مرورگر تغییر دهید.

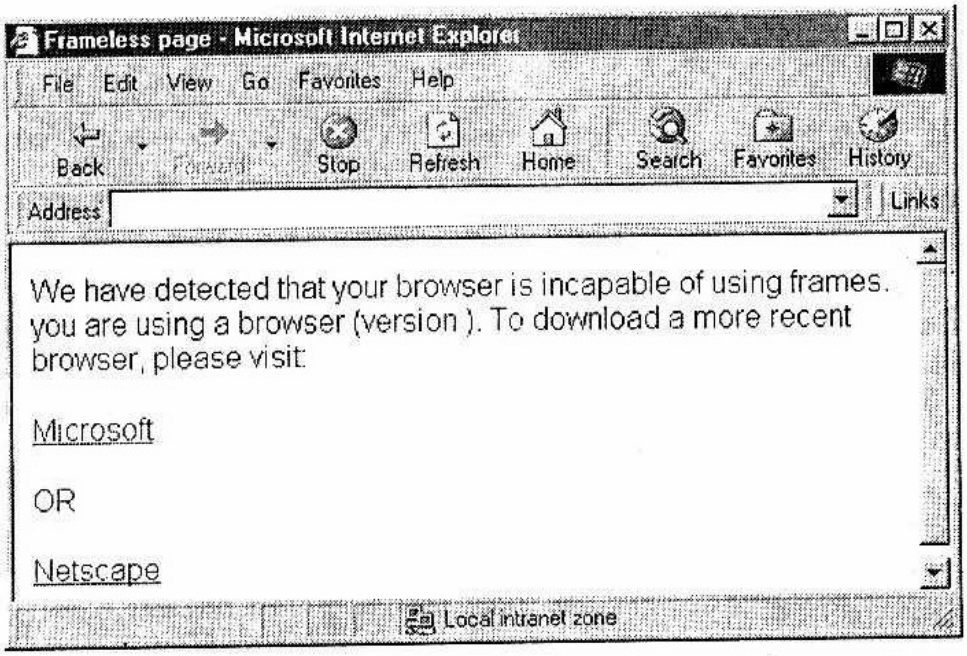
۷-۵-۳ یک برنامهٔ کاربردی برای جزء امکانات مرورگر

در این بخش به یک برنامهٔ کاربردی قابل استفاده ازجزء امکانات مرورگر، می پردازیم. با مشاهدهٔ این مثال نه تنها چگونگی استفاده از جزء را می بینیم بلکه محدودیتهای آنرا نیز درخواهیم یافت. در ASP نشان داده شده در لیست زیر جزء امکانات مرورگر برای پیدا کردن اینکه یک مرورگر می تواند از فریمها استفاده کند بکار رفته است. اگر مرورگر بتواند از فریمها استفاده کند یک نسخه فریم دار صفحه، نمایش داده می شود و در غیر اینصورت کاربران را آگاه می کند که برای دیدن این سایت باید از یک مرورگر سازگار با فریم استفاده کنند.

```
<%
Set MyBrow = Server.CreateObject ("MSWC.Browsertype")
IF MYBrow.Frames THEN
%>
<HTML>
<HEAD ><TITLE> Framed page  </TITLE> </HEAD>
< FRAMESET COLS ="100,*">
<FRAME    SRC ="menu.asp">
<FRAME  SRC="body.asp">
</ FRAMESET>
</HTML>
<%ELSE%>
<HTML>
<HEAD ><TITLE> Frameless    page </TITLE> </HEAD>
<BODY>
We have detected that your browser is incapable of using frames.
you are using a <% =MyBrow.browser%>browser
(version <%=MyBrow.version%>).
To download a more recent browser, please visit:
<p> <A  HREF="http ://www.microsoft.com">Microsoft </A>
<P> <OR>
<P> <A   HREF="http://www.netscape.com"> Netscape </A>
```

```
</BODY>
</HTML>
<% END IF%>
```

این ASP به صورت شرطی در دو صفحه مختلف نمایش داده می شود . اگر جزءامکانات مرورگر تشخیص دهد که مرورگر کاربر برچسب <Frame> را می شناسد صفحه اول نمایش داده می شود. که این صفحه دو فریم دارد و در غیر اینصورت صفحه دوم که دارای فریم نیست نمایش داده می شود. توجه کنید که یک جزء امکانات مرورگر به چه صورت برای گزارش نام ونسخه مرورگربکار رفته است .دستورات مثال قبل تنها برای نشان دادن چگونگی استفاده از جزء امکانات مرورگر نمی باشد و به طور معمول نباید برای این هدف استفاده شود . این مورد روشن کننده یک مشکل جدی با اجزاء مرورگر است .



شکل ۷-۳ نتایج بد از جزء امکانات مرورگر

مشکل این ASP در اینجا است که معمولاً صفحه دوم را وقتی که مرورگر را تشخیص ندهد ، نمایش می دهد . به عبارت دیگر حتی اگر مرورگری بتواند.از فریم ها حمایت کند ولی تشخیص داده نشود صفحه دوم نمایش داده می شود . به عنوان مثال ، براساس آنچه دراین کتاب آمده، نسخه جاری از فایل browscap.ini جدیدترین نسخه مرورگر Netscape را تشخیص نمی دهد . بنابراین حتی اگر این مرورگر بتواند از فریم ها استفاده کند ، دومین صفحه نمایش داده می شود.

نقص اصلی جزء امکانات مرورگر این است که به اجباروابسته به اطلاعات قرارداده شده بوسیله مایکروسافت در فایل browscop.ini شما می باشد.بهتراست از خود HTML برای نمایش محتویات مختلف ، بسته به امکانات مرورگرها استفاده شود . به عنوان مثال ، بهترین راه برای تعیین اینکه آیا مرورگر می تواند از فریم ها استفاده کند استفاده از برچسب <NOFRAMES> است . امتیاز این روش آن است که کارش را با مرورگرهای جدید ادامه می دهد .

فصل هشتم:

کار بافایلها ، درایوها وپوشه ها

در این فصل بطور کلی به جزء سیستم فایل[1] ASP پرداخته خواهد شد. با استفاده از این جـزء درون ASP ، شـما می توانید به کنترل کاملی روی سیستم فایل کامپیوترتان نائل آیید. بخش اول فصل دیدگاهی از اشیاء استفاده شـده توسط این جزء را به شما می دهد. در بخش دوم شما چگونگـی خوانـدن و نوشـتن درون یـک فـایل متنـی را فرامـی گیرید. در بخش سوم می فهمید که چگونه با متدها، خصوصیات و مجموعه های فایل کـار کنیـد. و در بخـش آخـر متدهای دستکاری فهرستها و پوشه ها[2] ، بررسی می شود.

۸-۱ بررسی جزء دستیابی فایل:

درنسخه های قبلیIIS دسترسی به سیستم فایل بسیار محدود بود؛تا جائیکه شما کار ی جـز خوانـدن و نوشـتن در یک فایل را نمی توانستید انجام دهید. این محدودیت کاملاً مشهود بود به طور نمونه شما هیچ روش مستقیمی جـهت بررسی این که یک فایل وجود دارد یا نه ، در اختیار نداشتید. با آمدن نسخه های جدید IIS این مشکل نیز حل شد. با استفاده اسکریپت هایASP نسخه های جدید IIS، اکنون تقریباً روی تمام جنبه های سیسـتم فـایل نظـارت و تسـلط ایجاد شده است . برای کار با فایلها شما باید از جزء سیستم فایل استفاده کنید که این جزء اشیاء زیر را بکار می برد:

FileSystemObject : این شیء شامل تمام روشهای اصلی جهت کار با سیستم فایل می باشد برای نمونه می تـوان از متدهای این شیء جهت کپی و حذف پوشه ها یا فایلها استفاده نمود.

TextStream : برای خواندن و نوشتن روی یک فایل از این شیء استفاده می شود.

File : خواص و متدهای این شیء کار با فایلهای ویژه را ممکن می کند.

Folder : خواص و متدهای این شیء کار با پوشه های فایل را ممکن می کند.

۸-۲ نوشتن و خواندن روی یک فایل:

کاربردهای زیادی برای فایل متنی وجود دارد که متداول ترین آنها از قرار زیر است:

A custom log: بکار بردن یک فایل متنی برای ثبت فعالیتهایی که بازدیدکنندگان ازسایت وب شما انجـام مـی دهند . شما می توانید اطلاعاتی مانند آدرسهای IP آنها نوع مرورگر آنها و مقـدار زمـانی کـه آنـها در سـایت وب شما بسر می برند را ثبت کنید.

FORM Data : بکار بردن یک فایل متنی جهت ذخیره نمودن اطلاعات جمـع آوری شـده ازیـک فـرم HTM بـه عنوان مثال اگر یک کار بر اطلاعات ثبت نام در یک فرم HTML وارد کند شما می توانید آن اطلاعات را در یـک فایل متنی ذخیره کنید.

Tip Of The Day : ذخیرهٔ لیستی از TIP ها را برای استفاده سایت وب شما در یک فایل متنی و دریافت و نمایش تصادفی آنها روی یک صفحهٔ وب.

۸-۲-۱ نوشتن روی یک فایل متنی:

برای ایجاد و نوشتن در یک فایل متنی می توانید از اشیاء FileSystemObject و TextStream استفاده کنید. ابتدا باید یک نمونه شیء FileSystemObject ایجاد کنید سپس متد( )CreateTextFile ازشیء FileSystemObject را برای برگرداندن نمونه شیء TextStream صدا بزنید. سرانجام از متد ( )WriteLine از شیء TextStream برای نوشتن اطلاعات درفایل استفاده کنید. مانند مثال زیر:

```
%>
Set MyFileObject =Server.CreateObject ("Scripting.FileSystemObject")
Set MyTextFile = MyFileObject.CreatTextFile ("c:\my dir \ test.txt ")
MyTextFile.WriteLine ("Hello There! ")
MyTextFile.Close
%>
```

این مثال یک فایل به نام test.txt با مسیر c:\mydir \ test.txt ایجاد می کند . برای فرستادن متن تک سطری Hello There!به فایل از متد ( ) WriteLine استفاده می شود در انتها نمونه شیء TextStream برای محافظت منابع سیستم بسته می شود . در ادامه می توان با جزئیات بیشتری این کار را مرور کرد :

متد ( ) CreateTextFileبرای ایجاد فایل متنی جدید استفاده می شود . زمانی که این متد درخواست می شود یک شیء TextStream برگردانده می شود.این متدیک پارامترالزامی ودوپارامتراختیاری داردکه آنها از قرار زیرند:

FileSpecifier : که مسیر فایل ایجاد شده را مشخص می کند. اگر فهرست داده شده در مسیر موجود نباشد پیغام خطای File not found برمی گردد.

Overwrite :این پارامتر اختیاری است و به طور پیش فرض مقدار آن TRUEاست. در اینصورت با فراخوانی متد ( )CreateTextFile به طور خودکار روی هر فایلی که از قبل موجود بوده و نامش مشابه فایل جدیدی که قرار است ایجاد شود، نوشته خواهد شد و اگر این پارامتر مقدارش False باشد در صورت وجود با نام یکسانی پیغام خطا داریم. که نشان دهندهٔ این است که فایل باید با استفاده از مجموعهٔ کاراکترهای اسکی ایجاد شود. و اگر TRUE باشد فایل باید توسط مجموعهٔ کاراکترهای Unicode ایجاد شود.

Unicode: این پارامتر نیز اختیاری است، بطور پیش فرض مقدار آن FALSE است پس از این که یک فایل با استفاده از متد( )CreatTextFile ایجاد شد می توان از شیء TextStream جهت نوشتن روی فایل استفاده نمود . جهت نوشتن می توان از متدهای زیر استفاده کرد.

([رشته]) Write :این متد یک رشته را در فایل می نویسد.

([رشته]) WriteLine :این متد یک رشته را در فایل می نویسدهمچنین یک کاراکترخط جدید[1] را نیزاضافه می کند.

(تعدادخطوط)WriteBlankLines:این متد به تعداد معین خطوط خالی(باکاراکترهای خط جدید)درفایل می نویسد.

Close: این متد برای بستن یک فایل TextStream و آزاد کردن منابع به کار می رود . به عنوان مثال برای ایجاد فایلی حاوی متنی که کلمه !Hello World، ۳۲ مرتبه در یک سطر آن تکرار شده باشد می توان از دستورات زیر زیر استفاده کرد:

```
<%
Set MyFileObject = Server.CreatObject ("Scripting.FileSystemObject")
Set MyTextFile = MyFileObject.CreatTextFile ("c: \my dir \ test.txt ")
FOR i = 1 to 32
  MyTextFile.WriteLine ("Hello World! ")
NEXT
MyTextFile.Close
% >
```

۸-۲-۲ خواندن و افزودن اطلاعات به یک فایل متنی

برای خواندن اطلاعات از یک فایل متنی شما ابتدا نیاز به ایجاد نمونه ای از یک شیءFileSystemObject دارید سپس می توانید از متد ( )OpenTextFile برای بازگرداندن یک نمونهٔ شیء TextStream استفاده نمایید. سرانجام می توانید از متد ReadLine متعلق به شیء TextStream برای خواندن فایل استفاده کنید.

نمونه ای از این کار درذیل آمده است :

```
<%
Set MyFileObject = Server.CreateObject ("Scripting.FileSystemObject")
Set MyText File = MyFileObject.OpenText File ("C:\Mydir \ Test.Txt")
WHILE NOT MyText File.At EndOf Stream Response.Write (MyTextFile.ReadLine)
WEND
MyTextFile Close
%>
```

اسکریپت بالا هر چیزی را از فایل Test . Txt می خواند و محتویات فایل را به مرورگر می فرستد. اگر فایل وجود نداشته باشد پیغام خطای File Not Found را باز می گرداند.

حلقهٔ WEND ... WHILE باعث می شود آنقدر در میان محتویات فایل حرکت کنیم تا به انتهای آن برسیم. خاصیت AtEndOf Stream مادامی که حلقه به انتهای فایل نرسیده مقدارش FALSE خواهد بود.در هنگام خواندن از یک فایل متنی خواص زیر که مربوط به شیء TextStream هستند مفید واقع می گردند:

AtEndOfLine: این خصوصیت نشانگر این است که آیا در یک فایل متنی به انتهای سطر رسیده ایم یا خیر ؟ زمانیکه کاراکترسطر جدید بدست آمد این خصوصیت مقدار TRUE دارد.

AtEndOfStream : این خصوصیت نشانگر این است که به پایان فایل متنی رسیده ایم یا خیر. این خصوصیت نیز می تواند دارای مقدار TRUE یا FALSE باشد.

Column :این خصوصیت موقعیت کاراکتر جاری در یک خط رانشان می‌دهد. این خصوصیت یک عدد صحیح برمی گرداند.

Line:این خصوصیت موقعیت خط جاری را نشان می دهد این خصوصیت نیز عدد صحیح بر می گرداند. . بجای استفاده از متد ( ) Readline برای خواندن ابتداء تا انتهای محتویات فایل ، شما می توانید از متد ( )Read استفاده

نمائید. متد ( ) Read تعداد مشخصی کاراکتر را از یک فایل متنی باز می گرداند . مثال زیر چگونگی آن را بیان می دارد:

```
<%
Set MyFileObject = Server.CreatObject (" Scripting.FileSystemObject ")
Set MyTextFile=MyFileObject.OpenTextFile (" c:\my dir \test.txt")
WHILE NOT MyTextFile.AtEndOfLine
Response.Write(MyTextFile.Read (1))
WEND
MytextFile.Close
%>
```

این اسکریپت اولین خط فایل را کاراکتر به کاراکتر می خواند. خاصیت AtEndOfLine زمانی که به انتهای اولین خط فایل متنی رسیده باشیم Trne رابرمی گرداند. برای خواندن یک کاراکتر از فایل متنی از متد( ) Read استفاده می‌نماییم. متدهای زیر هنگام خواندن داده از یک فایل متنی بسیار مفید هستند:

(تعداد کاراکترها) Read: این متد تعداد معینی از کاراکترها را از یک فایل متنی می خواند.

ReadLine: این متد یک خط از فایل متنی را می خواند. (کاراکترهای خط جدید خوانده نمی شوند).

ReadAll: این متد کل اجزاء فایل Textstream را بدست می آورد.

(تعداد کاراکترها) Skip : این متد تعداد معینی کاراکتر را از یک فایل متنی حذف می کند.

SkipLine: این متد یک خط از یک فایل متنی را حذف می کند.

Close :این متد یک فایل TextStream باز را می بندد ومنابع را آزاد می کند.

در حالت عادی از متد ( )OpenTextFileجهت دریافت داده از یک فایل متنی استفاده می شود. البته شما می توانید از این متد جهت الصاق دادهٔ جدید به یک فایل متنی نیزاستفاده کنید. مانند مثال زیر:

```
<%
Set MyFileObject =Server.CreatObject (" Scripting.FileSystemObject ")
Set MyTextFile=MyFileObject.OpenTextFile("c:\mydir \bowser.log",8,TRUE)
MyTextFile.WriteLine (Request.ServerVariables ("HTTP_USER_AGENT")
MyTextFile.Close
%>
```

این اسکریپت یک log در مرورگرهایی که در یک سایت وب استفاده شده اند بوجود می آورد .هر زمان که اسکریپت اجرا گردد نوع مرورگری که صفحه ای را در سایت درخواست کرده در یک فایل متنی ثبت می گردد. این اطلاعات از مجموعهٔ ServerVariables دریافت می شود. اسکریپت قبل نوع مرورگر را به انتهای یک فایل متنی به نام browser.log اضافه می نماید. اگر فایل browser.log موجود نبود، اولین مرتبه ای که این اسکریپت اجرا شود این فایل به طور خودکار ایجاد می گردد. این مورد توسط دو پارامتر متد( )OpenTextFile یعنی Createو IOMode انجام می گیرد.

لیست زیر تمام پارامترهای متد OpenTextFile را توضیح می دهد:

FileSpecifier : مسیر فایل برای بازکردن را تعیین می کند.

IOMode : این پارامتر اختیاری است و هرگاه که فایل برای خواندن و یا الصاق ویا نوشتن ، باز می شود می تواند بکار برود. مقدار پیش فرض آن ۱ و نشانگر عمل خواندن است. هنگام باز کردن یک فایل جهت نوشتن مقدار آن ۲ است. هنگام باز نمودن یک فایل جهت الصاق مقدار آن ۸ است.

Create :این پارامتر اختیاری نشان می دهد که آیا فایلی که باید ایجاد شود موجود است یا نه. پیش فرض مقدار این پارامتر FALSE است.

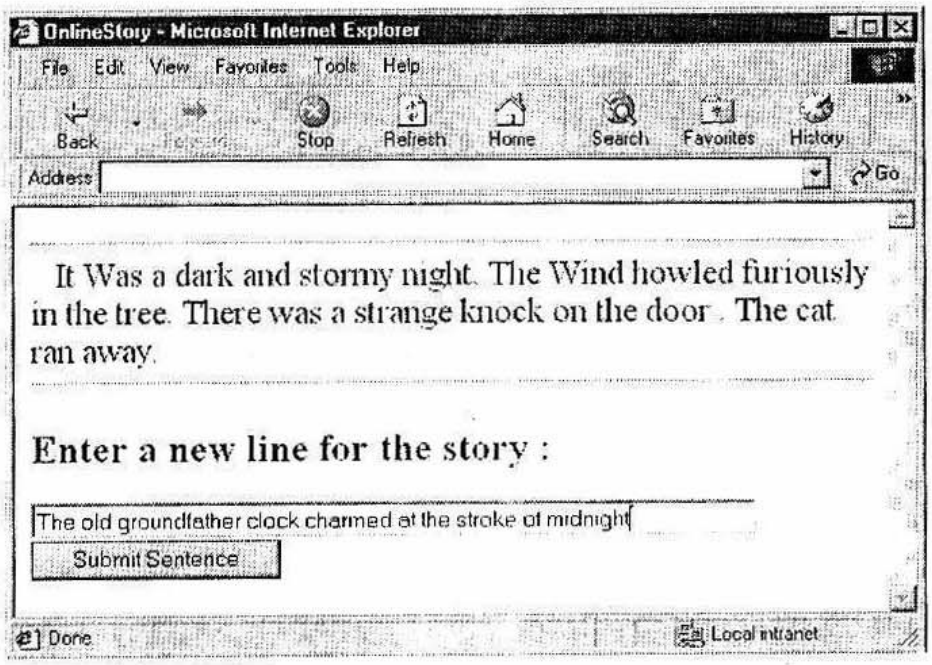
Format : این پارامتر اختیاری فرمت فایل را مشخص می نماید. بطور پیش فرض یک فایل از مجموعه کاراکترهای اسکی استفاده می نماید لیکن می توانید از مجموعه کاراکترهای Unicode با دادن مقدار ۱ یا مجموعهٔ کاراکترهای اسکی با دادن ۲ استفاده کنید.

۸-۲-۳ برنامهٔ کاربردی نمونه :

در این بخش به یک برنامهٔ کاربردی می پردازیم که در آن از متدهایی برای خواندن و نوشتن روی فایلها استفاده شده، است. فرض کنید که سایت Amazon.Com (که یک سایت فروش کتاب است) یک مسابقه برای نوشتن یک داستان همگانی به راه می اندازد. اولین قسمت داستان توسط شخصی به نام جان آپدیک[1] وارد شده. هر روز بازدید کنندگان سایت وب Amazon.Com ، جملات جدیدی به این داستان اضافه می کنند و هر روز یک جمله انتخاب می شود و برنده ۱۰۰۰ دلار دریافت می کند وبدین ترتیب این داستان با کمک همه نوشته می شود. ایجاد این رقابت یک ایده بزرگ است. این کار بازدیدکنندگان را مکرراً جذب سایت وب Amazon.Com می کند ، عمومیت چشمگیری را به همراه دارد و افراد بسیار زیادی را به پای این شبکه فرا می خواند. ممکن است بخواهید چیزهایی شبیه به این را به سایت خود اضافه نمائید. مثال زیر نمونهٔ ساده ای از چگونگی انجام این کار است.

```
<%
IF NOT Request.Form("NextLine") ="  "  THEN
Set MyFileObject = Server.CreateObject ("Scripting.FileSystemObject")
Set MyTextFile=MyFileObject.OpenTextFile("c:\mydir\TheStory.txt",8,TRUE)
MyTextFile.WriteLine(Request.Form("NextLine"))
MyTextFile.Close
END IF
%>
<HTML >
<HEAD> <TITLE> OnlineStory </TITLE> </HEAD>
<BODY>
<HR>
<%
Set MyFileObject = Server.CreateObject ("Scripting.FileSystemObject")
Set  MyTextFile = MyFileObject.OpenTextFile ("c:\mydir \ The story .txt ")
WHILE NOT MyTextFile.AtEndOfStream
Response.Write ("& nbsp; & nbsp ; "&MyTextFile .ReadLine)
WEND
MyTextFile.Close
%>
<HR>
<H3> Enter a new line for the story : </H3>
<FORM  ETHOD =" POST"  AGTION = "story .asp " >
<INPUT   NAME ="Next Line "   TYPE ="TEXT"  SIZE =70">
```

```
<INPUT  TYPE =" SUBIT"  VALUE= " Submit Sentence">
</FORM>
</BODY>
</HTML>
```



شکل ۸-۱ یک داستان مشارکتی

این ASP حاوی دو اسکریپت است. اسکریپت اول زمانی اجرا می گردد، که جمله جدیدی اضافه شده باشد. اگر جمله جدیدی ایجاد گردد، این جمله به فایلی به نام TheStory. txt افزوده می شود. اسکریپت دوم محتویات فایل TheStory.txt را نمایش می دهد(رجوع شود به شکل ۸-۱). خطوط توسط دو کاراکتر فاصله از یکدیگر جدا می شوند بنابراین وقتی جملات به اجرا در می آیند توسط فضاهای خالی تقسیم و طبقه بندی می گردند. بقیهٔ ASP حاوی یک فرم HTML برای ارائهٔ خط بعدی داستان است. برای اینکه Asp شما کار کند باید نام آن را story.asp بگذارید. برای اولین بار قبل ازاینکه از این ASP استفاده نمائید لازم است که یک فایل متنی به نام TheStory.txt ایجاد کنید . زیرا اولین جمله این داستان باید در این فایل وارد شود واگر بخواهید داستان را مجدداً شروع نمائید، بسادگی می توانید محتویات این فایل را پاک کنید وجمله جدیدی را وارد نمائید.

۸-۳ کار با فایلها:

این بخش شامل چگونگی کار با فایلها از قبیل چگونگی کپی، انتقال ، حذف و تشخیص موجود بودن آنها و دریافت ویژگیهایشان می باشد.

کپی ، انتقال و پاک کردن فایلها:

برای کپی ، حذف ، و انتقال یک فایل چندین متد وجود دارد. برای این کار شما می توانید ازمتدهای شیء File و یا متدهای شیء FileSystemObject استفاده نمائید. متدهای FileSystemObject کمی انعطاف پذیرتر است زیرا در آن شما به کار با یک فایل تکی محدود نمی شوید. لیست زیر متدهای FileSystemObject برای دستکاری فایلها را نشان می دهد:

[جانویسی[1]] مقصد ، مبداء ، CopyFile : این متد یک فایل را از یک محل به جای دیگر کپی می کند. شما می توانید از وایلدکارد[2] در پارامتر مبدأ استفاده نمائید تا بیشتر از یک فایل در یک زمان کپی شود. پارامتر اختیاری جانویسی معین می کند که آیا فایل موجود رویش جاوانویسی شود یا خیر و می تواند مقدار True یا False داشته باشد.

مقصد ، مبدأ MoveFile : این متد یک فایل را از یک محل به جای دیگر انتقال می دهد. برای این کار می توانید از وایلدکاردها در پارامترمبدأ استفاده نمائید وبیشتر از یک فایل را انتقال دهید. اگر قبلاً فایلی در مقصد با نام مشابه موجود باشد پیغام خطا می دهد.

مشخصهٔ فایل DeleteFile:این متد فایل تعیین شده را حذف می کند . شما می توانید از وایلدکارد برای حذف بیش از یک فایل در یک زمان استفاده کنید. اگر وایلدکارد استفاده کنید و هیچ مطابقتی وجود نداشته باشد، پیغام خطا می دهد. پیش از استفادهٔ هر کدام از این متدها ابتدا لازم است که یک نمونه از شیء FileSystemObject را ایجاد نمائید. مثال زیر طرز استفادهٔ هر کدام از متدهای بالا را نشان می دهد. همچنین می توانید به جای استفاده از شیء FileSystemObjectجهت کپی کردن، انتقال دادن و یا حذف فایلها از شیء File استفاده نمائید. در ذیل متدهای معادلی که می توانید در شیء Fileاستفاده کنید، بیان شده است:

```
<%
'Create an Instance Of The File System Object Object
Set My File Object=Server.CreatObject( "Scripting.FileSystemObject" )
' Create a file to Manipulate
Set MyFile=MyFileObject.CreateTextFile("C:\Test. txt" )
My File.Write Line ("Hello")
My File.Close
'Copy The File
My FileObject.Copy File " C:\Test.Txt , C:\Test 3.Txt" Delete Both Files
My FileObject.Delete File " C:\Test.Txt
My File Object.Delete File " C:\Test.Txt
%>
```

[جانویسی] ، کپی جدید Copy : این متد یک کپی جدید از فایل جاری را ایجاد می کند. اگر چنانچه پارامتر اختیاری جانویسی مقدار TRUE داشته باشد فایل موجود جاوا نویسی می شود. کپی جدید نیز نام و مسیر فایل جدید است.

کپی جدید.Move : این روش فایل جاری را انتقال می دهد. و فایل جاری به این فایل مرجوع می گردد.

Delete: فایل جاری را حذف می کند.

پیش از اینکه از این روشها استفاده کنید باید یک نمونه از شیء File را ایجاد کنید. برای این امر می توانید از متد ( )GetFile شیء FileSystemObject استفاده نمائید. در ذیل مثال قبلی به کمک متدهای شیء File تکرار شده است.

```
<%
' Create an Instance Of The Files System Object Object
Set MyFileObject = Server.CreatObject ( "Scripting.FileSystemObject" )
' Create afile to Manipulate
Set MyFile=MyFileObject.CreateTextFile ( "C:\ Test.Txt" )
MyFile.Write Line ("Hello")
MyFile.Ciose
'Create an Instance Of The Files System Object
Set MyFile Object = Server.CreatObject ( "Scripting.FileSystemObject" )
Copy The File
AFile.Copy "C:\Test 2.Test "
' Move The File
AFile.Move "C:\Test3.Test "
Delete The Original File
Afile.Delete
%>
```

۸-۳-۱ تعیین وجود فایل:

برای فهمیدن اینکه آیا یک فایل خاص موجود است یا نه می توانید از متد FileExists() ازشیء FileSystemObject استفاده نمائید. می توانید به سادگی مسیر فیزیکی فایل را به این متد منتقل کنید تا متد مقدار TRUE یا FALSE برگرداند. به یک مثال از چگونگی اجرای آن توجه کنید:

```
<HTML>
<HEAD> <TITLE> FileExsits Example </TITLE> </HEAD>
<BODY>
<%
MySelf =Request.ServerVariables ("PATH_TRANSLATED")
' Creat an inst ince of the FileSystemObject object
Set MyFileObj ect=Server.CreateObject ("Scripting.FileSystemObject ")
IF MyFileObject.FileExists(Myself)  THEN
   Response.Write("I exist! ")
ELSE
   Response.Write("I do not exist. ")
END IF
%>
</BODY>
</HTML>
```

این ASP بررسی می کند که فایل مورد نظر وجود دارد یا نه. متغیر محیطی سرویس دهنده PATH_TRANSLATED برای بازگرداندن مسیر فیزیکی فایل جاری مورد استفاده قرار می گیرد .متد FileExists موجودیت فایل را بررسی می کند.

۸-۳-۲ دریافت خصوصیت های فایل ها :

شیء File حاوی شماری از خصوصیات است که هنگام کار با فایلهای مفید واقع می شوند. لیست زیر این خاصیتها را توضیح می دهد:

Attributes:این خصوصیت مشخصات فایل جاری را باز می گرداند. (مثل دستور ATTRIB در Dos عمل می کند) برای مثال می توانید از این خاصیت برای تعیین مخفی بودن[1] یا فقط خواندنی بودن[1] فایل استفاد. کنید.

DateCreated:این خصوصیت، تاریخ و ساعتی که فایل ایجاد گردیده است را باز می گرداند.

DateLastAccessed:این خصوصیت آخرین تاریخ و ساعتی که به فایل دسترسی انجام شده را باز می گرداند.

DateLastModified:این خصوصیت آخرین تاریخ و ساعتی را که فایل اصلاح شده باز می گرداند.

Drive:این خصوصیت فهرستی که فایل در آن قرار دارد را باز می گرداند.

Name:این خصوصیت نام فایل را باز می گرداند

ParentFolder:این خصوصیت پوشه ای که فایل در آن قرار دارد را باز می گرداند.

Path:این خصوصیت مسیر فایل را باز می گرداند.

Size:این خصوصیت اندازه فایل را به بایت باز می گرداند.

Type:این خصوصیت نوع فایل را باز می گرداند.

قبل از استفاده از هر کدام از این خواص ابتدا باید یک نمونه شیء File ایجاد کرد. در ادامه مثالی در این رابطه بیان شده است. خصوصیت Attributesنیاز به کمی توضیح دارد. این خصوصیت به ازای هر ویژگی تنظیم شده یک شماره برمی گرداند. جدول زیر مقادیر ویژگیهای فایل را لیست کرده است:

ویژگی	مقدار
Normal	0
Read – Only	1
Hidden	2
System	4
Volume	8
Directory	16
Archive	32
Alias	64
Compressed	128

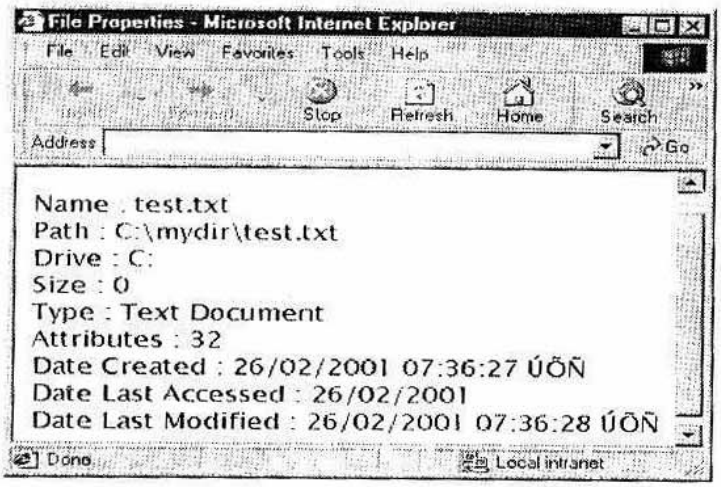
فایل نشان داده شده در شکل (۸-۲) دارای ویژگیهای Hidden , Archive می باشد. بعضی از این ویژگیها را علاوه برخواندن می توان تنظیم نیز نمود. شما می توانید خصوصیات Archive , System , Hidden را تنظیم کنید. مثلاً برای Hidden کردن فایل C:\Test.txt شما می توانید از اسکریپت زیر استفاده کنید.

```
<%
' Create an Instance of the File System Object Object.
Set MyFileObject = Server.CreateObject ("Scripting.FileSystemObject")
' Create an Instance of the File Object.
Set  AFile = Server.MyFileObject Get File ("C:\Test.txt") Makeit Hidden
AFile.Attribute = 2
%>
```

همهٔ خصوصیات برای فایلی با مسیر C:\Test.txt درشکل ۸-۲ نشان داده شده است.

```
<HTML>
<HEAD><TITLE>File Properties </TITLE></HEAD>
<BODY>
```

```
<%
'Create an Instance of the File System Object Object.
Set MyFileObject = Server.CreateObject ("Scripting.FileSystemObject")
'Create an Instance of the FileSystemObject Object.
Set aFile =MyFileObject.etfile("C:\Test.txt")
%>
<BR>Name : <% = aFile. Name %>
<BR>Path : <% = aFile.Path %>
<BR>Drive : <% = aFile.Drive %>
<BR>Size : <% = aFile.Size %>
<BR>Type : <% = aFile.Type %>
<BR>Data Created : <% = aFile.DateCreated %>
<BR>Last Accessed : <% = aFile.LastAccessed %>
<BR>Last Accessed : <% = aFile.LastAccessed %>
<BR>Last Modified : <% = aFile.LastModified %>
</BODY>
</HTML>
```



شکل ۸-۲ خصوصیات فایل

۸-۴ کار با درایوها :

دو شیء برای دریافت اطلاعات دریواهای ماشین محلی وجود دارند:

یکی شیء FileSystemObject و دیگری شیء Drive . به عنوان مثال ASP زیر لیستی از تمام درایوهای سرویس دهنده و کل اندازهٔ آنها واندازهٔ قابل دسترسی را نشان می دهد: (رجوع شود به شکل ۸-۳)

```
<HTML>
<HEAD> <TITLE> Drive List </TITLE > </HEAD>
<BODY>
<%
' Create an instance of the FileSystemObject object
Set MyFileObject=Server.CreateObject (" Scripting.FileSystemObject")
' Loop through the Drives collenction
FOR EACH thing in MyFileObject.Drives
%>
<BR> Drive Letter :  <% =thing.DriveLetter %>
<BR> Drive Total Size : <%= thing.TotalSize %>
```

```
<BR> Drive Availables Space:  <%= thing.AvailableSpace%>
<HR>
<%
NEXT
%>
</BODY>
</HTML>
```

شکل ۸-۳درایوهای روی ماشین محلی

مجموعهٔ Drives از شیء FileSystemObject حاوی مجموعهٔ تمام درایوهای موجود در سرویس دهنده است. البته این مجموعه تنها حاوی آندسته از درایوهایی است که با نام درایو(که یک حرف می باشد) نگاشت یافته اند. در زیر متدهای شیء FileSystemObject که مرتبط با درایوها می باشند آمده است:

(مشخصهٔ درایو[۱]) DriveExists : اگر درایو مشخص شده موجود باشد TRUEبر می گرداند.

Drives:مجموعهٔ درایوها را برای ماشین محلی بر می گرداند.

(مشخصهٔ درایو) GetDrive:یک شیء Drive را که نماینده درایو مشخص شده است بر می گرداند.

(مسیر[۲]) GetDriveName: رشته ای که حاوی درایو مسیر مشخص شده است را باز می گرداند.

تعجب آور نخواهد بود اگر شیء Drive حاوی شماری از متدها و خصوصیت های مفید برای کارکردن با درایوها باشد. این متدها و ویژگیها در زیر آمداست.

AvailableSpace: فضای قابل دسترس روی درایو را به بایت بر می گرداند.

DriveLetter: نام درایو را باز می گرداند برای مثال D,C یاE.

DriveType: تعداد مشابه ای از انواع درایو به طور نمونه نوع CD - ROM یا درایو قابل انتقال را باز می گرداند.

FreeSpace:میزان فضای آزاد درایو را به بایت باز می گرداند(مثل AvaibleSpace).

IsReady: نمایانگر این است که آیا یک حجم قابل استفاده است این خصوصیت نشان دهندهٔ وضعیت درایوهای قابل انتقال می باشد.

Path: نشان دهندهٔ مسیر درایو است.

RootFolder:این خصوصیت یک شیء Folder که نشان دهندهٔ شاخهٔ پوشه روی درایواست را بر می گرداند.

SerialNamber : شماره سریال درایو را باز می گرداند.

ShareName: نامی که برای به اشتراک گذاشتن درایو در شبکه استفاده شده است را باز می گرداند.

TotalSize:حجم کل درایو را به بایت باز می گرداند

VolumeName:یک رشته که نشانگر نام درایو است را باز می گرداند.

جهت استفاده از این خواص و متدها لازم است که یک نمونه ازشیء Drive ایجاد نمایید. می توانید این کاررا با استفاده از متد ( ) GetDriveاز شیءFileSystemObject انجام دهید. مثال زیر نام مستعار :C را باز می گرداند:

```
<%
' Create an instance of the FileSystemObject object
Set MyFileObject = Server.CreateObject ("Scripting .FileSystemObject ")
 'Create an instance of the Drive object
Set MyDrive=MyFileObject.GetDrive("c:")
Response.Write(MyDrive.VolamuName)
%>
```

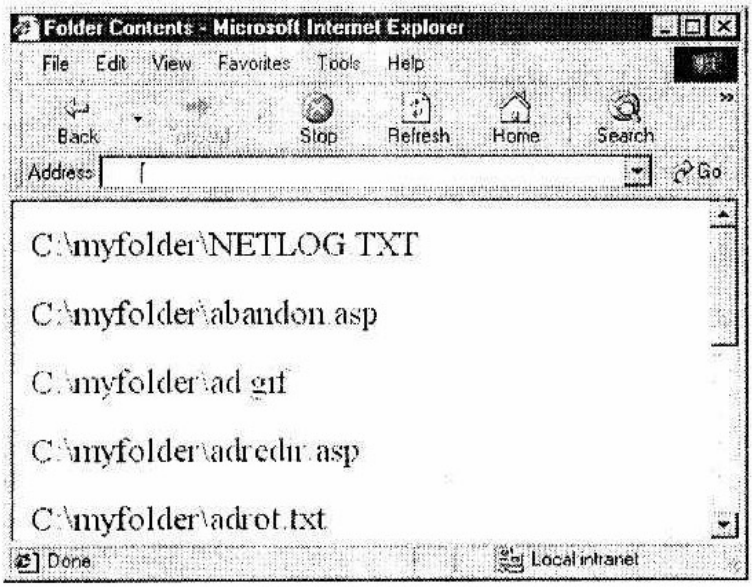
۸–۵ کار با پوشه ها:

این بخش طریقهٔ دستکاری پوشه ها و نمایش محتوای آنها را نشان می دهد. برای کار با پوشه می توانید ازدوشیءFileSystemObject وFolder استفاده نمائید. در این مثال تمام فایلهای پوشه ای با مسیر c:\ MyFolder نمایش داده شده است:

```
<HTML>
<HEAD> <TLTLE> Folder Contents </ TLTLE> </ HEAD>
<BODY>
< %
' Creat an instance of the FileSystemObject object
Set MyFileObject=Server.CreatObject ("Scripting.FileSystemObject")
' Create a folder object
Set MyFolder=MyFileObject.GetFolder("c:/ mydir")
' loop through the Files collection
FOR EACH thing in MyFolder.Files
Responde.Write("<P>"&thing)
NEXT
%>
</ BODY>
</ HTML>
```

در مثال قبل یک شیء Folder با متد ( )GetFolderاز شیءFileSyste-mObject به محض این که شیء Folderایجاد شد استفاده می گردد.

شیء FileSystemObject ایجاد شده است. به محض ایجاد شیء Folder از حلقهٔ FOR...NEXT بـرای قـرار دادن مجموعه فایلهای آن درون حلقه استفاده می شود.



شکل ۸-٤ خصوصیات فایل

شیء FileSystemObject شامل تعدادی از متدهای کار با پوشه ها است لیسـت زیـر توضیحـات مختصـری در مورد چگونگی استفاده از این روشها آورده است:

[جانویسی] مبدأ ,مقصدCopyFolder : این متد پوشه را از جائی به جای دیگر کپی می کند. می توانید از وایلدکارد در پارامتر مبدأ برای کپی کردن هم زمان چندین پوشه استفاده کنیـد. بـه طـور پیـش فـرض درصـورت وجـود پوشـهٔ مقصد آن پوشه جانویسی می شود مگر اینکه با قرار دادن مقدار False برای پارامتر جانویسی از آن جلوگیری کنیم.

مشخصهٔ پوشه CreateFolder : پوشه مشخص شده را ایجاد می کند.

مشخصهٔ پوشه DeleteFolde: پوشهٔ مشخص شده و محتویاتش را حذف مـی کنددراینجـا نـیز از وایلدکاردهـا مـی توان برای حذف کردن هم زمان چندین پوشه استفاده کرد.

مشخصهٔ پوشه GetFolder: یک شیء Folder که نشانگر پوشه مشخص شده است را بر می گرداند.

مشخصهٔ پوشه MyFolderExists: اگر پوشه موجود باشد TRUE و در غیر این صورتFALSE را بر می گرداند.

(مسیر) GetParentFolderName: محتویات یک رشته از مسیر پوشه پدر[1] را بر می گرداند.

مبدأ, مقصد MoveFolder:یک پوشه را از یک محل به محل دیگر انتقال می دهد می تـوان با اسـتفاده وایلدکـارد بیش از چندین پوشه را بطور هم زمان انتقال داد.

برای استفاده از هر کدام از این روشها ابتدا باید یک نمونه شیء FileSystemObject ایجـاد کـرد مثـال بعـدی یک پوشه را ایجاد می کند، آن را انتقال می دهد و بعد آن را حذف می نماید:

```
<%
'Create an instance of the FileSystemObject object
```

```
Set MyFileObject = Server.CreatObject (" Scripting. FileSystemObject ")
' Create a new folder
MyFileObject.CreatFolder "c:\snew folder"
' Move the Folder
MyFileObject.Move the Folder "c:\new Folder ","c:\oldfolder"
Delete the folder
MyFileObject.Delete Folder "c:\old folder"
% >
```

روشها و خصوصیات شیء Folder برا ی دستکاری پوشه ها نیز می تواند مورد استفاده قرار گیرد. در این قسمت توضیح مختصری در مورد خصوصیات و متدهای شیء Folder وجود دارد.

[جانویسی] ،کپی جدید CopyFolder :پوشه جاری را از یک محل به محل دیگر کپی می کند.اگر مقدار پارامتر جانویسی FALSE باشد و آن پوشه یا پوشه مشابه نام آن موجودباشد ، پیغام خطا خواهید داشت.

DeleteFolder: پوشه جاری و محتویات آن را حذف می نماید.

Files: مجموعهٔ Filesای که در پوشه وجود دارد را باز می گرداند. در این حالت فایلهای مخفی نمایش داده نخواهد شد.

IsRootFolder : اگر پوشه، پوشه اصلی (ریشه) باشد مقدار TRUE را باز می گرداند.

مشخصهٔ پوشه MoveFolder: پوشه را از یک محل به محل دیگر انتقال می دهد.

Name: نام پوشه را باز می گرداند.

ParentFolder:پوشه پدر را باز می گرداند.

Size: اندازه پوشه و تمام زیر پوشه ها را به بایت باز می گرداند.

SubFolder: اندازه زیر پوشه های پوشهٔ جاری را باز می گرداند.

برای استفاده از هر یک از این متدها مثل همیشه لازم است که ابتدا یک نمونه شیء Folder را ایجاد کنید. مثال زیر لیستی از تمام زیر پوشه های پوشه ای با مسیر c:\ myfolder را مشخص می نماید:

```
<%
' Create an instance of the FileSystemObject object
Set MyFileObject = Server.CreatObject (" Scripting.FileSystemObject ")
' Create an instance of the Folder Object
Set MyFolder=MyFileObject.GetFolder "c:\myfolder"
FOR  EACH  thing IN Folder.SubFolders
  Response.Write(thing)
NEXT
% >
```

فصل نهم :

فراهم نمودن هدایتگر Site Wide

این فصل شامل چگونگی بکارگیری دو جزء از اجزاء اکتیوایکسی است که در Asp ها گنجانده شده اند. در بخش اول چگونگی استفاده از جزء محتوی پیوند[1] را فرا می گیرید. این جزء می تواند برای آسانتر کردن هدایت سایت وب شما مورد استفاده قرار گیرد. همچنین شما طرز استفاده از جزء محتوی پیوند را برای ایجاد یک گروه خبری ساده[2] فرا می گیرید. در انتهای فصل شما چگونگی استفاده از جزء بررسی کنندهٔ اجازه[3] را فرا می گیرید. این جزء تنها هنگامی پیوند به یک صفحه را نمایش می دهد که کاربر اجازهٔ دسترسی به آن صفحه را داشته باشد.

۹-۱ جزء پیوند محتوی

جزء محتوی پیوند درحالتی که لازم باشد یک سری صفحه را به هم پیوند کنید ، مفید واقع می شود. مثلاً شما می توانید از این جزء برای پیوند دادن صفحات یک کتاب Online نمایش اسلایدی و یا حتی پیغامهای یک گروه خبری ، استفاده کنید. مثالی از چگونگی استفاده از این جزء برای ایجاد یک گروه خبری ساده را در همین فصل خواهید دید.

در حالت عادی برای پیوند دادن یک سری صفحه شما مجبورید که یک پیوند فرامتن را به هر صفحه اضافه کنید. جزء محتوی پیونداین فرایند را برای شما ساده می کند . شما می توانید با استفاده از این جزء لیستی از صفحات را در یک فایل ایجاد کنید و به محض اینکه این فایل ایجاد شد می توانید از متدهای این جزء برای اختصاص پیوند ها به هرصفحه استفاده کنید .جزء محتوی پیونددارای متدهای زیر می باشد:

(فایل شامل محتوی لیست پیوند [4])GetListCount : این متد تعداد تمام صفحات گنجانده شده را در فایل محتوی لیست پیوند باز می گرداند .

(فایل شامل محتوی لیست پیوند) GetListCount : این متد موقعیت صفحه جاری در فایل محتوی لیست پیوند را باز می گرداند .

(فایل محتوی لیست پیوند) GetNextDescription :این متد توضیحات مربوط به صفحه ای با اندیس معین در فایل محتوی لیست پیوند را باز می گرداند .

(فایل محتوی لیست پیوند) GetNextURL : این متد مسیر صفحه ای با اندیس معین در فایل محتوی لیست پیوند را بازمی گرداند.

(فایل محتوی لیست پیوند) GetNthDescription : این متد توضیحات مربوط به صفحه ای با اندیس معین در فایل محتوی لیست پیوند را باز می گرداند .

(فایل محتوی لیست پیوند) GetNthURL : این متد مسیر صفحه ای با اندیس معین در فایل محتوی لیست پیوند را باز می گرداند.

(فایل محتوی لیست پیوند) GetPreviousDescription : این متد تشریح صفحهٔ قبلی در فایل محتوی لیست پیوند را باز می گرداند.

(فایل محتوی لیست پیوند) GetPreviousURL: این متد مسیر صفحه قبلی در فایل محتوی لیست پیوند را باز می گرداند.

برای مثال فرض کنید که می خواهید پختن شیرینی را(که در مثال زیر شیرینی پاستا می باشد) روی سایت وب به صورت قدم به قدم آموزش دهید. شما می خواهید برای هر مرحله یک Asp متمایز اختصاص دهید و مراحل را به ترتیب نمایش دهید. برای این کار ابتدا احتیاج دارید که یک فایل ویژه به نام فایل محتوی لیست پیوند ایجاد کنید . این فایل ، یک فایل متنی معمولی است که می توانید با هر ویرایشگر متنی ایجاد نمائید وبه این ترتیب این فایل به سادگی حاوی لیستی از صفحاتی که قرار است به هم پیوند داده شوند ، خواهد بود. در زیر مثالی از آنرا داریم:

/pasta/ grabpot.asp	Grab a pot form the cupboard.
/pasta/ boilwater.asp	Boil some water in the pot.
/pasta/ openbox.asp	Open box of pasta.
/pasta/ dumpcontents.asp	Dump Content of box in pot.
/pasta/ wait.asp	Wait ten minutes.
/pasta/ home.asp	Return to home page.

وقتی که فایل را ایجاد کردید می توانید آنرا به نام دلخواه خود، مثلاً Pasta.txt ذخیره کنید. این فایل نمونه شامل دو ستون است ؛ ستون اول شامل لیستی از فایلهای مورد پیوند است ، که می توانند فایلهای HTML معمولی ویا فایلهای ASP باشند ؛ ستون دوم شامل توضیحات این فایلها است . این دو ستون باید به جای گذاشتن فاصله توسط یک تک کاراکتر Tab کاملاً از یکدیگر جدا شوند. در غیر اینصورت جزء محتوی پیوند قادر به تشخیص دو ستون از هم، نخواهد بود.

پس از اینکه فایل محتوی لیست پیوند را ایجاد کردید می توانید از جزء محتوی پیوندبرای اضافه کردن لینکهای هدایت به ASP خودتان استفاده نمائید . برای نمونه زمانیکه شما بخواهید یک لیست ازتمام مراحل لازم در آماده نمودن شیرینی را روی صفحه خانگی درسایت وب خود بیاورید؛می توانید این کار رابا ASP زیرانجام دهید:

```
<HTML>
<HEAD> <TITLE> Home Page </TITLE> </HEAD>
<BODY>
<H2> Welcome  To The Pasta web site!</H2 >
<%
Set  mylinks=Server.CreateObject ("MSWC.NextLink")
%>
Here are the
<%= mylinks.GetListCount ("pasta.txt ") -1 %>
steps for perparing  pasta:
<OL>
%>
FOR i=1 TO myLinks.GetListCount("pasta.txt")  - 1
%>
<LI> <A  HREF =" < %=mylinks.GetNthURL (" pasta.txt", i) % >">
 <%=mylinks.GetNthDescription ("pasta.txt ") -1 %>
%>
NEXT
%>
</OL>
</BODY>
</HTML>
```

این ASP لیستی از پیوندهای درون فایل محتوی لیست پیوند را نشان می دهد (رجوع شود به شکل ۹-۱). این کار توسط ایجاد نمونه ای از جزء محتوی پیوندانجام می شود . نمونه ای از این جزء به متغییری به نام mylinks اختصاص داده می شود .

شکل ۹-۱سایت وب Pasta

سه متد از جزء گفته شده در این مثال بکار رفته که به قرار زیرند :

الف – متد ( )GetListCount: شمارهٔ تعداد ورودیها در یک فایل محتوی لیست پیوند را دریافت می کند. هر موقع که متد جزء محتوی پیوند را صدا بزنید باید نام فایل محتوی لیست پیوند را با آن بیاورید. در این مثال متد توسط mylinks.GetListCount (" pasta. txt") صدا زده می شود .

نکته : آخرین حلقه FOR از ۱ تا 1-( )GetListCount می شمارد، منهای یک برای صرف نظر کردن از item آخرمی باشد . item آخر در فایل لیست برای باز گشت به صفحه خانگی می باشد .

ب – متد ( ) GetNthURL ورودی URL آخردر فایل محتوی لیست پیوند را دریافت می کند. این متد دو پارامتر دارد . پارامتر اول لیست حاوی نام فایل محتوی لیست پیوند می باشد . پارامتر دوم به ورودی جاری دریافت شده از این فایل اشاره دارد. برای مثال اگر شما (2," Pasta.Txt") GetNthURL را صدا بزنید URL ای که به عنوان دومین ورودی در فایل Pasta.Txt لیست شده برگردانده می شود.

ج – متد ( ) GetNth Descrition برای دریافت شرح صفحه از فایل محتوی لیست پیوند صدا زده می شود. این متد هم دو پارامتر دارد. اولین پارامتر نام فایل محتوی لیست پیوند را تعیین می کند، و پارامتر دوم نشانگر ورودی دریافت شده از این فایل است. برای مثال اگر (2 ،" Pasta.Txt") GetNthDescrition را صدا بزنید شرح دومین ورودی فایل Pasta.Txt برگردانده می شود. متدهای GetNthURL و ( )GetNth Descrition وقتی در یک حلقهٔ FOR.... NEXT قرار می گیرند، تمام ورودیهای فایل محتوی لیست پیوند به استثناء ورودی آخر را نمایش می دهند. آخرین ورودی برای این مستثناً است که باید به صفحهٔ خانگی (Hame Page) اشاره کنددر مثالی که گذشت ، متدهای جزء محتوی پیوند برای لیست کردن یک سری صفحه بکار رفته بود. شما می توانید از متدهای این جزء برای پیوند صفحات تکی نیز استفاده کنید. همانند مثال زیر:

```
<HTML>
<HEAD><TITLE> Setp One </TITLE></HEAD>
<BODY>
<H1> Step 2: Boil Water </H1>
<H3> Boil Some Water in a Pot. </H3>
<HR>
```

```
<%
Set My Links = Server.Creat Object ( “MSWC. Next Link “)
IF My Links.Get List Index (“ Pasta.Txt “)>1 THEN
%>
<A HREF = “ <% MyLinks.Get Next URL ( “pasta.Txt “)% “>
Next Step </A>
</BODY>
</ HTML>
```

دو متد جزء محتوی پیوند در Asp بالا استفاده می شود.اول متد ( )Get PreviousURL که مسیر صفحهٔ قبلی را باز می گرداند وسپس متد ( ) GetNext URL که مسیر صفحهٔ بعدی را بـاز مـی گردانـد. ایـن متدهـا پیوندهـایی بـه صفحهٔ بعد و قبل ایجاد می کنند.

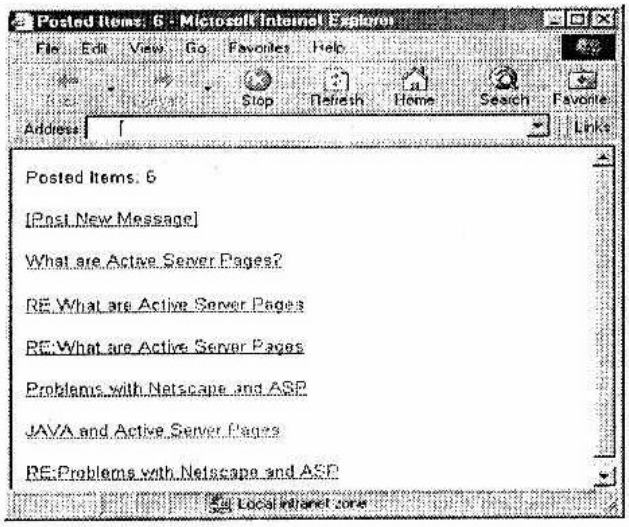
این دو متد بسته به صفحهٔ جاری مقادیر مختلفی را بر می گردانند. هنگامی کـه آنـها صـدا زده مـی شـوند، مسـیر صفحهٔ جاری با ورودیهای فایل محتوی لیست پیوند مقایسه می شود و ( )GetPreviousURL اطلاعات خط بـالائی ورودی صفحهٔ جاری را بر می گرداند و ( ) GetNext URL نیز اطلاعات بعدی آنرا برمی گرداند.

اگر صفحهٔ جاری در فایل محتوی لیست پیوند موجود نباشد متد ( ) GetPreviousURL اولین ورودی فـایل و متـد ( ) GetNext URL آخرین ورودی را بر می گرداند.

۹-۲ برنامهٔ کار بردی مربوط به جزء محتوی پیوند

در این بخش چگونگی استفاده از جزء محتوی پیوند بـرای ایجـاد یـک گـروه خـبری سـاده را فـرا مـی گـیرید. کاربرهای گروه خبری می توانند پیغامهای جدید بفرستند و پیامهای ارسال شده توسط کاربرهای دیگر را بخوانند.

جزء محتوی پیوند استفاده شده در مثال زیر پیغامهای ارسالی در گروه خبری را مرتب می کند. این جزء کاربران را قادر می سازد که لیستی از تمام موارد ارسالی موجود در گروه خبری را ملاحظه و بتوانند بـه راحتـی در میـان آنـها حرکت کند. (رجوع شود به شکل ۹-۲)



شکل ۹-۲ یک گروه خبری ساده

برای ایجاد گروه خبری لازم است که فایلهای زیر را ایجاد کنید.

صفحهٔ post : این صفحه برای ارسال پیغام جدید مورد استفاده قرار می گیرد .

صفحهٔ Include :این فایل در پایان (پائین) هر پیغامی که فرستاده شده گنجانده می شود .

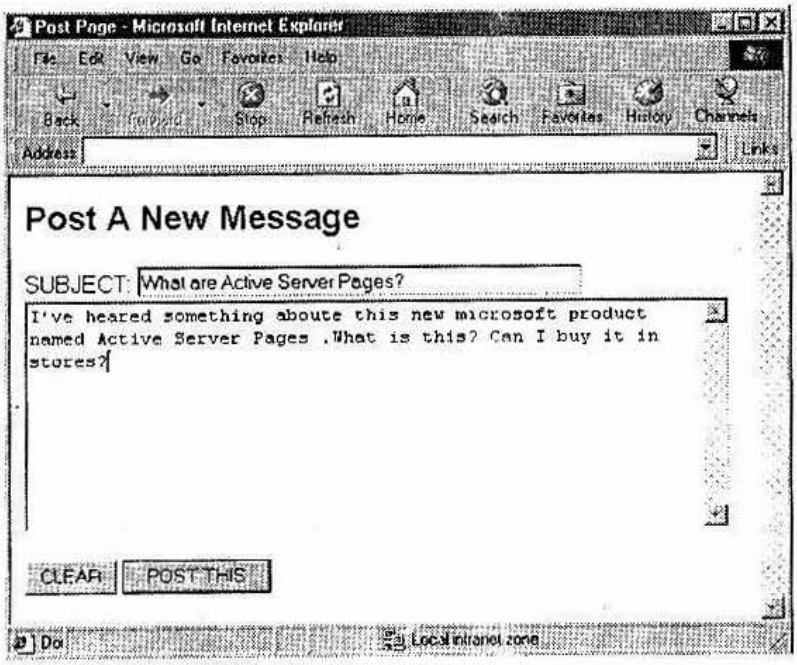
صفحهٔ New Item :از این صفحه برای ایجاد پویایی یک Asp که حاوی مرسولات جدید باشد استفاده می شود.

صفحهٔ New Page :این صفحه شامل لیستی از تمامی پیغامهای فرستاده شده به گروه خبری است .

۹-۲-۱ صفحه POST (post.htm) :

صفحهٔ Post همان صفحه ای است که کاربر می تواند پیغام خود را ارسال کند. (رجوع شود به شکل ۹-۳) این صفحه همانطوری که در برنامهٔ زیر مشاهده می کنید یک صفحهٔ HTML ساده است که حاوی یک فرم HTML می باشد. همچنین این فرم یک خط موضوع برای وارد کردن موضوع چیزی که کاربر ارسال می کند به همراه دارد. ویژگی WRAP=VIRTUAL اجازهٔ پنهان سازی کلمه را در ناحیهٔ متنی می دهد.

```
<HTML>
<HEAD><TITLE>Post Page</TITLE><HEAD>
<BODY>
<H2>Post A New Message</H2>
<FORM METHOD="POST" ACTION="newitem.asp">
SUBJECT: <INPUT NAME="subject" TYPE="TEXT" SIZE="50" MAXLENGHT=50>
 <BR>
<TEXTAREA NAME="posting" COLS=60 ROWS=10 WRAP="VIRTUAL" ></TEXTAREA>
<P>
<INPUT TYPE="RESET" VALUE="CLEAR">
<INPUT TYPE="SUBMIT" VALUE="POST THIS">
</FORM>
</BODY>
</HTML>
```

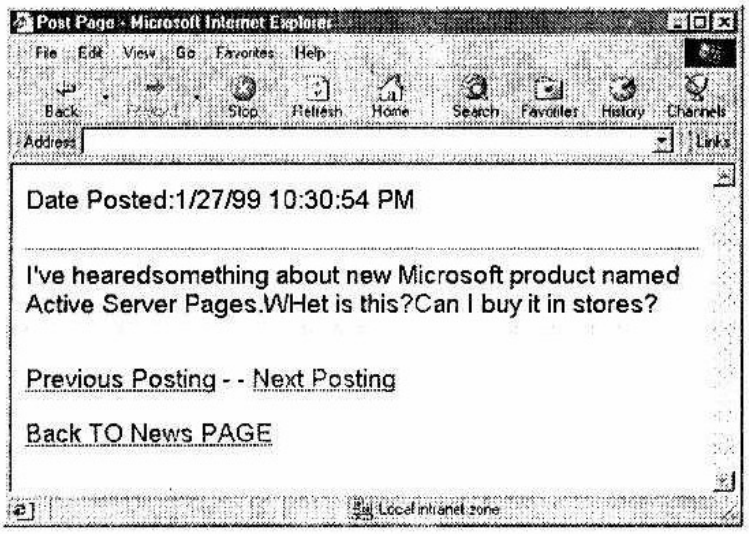


شکل ۹-۳ صفحهٔ Post

۹-۲-۲فایل InClude (Include.htm) :

همانگونه که در برنامهٔزیر آمده است ، هر پیغام در گروه خبری حاوی یک پیوند به پیامهای بعدی و قبلی خواهد بود . هم چنین شامل یک پیوند به صفحه News(رجوع شود به شکل ۹-٤)است.این پیوندها به پیغام درون فایل Include اضافه می گردند. این فایل در هر ASP ای که بوسیله صفحه NewItem تولید شده ، به طور خودکار وجود دارد. در بخش بعدی چگونگی انجام این کار را فرا می گیرد.

```
<HR>
<%
Set MyLinks=Server.CreateObject(MSWC.NextLink")
IF mylinks.GetList Index("news.txt")>1 THEN
%>
<A HREF"<%mylinks.GetPreviousURL("news.txt")%>">
Previous Posting</A> - -
<% END IF %>
<A HREF"<%=mylinks.GetNextURL("news.txt")%>">
Next Posting </A>
<P><A HREF="news.asp">Back TO News PAGE</A>
```



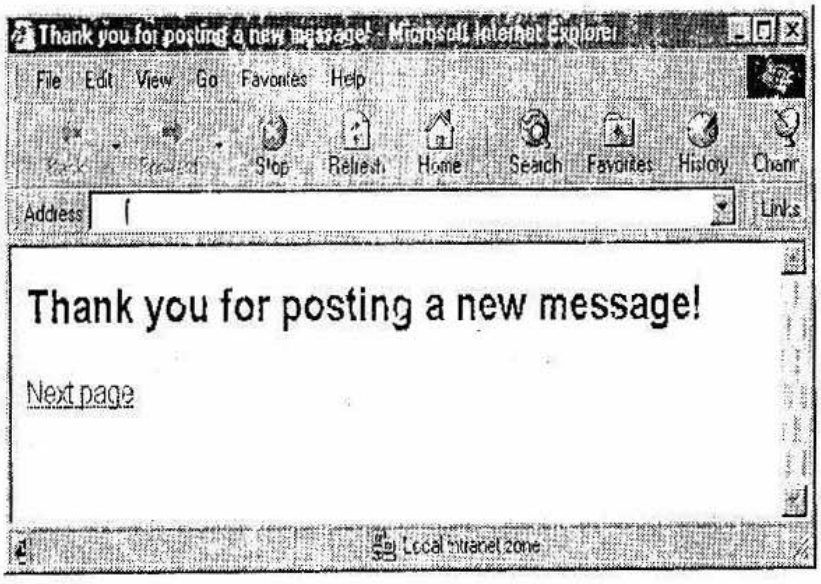
شکل ۹-٤ پیغامی در گروه خبری

۹-۲-۳ صفحه New Item :

پس از اینکه کار بر یک پیام را ارسال می کند او به صفحه New Item آورده می شود . با اینکه در پشت پرده صفحه کاملاً بزرگتر و متعددتر می گرددولی در ظاهر خالی به نظر می رسد (رجوع شود به شکل ۹-۵). صفحه New Item برای تولید پویای ASP های جدید مورد استفاده قرار می گیرد . زمانیکه یک کاربر پیغام جدید می فرستد پیغام در یک فایل متنی ذخیره می گردد . این فایل متنی یک فایل معمولی نیست ، بلکه یک ASP است . صفحه New Item ، ASP را از اطلاعاتی که کاربر وارد فرم HTML نموده است ایجاد می نماید . در ضمن صفحه New Item فایل محتوی لیست پیوند را به هنگام می نماید . این مسئله جزء محتوی پیوند را قادر می سازد که نسبت به چیزی که جدیداً ارسال شده به طور خودکار ودقیق عکس العمل نشان دهد.

```
<%
' Createthe Posting
Thesubj= Server.HTMLEncode ( Request.Form (" subject"))
```

```
IF TheSubj = " "  THEN  TheSubj = " NO Subject"
The Post  = " < HTML > < HEAD > < TITLE > " &TheSubj & " < / TITLE>
<HEAD > </BODY>

The Post = The Post & " Date Posted : " & NOW & " < HR>"
The Post = The Post & Server.HTMLEncode ( Request.From (" posting"))
The Post = The Post & "< ! -- # INCLUDE VIRTUAL = " " news.inc" "-- >"
The Post = The Post & " < / BODY > < / HTML>"
' Createunique file name
Set  mylinks=Server.CreateObject ("MSWC.Next Link")
TheName =   item " & mylinks.GetListCount(" news.txt " ) + 1 &".asp"
TheNamePath = Server.MapPath (The Name)
' Save the new posting file
Set  MyFileObj=Server.CreateObject ("Scripting.FileSystemObject")
Set  MyOutStream=MyFileObj.CreateTextFile( TheNamePath)
MyOutStream.Write ThePost
MyOutStream.Close
' Update the Content Linking List File
TheNews = Server.MapPath ("news.txt")
Set  MyNews=MyFileObj.OpenTextFile( TheNews , 8, TRUE)
MyNews.Write line TheName & vbTab & TheSubj
MyNews.Close
%>
< HTML >

< HEAD > < TITLE > New Item < / TITLE > < / HEAD >

< BODY >

< H1 > Thank you for posting a new message ! < / H1 >

< A HREF = " news.asp " > News Page < /A >

</ BODY >

</ HTML >
```



شکل ۹-۵ صفحه New Item

در قسمت اول صفحه متغیری به نام ThePost ایجاد گردیده است . محتوای این متغیر در واقع یک ASP است. ایجاد این ASP مانند ایجاد یک رشته طولانی است. ASP ایجاد شده حاوی متنی است که کاربردر فرم HTML وارد کرده است و همچنین ساعت و تاریخ جاری می باشد. سر انجام دستور INCLUDE برای داخل کردن خودکار فایل news.inc (که به موقع خود درباره اش بحث می کنیم) اضافه می شود. قسمت بعدی اسکریپت یک ASP جدید با نام منحصر به فرد آماده می کند. این کار به این دلیل است که شاید شما نخواهید پیغام ارسال شده توسط یک کاربر روی فایل حاوی پیغام ارسال شدهٔ کاربر قبلی نوشته شود (در اصل جانویسی شود). به همین دلیل قبل از اینکه Asp جدید ذخیره گردد ، باید یک نام منحصر به فرد تولید شود. برای تولیدیک نام منحصر به فرد از جزء محتوی پیوند استفاده می شود. هر ارسال جدید باید در فایل محتوی لیست پیوند جای بگیرد با دریافت شمارهٔ تعداد ورودیهای این فایل به علاوهٔ یک ، نامی منحصر به فرد برای Asp جدید تولید می کند. مثلاً اولین پیام ارسال شده نام item1.asp ، پیام دوم item2.asp خواهند بود ؛ به این ترتیب هر پیام ارسالی متناظر با یک ASP منحصر به است. قسمت سوم اسکریپت صفحه New Item ، ASP جدید را ذخیره می کند . مقدار ThePost در یک فایل متنی ذخیره می شود . این فایل ذخیره شده با نام منحصر به فردی که داخل متغیر TheNamePath قرار دارد ، ذخیره می شود. FileSystemObject برای ذخیره فایل روی دیسک استفاده می شود. قسمت نهایی اسکریپت، فایل محتوی لیست پیوند را به روز در می آورد . مسیر و نام فایل ASPجدید به فایل محتوی لیست پیوند ضمیمه شده است . اگر فایل محتوی لیست پیوند موجود نباشد این اسکریپت،اسکریپت آنرا به طور خودکار ایجاد می کند. لیست پیوند محتوایی که اسکریپت، آنرا ایجاد کرده است News.txt نام دارد.در این فایل هنگامی که اولین پیام برای گروه خبری ارسال می شود ، لیست محتویات قرار می گیرد.

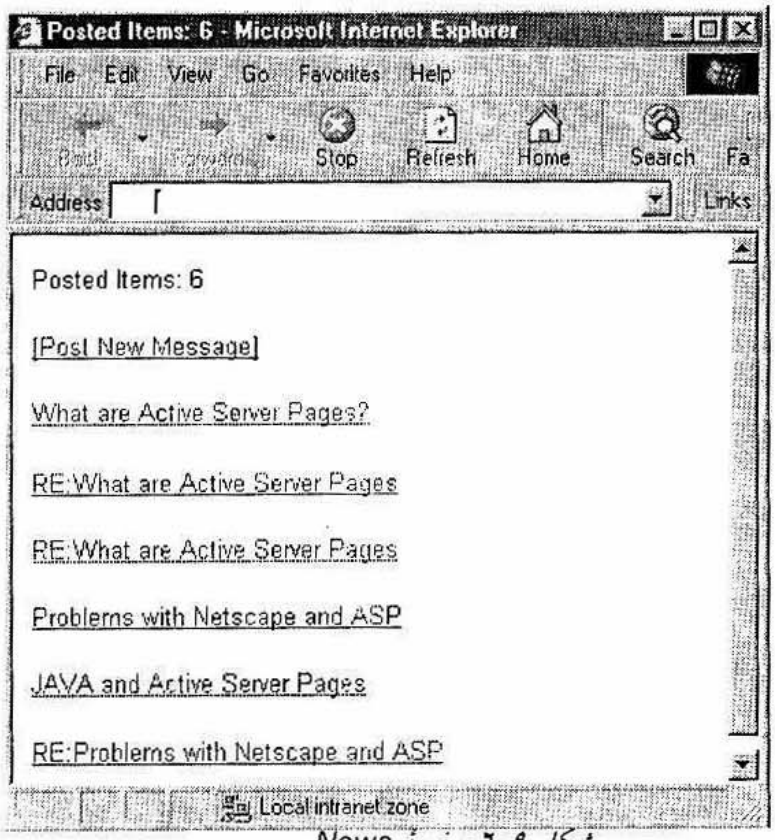
۹-۲-٤ صفحه News :

آخرین صفحه ای که گروه خبری لازم است ایجاد کند صفحهٔ News است . صفحهٔ News تعداد پیغامها درگروه خبری را نشان می دهد و تمام پیامها را بر اساس عنوانشان لیست می کند . برنامهٔ زیراسکریپت صفحه News را نشان می دهد . در این صفحه برای دریافت تعداد ورودیها در فایل محتوی لیست پیوند از جزء محتوی پیوند استفاده گردیده است .

تعداد ورودیها به تعداد پیامهای ارسال شده بستگی دارد. در مرحلهٔ دوم تمام ورودیهای فایل لیست پیوند محتوی نمایش داده می شوند. اگر کاربری روی یکی از ورودیها کلیک کند به صفحهٔ مرتبط به آن ورودی وصل خواهد شد.

```
<%' Create The Content Linking Componet
Set  mylinks = Server.CreateObject (" MSWC.NextLink")
%>
< HTML >
< HEAD > < TITLE > News Page < /TITLE > < / HEAD >
< BODY >
PostedItems : < % = mylinks.GetListCount (" news.txt")
< HR >
[ < A HREF = " Post.htm " > Post New Message < /A > ]
```

```
< UL >
%>
' Display the list of messages
FOR  i = 1 TO mylinks.GetListCount (" news.txt")
%>
< LI >
< A  HREF = " < % = mylinks.GetNthURL (" news.txt " , i )%>">
<%=mylinks.GetNthDescription (" news.txt" , i ) % > < /A >
<%NEXT %>
</ UL >
</ BODY >
</ HTML >
```



شکل ۹-۶ صفحهٔ News

۹-۲-۵ بسط مثال فوق :

یک گروه خبری ساده همانند مثال قبل می تواند فقط تعداد کمی مراجعات و کار بر را پشتیبانی کند. اگر چندین کاربر به ارسال پیامهای جدید در یک زمان مبادرت ورزند مشکلات متعددی رخ خواهد داد. جزء محتوی پیوند برای این منظور طراحی نگردیده است.بهترین راه برای ایجاد یک گروه خبری ذخیره کردن پیغامها در یک پایگاه داده همانند SQL Server مایکروسافت است. برای انجام این کار شما می توانید از ADO استفاده نمائید .

بر خلاف جزء پیوند محتوی، پایگاه داده ها برای ذخیره وباز یابی مؤثر اطلاعاتی با حجم بالا طراحی شده اند. هرچند این برنامهٔ کاربردی نمونه تشریح کرده که چگونه جزء محتوی پیوندمی تواند برای پیوند دادن آسان تعداد زیادی ASP به هم استفاده گردد؛ همه اقلام وارده در گروه خبری بطور خودکار و متوالی به هم پیوند می شوند؛ با اینحال جزء محتوی پیوند انجام این کار را آسان می سازد .

۹-۳ کاربرد جزء Permission Checker :

از این جزء می توان برای نشان دادن پیوند یک صفحه زمانیکه به کاربر اجازهٔ دسترسی به آن صفحه را داده باشند ، استفاده نمود. این جزء تنها یک خاصیت به نام HasAccess دارد. زمانیکه یک کاربر دستیابی به یک فایل داشته باشد خاصیت مربوطه مقدار TRUE را بر می گرداند. اگر کاربر دستیابی به فایل نداشته باشد یا اینکه فایل موجود نباشد خاصیت مربوطه FALSE را بر می گرداند.

مثال زیر چگونگی استفاده از این جزء را نشان می دهد :

```
<%
Set   Permit = Server.CreateObject (" MSWC.PermissionChecker")
%>
< HTML >
< HEAD > < TITLE > AdministrationPage < /TITLE > < / HEAD >
< BODY >
<%
IF  Permit.HasAccess ("DestroyAll.asp " ) THEN
%>
<A HREF = " DestroyAll.asp">
' Click here to delete all files on The hard  drive.
</ A >
<%ELSE %>
. You can not delete all the files on the hard drive '
< ٪ END IF ٪ >
</ BODY >
</ HTML >
```

نکته : جزءPermission Checker رسماً توسط مایکروسافت تضمین و حمایت نمی گردد . اگر چه در نسخه جاری IIS گنجانده شده است. همچنین شما می توانید نسخهٔ جدید این جزء را توسط آدرس http://www.microsoft. com/iis بدست آورید.

در این مثال پیوند مربوط به فایلها در روی دیسک سخت جهت حذف کردن، تنها به کاربرانی نشان داده می شود که اجازه دستیابی به آن فایلها را داشته باشند. کاربرانی که حق دستیابی به این فایلها را ندارند حتی پیوند آنها را هم نمی توانند ببینند. برای تعیین مجوزها ی یک فایل خاص، روی نام آن کلیک راست کنید. سپس Propertiesرا انتخاب کنید روی دکمه Security و سپس روی دکمه Permission کلیک کنید . در جعبه محاوره File Permissions ای که ظاهر می شود شما می توانید گروهها ویا کاربرانی که اجازه دستیابی به فایل را دارند مشخص کنید.

جزء Permission Chekerبرای تنظیمات مجوز جهت تعیین کاربرانی که دستیابی به فایل دارند ، استفاده می شود. بنابراین جزء تنها هنگامی این کار را می کند که شناسهٔ کاربر را بداند. اگر کار بران سایت وب شما مجبور نباشند که log in کنند این جز نمی تواند مفید واقع گردد . در این قسمت دو روش برای مجبور نمودن کاربر به log in کردن در سایت وب شما وجود دارد . اولین متد استفاده از بخش Internet Service Manager برای انجام تعیین اعتبار مبتدی یا Windows NT Challenge/Response یا Basic است. هنگامی که یکی از دو روش بالا

فعال باشد شما می توانید از طریق یک کاربر ناشناس[1] Log in کنید. بطور پیش فرض تمام کاربران ناشناس سایت وب ما از یک شماره استفاده می کنند. از نقطه نظر WindowsNT هر بازدید کننده سایت وب شما از شمارهٔ IUSR-Machine استفاده می کند . برای مثال اگر نام ماشین شما Plato است تمام بازدیدکنندگان بی نام شبکه از IUSR-Plato استفاده می نمایند. حال کاربرانی که با این شماره وارد سایت شما می شود ممکن است از وصل شدن به بعضی صفحات محروم شوند.

مثال زیرنیز مانند مثال قبلی است بجز خطوط اول اسکریپت که متفاوت است. متد Status از شیء Response برای ارسال یک کد وضعیتUnauthorized به مرورگر ، استفاده می شود. سپس به طور خودکار جعبه محاورهٔ کلمه عبور ظاهر می گردد و کاربر را مجبور به Login می کند. پس از اینکه کاربر login کرد جزء Permission Checker می تواند برای تعیین فایلهایی که کاربراجازه دستیابی به آنها را دارد استفاده شود:

به کاربران مختلف بسته به نقش آنها اجازه های مختلفی داده می شود.

```
<%LOGON = Request.ServerVariables ( " LONGON – USER")
IF  LOGON = " " OR ISNULL ( LOGON ) OR ISEMPTY ( LOGON)  Then
Re-sponse.Status = " 401 Unauthorized"
Response.End
End  If
Set  Permit = Server.CreateObject (" MSWC.Permission Checker") %>
< HTML >
< HEAD > < TITLE > Administration  Page < / TITLE > < / HEAD >
< BODY >
<%
IF  Permit.HasAccess (" DestroyAll.asp")  THEN
%>
<A HREF = "DestroyAll.asp">
' Click here to delete all files on the hard drive .
</ A >
<%ELSE %>
' You can not delete all the files on the hard drive .
< ٪ END  IF ٪ >
</ BODY >
</ HTML >
```

جزء Permission Checker می تواند برای نمایش گزینه های مناسب با نقش خاص ، مورد استفاده قرار گیرد. به عنوان مثال ممکن است شما چندین مدیر با اجازه های مختلف درسایت وب خود داشته باشید. ممکن است که قصد داشته باشید به مدیران خاصی فقط اجازه حذف پیام گروههای خبری را بدهید یا اینکه بخواهید به گروه دوم از مدیران اجازه انجام بعضی کارها مانند حذف تمام فایلهای روی درایو دیسک سخت را بدهید. بوسیله استفاده از جزء Permission Checker، می توانید مانع انجام کارهای اضافی و غیر مجاز توسط افراد مختلف بشوید.

فصل دهم :
کار با آگهی ها

این فصل شماری از اجزاء اکتیوایکس اضافی را بررسی می کند. در بخش اول فصل چگونگی استفاده از جزء Adrotator را فرا می گیرید. این جزء می تواند برای نمایش اعلانهای آگهی[1] در سایت وب شما مورد استفاده قرار گیرد. در بخش سوم نیز جزء محتوی چرخش[2] مورد بررسی قرار گرفته است. این جزء در اصل بطور اتفاقی محتویات فایلهای HTML مختلف را در صفحهٔ وب نمایش می دهد. و در آخرین بخش شما طریقهٔ استفاده از اجزاء شمارنده و شمارندهٔ صفحه را فرا می گیرید. از این دو جزء می توانید برای پیدا کردن تعداد بازدید کنندگان سایت وب خود استفاده کنید.

۱۰-۱ جزء Ad Rotator :

بسیاری از سایتهای وب تجاری بر این فرض بنا شده اند که با آگهی های تبلیغاتی کسب درآمد کنند. فرض و تصور بر مبنای این است که یک سایت وب بسیار شبیه یک مجله یا یک برنامه تلویزیونی است . مانند یک برنامهٔ تلویزیونی یک سایت وب با محتویات خود جمعیت بیننده را به سمت خود جلب می نماید . البته تاکنون این استراتژی و روش در بیشتر سایتهای وب کارایی نداشته است. مشکل از اینجا بوده است که تعداد بسیار کمی از شرکتها جهت آگهی های اینترنت پول خرج می کنند از طرفی آندسته از شرکتهایی که می خواهند آگهی بدهند متمایل به آگهی در گروهی خاص از سایتهای وب هستند. با این وجود روش اعلان آگهی ها اولین راه پول درآوردن در اینترنت است . مگر اینکه تصمیم داشته باشید که بطور مستقیم از سایت وب خود کسب درآمد کنید. در این صورت نیازمند آگهی های تبلیغاتی خواهید بود.

۱۰-۱-۱ استفاده از جزء Ad Rotator :

جهت استفاده از جزء Ad Rotator می توانید صفحه وبی را ایجاد کنید تا هر بار که دیده می شود حاوی آگهی تبلیغاتی متفاوتی باشد.می توانید به این آگهی ها درجهٔ اهمیت بدهید تا تکرار نمایش آنها بر اساس اهمیتشان باشد. همچنین می توانید تعداد دفعاتی که روی یک آگهی کلیک شده است را برای تعیین میزان click-through آن آگهی ثبت کنید.

نکته: آگهی و اعلان دهنده ها عموماً میزان تاثیر و سودمندی یک سایت وب را توسط میزان click-through تعیین می کنند . میزان click-through نشان دهندهٔ این است که یک بیننده سایت چه وقتهایی به آگهی توجه نشان

می دهد . میزان chick-through توسط تقسیم نمودن تعداد دفعات نمایش یک آگهی بر تعداد دفعات کلیک روی آن آگهی بدست می آید و هر عدد بالای ۱۰ ٪ خوب است .جزءAdRotator تنها یک متد دارد و آن متد GetAdvertisement() می باشد که برای دریافت اطلاعاتی در مورد یک آگهی بکارمی رود.

در این قسمت یک مثال از چگونگی استفاده از آن بیان شده :

```
< HTML >
< HEAD > < TITLE > Home Page < / TITLE > < / HEAD >
< BODY >
< CENTER > < H 1 > Welcome to our web site ! < / H 1 > < / CENTER >
< HR >
<%
Set MyAd = Server.CreatObject ("MSWC.AdRotator")
%>
< CENTER > < % = MyAd.GetAdvertisement (" adort.txt " ) % > < / CENTER >
</ BODY >
</ HTML >
```

این ASP یک اعلان آگهی را در زیر صفحه نشان می دهد (رجوع شود بـه شـکل ۱-۱۰) در اینجا اسکریپت نمونه ای از جزء Ad Rotator را با استفاده از متد ( ) Server.CreateObjectایجاد می کند .سپس اعلان آگهی عملاً با صدازدن متد ( ) GetAvertisementنمایش داده می شود.

شکل ۱-۱۰ مثالی از اعلان آگهی

توجه نمایید که متد ( ) GetAdvertisement یک پارامتر دارد که این پارامتر فایلی است که حـاوی اطلاعـاتی در مورد آگهی های مورد نمایش ، می باشد. درمثال قبل نام این فایل adort.txt بود.

۱۰-۱-۲ فایل زمانبندی[1] Rotator:

پیش از اینکه بتوانید از جزء Ad Rotator استفاده نمایید باید فایل ویـژه ای بـه نـام فـایل زمـانبندی Rotator را ایجاد نمایید. فایل زمانبندی Rotator حاوی اطلاعات کاملی از اعلانهای آگـهی اسـت و یـک فـایل متنـی معمولـی است که می توانید آن را با هر برنامهٔ ویرایشگرمتنی ایجاد و ویرایش کنید. فایل زمـانبندی Rotator دارای دو بخـش است، در بخش اول آن شما اطلاعات جامعی را در مورد همه اعلانهایی که می خواهید نمایش دهید ، تهیه می کنید و در بخش دوم اطلاعات متعلق به هر آگهی را مشخص می کنید. لیست زیر یک مثال از این نمونـه فـایل را نمـایش می دهد که نام آنرا adrot.txt گذاشته ایم :

```
REDIRECT / adredir.asp
WIDTH 200
HEIGHT 30
BORDER 0
*
bannerad.gif
http:// www.aspsite.com
The Active Server Page Site
80
http:// www.collegescape.com /gifs/csad.gif
http:// www.collegescape.com
Collegescape
20
```

دو بخش اطلاعات فایل توسط علامت (*) از هم جدا شده اند. اولین قسمت شامل چـهار پـارامتر اسـت کـه بـر تمام آگهی های فایل تاثیر می گذارد چگونگی بکارگیری این پارامترها را ببینید :

REDIRECT : یک فایل تعیین مسیر برای آگهی ها تعیین می کند و زمانیکه یک اعـلان آگـهی کلیـک مـی شـود کاربر به این فایل آدرس دهی می شود .

WIDTH : عرض تصویر اعلان آگهی را به پیکسل تعیین می کند . اگر شما این پارامتر را نـادیده بگـیرید مقـدار ایـن پارامتر ٤٤٠ پیکسل فرض می شود .

HEIGHT : ارتفاع تصویر اعلان آگهی به پیکسل تعیین می گردد . اگر این پارامتر را نادیده بگیرید ایـن پـارامتر ٦٠ پیکسل فرض می شود.

BORDER : اندازه حاشیه تصویر اعلان آگهی است . مقدار پیش فرض آن یک پیکسل است .

آنچنان که در فایل adrot.txt در لیست قبـل دیدیـد. پـارامتر REDIRECT بـه Asp ای بـه نـام adredir.asp اشاره می کند. پارامترهای WIDTH و HEIGHT به ترتیب پهنای تصویر اعلان آگهی را ۲۰۰ پیکسل باشد و ارتفاع آن را ۳۰ پیکسل ، تعیین می کنند . در انتها پارامتر BORDER مقدار صفر را می گیرد که در نتیجه هیـچ حاشـیه ای به آن داده نمی شود .

قسمت دوم شامل اطلاعات ویژه هر آگهی است . در این قسمت فایل زمانبندی Rotator حاوی اطلاعات دو آگهی است . اولین اعلان آگهی برای سایت Asp است. دومین آگهی برای سایت وبی به نام Collegescape است. برای هر آگهی شما چهار خط اطلاعات ایجاد می کنید. که اولین خط آن مسیر تصویر آگهی را نشان می دهد البته ممکن است تصویر مزبور در جایی روی کامپیوتر محلی یا سایتی از اینترنت قرار گرفته باشد.

خط دوم حاوی URL صفحه خانگی اعلان دهنده است . زمانیکه کاربران روی یک آگهی کلیک می کنند او به این صفحه سوق داده می شود. اگر شما یک خط فاصله[1] (-) اول خط دوم بگذاریدآگهی ها مانند یک پیوند عمل نخواهند کرد. خط سوم متن جایگزینی که هنگام عدم پشتیبانی مرورگر از صفحات گرافیکی باید نمایش داده شود، در خود دارد. این خط معادل ویژگی ALT از برچسب IMAGE در HTML است. شما می توانید هر متن غیر HTML ای را که می خواهید اینجا قرار دهید.

سرانجام چهارمین خط مشخص می کند که یک آگهی خاص چه وقتهایی باید نمایش داده شود و همچنین نشانگر وزن نسبی[2] تعیین شده برای آگهی است. درمثال قبل اولین آگهی ۸۰ درصد مواقع نمایش داده خواهد شد.

نکته: با تعیین وزنهای گوناگون برای هر اعلان می توانید آگهی ها را به نرخ های مختلف به فروش بگذارید معمولا سایت های وب این آگهی ها را قیمت مبنا (CPM) را بر اساس یک هزار بار نشر تعیین می کنند. به عبارت دیگر خریداران یک مقدار معین از پولشان را که بستگی به تعداد دفعاتی که اعلان آگهی آنها دیده می شود خرج می کنند و فرمول مورد استفاده برای تعیین هزینه نمایش یک آگهی CPM*(1000 /تعداد چاپها) است . به عنوان مثال تصور نمایید که سایت وب شما به طور متوسط ۱۰۰۰۰۰ باز دید کننده در ماه دارد و می خواهید فضای اعلان آگهی را در یک CPM ۱۰۰۰ دلاری بفروشید . اگر یک خریدار ۱۰۰ دلار بهای نشر آگهی های شما را بپردازد باید آگهی او برای ۱۰۰۰۰ بار نمایش داده شود به عبارت دیگر اعلان آگهی باید در ۱۰ درصد از زمان نمایش داده شود که این کار با استفاده از فایل زمانبندی Rotatorصورت می گیرد .

۱۰-۳-۱ فایل تعیین مسیر :

شما نیز می توانید یک فایل تعیین مسیر را مشخص کنید و برای تمام آگهی ها در فایل ویژه زمانبندی Rotator بکار ببرید. زمانیکه کاربران روی یک آگهی کلیک کنند، به این فایل سوق داده می شوند. این فایل می تواند یک فایل ASP باشد . تابع اصلی این فایل تعداد دفعاتی را که یک آگهی ویژه کلیک می شود ثبت می کند. به محض این که اطلاعات ثبت گردید ،کاربر به طور نمونه به صفحهٔ خانگی آگهی سوق داده می شود. لیست زیر یک مثال از این مورد را بیان می دارد :

```
<%
Response.AppendTolog Rrequest.QueryString("url")
Response.Redirect Requst.QueryStringg ("url")
%>
```

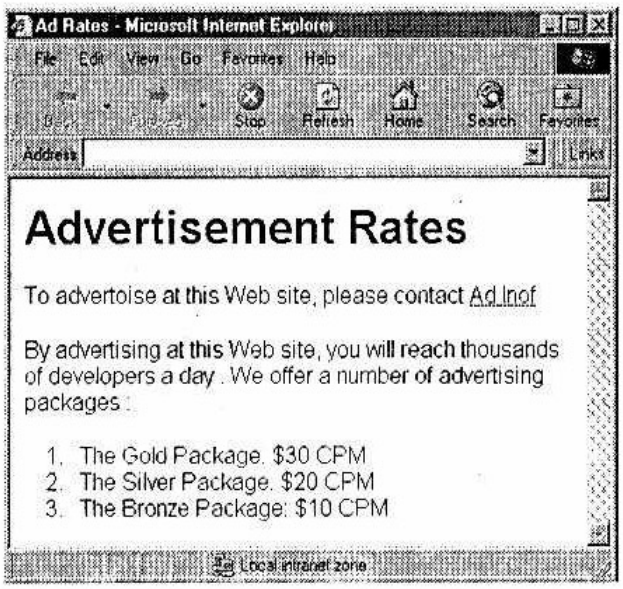
---

[1] Hyphen

[2] relative weight

این فایل تعیین مسیر شامل یک اسکریپت Asp دو خطی است که اولین خط اطلاعاتی در مورد اینکه کدام آگهی موجود در سرویس دهنده کلیک شده است را ثبت می کند. در خط بعدی اسکریپت از متد Redirect جهت فرستادن کاربر به صفحهٔ خانگی صاحب آگهی استفاده می کند

هر زمان که فایل تعیین مسیر صدا زده شود ، دو رشته در خواست رد و بدل خواهند شد. رشته درخواست url که حاوی مسیر صفحهٔ خانگی صاحب آگهی است ، این همان مسیری است که شما به عنوان مسیر صفحهٔ خانگی آگهی دهنده در فایل زمانبندی Rotator وارد نموده اید . رشته درخواست دوم image نامیده می شود . رشته درخواست image شامل مسیر تصویر آگهی است و محتوای این رشته بیانگر مسیری است که شما برای تصویر آگهی در فایل زمانبندی Rotator وارد نموده اید.در واقع شما هر چیزی را که می خواهید ، می توانید در فایل زمانبندی قرار دهید.برای مثال ،می توانید یک فایل HTML معمولی را به عنوان فایل مسیر یابی قرار دهید که بهای تبلیغات از طریق سایت شما را نشان دهد. به مثال توجه کنید:



شکل ۲- ۱۰ اطلاعات آگهی

```
< HTML >

< HEAD > < TITLE > Ad Rates < / TITLE > < / HEAD >

< BODY >

< H1 >Advertisement Rates < / H1 >

To advertoise at this Web site, please contact
< A HREF = " mailto : admaster @ mysite.com " > Ad Inof < / A >

<P>

By advertising at this Web site, you will reach.
thousands of developers a day.
We offer a number of advertising packages:
< OL >

<LI > The Gold Package : $30 CPM
<LI > The Silver Package : $20 CPM
```

```
<LI > The Bronze  Package : $10 CPM
< /OL >
< /BODY >
< /HTML >
```

۱۰-۱-۴ خصوصیت های Ad Rotator :

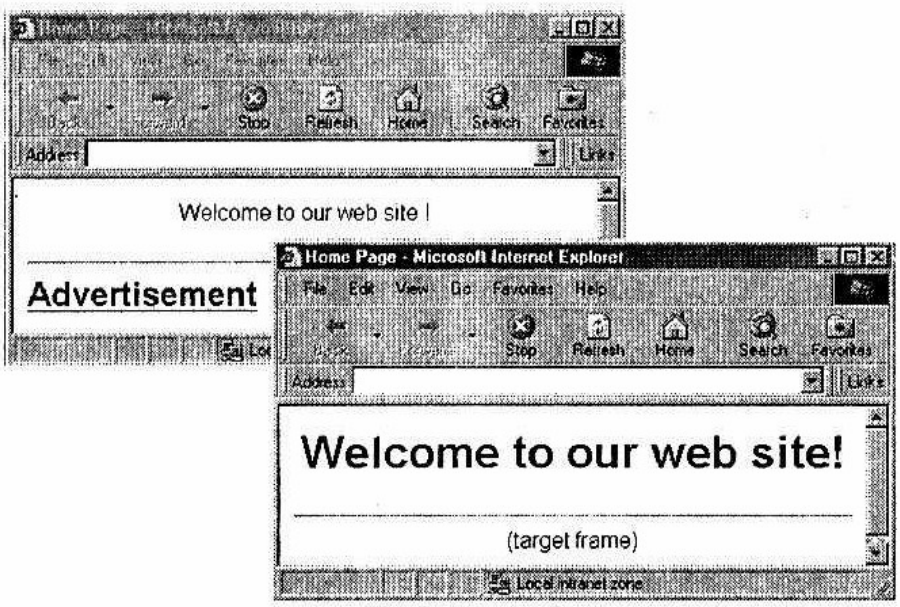
جزء Ad Rotator دارای سه خصوصیت است . قبل از فراخوانی متد ( )GetAvertisement شما می توانید این خصوصیت ها را برای کنترل چگونگی نمایش آگهی ، تنظیم کنید. در ادامه تک تک خصوصیات توضیح داده شده :

Border : این خصوصیت دربرگیرندهٔ پارامتر BORDER مشخص شده در فایل زمانبندی Rotator است. می توانید از این خاصیت برای تعیین اندازهٔ حاشیه آگهی (به پیکسل) برای یک صفحه مشخص استفاده نمایید .

Clickable : این خصوصیت مشخص می کند که آیا آگهی باید بصورت یک پیوند هم عمل کند یا نه . و مقدار آن True یا False است.

TargetFrame : این خصوصیت نام فریمی که پیوند اعلان باید از آن بار شود را مشخص می کند.خصوصیت TargetFrame بسیار مفید است. اگر شما خصوصیت TargetFrame را با نام یک فریم جدید تنظیم کنید ، زمانیکه یک کاربر روی آگهی کلیک کند صفحهٔ خانگی صاحب آگهی را در فریم مربوطه بار می کند . با این کار شما مانع از این می شوید که کاربر به طور کلی از سایت شما خارج شود. به یک مثال از چگونگی استفاده از این خصوصیت توجه نمایید :

```
< HTML >
< HEAD > < TITLE > Home Page < / TITLE > < / HEAD >
< BODY >
< CENTER > < H1 > Welcome to our web site ! < / H 1 > < / CENTER >
< HR >
<%
Set MyAd = Server.CreateObject (" MSWC.AdRotator")
MyAd.TargetFrame (" AdFrame")
%>
< CENTER > < % = MyAd.GetAdvertisement (" adrot.txt " ) % > < / CENTER >
</ BODY >
</ HTML >
```



شکل ۱۰-۳ استفاده از خصوصیات TargetFrame

۱۰-۲ جزء Content Rotator:

جزء Content Rotator بسیار شبیه جزء Ad Rotator است . البته برخلاف جزء Ad Rotator این جزء برای نمایش محتوای HTML روی صفحات وب به طور تصادفی ، استفاده می شود. درادامهٔ این قسمت چند ایده برای چگونگی استفاده از این جزء آمده است:

Tip of the Day: می توانید از این جزء برای نمایش تصادفی نماهای مختلفی از سایت وب خود، استفاده نمایید.

News Flash : این جزء می تواند جهت گردش در میان لیست رویدادهای خبری مورد استفاده قرار گیرد .

Random Link : می توانید از این جزء برای نمایش اتفاقی یک پیوند گرفته شده از سایتهای وب که در Favorite قرار داده شده ،استفاده کنید.

Bannbr advertisement : همانند جزءAd Rotator این جزء می تواند برای نمایش اعلان آگهی استفاده شود . البته با استفاده از این جزء شما انعطاف پذیری بیشتری در چگونگی نمایش آگهی خواهید داشت .

نکته: توجه داشته باشید که این جزء رسماً توسط نسخه های اولیهٔ مرورگرهای مایکروسافت حمایت نمی گردد . با این حال می توانید این جزء را از سایت مایکروسافت به آدرس http://www.microsoft.com/iis بار کنید.

جهت استفاده از جزء Content Rotatorبرای نمایش یک رشته HTML از متد ( ) ChooseContent استفاده نمایید. متد( )ChooseContent یک رشته HTML از یک فایل ویژه که فایل محتوی زمانبندی[1] نامیده می شود را دریافت می کند وآن را دریک ASP نشان می دهد . به مثال زیر توجه فرمایید :

```
< HTML >
< HEAD > < TITLE > Home Page < / TITLE > < / HEAD >
< BODY >
<%
```

---

[1]Content Scheduler file

```
Set MyContent = Server.CreateObject (" MSWC.ContentRotator")
%>
< %=MyContent.ChooseContent ("  content.txt") % >

</ BODY >

</ HTML >
```

در این مثال ابتداء یک نمونه از جزء ایجاد می گردد. این عمل توسط فرا خواندن متد ( )Create Object انجام می شود. سپس رشته HTML را از یک فایل محتوی زمانبندی به نام content.txt دریافت می کند و در صفحه وب نمایش می دهد و هر زمان که نیاز به صفحه ای باشد ممکن است که یک رشته HTML متفاوت نمایش داده شود .

۱۰-۲-۱فایل محتوی زمانبندی :

فایل محتوی زمانبندی برای گنجاندن تمام رشته های HTML ، بکار می رود. این فایل یک فایل متنی معمولی است که می تواند با هر ویرایشگر متنی ایجاد و ویرایش شود و همچنین می تواند هر نامی داشته باشد. به مثال زیر توجه نمایید :

```
%%#2 //here is the first entry
< FONT  COLOR = " RED " > Visat Our News Group ! < / FONT >
%%#3 //Here is the second entry
< B > Don't Forget To Bookmark This Web Site.  < / B >
%%#5 //Here is the third entry
Download the following free software from our Download page:
< UL >
<LI> ActiveX Components
<LI > Link Chekcer
<LI > HTML Validator
</UL>
```

این فایل محتوی زمانبندی شامل سه ورودی است. شروع هر ورودی بوسیله یک علامت درصد دوتایی(%%) مشخص شده است. هرگاه متد ( ) ChooseContent صدازده شود، یکی از ورودیها دریافت می شوند . دراین مثال به هر ورودی یک وزن مخصوص داده شده است. این کار دفعاتی که یک ورودی مشخص توسط متد ()ChooseContentانتخاب می شود را تعیین می کند. وزن را بوسیله گذاشتن یک عدد پس از علامت (#) مشخص می نمایید . به عنوان مثال ورودی اول وزنش ۲ است. وزن ورودی می تواند هر رقم بین ۰ تا ٦٥٥٣٥ باشد. اگر وزن یک ورودی صفر باشد آن ورودی هرگز نمایش داده نخواهد شد . این قابلیت برای از کار انداختن موقتی یک ورودی مفید است. اگر شما وزن ورودی را مشخص نکنید ورودی وزن به طور پیش فرض یک خواهد داشت. در این مثال اولین ورودی از هر ده مرتبه ای که متد ( ) ChooseContent فراخوانی می شود دو مرتبه نمایش داده خواهد شد . ورودی دوم از هر ۱۰ مرتبه سه مرتبه نمایش داده می شود و ورودی سوم نیز از هر ۱۰ مرتبه ٥ مرتبه نمایش داده خواهد شد . اغلب برای اینکه تعیین کنید که هر ورودی چه وقتهایی نمایش داده می شود ، وزن آن ورودی را بر مجموع وزنها تقسیم کنید. هر ورودی شامل یک توضیح است به عنوان مثال ورودی اول شامل توضیح

"Here is the first entry" است. برای گنجاندن یک توضیح به راحتی می توان از دو کاراکتر (,) و به عبارتی (//) قبل از هر توضیح استفاده کرد این توضیحات روی صفحه وب هنگام دریافت رشتهٔ HTML نشان داده نخواهند شد .

در انتها نیز هر ورودی حاوی یک رشته HTML است که این رشته می تواند به اندازهٔ یک خط باشد ، وهم چنین می تواند شامل هر برچسبی از HTML بشود. به عنوان مثال اولین ورودی در رشته Visit Our News Group! به رنگ قرمز نمایش می دهد. آخرین ورودی حاوی لیستی از نرم افزارهایی است که می توان از یک سایت وب بار کرد[1].

توضیحات و وزنها اختیاری هستند. کوچکترین فایل Content Schdule شامل چیزی نیست اما رشته های مندرجات HTML با کاراکترهای %% مجزا گردیده اند .دراین موردهر رشته مندرجات با همان سرعت نمایش داده خواهند شد. البته توضیحات و هم وزنها اختیاری اند. یک فایل محتوی زمانبندی حداقل باید حاوی رشته های HTML ای که با کاراکتر های %% از هم جدا شده اند باشد.

۱۰-۲-۲ کپی گرفتن از محتویات فایل زمانبندی :

جزء Content Rotator شامل یک متد الحاقی است. با استفاده از متد ( ) GetAllContent شما می توانید همه رشته های HTML فایل Content Rotator را دریافت کنید. این قسمت چگونگی استفاده از این متد را بیان می دارد :

```
< HTML >
< HEAD > < TITLE > Content  Schedule File Contents < / TITLE > < / HEAD >
< BODY >
<%
Set MyContent=Server.CreateObject ("MSWC.ContentRotator")
%>
< %= MyContent.GetAllContent (" content.txt") % >
< /BODY >
</ HTML >
```

زمانیکه این ASP نمایش داده می شود تمام ورودیها در فایل زمانبندی محتوی که اینجا content.txt نام دارد در این صفحه گنجانده می شوند. ورودیها بطور خودکار توسط برچسب <HR>از یکدیگر متمایز می گردند . در بعضی از موارد ممکن است این متد مفید واقع گردد . برای مثال اگر شما از جزء Content Rotator برای نمایش اتفاقی پیوندها در سایت وب استفاده می کنید ممکن است که بخواهید گزینه ای به کار بر جهت دیدن تمام پیوندها بدهید. همچنین برای این کار متد ( ) GetAllContent برای دیباگ یک فایل محتوی زمانبندی مفید است. اگر بخواهید

حضور تمام ورودیهای این فایل را تست کنید از این متد می توانید برای ملاحظهٔ محتویات آن فایل قبل از اینکه در تمام جهان منتشر شود استفاده کنید.

شمارش بازدید کنندگان

دو جزء گنجانده شده در Asp می تواند برای ایجاد شمارنده صفحه مورد استفاده قرار گیرند. با استفاده از شمارنده های صفحه ، شما می توانید تعداد دفعاتی که یک صفحهٔ مشخص درخواست می شود را تعیین کنید. و این اطلاعات را روی همان صفحه نمایش دهید ، یا از آنها برای اهداف مورد نظرخودتان استفاده کنید.

۱۰-۳ جزء Counters :

این جزء می تواند برای شمارش تعداد دفعاتی که یک صفحهٔ مشخص درخواست شده است استفاده گردد . اما برای شمارش چیزهای دیگر نیز بکار می رود . برای مثال می توانید برای شمارش تعداد بازدید کنندگان از سایت وب خود یا تعداد دفعاتی که یک آگهی کلیک می شود ویا حتی تعداد دفعاتی که یک شخص یک صفحه وب را بامرورگر نت اسکیپ درخواست می نماید استفاده کنید.

شما می توانید تنها یک نمونه از این جزء را ایجاد کنید. اگر چه وقتی که بتوانید چنین جزئی را ایجاد کنید، می توانید بسیاری از شمارنده های تکی دیگر را به هر اندازه که نیاز دارید ایجاد کنید. جزء شمارندهٔ تکی می تواند شامل شمارنده های تکی بسیاری تحت عناوین مختلف باشد .

از آنجائیکه شما می توانید فقط یک جزء Counters ایجاد نمائید . بهتر است که آنرا در فایل Global.asa ایجاد کنید. این کار تضمین خواهد کرد زمانیکه سرویس دهنده سایت وب شما شروع به کار می کند فقط یک نمونه جزء Counters ایجاد گردد . در اینجا یک مثال از چگونگی ایجاد فایل Global.asa را به شما نمایش می دهد :

```
<OBJECT RUNAT=" Server " SCOPE="Application" ID="MyCount"
PROGID="MSWC.Counters" > < / OBJECT>
```

HTML توسعه یافتهٔ ماکروسافت دارای برچسبی است به نام <OBJECT> که در این مثال برای ایجاد جزء Counters ای به نام MyCount با ناحیهٔ عمل کل برنامهٔ کاربردی ، استفاده شده است. استفاده از برچسب <OBJECT> در خارج اسکریپتها در فایل Global.asa را بخاطر بیاورید. مادامی که یک نمونه از جزء Counters به این طریق ایجاد شود ، این متد توسط هر صفحه ای از طریق برنامهٔ کاربردی مشخص شما ، قابل دسترسی خواهد بود.

جزء Counters دارای چهار متد است .در زیر چگونگی استفاده از هر متد تشریح شده است:

(نام شمارنده)Get :این متد محتوای جاری شمارنده را باز می گرداند.اگر شمارنده موجود نباشد ،ایجاد شده و مقدار آن صفر می شود.

(نام شمارنده)Increment : این متد یک شماره به مقدار فعلی شمارنده می افزاید و اگر شمارنده وجود نداشته باشد،این متد آنرا ایجاد و مقدار اولیه یک را به آن می دهد.

(نام شمارنده) Remove : این متد یک شمارنده را از بین می برد.

(عدد صحیح ، نام شمارنده) Set: این متد دو آرگومان دارد که با کاما از هم جدا می شوند. اولین آرگومان نام یک شمارنده و دومین آرگومان یک عدد صحیح است. این متد مقدار عدد صحیح را به مقدار شمارنده اضافه می کند. اگر شمارنده ای وجود نداشته باشد این متد یک شمارنده با مقدار اولیه عدد صحیح موجود در آرگومان دوم ایجاد می کند. به محض اینکه یک نمونه جزء Counters در فایل Global.asa ایجاد گردید شما می توانید شمارنده های تکی را از طریق ASP کاهش و یا افزایش دهید. شمارنده ایجاد شده دریک صفحه می تواند افزایش و کاهش یابد و یا به صفحه دیگر انتقال یابد. در ذیل مثالی از چگونگی استفاده از جزء Counters برای نگه داشتن تعداد دفعاتی که صفحه ای مشخص درخواست شده آمده است:

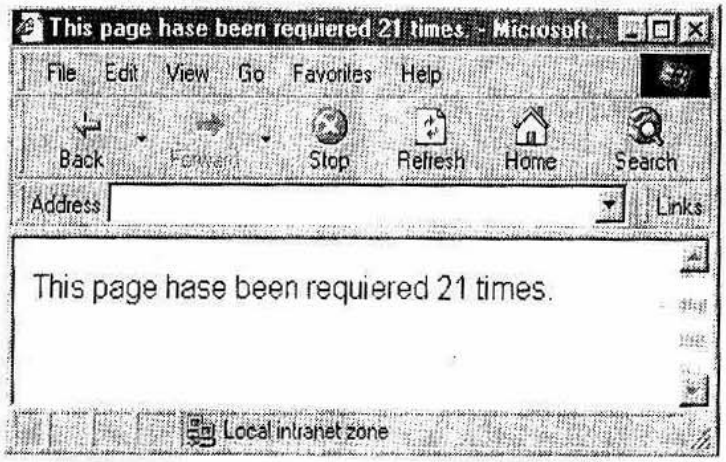
```
< HTML >
< HEAD > < TITLE > Some Page < / TITLE > < / HEAD >
< BODY >
This page has been requested
<%= MyCount.Increment (" PageCnt")% >
times.
< /BODY >
< /HTML >
```

بار اول که این صفحه درخواست می گردد شمارنده ای به نام PageCnt ایجاد می گردد و به آن مقدار یک داده می شود. درخواستهای بعدی مقدار این شمارنده را یکی یکی اضافه می کند. شمارنده PageCnt تعداد دفعاتی که صفحه درخواست می شود را منعکس می کند .

اگر ناگهان سرویس دهنده شما خاموش شود چه رخ می دهد ؟ پاسخ این است که شمارنده هائی را که شما با جزء Counters ایجادکرده اید پا برجا هستند. آنها در فایلی به نام Counters.txt ذخیره می گردند . اگر سرویس دهنده شما ناگهان خاموش گردد مقدار شمارنده ها تا وقتی که مجدداً دوباره شروع به کار کنند در این فایل موجود است. بیشتر شمارنده هایی را که شما در سایت وب می بینید از تصاویر برای نمایش شماره ها استفاده می کنند. شما نیز می توانید این کار را به صورت زیر انجام دهید:

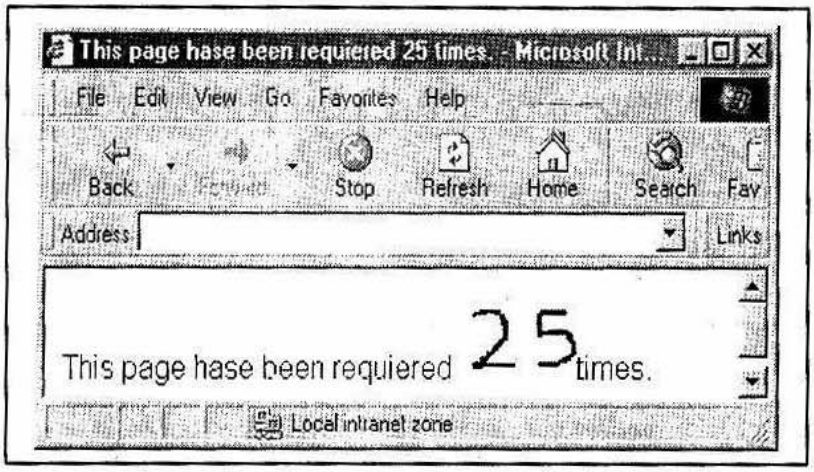
```
<%
SUB ShowImageCnt (TheNum)
CntStr = CSTR(TheNum)
FOR  i =1  TO  LEN( CntStr)
CntPart = MID( CntStr , i, 1)
%>
< % IMG SRC = " < % CntPart % >.gif "    ALT = " < % = CntPart >%>
<%
NEXT
END SUB
%>
< HTML >
< HEAD > < TITLE > Some Page < / TITLE > < / HEAD >
```

```
< BODY >
This page has been requested
<%
ShowImageCnt  MyCount.Increment (" PageCnt")
%>
times.
</ BODY >
</ HTML >
```



شکل ٤-١٠ صفحه ای با یک شمارنده

این ASP جدید نیز یک شمارندهٔ صفحه را نشان می دهد. البته شمارنده های صفحه شمارها بیشتر توسط تصاویر نمایش داده می شوند تا متن (رجوع به شکل ١٠-٥) رویه ای به نام ShowImageCnt ابتدا عدد شمارندهٔ صفحه را به یک رشته تبدیل می کند. سپس توسط یک حلقهٔ FOR ... NEXT به تعداد کاراکترهای رشته (به عبارتی ارقام شمارنده) به سراغ تصاویری که متناسب با شمارهٔ هر کاراکتر ذخیره شده است ، می رود و آنها را نمایش می دهد.



شکل ١٠-٥ شمارندهٔ صفحه با تصویر

جهت استفاده از این مثال لازم است که ١٠ تصویر به نامهای 0.gif،1.gif تا 9.gif ایجاد نمایید. می توانید این تصاویر را خودتان ایجاد نمایید . البته شماری از سایتهای اینترنت کتابخانه هایی برای تصاویر شمارنده ها دارند که شما می توانید به راحتی این تصاویر را انتقال دهید.

۱۰-۳-۱ جزء Page Counter :

جزء ثانویه ای وجود دارد که شما می توانید برای نمایش یک شمارنده صفحه در صفحه وب از آن استفاده کنید. با استفاده از این جزء می توانید تعداد دفعات باز شدن یک صفحه وب مشخص را دنبال کنید.

نکته: این جزء رسماً توسط مایکروسافت حمایت نمی شود . ولی می توانید آن را از سایت اطلاعاتی http://www.microsoft.com/iis بار کنید.

این جزء از جزء Countres انعطاف پذیر ی کمتری دارد و از آن جز برای Hit Counter برای کار دیگری نمی توان استفاده نمود. این جزء دارای دو متد است :

(مسیر)Hits : این متد تعداد دفعات باز شدن یک صفحه با مسیر معین شده را باز می گرداند . اگر هیچ مسیری از قبل تعیین نشده باشد این مقدار را برای صفحه جاری باز می گرداند .

(مسیر) Reset : این متد شمارنده را برای صفحهٔ صاحب مسیر مشخص شده صفر می کند. اگر هیچ مسیری تعیین نشده باشد مقدار شمارنده صفحه جاری را با صفر مقداردهی می کند.

بر خلاف زمان استفاده از جزء Counters شما هیچ احتیاجی به ایجاد نمونه جزء PageCounter در فایل Global .asa ندارید . می توانید نمونهٔ جزء را در همان صفحه ای که از آن استفاده می کنید ، ایجاد نمائید ، همانند مثال زیر:

```
<HTML>
<HEAD> <TITLE> Page Counter Example </TITLE> </HEAD>
<BODY>
<%
Set  MyHits=Server.CreateObject(" MSWC.PageCounter")
%>
This page has been viewed
<%= MyHits.Hits %>
times.
</BODY>
</HTML>
```

این Asp به سادگی تعداد دفعاتی که صفحهٔ جاری نمایش داده می شود ، را نمایش می دهد. متد Hits بدون پارامتر فراخوانی می شود، نتیجه را در Hit Count متعلق به صفحهٔ جاری قرار می دهد.

فصل یازدهم :

ADO

مقدمه:

این فصل اشیاء داده ای اکتیوایکس[1] (ADO) را معرفی می کند. در بخش نخست بـر ایـن اشـیاء نظـری خواهیـم داشت. در بخش دوم به شما یک راهنمائی مرحله به مرحله در استفاده از ADO برای ذخـیره و بازیـابی اطلاعـات در یک پایگاه داده ، ارائه می شود. در آخر فصل یعنی بخش سوم شما رسماً با شیء Connection که از مهمترین اشیاء ADO است آشنا می شوید.

۱۱-۱ مروری بر ADO

توسط ADO شما می توانید داده را بواسطهٔ فراهم کنندگان داده گوناگون ذخیره و بازیابی کنید. بـه عنـوان مثـال می توانید برای دستیابی به اطلاعات از پایگاه داده اکسس یا سـرویس دهنـدهٔ SQL واُراکـل از ADO اسـتفاده کنیـد. حتی می توانید از آن برای دریافت اطلاعات از نرم افزار صفحه گستردهٔ Excel استفاده نمـایید. ADO حـاوی هفـت شیء است . در لیست زیر نام این اشیاء و شرح وظایف آنها آمده است:

۱- شیء Connection: ارتباط با یک پایگاه دادهٔ خاص را برقرار می سازد. به عنوان مثـال شـما مـی توانیـد از یـک شیء Connection برای گشودن یک ارتباط با سرویس دهنده SQL مایکروسافت استفاده نمائید .

۲- شیء RecordSet: جهت کار روی مجموعه رکورد ها از این شیء استفاده می شـود. بـه عنـوان مثـال شـما مـی توانید از این شیء برای تغییر محتویات رکوردها دریک جدول سرویس دهنده SQL استفاده کنید .

۳- شیء Field: یک فیلد انفرادی را در یک مجموعه رکورد[2] نشان می دهد .

٤- شیء Command: یک دستور را نشان می دهد . برای مثال شما می توانید ازاین شیء برای اجرای یک SQL یـا یک درخواست استفاده نمائید .

٥- شیء Parameter: یک پارامتر انفرادی را در یک دستور یا رویهٔ SQL یـا پارامترهـای درخواسـت را نشـان مـی دهد .

٦- شیء Property: خواص ویژه فراهم کننده اطلاعات رانشان می دهد .

۷- شیء Error: خطاهای ADO رانشان می دهد.

هنگام استفاده از ADO شما مستقیماً با اشیاءConnection, Recordset,Command کار می کنید.

کاربرد ADO ها :

این بخش شما را در استفاده از ADO ها در ASP به طور مرحله به مرحله راهنمائی می کند. در ابتداء شما چگونگی پیکر بندی سرویس دهنده برای استفاده از ADO را فرا می گیرید. سپس در بخش بعدی با مثال ساده ای که چگونگی استفاده از ADO برای ذخیره و بازیابی اطلاعات روی یک پایگاه داده را نشان می دهد ، آشنا می شوید. و در آخر نیزدر صورتی که در عمل با مشکل مواجه شدید ، قسمتی هم برای رفع اشکال آمده است.

۱۱-۲ پیکر بندی سرویس دهنده با استفاده از ADO :

در این کتاب فرض شده که شما ADO را با سرویس دهندهٔ SQL مایکروسافت استفاده می کنید. ابتدا باید سرویس دهنده SQL مایکروسافت را روی همان ماشینی که سرویس دهنده وب تان نصب شده یا ماشین مشابهی که در جایی روی شبکه شما که سرویس دهندهٔ وب شما هم آنجا قرار دارد نصب نمائید . قبل از این که بتوانیداز ADO استفاده کنید باید یک منبع داده[1] حاوی اطلاعاتی در مورد چگونگی اتصال به فراهم کننده داده را ایجاد کنید . دراین مورد شما از منبع داده برای اتصال به سرویس دهنده SQL مایکروسافت استفاده خواهید کرد . سه نوع منبع داده وجود دارد ؛ شما می توانید یک منبع داده کاربر ،یک منبع داده سیستم و یک منبع داده فایل را ایجاد کنید. هنگام ایجاد یک منبع داده برای استفاده به همراه سرویس دهنده وب شما باید یک منبع دادهٔ فایل نیز ایجاد کنید. مزیت ایجاد یک منبع دادهٔ فایل این است که اطلاعات ارتباطی دریک فایل واقعی ذخیره می شوند. و بیش ازیک کاربر به این فایل دسترسی خواهند داشت. همچنین اگر نیاز باشد که برنامه های سرویس دهندهٔ وب خود را به سرویس دهنده وب منتقل کنید . می توانید به سهولت این فایل را منتقل نمائید.

نکته : قبل از ایجاد یک منبع اطلاعاتی جدید ، اطمینان حاصل کنید که سرویس دهنده SQL فعال باشد. از SQL Service Manager در مجموعه برنامه Microsoft SQL Server استفاده نمائید ، و نشان دهید که آیا SQL Service فعال می باشد. برای ایجاد یک منبع دادهٔ فایل، مراحل زیر را پی بگیرید:

۱- در سرویس دهندهٔ ویندوزNT ، ControlPanel را باز کنید .

۲- روی آیکون ODBC کلیک کنید .

۳- روی FileDSN کلیک کنید .

٤- روی دکمه ADO کلیک کنید ، جعبه محاورهٔ Create New Data Source ظاهر می شود .

٥- در جعبه محاورهٔ مربوطه ، فهرست سرویس دهنده SQL را انتخاب وروی دکمه Next کلیک کنید . جعبه محاورهٔ Create New Data Source باز می شود .

٦- یک نام برای منبع دادهٔ فایل جدیدتان وارد کنید . برای مثال نام MyData.dsn را وارد کنید سپس روی Next وبعد روی Finish کلیک کنید. ویزارد Create New Data Source To SQL Server می آید.

۷- درجعبه متن Description یک توصیف مانند My Data Source را وارد کنید. در جعبه متن سرویس دهنده نام سرویس دهنده ای راکه سرویس دهنده SQL مایکروسافت نصب شده بنویسید وروی Next کلیک کنید .

۸- یک سری از جعبه های محاوره ای از شما خواص مختلف پیکر بندی منبع اطلاعاتی را می خواهند . شما باید یک پایگاه داده پیش فرض انتخاب کنید ونام پایگاه داده را جایی که جداول شما قرار دارند تعیین کنید . می توان بقیه انتخابها را با مقادیر پیش فرض شان انتخاب کرده و روی Next کلیک کنید.

۹- سرانجام به شما امکان آزمایش منبع داده جدیدتان داده خواهد شد . اگر آزمایش شما موفقیت آمیزبود روی دکمه OK کلیک کنید ومنبع دادهٔ جدید را اضافه نمائید .

شما اکنون یک منبع دادهٔ فایل جدید به نام MyData.dsn دارید که می توانید از آن برای اتصال به سرویس دهنده SQL مایکروسافت استفاده کنید.

۱۱-۲-۱ استفاده از ADO برای ذخیره و بازیابی اطلاعات دریک پایگاه داده :

این قسمت مثالی از چگونگی استفاده ADO برای ذخیره و بازیابی داده در یک سرویس دهندهٔ SQL مایکروسافت را نشان می دهد . این مثال دو منظور دارد یکی اینکه روشهای اساسی دستیابی سرویس دهنده SQL مایکروسافت از یک ASP را توضیح می دهد و دیگری اینکه شما می توانید از مثال زیربرای بررسی پیکر بندی سرویس دهنده تان استفاده نمائید . ASP مربوط به لیست زیر متن !Hello World را در یک جدول پایگاه داده وارد می کند. سپس متن !Hello World از جدول دریافت و به مرورگر کاربر فرستاده می شود.

```
<HTML>

<HEAD> <TITLE> ADO Example </TITLE > </HEAD>

<BODY>

<%
Set MyConn=Server.CreateObject ("ADODB.Connectio n")
MyConn.Open "FILEDSN = d:\Program Files\Common Files\ODBC\
Data Sources \MyData.dsn"
MyConn.Execute   " INSERT MyTable(MyColumn) VALUES (' Hello World!')"
Set RS=MyConn.Execute ("SELECT * FROM MyTable")
Response.Write (RS ("MyColumn"))
MyConn.Close
%>
</BODY>

</HTML>
```

قبل از اینکه بتوانید از این مثال استفاده کنید لازم است که جدولی به نام MyTable ایجاد کنید. این کار رامی توانید توسط ISQL/w انجام دهید. ابتدا برنامه را اجرا کنید ، پایگاه داده پیش فرض خودرا انتخاب نمائید و دستور SQL زیر را اجرا کنید:

```
CreateTable MyTable(MyColumn VARCHAR(255))
```

اولین خط برنامهٔ ASP بالا یک نمونه شیء Connection را ایجاد می کند .سپس متد Open از شیء Connection برای برقراری ارتباط با پایگاه داده صدا زده می شود. در خواست منبع دادهٔ فایل که در قسمت قبلی

ایجاد نمودید توسط متد Open برای برقراری ارتباط استفاده می شود . (جای مسیر منبع دادهٔ فایل موجود در اسکریپت را با مسیر صحیح آن روی ماشین خودتان تصحیح کنید.)

به محض این که یک ارتباط ایجاد شد می توانید جملات SQL را توسط متد Execute از شیء Connection اجرا نمائید . متد Execute برای اجرای دو جمله SQL در این اسکریپت استفاده می شود. ابتدا از دستور INSERT برای وارد کردن رشته !HelloWorld به جدول پایگاه داده استفاده می شود. سپس از دستورSELECT برای دریافت این رشته از جدول پایگاه داده استفاده می شود. اگر برنامهٔ شما بطور صحیح اجرا گردد رشته !HelloWorld باید در پنجره مرورگر شما نشان داده می شود. در صورتی که رشتهٔ مورد نظر نمایش داده نشد به بخش بعدی توجه کنید.

۱۱-۲-۲ رفع اشکال ADO:

اگر مشکلاتی در دستیابی به سرویس دهنده SQL مایکروسافت با ADO دارید این بخش شاید مفید واقع شود. ASP شرح داده شده در بخش قبل به دلایل مختلفی می تواند بی نتیجه باشد. در لیست زیر آثار و علل شماری از مشکلات متداول در این راه آمده است:

علامت: شما پیغام خطای Unable Create File Buffer را دریافت کنید.

علت: منبع دادهٔ فایل نادرست است. این پیغام خطا را هنگامی دریافت می کنید که منبع دادهٔ فایل مسیر یا نام اشتباهی دارد و یابه طور کلی وجود ندارد. مطمئن شوید که مسیر منبع دادهٔ فایلی که روی ماشین خود بکار برده اید با مسیری که در متد Open در Asp بکار برده اید یکی است یا نه.

علامت: پیغام خطای " Invalid Object Name " My Table را دریافت می کنید.

علت : جدول MyTable در پایگاه داده شما موجود نیست . این جدول را با استفاده از ISQL/w ایجاد کنید .

علت دیگر: اگر جدول MyTable در پایگاه داده پیش فرض شما قرار نگرفته باشد احتیاج دارید که یک پایگاه داده پیش فرض برای خودتعیین کنید . وارد Control Panel شده و سپس به ترتیب آیکون ODBC وفایل DSN را کلیک کنید . نام منبع داده فایل را انتخاب کنید وسپس به ترتیب Configure وبعد Optiens را کلیک کنید . می توانید نام پیش فرض پایگاه دادهٔ خود را درجعبهٔ محاوره موجود تعیین کنید .

علامت: پیغام خطای The Server appears to be not availble را دریافت کرده اید.

علت : SQLسرویس دهنده در حال اجرا نیست. از گروه برنامه مایکروسافت SQL سرویس دهنده SQL Service Manager را باز و MSSQLسرویس دهنده انتخاب کنید . و سپس روی چراغ سبز کلیک کنید.

علامت: پیغام خطای Login failed را دریافت می کنید.

علت : شما از تعیین نشانی ویندوز NT استفاده نکرده اید و بدین ترتیب یک Loggin ID یا Password اشتباه موقع ایجاد منبع دادهٔ فایل وارد شده است. وارد Control Panel شویدو سپس به ترتیب آیکون ODBC وفایل DSN را کلیک کنید . نام منبع دادهٔ فایل را انتخاب نموده و سپس روی Cofigure کلیک کنید. می توانید Use Trusted Connection را انتخاب نمائید ویا یک loggin ID و کلمه رمز معتبر را وارد کنید .

علامت: شما پیغام خطای SELECT IINSERT permission denied on object MyTable

یا permission denied on object MyTable را دریافت می کنید.

علت : هنگام ایجادمنبع دادهٔ فایل login ID ای مشخص کرده اید که از اجازه کافی برای دسترسی به جدول MyTable برخوردار نیست. لازم است که شما login ID بکار رفته را توسط منبع دادهٔ فایل را تعویض نموده ویا با این Login به سرویس دهندهٔ SQL ، اجازه های اضافی به گروه یاکاربر بدهید.

برای دادن اجازه های بیشترمورد نیاز ، SQL Enterprise Manager از گروه برنامه Microsoft SQL Server را انتخاب کنید. به جدولی با نام MyTable بروید وروی آن کلیک راست کنید .سپس Permission راانتخاب کنید . حالا می توانید اجازه ها رابرای کاربران مختلف یا جداول مختلف ایجاد کنید. باید به یاد داشت که برای قطعیت بخشیدن به تمام تغییراتی که روی اجازه ها صورت گرفته است در نهایت باید روی دکمه Set کلیک نمود .

۱۱-۳ بکارگیری شیء Connection:

تمام عملیات مربوط به پایگاه داده در میان یک ارتباط باز جای می گیرد. قبل این که بتوانید هرگونه اطلاعاتی راجایگزین ویا وارد پایگاه داده نمائید باید ارتباط با پایگاه داده را باز کنید. این فرایند بازشدن وبسته شدن ارتباط، از طریق متدهای مختلف شیء Connection صورت می گیرد.

۱۱-۳-۱ گشودن وقطع ارتباط با پایگاه داده :

برای باز نمودن یک ارتباط با پایگاه داده می توانید یک نمونه[1] از شیء Connection را ایجاد کنید. به محض این که نمونه شیء ایجاد شد. می توانید متد Open از شیء Connection برای گشودن یک ارتباط حقیقی استفاده کنید. در مثال زیر توضیحات کافی داده شده است :

```
<%
Set MyConn=Server.CreateObject ("ADODB.Connection")
MyConn.Open    " FILEDSN = d:\Program Files\Common Files\ODBC\
Data Sources\MyData.dsn"
MyConn.Execute "INSERTMyTable(MyColumn)VALUES("Hello Wold!")
MyConn.Close
%>
```

دراین مثال یک نمونه از شیء Connection به نام MyConn ایجاد گردیده است . متد Open با نام یک منبع دادهٔ فایل فرا خوانده می شود . سپس متد Execute ازشیء Connection فرا خوانده می شود و یک جمله SQL اجرا می گردد ، سرانجام اتصال بسته می شود .

هر شخصی ممکن است از اینکه هر بار که نیاز به باز شدن یک ارتباط دارد باید ، نام منبع دادهٔ فایل را وارد کند خسته شود. شما باید این رشته را به یک متغیر Session تخصیص دهید ویا یک ثابت در فایل Includeبسازید. برای اینکار شما فقط باید نام آن متغیر را متناسب با نام کامل منبع دادهٔ فایل وارد کنید. برای ایجاد این متغیر Sessionکه

حاوی نام منبع دادهٔ فایل است شما باید این متغیر را در فایل Global .asp ایجاد کنید. برای مثال شما می توانید خط زیر را به قسمت OnStart-Session در فایل Global .asp اضافه کنید :

```
Session (" connectionstring ") = "FILEDSN =d:\Program Files \Common
Files \ODBC \Data Sources\MyData.dsn"
```

پس از این که متغیرSession ایجاد گردید می توانید یک اتصال با استفاده از اسکریپت زیر بازکنید :

```
<%
Set MyConn= Server.CreateObject ("ADODB.Connectio n")
MyConn.OpenSession("connectionstring")
```

یکی از فواید دادن نام منبع دادهٔ فایل به متغیرSession این است که می توان به راحتی منبع داده را تغییر شکل داد. اگر شما به منبع دادهٔ متفاوتی نیاز داشته باشید به آسانی می توانید مقدار متغیر Session تکی را در فایل Global.asa تغییر دهید. هنگامی که استفاده از ارتباط تمام شد، شما باید این ارتباط را ببندید. متد Close از شیء Connection اینکار را انجام می دهد. پس از بسته شدن ارتباط، شما نمی توانید با پایگاه داده ارتباط داشته باشید.

۱۱-۳-۲ اجرای جملات SQL با یک ارتباط باز :

برای اجرای جملات SQL با یک ارتباط باز از متد Execute استفاده نمائید . این متد دو حالت دارد : اولین شکل زمانی است که نتیجهٔ دستور SQL خروجی باشد که از پایگاه داده ، باز گردانده می شود ودومین شکل زمانی است که دستور SQL هیچ نتیجهٔ خروجی نداشته باشد. مثال زیر چگونگی استفاده از متد Execute را نشان می دهد:

```
<%
Set MyConn =Server.CreateObject ("ADODB.Connectio n")
MyConn.Open  " FILEDSN = d :\ Program  Files\Common Files\ODBC\
Data Sources \MyData.dsn"
MyConn.Execute  "INSERT  MyTable(MyColumn) VALUES (' Hello World!')"
MyConn.Close
%>
```

در این مثال متد Execute برای اجرای دستور INSERT استفاده شده است. از آنجائیکه هیچ نتیجه ای بازگردانده نمی شود ، متد Execute از پرانتز ها استفاده نمی کند. همچنین می توانید از متد Execute برای بازگرداندن نتیجه یک Recordset از یک درخواست SQL استفاده کنید ، مانند مثال زیر:

```
<%
Set MyConn=Server.CreateObject ("ADODB.Connection")
MyConn.Open   " FILEDSN = d:\Program Files\Common
Files \ODBC\Data Sources\MyData.dsn
Set  RS=MyConn.Execute ("SELECT  * FROM  MyTable")
MyConn.Close
%>
```

در این مثال متد ( ) Execute برای بازگرداندن نتیجه از یک درخواست SELECT استفاده شده است. دراین مثال متد Execute از پرانتزها استفاده می شود. هنگام بازگرداندن نتایج پرانتزها را نباید فراموش کرد و گرنه پیغام خطای Expected end of statement دریافت خواهید کرد.

نتیجه دستورSQL بصورت یک شیء RecordSet به نام RS دریافت می شود. این مجموعه رکورد بطور خودکار ایجاد می گردد. این عمل توسط متد ( )Execute ایجاد می گردد. این متد حاوی دو پارامتر اختیاری است که می توانید از پارامتر RecoerdsAffected که تعداد رکوردهای یک جمله SQL را نشان می دهد، استفاده

کنید. همچنین می توانید از پارامتر اختیاری Options که اطلاعاتی در مورد نوع جمله اجرا شده می دهد ، استفاده کنید. این مثال هر دو پارامتر اختیاری بالا را برای شما نشان می دهد:

```
<!--# INCLUDE  VIRTUAL ="ADOVBS.inc"-- >
<%
Set MyConn=Server.CreateObject ("ADODB.Connection")
MyConn.Open "FILEDSN=d :\Program Fils\Common
Files\ODBC\Data Sources \MyData.dsn"
MyConn.Execute  "UPDATE MyTable Set
MyColumn = ' Goodbye! '", HowMany,adCMDText
Response.Write (HowMany)
MyConn.Close
%>
```

در این اسکریپت جمله UPDATE برای تغییر مقادیر تمام سطرهای جدول MyTable استفاده می شود.متد Execute دو پارامتر اضافی برقبلی دارد؛پارامتر اول پارامتر RecordsAffected است . دراین مثال متغیری به نام HowMany به عنوان پارامتر RecordsAffected عبور داده می شود. پس از این که جمله SQL اجرا گردید این متغیر حاوی تعداد رکوردهای موثر جمله SQL خواهد شد .

دومین پارامتری که در متد Excute وجود دارد پارامتر Options است ، که در این مثال مقدار ثابت adCMDText تعیین شده است. با این ثابت ADO را مطلع می سازید که مندرجات رشته دستور چیزی بیش از نام یک جدول یا رویهٔ ذخیره شده می باشد و باید تفسیر شود. با اخطار به ADO در مورد محتویات رشته در حال اجرا ، ثابت مزبور دستور اجرایی ADO را موثرتر می کند . شما می توانید ثابتهای زیر را برای پارامتر Options استفاده نمائید:

۱- adCMDTable :رشته ای که اجرا می گردد محتوی نام یک جدول است .

۲-adCMDText: رشته ای که اجرا می گردد محتوی یک دستور متنی است .

۳- adCMDStoredProc :رشته ای که اجرا می گردد محتوی نام یک پروسیجر ذخیره شده است.

٤- adCMDUnkown : مندرجات رشته نامشخص است. (این ثابت مقدار پیش فرض است).

قبل از این که شما بتوانید از هر کدام از این ثابتها در یک ASP استفاده نمائید باید یک فایل ویژه ومخصوص به نام ADOVBS.inc را برای خودتان فرض بگیرید. فایل ADOVBS.inc حاوی تمام ثابتهای VBScript ای که با ADO استفاده می گردند، است. اولین سطر مثال قبلی ، حاوی یک دستور INCLUDE است که حاوی فایل ADOVBS.inc است .

وقتی شما ASP رانصب کردید این فایل باید به طریق خودکار نصب گردد . عموماً فایل در شاخهٔ c:\Program File\Common File\System \ADO به طور مستقیم ساخته می شود. به هرحال ممکن است که شما برای یافتن محل دقیق این فایل به دستور Findاز منو Start،ویندوز NT نیاز داشته باشید .پس از یافتن ، آن فایل را در ASP خودتان به طور مستقیم کپی نمائید. مانند زمانهایی که شما احتیاج به بازنمودن یک ارتباط داشتید برای این کار نیز می توانید از متد Execute استفاده نمائید. به عنوان مثال اسکریپت زیر ۳۲ رشته مختلف را به جدولی به نام MyTable وارد می کند ، توجه نمائید:

```
<!--# INCLUDE  VIRTUAL="ADOVBS.inc"-- >
<%
Set  MyConn =Server.CreateObject ("ADODB.Connection")
MyConn.Open " FILEDSN = d :\Program Files\Common
Files\ODBC\Data Sources\MyData dsn"
FOR  i = 1    To   32
MySQL = "INSERT MyTable(MyColumn) VALUES ('This is entry "&i& "')"
MyConn.Exectue HowMany,adCMDText,MySQL
NEXT
MyConn.Close
%>
```

این اسکریپت از یک حلقه FOR...NEXT استفاده می کند تا ۳۲ رکورد را در جدول MyTable ضبط نمائید. متغیر MySQL حاوی رشته دستورSQL است که از متد Exectue استفاده کرده است. دقت کنید که چطور از علامت گیومهٔ تکی و گیومهٔ دوتایی زمان تعیین رشته SQL استفاده می شود . دراین قسمت گیومهٔ دوتایی (") شروع وپایان رشته را در VBscript مشخص می کند . شما می توانید با متد Exectue تقریباً هر دستور SQLای را انجام دهید به عنوان مثال اسکریپت زیر یک جدول را ایجاد کرده آنرا پروخالی می کند و سپس آنرا بر می دارد :

```
<!--# INCLUDE  VIRTUAL="ADOVBS.inc"-- >
<%
Set  MyConn =Server.CreateObject ("ADODB.Connection")
MyConn.Open  " FILEDSN=d :\Program Files\Common
Files\ODBC\Data Sources\MyData.dsn"
'Create  a  new  table
MySQL = "CREATE TABLE NewTable(MyColumn VARCHAR (255))"
MyConn.Exectue MySQL
'Populate  the  table
MySQL = "INSERT NewTable(MyColumn)  VALUES  (' hello')"
MyConn.Exectue MySQL
'Truncate the table
MySQL = "TRUNCATE TABLE NewTable"
MyConn.Exectue MySQL
'Drop the table
MySQL= "DROP TABLE NewTable"
MyConn.Exectue MySQL
MyConn.Close
%>
```

۱۱-۳-۳ ایجاد تراکنشها[1] :

زمانیکه یک گروه از جملات ، یک تراکنش را شکل می دهند، اگر یک جمله پذیرفته نشود تمامی آن جملات پذیرفته نخواهد شد. تراکنشها زمانی مفید است که به هنگام کردن اطلاعات روی بیش از یک جدول را نیاز دارید و می خواهید همگی با هم انجام شوند. به عنوان مثال زمانی را تصور نمائید که شخصی از سایت وب شما چیزهایی را خریداری کند اطلاعات خرید در دو جدول ،که یکی جدول حاوی لیست کارت های اعتباری بدهکار و دیگری حاوی لیستی از اقلام خریداری شده است،می باشد.حال تصور کنید شخصی می خواهد از سایت وب شما خرید نماید . شماره کارت اعتبار ی شخص در اولین جدول شما وارد می گرددودر یک لحظهٔ هر چند کوتاه ، اتفاقی می

افتد مثلاً ناگهان برق دستگاه قطع می شود . به این ترتیب جدول دوم شما به هنگام نمی شود. در این موقعیت بهتر است که شما اطلاعات موجود را به هنگام نکنید چرا که شما پول اقلامی که هنوز خریداری نشده است را می گیرید. پس بهتر است مادامی که اقلام در جدول به هنگام نشده شمارهٔ کارت اعتباری به هنگام نشود. به مثال زیر توجه کنید:

```
<%
Set MyConn=Server.CreateObject("ADODB.Connection")
MyConn.Open " FILEDSN=d :\Program  Files\Common
Files\ODBC\Data Sources\MyData.dsn"
MyConn.BeginTrans
MyConn.Execute "INSERT  CreditCard(CCNum) VALUES (' 5555-55-555-55-5555')"
MyConn.Execute "INSERT   Shipping (Address) VALUES ('Paris , France')"
MyConn.CommitTrans
MyConn.Close
%>
```

در این مثال از متدهای CommitTrans و BeginTrans استفاده شده است که برای علامتگذاری پایان و شروع هر تراکنش بکار می رود.بعد ازاینکه متد BeginTrans صدا زده شد اگر مورد اشتباهی قبل از اینکه CommitTrans صدا زده شود ، رخ دهد جداول به هنگام نمی شوند . همچنین می توانید این عمل را به طور صریح رو به عقب انجام دهید . برای انجام این کار از متد RollBackTrans می توانید استفاده کنید .به قسمت زیر دقت کنید :

```
<%
Set My Conn = Server.Create Object (“ADDB. Conection”)
MyConn.Open”FILEDSN=d:\ProgramFiles\CommonFiles\ODBC\ DataFource\ MyData.Dsn ”
My Conn.BeginTrans
My Conn.Execute “ INSERT CreditCard (CCNUM) VALUES(“5665 – 65 – 555 – 56 – 5555 ”)
My Conn.Execute “ INSERT Shipping (Address) VALUES(“ Paris,France “)
IF WEEKDAYNAME(WEEKDAY (DATE)) = “Sunday ” THEN
MyConn.RollBack Trans
ELSE
MyConn.CommitTrans
END IF
MyConn.Close
%>
```

دراین مثال از متد RollBackTrans استفاده گردیده که صریحاً تراکنش را برای روز یکشنبه مرجوع می کند در روز یکشنبه نه جدول کارت اعتباری ونه جدول محموله هیچکدام به هنگام نخواهند گردید .

فصل دوازدهم :
کار با مجموعه رکوردها[1]

مقدمه

این فصل چگونگی استفاده از شیء Recordset را نشان می دهد. در بخش اول متدهای اصلی نمایش اطلاعات توسط این شیء را فرا می گیرید.در بخش دوم چگونگی باز کردن یک مجموعه رکورد با انواع مختلف مکان نما[2] و قفل[3] را فرا می گیرید. و نهایتاً در بخش سه شماری از مزایای متدها برای کار با رکوردهای داخل مجموعه رکورد ، تشریح شده است.

۱۲-۱ استفاده از یک مجموعه رکورد برای نمایش رکوردها :

یک مجموعه رکورد می تواند برای پیاده کردن رکوردها در یک جدول پایگاه داده مورد استفاده قرار گیرد . مانند یک جدول ، یک مجموعه رکورد حاوی یک یا چند رکورد (ردیفها) است و هر رکورد حاوی یک یا چند فیلد (ستونها)است . که در هر زمان فقط یک رکورد ، رکورد جاری است . برای ایجاد یک نمونه شیء Recordset جدید شما می توانید متد ( ) Execute از شیء Connection را به کار گیرید . زمانیکه شما از متد ()Execute برای باز گرداندن نتایج از یک درخواست پایگاه داده استفاده کنید به طور خودکار یک مجموعه رکورد جدید ایجاد خواهد گردید به مثال زیر توجه فرمایید :

```
<%
Set MyConn=Server.CreatObject("ADODB.Connection")
MyConn.Open  "FILEDSN= d:\Program Files\Common Files\ODBC\
Data Sources\MyData.dsn"
Set RS=MyConn.Execute("SELECT *FROM MyTable")
RS.Close
MyConn.Close
%>
```

در این مثال از جمله SELECTجهت دریافت کل جدولی به نام MyTable استفاده می گردد . متد ()Executeیک مجموعه رکورد را بر می گرداند. در این اسکریپت مجموعه رکورد تخصیص داده شده به متغیر ، RS نام دارد سپس مجموعه رکورد بسته می شود، در نهایت ارتباط قطع می گردد .

هر رکورد از مجموعه رکورد RS متناظر با یک رکورد در جدول MyTable است. برای نشان دادن تمام رکوردها در مجموعه رکورد می توانید بسادگی مانند مثال زیر رکوردها را در حلقه بیندازید:

```
<%
Set MyConn = Server.CreatObject ("ADODB.Connection")
MyConn.Open "FILEDSN=d:\Program Files\Common Files\ODBC \Data Sources
\MyData.dsn"
Set RS=MyConn.Execute("SELECT * FROM MyTable")
WHILE NOT RS.EOF
Response.Write ("< BR>"&RS(" MyColumn"))
RS.MoveNext
WEND
RS.Close
MyConn.Close
%>
```

در این مثال حلقه WHILE ... WEND برای انتقال تک تک رکورد های موجود در مجموعه رکورد RS مورد استفاده قرار می گیرد . فیلد MyColumn از هر رکورد برای مرورگر فرستاده می شود. این اسکریپت تمام مجموعه رکوردهایی که در جدول MyTable گنجانده شده اند را نمایش می دهد. در ابتدای اشغال مجموعه رکورد توسط رکوردها ، رکوردهای جاری به عنوان اولین رکورد مورد استفاده قرار می گیرند. در مثال قبل متد MoveNext از شیء Recordset برای رفتن به رکورد بعدی استفاده می شود . زمانیکه تمام رکوردها نمایش داده شدند، خاصیت EOF از شیء Recordset دارای مقدار TRUE خواهد شد واز حلقه WHILE...WEND خارج می شود.

یک شیء Recordset دارای مجموعهٔ Fields است که حاوی یک یا چند شیء Field است . یک شیء Field یک ستون مشخص درجدول را نشان می دهد . برای مثال در اسکریپت قبل، ستون MyColumn بوسیله عبارت ("MyColumn") نمایش داده شده است . در واقع می توانید مقدار یک ستون را از چند طریق تعیین کنید. هر کدام از عبارات زیر مقدار ستونی بنام MyColumn را نشان می دهند :

```
RS ("MyColumn")
RS (0)

RS.Fields (0)

RS.Fields ("MyColumn ")
RS.Fields.Item ("MyColumn ")
RS.Fields.Item(0)
```

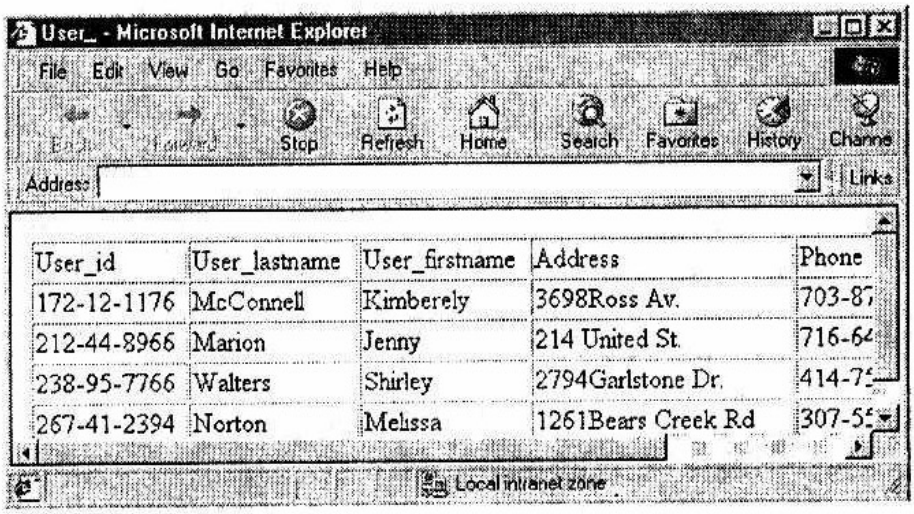
توجه داشته باشید که شما می توانید به هر فیلدی با نام یا شمارهٔ ترتیب[1] آن فیلد، رجوع کنید. برای مثال شما می توانید برای رجوع به فیلد MyColumn از ("RS("MyColumn و یا (0) RS استفاده نمائید. رجوع به این فیلد توسط شماره ترتیب آن هنگامی مفید است که شما نام فیلدهای مجموعه رکورد را نمی دانید . برای مثال اسکریپت Asp زیر تمام خطهای عمودی و افقی (سطرها وستونها)را در جدول نشان می دهد :

```
< HTML >

< HEAD > < TITLE > Show All Rows And Columns < / TITLE > < / HEAD >

< BODY >

<%
```

---

[1] Ordinal

```
Set MyConn=Server.CreateObject (" ADODB.Connection")
MyConn.Open"FILEDSN=d:\Program Files\Common Files\ODBC\Data Sources
\MyData.dsn"
Set RS=MyConn.Execute (" SELECT * FROM MyTable")
%>
<TABLE BORDER =1 >

< TR>

<% FOR  i = 0  to  RS.Fields.Count -1 % >

< TH > < % = RS (i ).Name % > < / TH>

< % Next % >

</ TR  >

< % While  Not  RS.EOF % >

< TR >

< % FOR  i = 0  TO RS.Fields.Count -1 % >

< TD > < % = RS ( i ) % > < / TD >

< % Next % >

</ TR >

<%
RS.MoveNext
WEND
RS.Close
MyConn.Close
%>
< /TABLE  >

</ BODY >

</ HTML  >
```



شکل ۱۲-۱نمایش تمام سطر ها و ستونها در جدول

در این مثال خاصیت Count از مجموعهٔ Fields شماره تعداد فیلدهای مجموعه رکورد را باز می گرداند.خاصیت Name برای دریافت نام هر فیلد مورد استفاده قرار می گیرد. هر دو حلقهٔ FOR ... NEXT جهت حرکت در میان کل فیلدهای مجموعه رکورد مورد استفاده قرار می گیرند . هیچ مهم نیست که جدول چند سطر و ستون دارد تمامی آنها بلا استثناء نمایش داده خواهند شد .

۱۲-۲ انواع مکان نماها و قفل های مجموعهٔ رکورد :

شما می توانید با یکی از چهار نوع مکان نمای موجود یک مجموعه رکورد را باز کنید . مکان نماانواع عملیاتی که شما می توانید با مجموعه رکورد اجرا کنید را تعیین می کند . لیست زیر انواع مختلف مکان نما را همراه با ساختارهایشان توضیح می دهد :

- adOpenForwardOnly: با استفاده از مکان نمای Forward-Only شما می توانید در میان رکوردهای یک مجموعه رکورد فقط به جلو حرکت دهید .
- adOpenKeyset : با استفاده از یک مکان نمای keyset شما می توانید در مجموعه رکورد هم به طرف جلو و هم به طرف عقب حرکت نمائید . اگر یک رکورد توسط کاربران دیگرحذف یا تغییر یابد ، این تغییرات به مجموعه رکورد نیز منعکس می شود ولی چنانچه رکوردی جدید بوسیله کاربر دیگر اضافه گردد رکورد جدید در مجموعه رکورد ظاهر نخواهد گشت .
- adOpenDynamic : توسط یک مکان نمای پویا شما می توانید در یک مجموعه رکورد به سمت جلو و عقب حرکت نمائید و هر تغییری که توسط کاربران دیگر در رکوردها ایجاد گردد به مجموعه رکورد انعکاس می یابد .
- adOpenStatic : توسط یک مکان نمای ثابت شما می توانید در یک مجموعه رکورد هم به جلو و هم به عقب حرکت نمائید. لیکن یک مکان نمای ثابت تغییرات اعمال شده توسط کاربران را به مجموعه رکورد منعکس نمی کند .

به طور پیش فرض زمانیکه شما یک مجموعه رکورد را باز می کنید، بوسیله یک مکان نمای ForwardOnly باز می شود. این بدین معنی است که می توانید تنها در رکورد های یک مجموعه رکورد با استفاده از متد MoveNext ، به جلو بروید. مزیت روش مکان نمای Forward-Only این است که بسیار سریع صورت می گیرد . هر زمان که بخواهید با آن کار کنید یا سرعت به کار خود دهید باید از متد مکان نمای forward-only استفاده نمایید . به هر حال اگر شما نیاز به باز کردن یک مجموعه رکورد با مناسبترین مکان نما را داشتید می توانید مانند اسکریپت زیر عمل نمایید :

```
<!--#INCLUDE VIRTUAL = "ADOVBS.inc" -->
<%
Set MyConn=Server.CreateObject("ADODB.Connection")
Set RS=Server.CreateObject("ADODB.Recordset")
MyConn.Open "FILEDSN =d:\Program Files\ODBC\Data Sources\MyData.dsn"
RS.Open " SELECT * FROM  MyTable ", MyConn,adOpenDynamic
RS.Close
MyConn.Close
```

%>

جهت باز نمودن یک مجموعه رکورد با یک مکان نمای خاص، باید به طور صریح یک مجموعه رکوردرا ایجاد نمائید و سپس آن را با مکان نمای مربوطه بازکنید . برای انجام این چنین عملی در ابتدا باید یک شیء Recordset ایجاد کنید ، سپس از متد Open مجموعه رکورد را توسط ارتباط و نوع مکان نمای مشخص باز نمائید . در این اسکریپت مجموعه رکورد RS بوسیله شیءارتباطی MyConn و مکان نمای پویا باز شده است . شما می توانید یکی از چهار مورد زیر را جهت قفل کردن رکورد انتخاب نمائید :

adLockReadOnly : نشان می دهد که شما نمی توانید رکوردهای موجود در مجموعه رکورد را تغییر دهید.

adLockPessimistic : نشان می دهد که یک رکورد باید به محض ویرایش سریعاً قفل شود.

adLockOptimistic : نشان می دهد که یک رکورد تنها زمانی که متد Updute از شیء Recordset صدا زده می شود، باید قفل شود.

adLockBatchOptihmsitic : رکوردهایی که دسته ای به هنگام[1] خواهند شد، را مشخص می نماید.

به طور پیش فرض یک مجموعه رکورد از قفل فقط خواندنی(Read Only) استفاده می کند. برای مشخص نمودن نوع دیگر قفل باید یکی از ثابتهای بالا را هنگام بازکردن مجموعه رکورد بکار بگیرد.با مشخص کردن یک قفل متفاوت شما می توانید یکی از این قفل ها را زمانیکه می خواهید مجموعه رکورد را باز کنید برای خود بکار برید . در اینجا یک مثال از چگونگی انجام آن وجود دارد :

```
<!--#INCLUDE VIRTUAL = "ADOVBS.inc" -->
<%
Set MyConn = Server.CreateObject(" ADODB.Connection")
Set RS=Server.CreateObject("ADODB.Recordset")
MyConn.Open"FILEDSN=d:\Program Files\Common Files\ODBC\Data Sources\MyData.dsn"
RS.Open " SELECT * FROM MyTable " ,MyConn,adOpenDynamic,adLock pessimistic
RS.Close
MyConn.Close
%>
```

این اسکریپت نیز مانند اسکریپت قبلی است و تنها در قسمت اضافه کردن نوع قفل با آن تفاوت دارد. مجموعه رکورد RS توسط قفل Pessimistic باز می شود. اینکار بدان معنی است که رکوردهای موجود در مجموعه رکورد می توانند تغییر کنند. سرانجام زمانیکه شما یک مجموعه رکورد را باز نمودید، می توانید یک پارامتر Options را مشخص کنید. پارامتر Options نوع رشته دستوری که در مجموعه رکورد برای باز کردن مجموعه رکورد بکار رفته را نشان می دهد . می توانید ثابتهای زیر را جهت پارامتر Options مورد استفاده قرار دهید :

adCMDTable : رشتهٔ اجرا شده حاوی نام جدول است .

adCMDText : رشتهٔ اجرا شده حاوی یک دستور متنی است .

adCMDStoredProc : رشتهٔ اجرا شده حاوی نام رویهٔ ذخیره شده است.

adCMDUnknown : محتوی رشته مشخص نشده است. (مقدار پیش فرض است) در اسکریپت زیر برای اینکه به ADO اخطار دهد که محتوی دستور رشته دستور متنی است از پارامتر Options استفاده شده است :

```
<!--#INCLUDE VIRTUAL = "ADOVBS.inc" -->
<%
Set MyConn = Server.CreateObject("ADODB.Connection")
Set RS=Server.CreateObject("ADODB. MyConn.Open " FILEDSN =d:\Program
Files\Common Files\ODBC\Data Sources\ MyData.dsn"
RS.Open " SELECT * FROM MyTable" ,MyConn,adOpenDynamic,adLock"
Pessimistic,adCMDText
RS.Close
MyConn.Close
%>
```

۱۲-۳ متدهای پیشرفته جهت کار با مجموعه رکوردها :

تا این مرحله شما تنها چگونگی استفاده از SQL برای تغییر رکوردها ی مجموعه رکورد را فرا گرفتید . شما می توانید ، شماری از متدها را برای تغییر رکوردهای مجموعه رکورد به صورت های مختلفی بکار بگیرید. در لیست زیر بطور خلاصه هر متد تشریح شده است:

AddNew : یک رکورد جدید به مجموعه رکورد اضافه می نماید .

CancleBatch : هنگامی که یک مجموعه رکورد در حالت به هنگام سازی دسته ای باشد، عمل به هنگام سازی را لغو می کند.

CancelUpdate : هر تغییر داده شده در رکورد جاری را قبل از اینکه متد Up-date در خواست گردد ، لغو می نماید .

Delete : یک رکورد از یک مجموعه رکورد راحذف می نماید .

Update : هر گونه تغییر ایجاد شده در رکورد جاری را ذخیره می نماید .

UpdateBatch : تمام تغییرات ایجاد شده در یک یا چند رکورد را اگر مجموعه رکورد در حالت به هنگام سازی دسته ای باشد، ذخیره می نماید . به عنوان مثال شما می توانید با استفاده از متد AddNew یک شاخه جدید رکورد به مجموعه رکورد باز اضافه نمائید . مانند مثال زیر :

```
<!--# INCLUDE VIRTUAL = "ADOVBS.inc" -->
<%
Set MyConn=Server.CreateObject("ADODB.Connection")
Set RS=Server.CreateObject("ADODB.Recordset")
MyConn.Open " FILEDSN =d:\Program Files\Common Files\ODBC\Data  Sources \
MyData.dsn"
RS.Open  "SELECT MyColumn MyConn,adOpenDynamic,adLockPessimistic,adCMDText
FROM MyTable"
RS.AddNew
Rs (" MyColumn ") ="A new column"
RS.Update
RS.Close
MyConn.Close
%>
```

در این اسکریپت برای ایجاد یک رکورد جدید از متد AddNew استفاده گردیده است . سپس به فیلد MyColumn از رکورد جدید مقدار Anewcolumn داده می شود. سرانجام متد Update برای ذخیره رکورد جدید مورد استفاده قرار می گیرد. برای استفاده چنین متدهایی مجموعه رکورد باید با یکی از انواع قفل به غیر از

فقط خواندنی باز گردد. بجای استفاده از متد AddNew برای اضافه نمودن یک رکورد جدید بـه جـدول، شـما مـی توانید از متد INSERT استفاده نمایید . به طور معمول بهتر است که بجای اسـتفاده از متدهـای توصیـف شـدهٔ بـالا از SQL استفاده کنید چرا که SQL دارای انعطاف پذیری بیشتری است.

۱۲-۳-۱ هدایت یک مجموعه رکورد :

شیء Rescordset شامل متدهای متعددی جـهت حرکـت میـان رکوردهـای یـک مجموعـه رکـورد اسـت . بسیاری از این متدها فقط می توانند زمانی مورد استفاده قرار بگیرند که یک مجموعه رکورد با نوع مشخصی از مکان نما باز شود. لیست زیرحاوی تعدادی از این متدها و توضیح وظایف آنها است :

تعداد رکوردها Move : مکان نما را به تعداد مشخصی دریک مجموعه رکورد به جلو ویا عقب می فرستد .

MoveFirst : مکان نما را به اولین رکورد در مجموعه رکورد انتقال می دهد .

MoveNext : مکان نما را به رکورد بعدی در مجموعه رکورد انتقال می دهد .

MovePrevious : مکان نما را به رکورد قبلی در مجموعه رکورد انتقال می دهد .

MoveLast : مکان نما را به آخرین رکورد در مجموعه رکورد انتقال می دهد .

همچنین یک شیء Recordset دارای خاصیت های متعددی است که برای هدایت رکوردها بسیار مفید هستند . این خصوصیات به مکان نمای خاصی نیاز دارند که در زیر آمده است:

AbsolutePosition : برای تنظیم یا خواندن موقعیت ترتیبی رکورد جاری در مجموعه رکورد مورد استفاده قرار می گیرد .

BOF : نشانگر موقعیت جاری رکورد قبل از اولین رکورد در مجموعه رکورد است .

EOF : نشانگر موقعت جاری رکورد پس از آخرین رکورد در مجموعه رکورد است .

RecordCount : نشانگر تعداد رکوردها در مجموعه رکورد است .

برای مثال تصور نمایید شما می خواهید رکوردها را در یک مجموعه رکورد به سمت عقب برانید شما می توانیـد این عمل را با استفاده از متدهای MoveLast و MovePrevious و خاصیت BOF انجام دهید. Asp زیر چگونگـی عمل را به شما نشان می دهد :

```
< HTML >
< HEAD > < TITLE > Backwards Recordset </ TITLE > < /HEAD >
< BODY >
<!--# INCLUDE VIRTUAL = "ADOVBS.inc" -->
<%
Set MyConn =Server.CreateObject("ADODB.Connection")
Set RS =Server.CreateObject("ADODB.Recordset")
MyConn.Open " FILEDSN =d:\Program Files\Common Files\ODBC\Data Sources\ My-
" Data.dsn
RS.Open " SELECT MyColumn FROM MyTable ",MyConn,adOpenStatic
RS.MoveLast
WHILE NOT RS.BOF
```

```
Response.Write (" < BR > " & RS (" MyColumn"))
RS.MovePrevious
WEND
RS.Close
MyConn.Close
%>
</ BODY>

</ HTML>
```

در این مثال مجموعه رکورد با یک مکان نمای ثابت باز شده است . ایـن استراتژی هـم بـرای متـد MoveLast وهم متد Moveprevious کارائی دارد . همینکه مجموعه رکورد باز می شود ، تمام رکوردهای آن نمایش داده می شود تا اینکه دوباره به اول مجموعه رکورد برگردد. خاصیت BOF برای مشخص کـردن زمـانی کـه ایـن اتفـاق مـی افتد، مورد استفاده قرار می گیرد. شما می توانید از این متدها برای حرکت به عقب در میان مجموعه رکـورد اسـتفاده کنید . هر چند درک اینکه چرا این عمل در این مرحله از کار مورد نیاز است ،کمی مشکل اسـت. اسـتفاده از خـود زبان SQL برای سامان دادن به نتایج یک درخواست بسیار موثرتر است. هر وقت که بتوانید ، باید رکوردهایتان را بـا استفاده از شرط ORDER BY متعلق به SQL، سامان بدهید.

۱۲-۳-۲ دریافت یک Recordcount :

جهت تعیین تعداد رکوردهـای یـک مجموعـه رکـورد مـی توانیـد از خـاصیت RecordCount متعلـق بـه شـیء Record-Set ، استفاده نمایید . البته در استفاده از این خاصیت باید خویشتنداری کنید. با آنکـه خـاصیت مذکـوردر بسیاری از مواقع بسیار موثر است ، شما نمی توانید آنرا با مجموعه رکوردی که ازیک مکـان نمـای Forward-Only استفاده می کند، بکار ببرید. جهت این امر شما باید یک مکان نما با تاثیر کمتر را باز کنید به مثال زیر توجه نمایید :

```
<!--#INCLUDE VIRTUAL = "ADOVBS.inc" -->
<%
Set MyConn=Server.CreateObject("ADODB.Connection")
Set RS=Server.CreateObject("ADODB.RecordSet")
MyConn.Open "FILEDSN=d:\Program Files\Common Files\ODBC\DataSources\MyData .dsn
 SELEC MyColumn FROM  MyTable " ,MyConn,adOpenStatic RS.Open"
Response.Write(RS.RecordCount)
RS.Close
MyConn.Close
%>
```

خروجی این اسکریپت تعـداد رکوردهـا ی جدولـی بـه نـام MyTable اسـت .خـاصیت RecordCount جـهت بازگرداندن این تعداد مورد استفاده قرار می گیرد. برای اینکـه خـاصیت RecordCount مـورد اسـتفاده قـرار گـیرد مجموعه رکورد با مکان نما ی ثابت باز شده است.

به طور معمول ، به شماره رکورد تنهابرای تعیین آخرین رکورد متناسب با ضابطه مشـخص شـده نیـازدارید. مثـلاً شاید بخواهید بررسی کنید که چه کسی کلمهٔ عبور معتبر را وارد کرده است. در این حالت ممکن اسـت درخواسـت یک جدول کلمهٔ عبوررا بدهید و از خاصیت Recordcount برای بررسـی وجودکلمـهٔ عبـور ، اسـتفاده کنیـد . اگـر

RecordCount بزرگتر از صفر باشد، کلمهٔ عبورموجود بوده ودر غیر اینصورت کلمه عبور موجود نبوده و صحیح نمی باشد.

گر چه خیلی بهتر است که برای بررسی اینکه یک درخواست نتیجه ای را بر می گرداند یا نه از خاصیت EOF استفاده کنید. این خاصیت را موقع استفاده از مجموعه رکوردی با مکان نمای Forward-Only، بکار ببرید. به مثالی در این مورد توجه نمائید:

```
<!--#INCLUDE VIRTUAL = "ADOVBS.inc" -->
<%
Set MyConn = Server.CreateObject("ADODB.Connection")
Set RS = Server.CreateObject(" ADODB.RecordSet")
MyConn.Open " FILEDSN =d:\Program Files\Common Files\ODBC\
Data Sources\ MyData.dsn"
RS.Open "SELECT * FROM Password _Table WHERE Password =" & Request.Form("
Password "),MyConn
IF RS.EOF THEN
Response.Write (" The Password you entered is invalid.")
ELSE
Response.Write (" Welcome to our web site!")
END IF
RS.Close
MyConn.Close
%>
```

در این مثال خاصیت EOF برای بررسی اینکه نتیجه ای برای درخواست آمده است، استفاده شده است. اگر خاصیت EOF دارای مقدار TRUE باشد کلمه عبوری که کاربر وارد نموده در جدول کلمه رمز وجود ندارد.

در برخی مواقع شما واقعا نیاز به دریافت شمارهٔ رکورد دارید. برای مثال ممکن است شما بخواهید تعداد کل کاربران ثبت شده در سایت وب خود را نمایش دهید . از آنجایی که باید در استفاده از خاصیت RecordCount پرهیز کنید. به جای آن می توانید از یک درخواس(*) COUNT متعلق به SQL استفاده نمایید . به مثال زیر توجه نمایید :

```
<%
Set MyConn=Server.CreateObject("ADODB.Connection")
Set RS=Server.CreateObject("ADODB.RecordSet")
MyConn.Open" FILEDSN=d:\Program Files\Common Files\ODBC
Data Sources\MyData.dsn"
RS.Open " SELECT COUNT(*) MyCount FROM Password _Table ", MyConn
%>
There are < % = RS (" MyCount ") % > registered users at this web site
<%
RS.Close
MyConn.Close
%>
```

به نام مستعار MyCount که در این درخواست SQL استفاده شده است توجه کنید. با فراهم آوردن تابع کلی Count (*) توسط یک نام ، شما می توانید آن نام را هنگام خروج نتایج درخواست استفاده کنید.

۱۲-۳-۳ صفحه بندی یک مجموعه رکورد :

تصور نمایید که می خواهید لیستی از محصولات فروشی در سایت وب خود را نمایش دهید . ممکن است شما هزاران قلم جنس بفروشید،در نتیجه شاید نخواهید تمام اقلام را در یک ASP منفرد نشان دهید. بهتر است اگر می توانید به کاربر اجازه دهید که لیست اقلام را صفحه به صفحه ببینید. (ورق بزنند)

شیء Recordset برای این منظورنیز سه خاصیت دارد. شما می توانید از این خاصیت ها برای تقسیم نمودن رکوردهای مجموعه رکورد به صفحات منطقی استفاده نمایید . با تقسیم رکوردها در مجموعه رکورد به صفحات جداگانه شما می توانید فقط قسمتی از مجموعه رکورد را در آن واحد نمایش دهید. لیستی از این خاصیتها در بخش زیر بیان شده است:

Absolute : صفحهٔ جاری از رکورد ها را مشخص می کند .

صفحهCount : تعداد صفحات که از یک Recordset بوجود می آید ، را بر می گرداند.

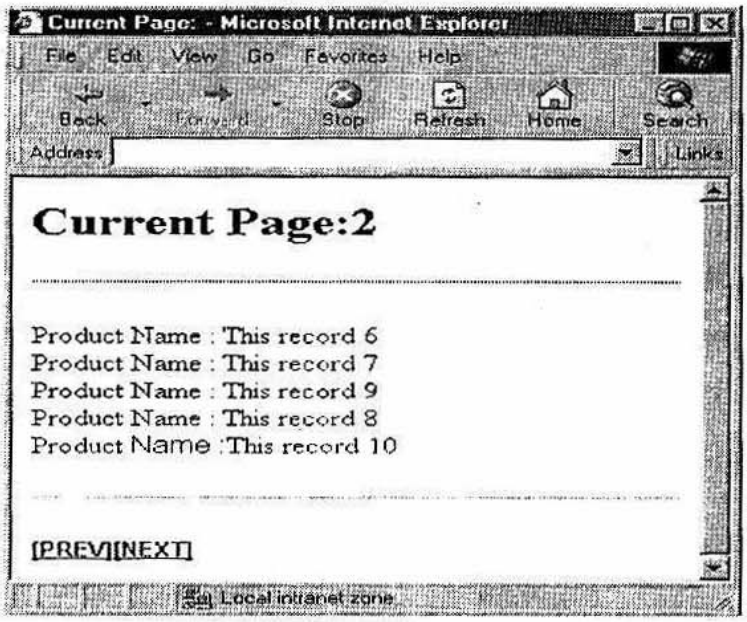
PageSize : تعداد رکوردها را در یک تک صفحه مشخص می نماید . و مقدار پیش فرض برای آن ۱۰ است.

برای تقسیم یک مجموعه رکورد به صفحات چند تایی ، از خاصیت PageSize جهت مشخص نمودن تعداد رکوردها در یک صفحهٔ منفرد استفاده نمائید. می‌توانید از خاصیت AbsolutePage جهت حرکت دادن رکوردهای صفحه مشخصی از رکوردها استفاده کنید. سرانجام خاصیت Page Count می تواند برای برگرداندن تعداد کل صفحات مورد استفاده قرار گیرد. لیست زیر چگونگی استفاده از این ویژگیها را نشان می دهد : (رجوع شود به شکل ۱۲-۲)

```
< HTML >
< HEAD > < TITLE > Recordset With Pages < / TITLE > < / HEAD >
< BODY >
<!--#INCLUDE VIRTUAL= "ADOVBS.inc"-->
<%
'Figure out the current page
IF Request.QueryString (" MOVE") = " NEXT " THEN
Session ("CurrentPage ") = Session (" CurrentPage ") + 1
END IF
IF Request.QueryString (" MOVE") = " PREV " THEN
Session ("CurrentPage ") = Session (" CurrentPage " ) - 1
END IF
IF Session (" CurrentPage ") = "  " THEN
Session (" CurrentPage ") = 1
END IF
%>
```

```
< H 1 > CurrentPage:< % = Session (" CurrentPage " ) % > < / H1 >

< HR >

<%
'Open a Connection and Recordset
Set MyConn = Server.CreateObject (" ADODB.Connection")
Set RS = Server.CreatObject (" ADODB.RecordSet")
MyConn.Open " FILEDSN =d:\Program Files\Common Files\ODBC\
Data Sources\MyData.dsn
'Retrieve the list of products
RS.Open " SELECT ProductName FROM Products " ,MyConn, adOpenStatic
'Set the number of records in a page
RS.PageSize = 5
'Set  the current page
RS.AbsolutePage = Session (" CurrentPage")
'Show the records for the current page
WHILE NOT RS.EOF AND NumRows <% RS.PageSize>
< BR > Product Name : < % = RS (" ProductName ")%>
<%
RS.MoveNext
NumRows = NumRows + 1
WEND
%>
< HR >

<% IF Session ("CurrentPage ") > 1 THEN %>
< A HREF = " Pages.asp? MOVE = PREV " > [ PREV ] < / A >
<% END IF %>
IF Session (" CurrentPage " ) < RS.PageCount  THEN
< A HREF = " page.asp ? MOVE = NEXT " > [ NEXT ] < /A >

<% END IF %>
<%
RS.Close
MyConn.Close
%>
</BODY>
</HTML>
```

این ASP رکورد های جدول Products را پنج تا پنج تا نمایش می دهد . یک متغیر Session با نام CurrentPage برای نگهداری جای صفحهٔ جاری در یک مجموعه رکورد مورد استفاده قرار می گیرد . زمانیکه کاربر روی دکمهٔ Next کلیک می کند صفحه بعدی از رکورد ها نمایش داده می شود و زمانیکه روی دکمهٔ PREV کلیک می کند صفحه قبلی نمایش داده می شود .



شکل۱۳-۲ صفحه بندی از طریق رکوردهای درون مجموعه رکورد

دریافت رکوردها از آرایه :

در حالتهای خاص ، شما نیازمند دریافت رکوردهای یک مجموعه رکورد از آرایه هستید. به طور مثال ، اگر نیازمند به محافظت از داده های ارائه شده توسط یک مجموعهٔ رکورد باشید، اما نخواهید رکوردها را درون خود مجموعه رکورد تغییر دهید ، می توانید رکورد را از آرایه دریافت کنید.

برای تخصیص رکوردهای یک مجموعه رکورد به یک آرایه ، باید از متد ( )GetRows متعلق به شیء Recordset استفاده کنید. در زیر مثالی از چگونگی آن آمده است :

```
<%
Set My Conn = Server.CreateObject ("ADODB.Connection ")
Set My Conn =Server.Create Object ("ADODB.Recordset ")
MyConn.Open " FILEDSN = d:\ Program Files\ODBC\ Data Source\ MyData.dsn "
RS.Open "SELECT MyFirst Col ، MySecond Col FROM MyTable", MyConn

MyArray=RS.Get Rows ( )
RS.Close
MyConn.Close
%>
```

در این اسکریپت ، همهٔ رکوردهای گنجانده شده در مجموعه رکورد RS به آرایه ای با نام MyArray تخصیص داده می شوند. این آرایه به طور خودکار توسط متد ( ) GetRows ، رویه ایجاد می شود. متد ( ) GetRows یک آرایهٔ دو بعدی ایجاد می کند. اولین قسمت از اسکریپت فیلد را شناسائی می کند و قسمت دوم اسکریپت شماره رکورد را نشان می دهد.اسکریپت زیر می تواند برای نمایش تمامی محتوای آرایه مورد استفاده قرار گیرد.

```
<%
For is To UBOND (MyArray, 2)
%>
<BR> First Column:<% My Array( 0 , i)%>
<BR> Secound Column:<% MyArray(1، i )%>

NEXT
%>
```

تابع VBOUND از وی بی اسکریپت برای تعیین اندازهٔ بعد دوم آرایه مورد استفاده قرار می گیرد. حلقهٔ FOR.....NEXT نیز به اندازهٔ اجزاء آرایه تکرار می شود. این آرایه مجموعه رکوردی با دو ستون را نشان می دهد. وقتی اولین اندیس آرایه مقدار ۱ دارد،ستون دوم نشان داده می شود.

۱۲-۳-٤ مشخص نمودن بیشترین مقدار یک Recordset :

تصور کنید که می خواهید تنها ۱۰ پیام آخری ارسال شده به سایت وب خود را نمایش دهید می خواهید تنها سه پیوند و نه بیشترفراهم کنید. چگونه می توانید رکوردهایی که از یک مجموعه رکورد دریافت می کنید را محدود کنید ؟ شیء Recordset برای این منظور یک خاصیت دارد. شما می توانید تعداد رکوردهای دریافت شده از یک مجموعه رکورد متعلق به درخواست پایگاه داده را با استفاده از خاصیت MaxRecords محدود کنید. مانند مثال زیر:

```
Set MyConn=Server.CreateObject (" ADODB.Connetion")
Set RS=Server.CreateObject ("ADODB.RecordSet")
MyConn.Open " FILEDSN =d:\Program Files\Common Files\ODBC\Data
Sources\MyData.dsn"
RS.MaxRecords = 10
RS.Open "SELECT MyColumn FROM MyTable", MyConn
WHILE NOT RS.EOF
Response.Write (" < BR > " & RS (" MyColumn")
RS.MoveNext
WEND
RS.Close
MyConn.Close
%>
```

در این اسکریپت به خاصیت MaxRecords مقدار ۱۰ می دهیم. با این کار زمانیکه مجموعه رکورد RSباز می شود بیشتر از ۱۰رکورد دریافت نخواهد شد. حتی اگر جدول حاوی ۱۰۰۰۰رکورد هم باشد فقط ده عدد از آنها دریافت خواهد شد.هنگامیکه از خاصیت Max Records استفاده می کنید باید قبل از باز کردن مجموعه رکورد، آنرا تنظیم کنید. پس از اینکه مجموعه رکورد باز شد خاصیت آن فقط خواندنی خواهد بود.

فصل سیزدهم :

کار با شیء Command

این فصل دربرگیرندهٔ چگونگی استفاده از شیء Command است. در ابتدا شما چگونگی استفاده از شیء Command برای اجرای رویه های ذخیره شده SQL و استفاده از پارامترها را، یاد می گیرید. باقیماندهٔ این فصل به دو برنامهٔ کاربردی نمونهٔ ADO اختصاص یافته است. در مقدمه شما چگونگی ایجاد یک صفحهٔ پیشرفته بازتاب برای سایت وب خود را فرا می گیرید. سپس چگونگی ایجاد یک سیستم محافظت کلمهٔ عبور را فرا می گیرید.

۱۳-۱ کار برد شی ء Command:

شی ء Command نشان دهندهٔ یک دستور است. فصل ۱۱ و ۱۲ به ترتیب چگونگی استفاده از متد Execute از شیء Connection و متد Open از شیء Recordest را برای اجراء یک رشته دستور را نشان داد. به دو مثال زیر توجه کنید :

```
RS.Open " SELECT * FROM MyTable ", MyConn
MyConn.Execute "UPDATE MyTable SET Mycolum =' Hello'"
```

هردو مثال بالا از یک رشته دستور SQL استفاده کرده اند. در مثال اول رشته دستور برای باز کردن مجموعه رکورد استفاده شده است و در مثال دوم رشته دستور اجرا می گردد. به جای استفاده ازیک رشته دستورمی توانید از شی ء Commandاستفاده نمائید. شیء Command می تواند برای نمایش صریح یک دستور بکار رود. می توانید از یک نمونه شیء Command برای بازگرداندن یک مجموعه رکورد یا اجرای دستور SQL ای که مجموعه رکوردی را باز نمی گرداند استفاده کنید .

درمثال زیر یک نمونه شی ء Command ایجاد شده است. سپس خاصیت ActiveConnection ، Command را با یک اتصال باز مرتبط می کند.(از آنجاییکه شما یک شیء را تخصیص می دهید،این کار توسط عبارت Set انجام می شود.) خاصیت CommandText، مشخص می کند کدام جمله SQL باید اجرا شود. خاصیت CommandType نوع دستور رانشان می دهد. سرانجام متد Execute برای اجرای Command فراخوانده می شود. مثال:

```
<!--#INCLUDE VIRTUAL="ADOVBS.inc"-- >
<%
Set MyConn=Server.CreateObject ("ADOBD.Connection")
Set MyCommand =Server.CreateObject ("ADODB.Command")
MyConn.Open"FILEDSN=d:\ProgramFiles\CommonFiles\ODBC\
DataSources\MyData.dsn
Set MyCommand.ActiveConnection=MyConn
MyCommand.CommandType =adCMDText
MyCommand.CommandText="SELECT *FROM MyTable"
Set RS= MyCommand.Execute( )

RS.Close
```

```
MyConn.Close
%>
```

در مثال فوق از شیء Command برای بازگرداندن یک مجموعه رکورد استفاده می شود. توجه کنید پرانتزها به دلیل اینکه متد برای بازگرداندن نتایج استفاده شده بکار رفته اند. پس از این که یک نمونه Recordset را توسط شیء Command ایجاد نمودید می توانید آنرا به طرق معمول دستکاری کنید. همچنین می توانید از شیء Command توسط یک مجموعه رکورد که از قبل موجود بوده استفاده نمائید مانند زیر :

```
<!--#INCLUDE VIRTUAL="ADOVBS.inc"-- >
<%
Set MyConn=Server.CreateObject ("ADOBD.Connection")
Set MyCommand=Server.CreateObject ("ADODB.Command ")
MyConn.Open"FILEDSN=d:\ProgramFiles\CommonFiles\ODBC\
DataSources\MyData.dsn"
Set MyCommand.Connection=MyConn
MyCommand.CommandText="SELECT *FROM MyTable"
MyCommand.CommandType =adCMDText
RS.Open MyCommand,adOpenStatic,adLockOptimistic
RS.Close
MyConn.Close
%>
```

مزیت عبور دادن یک شیء Command به سوی یک مجموعه رکورد از پیش موجود، این است که می توانید نوع قفل و مکان نمای مجموعه رکورد را تعیین کنید . در این مثال شیءCommand جهت باز نمودن مجموعه رکوردی که از مکان نمای ثابت و قفل Optimistic استفاده کرده ، بکار رفته است. توجه داشته باشید که برای این قسمت متدهای مختلفی وجود دارد اما متد شیء Command از همه بهتر است.

باید بدانید که هنگام استفاده از متد Open بوسیلهٔ شیء Command نباید مبداء را در شیء Connection وارد کنید:چرا که جای آن توسط شیء Command تعیین می شود. این مثالها شرح می دهد که شما چگونه می توانید از شیء Command استفاده کنید. استفاده از شیء Command یک مزیت اصلی دارد، اینکه شما می توانید شیء Command را به همراه یک رویهٔ ذخیره شدهٔ SQL بکاربرید.

نکته:در این قسمت یک سری از دلایل مختلف استفاده از رویه های ذخیره شدهٔ SQL وجود دارد:

- رویهٔ ذخیره شدهٔ SQL عموماً خیلی سریعتر از دستورات SQL اجرا می شوند.
- شما می توانید همان رویهٔ ذخیره شده را از چندین ASP فرا بخوانید .
- همچنین می توانید از رویه های ذخیره شده تمام پارامترهای یک تراکنش SQL - Transact را استفاده نمایید. شما می توانید متغیرها و شروط را بکار برید .یعنی می توانید یک رویه ذخیره شده را برای ایجاد بسیاری از درخواستها استفاده کنید و به پایگاه داده از متدهای مختلف به هنگام کنید.

۱۳-۱-۱ استفاده از شیء Command برای صدا کردن یک رویهٔ ذخیره شده:

فرض کنید که می خواهید همهٔ رکوردها را از جدول MyTabel دریافت ودر یک ASP ببینید. از سوی دیگر فرض کنید که می خواهید رکوردها را از این جدول از موثرترین راه ممکن دریافت کنید. دراین مورد شما باید از

رویهٔ ذخیره شده استفاده نمائید. برای ایجاد یک رویهٔ ذخیره شده ، ISQL/w از گروه برنامه Microsoft SQLServerرا اجرا وسپس متن زیر را در پنچره درخواست وارد کنید :

```
CREATE PROCEDURE sp_ myproc
SELECT   * FROM MyTable
```

دکمهٔ Execute Query را کلیک کنید تا رویهٔ ذخیره شده ایجاد شود. رویهٔ ذخیره شده جدید sp_myproc نام دارد. sp_myproc را از ASP فراخوانی کنید ، می توانید از نمونهٔ شیء Command استفاده نمائید. در زیر مثالی از این کار آمده است:

```
<!--#INCLUDE VIRTUAL="ADOVBS.inc"-- >
<%
Set MyConn=Server.CreateObject("ADODB.Connection")
Set MyConn=Server.CreateObject("ADODB.Command")
MyConn.Open  "FILEDSN =d:\Program Files\Comman Files\ODBC \DataSources
\MyData.dsn"
Set MyCommand.ActiveConection=MyConn
MyCommand.CommandType=adCMDStoredProc
MyCommand.CommandText ="sp_myproc"
Set RS=MyCommand.Execute()
WHILE NOT RS.EOF
Response.Write ("<BR> " & RS ("MyColum"))
RS.MoveNext
WEND
RS.Close
MyConn.Close
%>
```

این اسکریپت تمام رکوردهای جدولی به نام MyTable را نشان می دهد. رکوردها توسط فراخوانی رویهٔ ذخیره شده sp_myproc باز یابی می شوند . زمانیکه از یک شیء Command برای فراخوانی رویهٔ ذخیره شده استفاده می نمایید باید خاصیت CommandType به adCMDStroredProc مقداردهی شود.

۱۳-۱-۲ استفاده از مقادیر وضعیت بازگشت توسط شیء Command :

از شیءCommand برای دریافت مقدار وضعیت بازگشت ازیک رویهٔ ذخیره شده استفاده نمائید. برای مثال فرض کنید شمارهٔ تعداد رکوردهای یک جدول را می خواهید . مؤثر ترین راه انجام این کار ، ایجاد یک رویهٔ ذخیره شده است، مانند مثال زیر :

```
CREATE  PROCEDURE  sp_Count MyTable  AS
(RETURN (SELECT COUNT (*)  FROM  MyTable
```

این رویه ذخیره شده تعداد رکوردهای جدول MyTable را باز می گرداند.تابع ( ) Count از SQL رکوردهارا درجدول می شمارد ، و عبارت Return این شماره را باز می گرداند. برای دریافت مقدار وضعیت برگشت از یک رویهٔ ذخیره شده ، شما باید یک پارامتر برای شیءCommand ایجاد کنید. شی ءCommand مجموعه ای به نام Parameters دارد . مجموعهٔ Parameters مجموعه ای از اشیاء Parameter است . شما پارامتری را با استفاده از متد ( ) CreateParameter از شی ء Command ایجاد می کنید سپس ، متد Append از شی ء Commandرا برای ضمیمه کردن پارامتر به مجموعهٔ Parameters از شی ء Command استفاده می کنید:

```
<!--#INCLUDE VIRTUAL="ADOVBS.inc"-- >
<%
Set MyConn=Server.CreateObject(" ADODB.Connection")
Set MyCommand=Server.CreateObject("ADODB.Command")
MyConn.Open  "FILEDSN=d:\Program Files\Common Files\ODBC\Data Sources
\MyData.dsn"
MyCommand.ActiveConnection=MyConn
MyCommand.CommandType=adCMDStoredproc
MyCommand.CommandText="sp_CountMyTable"
Set MyParam=MyCommand.CreateParameter("RetBai"adInteger, adParamReturnValue)
MyCommand.ParametersAppend MyParam
MyCommand.Exeute
/>

There are <%=MyCommand("RetVal")%> records in MyTable.
<%
MyConn.Close
%>
```

این اسکریپت، یک شیء Parameterجدید را با استفاده از متد CreateParameter()ایجاد می کند. این متد سه آرگومان به قرار زیر دارد:

- آرگومان اول نامی را برای پارامترجدید مشخص می نماید.
- آرگومان دوم نوع داده را مشخص می نماید.
- سرانجام آرگومان آخر نوع پارامتر را تعیین می کند . در این مثال ثابت adParamReturnValue نشان می دهد که این پارامتر از نوع بازگشتی است.

پس از اینکه یک پارامتر جدید ایجاد شد، باید به مجموعهٔ Parameters از شیء Command ضمیمه شود. متد append برای اضافه کردن یک پارامتر جدید به این مجموعه استفاده می گردد. پس از اینکه Command اجرا شد ، مقدار پارامتر دریافت می شود . از آنجائیکه پارامتر عضو مجموعه Parameters از شیء Command است ، پارامتر مقدار می تواند توسط MyCommand("RetVal") برگردانده شود . هم چنین این مقدار را می توانید توسط هر یک از عبارات زیرنیز دریافت کنید:

```
MyCommand ("Reval")
MyCommand (0)
MyCommand.Parameters ("Reval")
MyCommand.Parameters (0)
MyCommand.Parameters. Item ("Reval")
MyCommand.Parameters. Item(0)
```

تمام این متدها کاردریافت مقدار پارامتر را انجام می دهند،چرا که یک پارامتر قسمتی از مجموعه Parameters از شیء Command است.

۱۳-۱-۳ استفاده از پارامترهای خروجی با شیء Command:

درمثال بخش قبل چگونگی دریافت یک وضعیت بازگشتی نشان داده شد.تصور کنید که جدولی به نام WebUsers دارید که شامل لیستی از اسامی کاربران ثبت شده در سایت وب شما است. این جدول یک ستون تک

واحدی به نامUserName دارد. حالا تصور کنید که می خواهید بلندترین و کوتاهترین کلمه از نظر تعداد حروف الفبا را به جدول باز گردانید . برای این کار می توانید از رویهٔ ذخیره شده زیر استفاده نمائید:

```
CRATE PROCEDURE sp _HighAmdLow
Highuser VARCHAR(30) OUTPUT, @LowUser VARCHAR(30) OUTPUT AS @)
SELECT @Highuser = MAX(UserName) FROM WebUsers
SELECT @LowUser = MIN(UserName) FROM WebUsers
```

این رویه ذخیره شده دو پارامتر خروجی به نامهای @LowUser و@HighUser دارد. برای صدا زدن این رویهٔ ذخیره شده در یک ASP می توانید از اسکریپت زیر استفاده نمائید:

```
<!--#INCLUDE VIRTUAL="ADOVBS.inc"-- >
<%
Set MyConn =Server.CreateObject ("ADODB.Connection")
Set MyCommand =Server.CreateObject ("ADODB.Command")
MyConn.Open"FILEDSN=d:\ProgramFiles\CommonFiles\ODBC\DataSources\MyData dsn"
Set MyCommand.ActiveConnection=MyConn
MyCommand.CommandType= adCMDStoredProc
MyCommand.CommandText="sp_HighandLow"
Set MyFirstParam=MyCommand.CreateParameter("HighUser",asVarChar,
as ParamOutput,30)
MyCommand.Parameters.Append MyFirstParam
MyCommand.Parmeters.Append MySecondParpm
MyCommand.Execute
%>
<P> The person With the alphabetically highes name is
<%=MyCommand("HighUser") %>
<P> Theperson with the alphabetically lowest name is
<% = MyCommand("LowUser ")%>
<%
MyConn.Close
%>
```

ساختار این اسکریپت بسیار شبیه به متد قبل است . در این اسکریپت دو شیءParameter توسط روش ()CreateParametert ایجاد می شوند. که هر دو با نوع داده ای VARCHAR تولید می گردند. برای نشان دادن این که پارامترها خروجی هستند، ثابت adParamOutPut مورد استفاده قرار می گیرد. سرانجام بیشترین مقدار هر پارامتریعنی مقدار ۳۰، در متد ( )CreateParameter گنجانده می شود. زمانیکه شما پارامترها را با اندازه داده متغییر مانند CHARیا VARCHAR ایجاد نمودید از آرگومان maximum size پشتیبانی می کنید.

۱۳-۱-٤ استفاده از پارامترهای ورودی با شیءCommand :

یک رویهٔ ذخیره شدهSQL می تواند پارامترهای ورودی را بپذیرد. پارامترهای ورودی شما را قادر می سازد که اطلاعات را به یک رویهٔ ذخیره کننده انتقال دهید.

برای مثال تصور کنید که می خواهید یک جدول شامل نام کاربران و کلمه های عبور داشته باشید. حالا فرض کنید که می خواهید یک رویهٔ ذخیره شده برای بررسی کلمه های عبور داشته باشید. شما می توانید بوسیله رویهٔ ذخیره شده زیر یک کلمه عبور معتبر متعلق به یک کاربر را بررسی کنید:

```
CREATE PROCEDURE sp_CheckPass
```

```
CHKNameVARCHAR(30) ,@CHKPass VARCHAR(30),@ISValid CHAR (4) OUTPUT)
AS
IF EXISTS(SELECT UserName  FROM WebUsers
WHERE UserName = @CHKName AND UserPass = @CHKPass )

SELECT  @ISValid ="Good"
ELSE
SELECT  @ISValid="Bad"
```

این رویهٔ ذخیره شده پارامترهای ورودی را قبول می کند. پارامتر ورودی @CHKName نـام کـاربر را بـه رویـه انتقال می دهد. پارامتر ورودی @CHKPass یک کلمه عبـور بـه رویـه وارد مـی کنـد. اگر کـاربری بـا کلمـه عبـور مشخصی موجود باشد، پارامتر خروجی مقدارGood برمی گرداند. در غیر این صورت، مقـدار Bad بـرگردانـده مـی شود. متد استفاده یک پارامتر ورودی بسیار شبیه به روش استفاده از پارامترهای خروجی است و تفـاوت آنـها در ایـن است که باید به یک پارامتر ورودی قبل از این که Command اجرا گردد مقدار اختصاص داده شود. مثال:

```
<!--#INCLUDE VIRTUAL="ADOVBS.inc"-- >
<%
Set MyConn=Server.CreateObject ("ADODB.Connection")
Set MyCommand =Server.CreateObject ("ADODB.Command")
MyConn.Open  "FILEDSN =d:\Program Files\CommonFiles
\ODBC\Data Sources \MyData .dsn
Set MyCommand.AcitveConnection=MyConn
MyCommand.CommandType=asCMDStoredProc
MyCommand.CommandText="sp _CheckPass"
Set MyFirstParam=MyCommand.CreateParameter("UserName",adVarChar,
adParamInput ,30)
MyCommand.Parameters.Append MySecondParam
Set MySecondParam=MyCommand.CreateParameter("UserPass",
adVarChar, adParamInput,30)
MyCommand.Parameters.Append MySecondParam
Set MyThirdParam=MyCommand.CreateParameter("RetValue",adChar,adParam
OutPut,4)
MyCommand.Parameters.Append MyThirdParam
MyCommand("UserName")="Bill Gates"
MyCommand("UserPass")= "Billions"
MyCommand.Execute
%>
The Passord is <%= MyCommand ("RetValue")%>.
MyConn.Close
%>
```

در این مثال نام Bill Gates با کلمهٔ عبور Billions برای رویه ذخیره شده ارسال می شود. اگر این ترکیب نـام و کلمه عبوردر جدول WebUsers موجود باشد، کلمه عبور به عنوان Good بازگردانده می شود. در غیر این صورت کلمه عبورBad بازگردانده می شود.

با توجه به این که به هر دو پـارامتر ورودی قبـل از ایـن کـه Command اجـرا شـود، مقـداری اختصـاص یافتـه است؛در این اسکریپت دو پارامتر ورودی با استفاده از ثابت adParamInput ارائه می شوند.

۱۳-۲ دریافت اطلاعات پارامتر:

شاید دریابید که به یک رویهٔ ذخیره شونده احتیاج دارید، اما ندانید که رویه به چه پارامترهایی احتیاج دارد. برای مثال شما ممکن است نوع دادهٔ پارامترها یا سایز آنها را ندانید. شما می توانید برای دریافت اطلاعات در مورد پارامترهای بکار رفته در یک رویه ذخیره شده از اسکریپت زیر استفاده کنید:

```
<!--#INCLUDE VIRTUAL="ADOVBS.inc"-- >
<%
Set MyConn =Server.CreateObject ("ADODB.Connection")
Set MyCommand=Server.CreateObject ("ADODB.Command")
MyConn.Open  "FILEDSN = d:\ Program Files\Common Files\ODBC
\DataSources \MyData.dsn
Set MyCommand.ActiveConnection=MyConn
MyCommand.CommandType=adCMDStoredProc
MyCommand.CommandText= "sp _myproc"
MyCommand.Parameters.Refresh
%>
<HTML>

<HEAD><TITIE> PARAMETER   INFORMATON><TITIE> </ HEAD>

<BODY>

<TABLE   BORDER=1>

<CAPTION> Parameter  Information</CAPTON>

<TR>

<TH> Parameter Name </TH>

<TH> DataType </TH>

<TH> Direction</TH>

<TH> Size  <TH>

</TR>

<%For Each thing in MyCommand. Parameters%>

<TR>

<TD><%=thing.name %></TD>

<TD><%=thing.type %></TD>

<TD><%=thing.direction %></TD>

<TD><%=thing.size %></TD>

</TR>

<%
Next
MyConn.Close
%>
```

```
</TABLE>
</BODY>
</HTML>
```

این مثال اطلاعات پارامتر برای رویه ای به نام sp_myproc را نمایش می دهد. نـام ، نـوع داده، جـهت و انـدازهٔ پارامتر در یک جدول نشان داده می شود. جهت نشان می دهد که آیـا پـارامتر ، یـک پـارامتر ورودی ، یـک پـارامتر خروجی ویا مقدار وضعیت برگشت است .

مهمترین عبارت در این مثال، MyCommand.Parameters.Refresh است. وقتی این عبارت اجرا می شود ، اطلاعات مربوط به پارامترهای رویه ذخیره شده از پایگاه داده بازیابی می شود.این اسکریپت ثابتها را بر نمی گردانـد، در عوض، مقادیر خام را باز می گرداند. برای تفسیر مقادیر بازگردانده شده توسط این اسکریپت ، شما نیاز به امتحـان فایل همراه ADOVBS دارید. در این فایل مقادیر عددی خام به ثابتهای صحیح پیوند خورده است.

۱۳-۳ برنامهٔ کاربردی نمونه :(یک صفحهFeedback پیشرفته):

اینکه سایت وب شما یک صفحهٔ feedback در خود داشته باشد ایدهٔ خوبی است. این صفحه به کاربرها اجـازهٔ انتقاد و پیشنهاد داده و نقطه نظر خود را بیان می کنند. شما می توانیـد از ایـن اطلاعـات بـرای سـازگـاری هوشـمندانه سایت خود با احتیاجات بینندگان آن اسـتفاده کنیـد. سـاده تریـن ,vروش جـهت ایجـاد ایـن صفحـه اسـتفاده از فـرم HTML ای است که Feedback را توسط یک آدرس پست الکترونیکی می فرستد. برای مثال صفحه HTML زیـر اطلاعات وارد شده به یک فرم را به آدرس پست الکترونیکیwebmaster@yoursite.com می فرستد:

```
<HTML>
<HEAD><TITIE> Feedback</TITIE> </ HEAD>
<BODY>
<H2> Please enter any suggestions for improving
this web site in the form below :</H2>
<FORM ACTION="MAIL TO:webmaster@yoursite.com">
<TEXTAREA NAME="Feedback" COLS=30 ROWS=10
WRAP=VIRTUAL></TEXTAREA>
</FORM>P><INPUT   TYPE= "SUBMIT"   VALUE" Submit Feedback>
</BODY>
</HTML>
```

این روش برای سایتهای وب کوچک خوب کار می کند. همـه Feedback بـه یـک پسـت الکـترونیکی منفـرد فرستاده می شوند. اما اگرسایت وب شما چندین مدیر داشته باشد ، ممکن است کـه بخواهیـد تمـام مدیـران قـادر بـه مشاهدهٔ این خروجیها باشند. همچنین ممکن است که بخواهید هرخروجی را ذخیره کرده یا در آینده تجزیه و تحلیـل کنید. در این خصوص شما باید اطلاعات کاربر را در یک جدول پایگـاه داده ذخیره کنیـد. مطـالب زیـر چگونگـی

استفاده از ADO برای ذخیره کردن و دریافت Feedback کاربر و چگونگی ایجاد یک فرم Feedback پیشرفته را توضیح می دهند:

جدول Feedback : یک جدول پایگاه داده سرویس دهنده SQL برای ذخیره hاطلاعات کاربران مورد استفاده قرار می گیرد.

صفحهٔ Feedback : یک صفحه HTML که اطلاعات کاربران را به سایت وب شما وارد می کنند.

صفحهAcknowledgment: یک Asp است که از کاربران برای ورود اطلاعات و ذخیره آنها در پایگاه داده قدردانی می کند.

صفحهٔ Display: یک Asp که اطلاعات کاربر را توسط دریافت آن از پایگاه داده نمایش می دهد.

۱۳-۳-۱ ایجاد جدول Feedback:

جدول Feedback دارای چهار ستون است : که شامل آدرس پست الکتریکی، آدرس IP، تاریخ Feedback ایجاد شده و مندرجات Feedback است. برای ایجاد این جدول باید ISQL/w از گروه برنامه Microsofot SQL Server را اجرا کنید. سپس متن زیر را در پنجرهٔ درخواست وارد کنید و آن را اجرا نمائید:

```
CREATE TABLE Feedback (Feed_Email VACHAR(50)
Feed_ IPVACHAR(20)
Feed _ Data DATETIME Default GetDate( ).
Feed _ Contents TEXT
```

۱۲-۳-۲ ایجاد صفحهٔ Feedback :

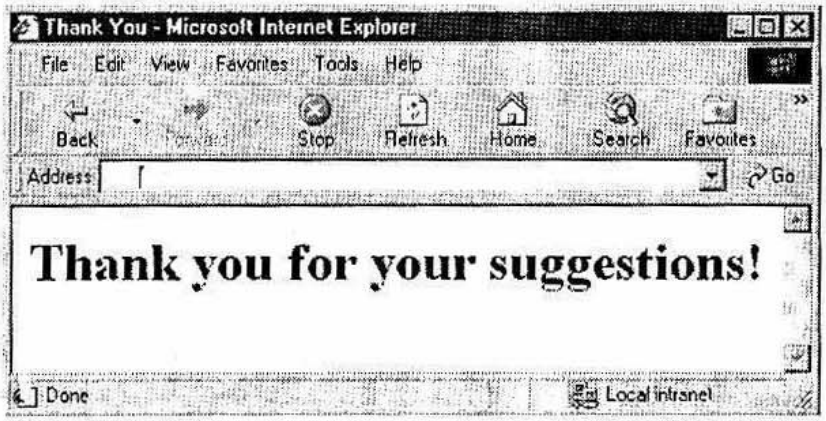
صفحهٔ Feedback یک صفحهٔ معمولیHTML است. شامل یک فیلد متنی که کاربران می توانند آدرس پست الکتریکی خودشان راوارد کنند و یک ناحیهٔ متنی که کاربرها می توانند اطلاعات را وارد کنند.همچنین دارای یک دکمهٔ Submit برای ارسال اطلاعات است. وقتی که یک Feedback ارائه می گردد، کاربران به صفحه Acknowledgment آورده می شوند. لیست زیراسکریپت صفحه Feedback را نشان می دهد:

```
<HTML>

<HEAD><TITLE> Feedback</TITLE></HEAD>

<BODY BGCOLOR=# FFFFFF>

<H2> Please enter any suggestions for improving
this web site in the form below:</H2>
<FORM  METHOD ="POST"  ACTION = "acknowledge.asp">
Please enter your email address:
<BR><INPUT NAME="Email" TYPE="TEXT" SIZE="30" MAXLENGTH ="50">
<P> <TEXTAREANAME="Contents" COLS=30 ROWS=10 WRAP= VIRTUAL>

</TEXTAREA>

<P><INPUT TYPE ="SUBMIT" VALUE="Submit Feedback">
</FORM>

</BODY>

</HTML>
```

شکل ۱۳-۱ صفحهٔ Feedback

۱۳-۳-۳ ایجاد صفحه Acknowledgment :

این صفحه دو منظور دارد. ابتدا جهت تشکر از کاربر برای آماده کردن Feedback استفاده می شود در مرحلهٔ بعد برای ذخیرهٔ واقعی اطلاعات در جدول Feedback استفاده می شود. اطلاعات توسط جملهٔ INSERT از SQL به این جدول اضافه می شود. همانطور که در مثال زیر مشاهده می شود:



شکل ۱۳-۲ صفحهٔ Acknowledge

```
<%
 'Retrieve form fields into variables
Email=Replace(Request.Form("Email") ، " " " ، " " ")
Contents=Replace(Request.Form("Contents") ، " " " ، " " ")
 'Check for empty content
```

```
IF Email="  " THEN  Email= "Unknown"
IF Contents  =" " THEN Contents =" None"
 'Grab the user's IP address

UserIP=Request.ServerVariables("REMOTE_ ADDR")
'Create the SQL command string
MySQL ="INSERT Feedback (Feed _ Email , Feed _ IP ,Feed _ Contents)"
VALUES (' " &Email& " ' , ' "  &UserIP& " ' , ' " &Contents&"')
'Insert the form data into  the Feedback table
Set MyConn = Server.CreateObject ("ADODB.Connection")
MyConn.Open  "FILEDSN= d:\Program Files\Common Files\
ODBC\Data Sources \ MyData.dsn
MyConn.Execute MySQL
%>
<HTML>

<HEAD><TITLE> Thank You  </TITLE></HEAD>

<BODY>

<H2> Thank you for your suggestions! </H2>

</BODY>

</HTML>
```

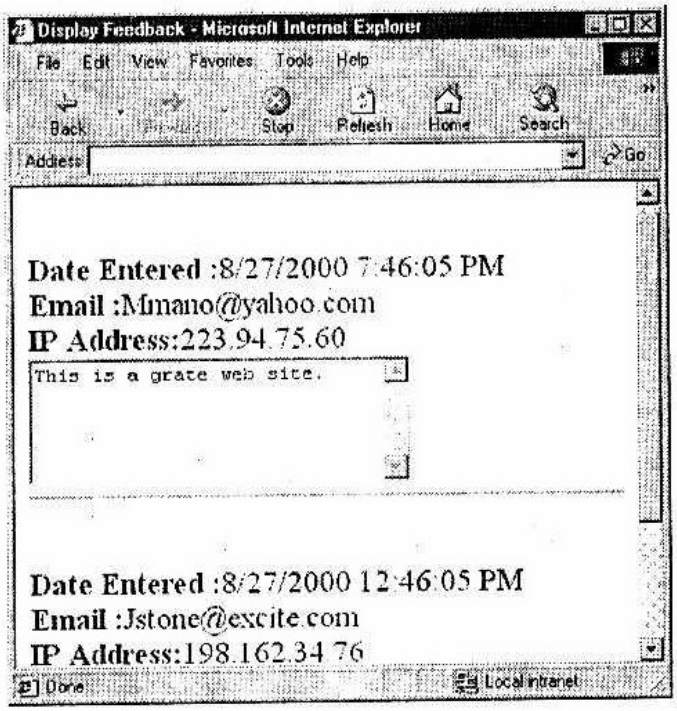
۱۳-۳-٤ ایجاد صفحه Display :

این صفحه برای نمایش اطلاعاتی که کاربران وارد می کنند مورد استفاده قرار می گیرد. اطلاعات از جدول Feedback دریافت می شود، از آنجائیکه ممکن است یک سایت وب هزاران اطلاعات را دریافت کند، صفحه Display نمی تواند تمام رکوردهای این جدول را نمایش دهد. فقط ۲۵ پیغام آخری که وارد شده اند نشان داده می شود و این کار با استفاده از خاصیت MaxRecords از شیء Recordset انجام می شود.

برای نشان دادن مندرجات هر پیام یک ناحیهٔ متنی استفاده می گردد. مزیت استفاده از این نواحی متنی این است که آنها دارای Scrol Bar می باشند، که اگر یک پیام طولانی وارد گردد، قابل دسترس باشند.

```
<%
'Create ADO objects
Set MyConn=Server.CreateObject("ADODB.Connection")
Set RS=Server.CreateObject ("ADODB.RecordSet")
MyConn.Open "FILEDSN=d:\Program Files \Common Files\ODBC\DataSources\MyData.dsn
'Set the maximum number of records to return
RS.MaxRecords=25
'Retrieve the records
RS.Open "SELECT * FROM Feedback ORDER BY Feed_Data DESC", MyConn
%>
<HTML>

<HEAD><TITLE>Display Feedback </TITLE></HEAD>

<BODY>

<FORM>
```

```
<%
'Display the records
WHILE NOT RS.EOF
%>
<BR><B>Data Entered :</B> <%=RS("Feed_Data")%>
<BR><B>Email :</B> <%=RS("Feed_Email")%>
<BR><B>IP Address:</B> <%=RS("Feed_IP")%>
<BR><TEXTAREA COLS=30 Rows=10>
<%=RS("Feed_Contents")%></TEXTAREA>
<HR>
<%
RS.MoveNext
WEND
%>
</FORM>

</BODY>

</HTML>

<%
'Close the Recordset and Connection
RS.Close
MyConn.Close
%>
```



شکل ۱۳-۳ صفحهٔ Display

۱۳-۴ برنامهٔ کاربردی نمونه :(ایجاد یک سیستم کلمهٔ عبور معمولی):

این قسمت چگونگی استفاده از ADO برای حفاظت کلمهٔ عبور در سایت وب و همچنین چگونگی ایجاد یک صفحهٔ ثبت برای ثبت ملاقات کنندگان سایت شما را نشان می دهد. همچنین به شما یاد می دهد که چگونه بازدید کنندگان را از ملاحظهٔ صفحاتی که اجازهٔ دیدن آنها را ندارند ، باز دارید.

در این قسمت با یک مشکل روبرو می شویم (در استفاده از Basic یا Windows NT Challenge/ Respanse) هر دوی این سیستمها قابل ثبت هستند و در قسمت امنیت Windows NT قرار دارند و این بدان معنی است که شمامی توانید هر زمان که یک کاربر جدیدثبت شد، درستی آن را بررسی کنید ولی نمی توانید به آسانی به کلمه های عبور و نامهای کاربرها دست پیدا کنید.

برای استفاده از ADO جهت ایجاد یک سیستم معمولی و اثبات صحت کلمه عبور ، شما نیاز به ایجاد دو فایل و جدول پایگاه داده دارید:

- جدول WebUsers: این جدول پایگاه داده حاوی اطلاعات ثبت است.
- صفحه Registration : این ASP یک فرم ثبت را شامل می‌شود. توسط کامل نمودن فرم، یک کاربر جدید می تواند به سایت وب شما دوباره دسترسی یابد.
- فایل Password Include: این فایل باید شامل ASP ای که می خواهیداز حفاظت کلمه عبور در آن استفاده نمائید، باشد.

ایجاد جدول Webusers :

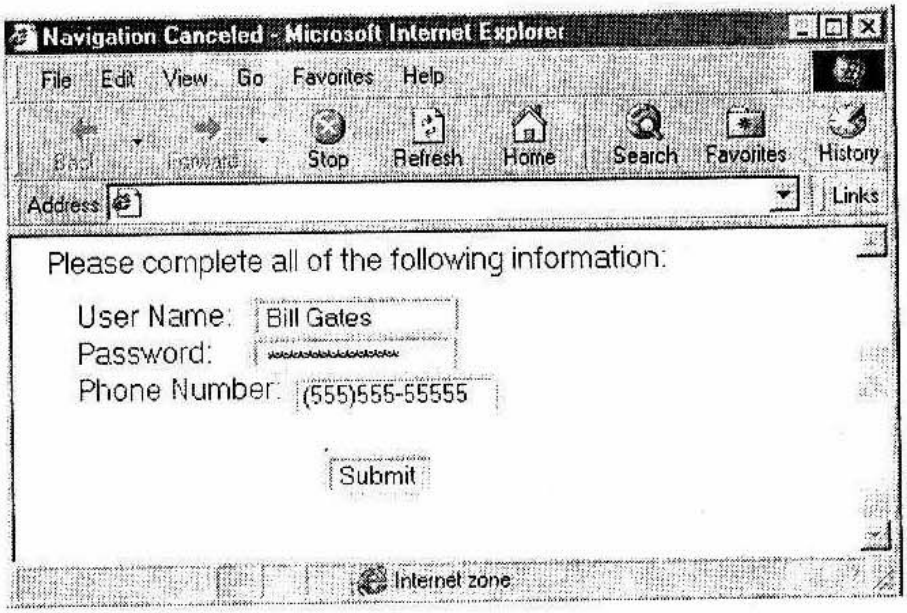
جدول Webusers یک جدول سرویس دهندهٔ SQL است که تنها سه ستون دارد. در اولین ستون اسامی کاربران ، در دومین ستون کلمهٔ عبور کاربران و سومی ستون شماره تلفنهای آنها نگهداری می شود. برای ایجاد این جدول ISQLYW را از گروه برنامه Microsoft SQL Server اجراء کنید. سپس متن زیر را در پنجرهٔ درخواست وارد کنید.

CREATE TABLE (User Name VARCHARB0)و User pass VARCHAR (30) و User phone VARCHAR (30)

هر وقت که یک بازدید کنندهٔ جدید برای دسترسی به یک صفحهٔ محافظت شده با کلمهٔ عبور تلاش کند ، نام کاربر و کلمهٔ عبور او با این جدول بررسی می شود.

ایجاد صفحهٔ Registration :

این صفحه برای اجازه دادن به کاربران جهت ثبت خود در سایت وب شما مورد استفاده قرار می گیرد. اگر شخصی که کلمهٔ عبور معتبر ندارد برای دسترسی به یک صفحهٔ محافظت شده با کلمهٔ عبور ، تلاش کند ، آن شخص به این صفحه راهنمائی می شود.



شکل ۱۳-٤ صفحه Registration

صفحهٔ Registration یک شرط اساسی را به کار می بندد،اینکه اگر تمام فیلد های فرم HTML کامل نشده باشد، آن فرم دوباره ظاهر می شود.در غیر اینصورت ،اگر تمامی اطلاعات وارد شده باشد ،سه حالت زیر رخ می دهد:

۱. اطلاعات ثبت در جدول پایگاه دادهٔ WebUsers وارد می شود.

۲. متغییرهای session ،با اسامی UserName و UserPass به نام کاربر و کلمهٔ عبور جدید تخصیص می یابند.

۳. در نهایت، کاربر به صفحه ای که کاربر از آن شروع کرده بازگشت داده می شود(در صورتی که آن صفحه نامشخص باشد کاربر به صفحهٔ خانگی برگردانده می شود.

صفحهٔ Registration بسیاری از اطلاعات جزئی را نیز از کاربر سؤال می کند. شما به سادگی می توانید مثال حاضر را برای گرفتن هر اطلاعاتی از کاربرتوسعه دهید.برای مثال،ممکن است بخواهید کاربر شمارهٔ کارت اعتباری یا آدرسی را قبل از استفاده از سایت وب شما وارد کند.برای انجام این کار، تنها یک فیلد دیگر به فرم HTML و جدول پایگاه دادهٔ WebUsers اضافه کنید.

```
<%
CONST  HomePage="/default.asp"
'Check If Registration Information Is Incomplete
IF Request.Form("UserName")=" "OR Request.Form("UserPass")=" "
OR Request.Form("Userphone")="" THEN
%>
<HTML>
<HEAD><TITLE>Registration Page</TITLE></HEAD>
<BODY BGCOLOR=#FFFFFF>
<H2>Please complete all of the following information:</H2>
<FORM METHOD="POST"
ACTION="<%=Request.ServerVariables("SCRIPT_NAME")%>">
<TABLE>
<TR>
<TD ALIGN=RIGHT>User Name:</TD>
<TD><INPUT NAME="User Name" TYPE="TEXT"
```

```
VALUE="<%=request.Form("UserName")%>"></TD>
</TR><TR>
<TD ALIGN=RIGHT>Password:</TD>
<TD><INPUT NAME="User Pass" TYPE="TEXT"
VALUE="<%=request.Form("UsertPass")%>"></TD>
</TR><TR>
<TD ALIGN=RIGHT>Phone Number:</TD>
<TD><INPUT NAME="User Phone" TYPE="TEXT"
VALUE="<%=request.Form("UserPhone")%>"></TD>
</TR><TR>
<TD ALIGN=RIGHT COLSPAN=2><INPUT TYPE="SUBMIT" VALUE="Continue">
</TD>
</TR>
</TABLE>
</FORM>
</BODY>
</HTML>
<%
ELSe
'Ready Database Objects
set MyConn=Server.createObject("ADODB.Connection")
MyConn.Open "FILEDSN=d:\Program Files\CommonFiles
\ODBC\Data sources\MyData.dsn"
```