

تعداد مراجعه به یک صفحه وب

برای مشخص نمودن تعداد دفعات مراجعه به یک صفحه وب روش های

متفاوتی وجود دارد که شاید خوانندگان گرامی با برخی از آنها نیز

آشنائی داشته باشند. هدف این مقاله معرفی یکی دیگر از روش های

موجود بکمک تکنولوژی **ASP** است. در این مقاله سعی می شود مطالب

بگونه ای بیان گردد که از ابعاد متفاوت بتواند مفید واقع شده و مخاطبان

خود را در رده های متفاوت اطلاعاتی ارضاء نماید. بنابراین لازم دیدم که

در ابتدا توضیحات مختصری در رابطه با اشیاء (**Objects**) و متدهائی

(**Methods**) که ما را جهت رسیدن به خواسته هایمان کمک می کند،

داشته باشم.

اشیاء استفاده شده

در برنامه ای که در ادامه مشاهده خواهید کرد از دو شیء عمده استفاده

شده است: **Scripting.FileSystemObject** و **TextStream**. دو

شیء فوق مربوط به خانواده بزرگ **Scripting Runtime Library**

می باشند. شیء **FileSystemObject** امکان دستیابی به سیستم فایل موجود بر روی کامپیوتر را فراهم می سازد. در این راستا امکان انجام عملیات متفاوتی بر روی فایل های متنی، فولدرها، درایوها و فراهم می گردد. شیء فوق را می توان از طریق کدهای نوشته شده به زبان **VBScript** که در یک فایل **ASP** قرار خواهند گرفت، استفاده نمود.

شیء **TextStream** تسهیلاتی جهت پردازش ترتیبی بر روی یک فایل را فراهم می نماید. بدیهی است فایل های متنی در این راستا جایگاه ویژه خود را خواهند داشت. معمولا از برخی متدها نظیر **CreateTextFile** و یا **OpenTextFile** استفاده می شود تا پس از فعال شدن بعنوان خروجی یک شیء **TextStream** را برگردانند. فکر می کنم که مثال زیر در این مقطع ما را با نحوه عملکرد این دو شیء بهتر آشنا کرده و یک قدم موثر جهت رسیدن به هدف اصلی این مقاله (نوشتن یک برنامه **ASP** برای مشخص نمودن تعداد دفعات مراجعه به یک صفحه) را برداشته باشیم.

```
/?>
```

```
Dim fso, MyFile
```

```
(fso = CreateObject("Scripting.FileSystemObject Set
```

```
(fso.CreateTextFile("c:\testfile.txt", True =Set MyFile  
(test MyFile.WriteLine("This is a  
MyFile.Close  
</.
```

در مثال فوق تابع **CreateObject** پس از فعال شدن یک شیء از نوع

FileSystemObject را ایجاد می کند. در ادامه متد

CreateTextFile صدا زده شده و فایل (**testfile.txt**) را بعنوان یک

شیء **TextStream** ایجاد می کند و در ادامه بکمک متد **WriteLine**

مربوط به شیء **TextStream** یک خط (**This is a test**) در فایل

ایجاد شده، نوشته خواهد شد.

متدهای استفاده شده

در برنامه ای که در ادامه خواهد آمد از چندین متد استفاده شده است

که لازم است در این مرحله با عملکرد یکایک این متدها آشنا شویم.

- متد **MapPath**: در زمانیکه با فایل های ذخیره شده بر روی یک وب

سایت کار می کنیم، اغلب علاقه مندیم که قادر به دستیابی مسیر واقعی و

فیزیکی فایل ها در مقابل مسیرهای منطقی باشیم. خوانندگان گرامی می دانند که مسیرها بر روی یک وب سایت می توانند بصورت فیزیکی و یا منطقی باشند. مسیرهای منطقی به مسیرهایی اطلاق می شود که بعنوان جزئی در ساختار سلسله مراتبی وب سایت که مدیریت وب (IIS یا PWS) ایجاد می کند، بوجود می آیند (Inetpub\wwwRoot).

مسیرهای فیزیکی دقیقا مشابه دادن مسیر مربوط به یک فایل بر روی هر نقطه از فضای ذخیره سازی خواهند بود. مثلا در صورتیکه متد فوق را بصورت زیر صدا بزنیم:

```
(Server.MapPath("/iishelp/default.htm
```

مقدار زیر برگردانده می شود:

```
C:\winnt\help\iishelp\default.htm
```

متد فوق یک رشته (مشخص کننده نام فایل است) را بعنوان ورودی گرفته و مسیر کامل آن را برمی گرداند.

- متد **OpenTextFile**: متد فوق با هدف باز نمودن یک فایل برای انجام عملیات نوشتن، خواندن و یا اضافه کردن به محتویات آن مورد

استفاده قرار گرفته و پس از فعال شدن یک شیء از نوع **TextStream**

را بر می گرداند. متد فوق دارای گرامری مشابه زیر است:

(OpenTextFile(filename,ioMode,create,format

متد فوق یک فایل با نام **filename** را ایجاد و یا یک فایل موجود با چنین نامی را باز خواهد کرد و یک شیء **TextStream** بر می گرداند.

ioMode، هدف از باز نمودن فایل را مشخص می کند. در صورتیکه

فایل برای خواندن باز شود **forReading = 1** (پیش فرض)، در

صورتیکه برای نوشتن باز شود **forWriting = 2** و اگر برای اضافه

کردن باز شود **forAppending = 8** خواهد بود. پارامتر **create**، در

صورتیکه قصد نوشتن و یا افزودن اطلاعات به فایلی را داشته باشیم که

موجود نباشد مقدار **TRUE** را گرفته و در غیر اینصورت مقدار آن

FALSE (پیش فرض) خواهد بود. و در نهایت پارامتر **format**، قالب

اطلاعاتی را که می بایست از فایل خوانده شده و یا در فایل نوشته شود،

مشخص می کند. در صورتیکه این پارامتر مقدار صفر را دارا باشد قالب

یک فایل **Ascii** در نظر گرفته شده و اگر مقدار ۱- را داشته باشد،

قالب فایل بصورت **Unicode** در نظر گرفته شده و در نهایت اگر این

پارامتر مقدار ۲- را دارا باشد قالب آن بر اساس فرمت پیش فرض سیستم در نظر گرفته خواهد شد.

- متد **CreateTextFile**: متد فوق با هدف ایجاد یک فایل بر روی محیط ذخیره سازی مورد استفاده قرار گرفته و پس از فعال شدن یک شیء از نوع **TextStream** را برمی گرداند. متد فوق دارای گرامری مشابه زیر است:

(**CreateTextFile(filename , overwrite , unicode**

متد فوق فایلی با نام **filename** را ایجاد می کند. در صورتیکه فایلی با همین نام موجود باشد و مقدار پارامتر دوم یعنی **overwrite** نیز **TRUE** باشد فایل قبلی حذف و فایل جدید ایجاد می گردد. مقدار پیش فرض پارامتر فوق **FALSE** است. پارامتر سوم اختیاری است و قالب فایل جدید را مشخص خواهد کرد. اگر مقدار این پارامتر **TRUE** باشد یک فایل متن **unicode** ایجاد می گردد. مقدار پیش فرض پارامتر فوق **FALSE** است.

parsi e-book
WWW.PARSIBOOK.4T.COM

- متد های مربوط به شیء **TextStream**: در برنامه ای که در ادامه خواهد آمد از دو متد **ReadLine** و **WriteLine(String)** استفاده شده است که اولی یک سطر از فایل را خوانده و بصورت یک رشته برمی گرداند و دومی رشته داده شده بعنوان پارامتر را در فایل به همراه یک کاراکتر **newline** اضافه می کند (نوشتن).

توابع استفاده شده

تابع **Trim**: در برنامه ای که در ادامه مشاهده خواهید کرد از تابع فوق استفاده شده است. این تابع جزء توابع کتابخانه ای **VScript** بوده و تمامی فضاهای خالی (**Spaces**) موجود در سمت چپ و راست رشته را حذف و رشته جدید را برمی گرداند. و اما برنامه **Hins.ASP** که تعداد دفعات مراجعه به یک صفحه را مشخص می سازد.

```
"/>  
  
= Set FileObject  
  
(Server.CreateObject("Scripting.FileSystemObject  
  
hits.txt\" & ("\") HitsFile = Server.MapPath
```

```
) Set InStream= FileObject.OpenTextFile
(HitsFile,1,false,0
(OldHits = Trim(InStream.ReadLine
\ + NewHits = OldHits
) Set OutStream= FileObject.CreateTextFile
(HitsFile,True,False
(OutStream.WriteLine(NewHits
</.
.times </.NewHits=/.> been accessed This page has
```



parsi e-book
WWW.PARSIBOOK.4T.COM