

iranphp articles

عنوان مقاله : Object Oriented Programming
نگارنده :
آدرس پست الکترونیک :
تاریخ نگارش :

Object Oriented Programming

در این مقاله می خواهیم به بررسی یکی از زیباترین، و در عین حال خطرناکترین مباحث برنامه نویسی بپردازیم، این مبحث از این جهت خطرناک هست که اگر شما اصول اولیه را یاد بگیرید و این نوع برنامه نویسی بر شما تاثیر بگذارد از آن پس دیگر به همه چیز به چشم یک شیء نگاه خواهید کرد و تمام روشهای برنامه نویسی گذشته خود را به کنار خواهید گذاشت. در این مقاله اصلا قصد ندارم که شروع کنم به آوردن مثالهایی از Object در دنیای واقعی و فرض می کنم که شما مثالهایی مثل رنگ ماشین و یا تلفن را بلد باشید!

در حالت کلی یک شیء شامل یک سری متغیرها و توابع می باشد که درون یک قالب کلی به نام کلاس قرار دارند، به متغیرهای درون کلاسها Properties و به توابع موجود در آن Method گفته می شود.

رویه کلی در ساخت یک class در PHP بسیار ساده است، کافی است شما با استفاده از کلمه class یک کلاس جدید ایجاد کنید، به این صورت:

```
class Item {  
    // properties  
  
    // methods  
}
```

در مثال بالا به این موارد توجه کنید:

ساخته شدن یک کلاس توسط کلمه کلیدی class صورت می گیرد.

شما در هر جای کلاس قادر به تعریف متغیرهای کلاس یا properties هستید اما بهتر است که آنها را در ابتدای class تعریف کنید.

بعد از آن به معرفی توابع کلاس یا methods بپردازید.

تمام این موارد بین دو {} تعریف کلاس صورت می گیرند.

خوب حالا سعی می کنم با چند مثال شما را بهتر با این مفاهیم آشنا کنم:

```
$obj = new Item();  
print "$obj is an ".gettype($obj);
```

اول برای این که ببینیم که یک کلاس ساختیم از PHP می خواهیم که نوع متغیرنسیبت داده شده را به ما برگرداند. در مثال بالا توجه به این نکته که ما برای استفاده از یک کلاس از کلمه new استفاده می کنیم ضروری است، همیشه این کار را برای نسبت دادن یک کلاس به متغیر انجام می دهیم. و در خط دوم با استفاده از تابع gettype نوع متغیر را بر می گردانیم، در زبانهای loosely type مثل PHP (زبانهایی که در آنها نوع متغیر همراه با تغییر متغیر، عوض می شود) این که بدانیم با چه نوع متغیری سروکار داریم اجتناب ناپذیر است، می بینیم که این عبارت رشته object را بر می گرداند. خوب ما تا اینجا یک کلاس ساختیم، ولی مسلما این کلاس هیچ کاربردی ندارد، حالا بیایید آنرا کاربردی تر کنیم.

```
class Item {  
    var $name = "item";  
}  
  
$obj = new Item();  
print "$obj->name<br/>";  
$obj->name = "widget";  
print "$obj->name<br/>";
```

خوب در اینجا دیدید که با استفاده از کلمه کلیدی var یک متغیر تعریف کردیم، با استفاده از عملگر -> به عناصر یک کلاس دسترسی پیدا کردیم و مقادیر آنها را چاپ کردیم، نکته قابل ذکر آنکه در استفاده از کلمه new ما Item() را به کار بردیم درسته که ما از () استفاده کردیم اما این به معنای یک تابع نیست. خوب حالا تا اینجا ما متغیرها رو تعریف کردیم و مقادیر آنها را تغییر دادیم و آنها را چاپ کردیم، اما خوب این کار کلاس رو فقط کمی جالب تر از آرایه های انجمنی (آرایه های که مقادیر به جای داشتن یک index در آرایه یک نام مشخص دارند یا همان associative array) نشون می ده، حالا با اضافه کردن متدها اونو کاربردی تر می کنیم.

```
class Item {  
    var $name="item";
```

```
function setName( $n ) {
    $this->name = $n;
}

function getName() {
    return $this->name;
}

$item = new Item();
$item->setName("widget");
print $item->getName();
```

خوب حالا از متدها هم در کلاسمون استفاده کردیم، فکر نمی کنم این کد به توضیح نیاز داشته باشه .
اما اگر ما بخوایم در هر بار نسبت دادن کلاس هم بتونیم متغیر name رو مقداردهی کنیم چکار باید بکنیم، مثلا در مثال بالا می بینید که با هر بار تعریف، مقدار \$name برابر با item می باشد و ما باید با استفاده از متد setName اونو تغییر بدیم. اینجاست که باید از سازنده ها یا Constructors استفاده کنیم. الان شما می تونید در PHP به دو شیوه Constructor تعریف کنید، قبل از PHP 5 شما باید از متد همانم با کلاس برای ساخت سازنده استفاده کنید، اما خوب با توجه به پیشرفتهای انجام گرفته در PHP 5 در زمینه OOP شما در این نسخه همچنین می تونین از تابع __construct() استفاده کنید. توجه به یک نکته ضروری است که اگر شما قصد نوشتن برنامه های کاربردی را دارید باید به این نکته توجه داشته باشید که تمام وب سرورها PHP 5 را پشتیبانی نمی کنند پس هنوز بهتر است از ویژگیهای عالی OOP در PHP 5 برای برنامه های کاربردی خود چشم پوشی کنید و فعلا اونا رو روی کامپیوتر خودتون آزمایش کنید. خوب حالا کد بالا را با استفاده از یک Constructor بازنویسی می کنیم :

```
class Item {
    var $name;

    function Item( $name="item" ) {
        $this->name = $name;
    }

    function setName( $n ) {
        $this->name = $n;
    }

    function getName() {
        return $this->name;
    }
}

$item = new Item("widget");
$item->getName();
```

خوب تا اینجا همه مفاهیم اولیه در OOP در PHP را یاد گرفتید، تاکید می کنم، فقط مفاهیم اولیه، در این قسمت قصد دارم یک مثال کاربردی تر بیارم تا شما بهتر با این مقوله آشنا بشید .
به قطعه کد زیر توجه کنید :

```
class Page {
    var $Title;
    var $Keywords;
    var $Content;

    function Display () {
        echo "<HTML>\n<HEAD>\n";
        $this->DisplayTitle();
        echo "\n";
    }
}
```

```
$this->DisplayKeywords();
echo "\n</HEAD>\n<BODY>\n";
echo $this->Content;
echo "\n</BODY>\n</HTML>";
}

function DisplayTitle() {
    echo "<TITLE>".$this->Title."</TITLE>";
}

function DisplayKeywords() {
    echo '<META NAME="keywords" CONTENT="'. $this->Keywords. '">';
}

function SetContent( $Data ) {
    $this->Content = $Data;
}
}
```

حالا این کد رو تفسیر می کنیم :

در ابتدا کلاسی با نام Page تعریف کردیم .

۳ متغیر یا properties را در ابتدای تعریف کلاس ساختیم .

در تابع Display() ابتدا خطوط اولیه یک صفحه HTML را نوشتیم، در خط دوم آن تابع DisplayTitle() را فراخوانی کردیم، که این تابع تگهای را در دو طرف عنوان سایت قرار میدهد، در خط چهارم متود Display() ما متود DisplayKeywords() را فراخوانی کردیم، در خط پنجم آن تگ را بستیم و تگ را باز کردیم، سپس مقدار متغیر Content را به طور مستقیم بر روی صفحه نمایش دادیم. این صفحه را با نام page.class.php نام گذاری کنید حالا از این کلاس برای ساخت یک صفحه استفاده می کنیم :

```
include "page.class.php";

$Sample = new Page();

$Content = "<P>This page was generated by the Page Class example.</P>";

$Sample->Title = "Using classes in PHP";
$Sample->Keywords = "PHP, Classes";
$Sample->SetContent($Content);

$Sample->Display();
```

این صفحه را با نام page.php نام گذاری کنید و در مرورگر خود اونو اجرا کنید، اگر همه چیز درست پیش رفته باشه شما باید تمام مفادیری رو که به متغیرها نسبت دادید در جای خود مشاهده کنید. خوب در اینجا هم ابتدا یک نمونه از کلاس Page رو با استفاده از کلمه کلیدی new به متغیر Sample نسبت دادیم، سپس متغیرهای متفاوت را مقداردهی کردیم و سپس تابع Display() را فراخواندیم، به همین سادگی، توجه داشته باشید که هنگامی که با استفاده از عملگر -> از متغیرهای کلاس استفاده می کنیم از آوردن علامت \$ در ابتدای نام متغیر خودداری می کنیم. اما در انتها پاسخ به یک سؤال ضروری به نظر می رسد و آن اینکه آیا شما مجبورید برنامه نویسی OO رو یاد بگیرید؟

شما به راحتی می تونین با گسترش مثال بالا اولین پروژه OOP خود را آغاز کنید! منتظر مقالات پیشرفته تر در این زمینه باشید .