در این بخش قصد داریم مهمترین مبحث زبان های برنامه نویسی را برای شمابازگو نمائیم و آن هم چیزی نیست جز شرط ها و حلقه ها ، شرط ها و حلقه باعث به وجود آمدن یک سیستم رایانه ای شدند ، انسان ها رایانه را ایجاد کردند تاز دوباره نویسی جلوگیری کنند ، و پشت سر هم اطلاعاتی را تکرار و یافراخوانی کنند ، بدون اینکه کاری را چندین بار انجام دهند

شرط ها و حلقه ها بسیار کاربرد دارند

شرط ها : شرط ها گاهی اوقات باعث تغییر مسیر برنامه می شوند ، شما هر اتفاق یا رویدادی راکه در سیستم عامل خود مشاهده می کنید یک شرط است مثلا وقتی درایوE شما فضای کافی برای کپی یک فایل را نداشته باشد ، یک پیغام برای شما ظاهر می شود و می گوید که فضای کافی ندارید و این نیز یک شرط است

دستور شرطی: if

زندگی بدون تصمیم گیری معنا و مفهومی ندارد ، PHPنیز اینگونه می باشد ، پرکاربردترین دستورات شرطی ، دستورif می باشد ، برای اینکه مفهوم شرط برای شما درک شود یک مثال می زنیم:

اگر (if) هوا گرم شد ، من خانه می مانم

در زبان php شرط بدین صورت نوشته می شود

```
<?php
if (it's hot) {
I Will Stay Home;
}
?>
```

همانطور که مشاهده نمودید ، شرط ما داخل پرانتز قرار می گیردو اگر نتیجه درست بود ، شرط ما که بین براکت ها نوشته‌می شود اجرا میگردد:

```
<?php
) {بود true اگر شرط if (
کدهای ما اجرا شود
}
?>
```

نکته : در زبان php همیشه و بعد از هر خط کد از علامت نقطه ویرگول(;) استفاده می شود ولی برای شرط ها از این علامت استفاده نمی شود .

کد های درون براکت ها تنها وقتی اجرا می شوندکه شرط ما صحیح یا True باشد ، در غیر اینصورت شما می توانید کاری دیگر انجام دهید و یا از برنامه خارج شوید

به مثال زیر توجه نمائید:

```
<?php
$i = 20;
if($i >= 20){
echo "salam!your login successfully!";
}
?>
```

در قطعه کد بالا ، یک متغیر با مقدار20 تعریف شده است و در قسمت شرط ، شرطی قرار داده ایم که اگر متغیر ما بزرگتریا مساوی 20 بود یک جمله را چاپ نماید

بعضی وقت ها شما فقط به دستورif نیاز دارید و اگر هم شرط شما false شد برای شما اهمیت چندانی ندارد ولی گاهی اوقات در صورتی که شرط شما صحیح نبود باید یک کار دیگر انجام شود ، در این حالت ازدستور if else استفاده می شود:

```
<?php
if (شرط شما)
{
;اگر شرط صحیح <strong>باشد </strong>، این کد اجرا می شود
}
else
{
;اگر شرط صحیح <strong>نباشد </strong>، این کد استفاده می شود
}
?>
```

به مثال زیر توجه نمائید:

```
<html>
<body>

<?php
$d=date("D");
if ($d=="Fri")
{
echo "Have a nice weekend!";
}
else
{
echo "Have a nice day!";
}
?>

</body>
</html>
```

در قطعه کد بالا ، ما یک متغیر به نامd تعریف نموده ایم و با استفاده از تابعdate ، روز کنونی را به متغیرd مقدار دهی کرده ایم ، حال در قسمت شرط ، شرط قرار داده ایم که اگر روزکنونی برابر با جمعه(Fri) بود ، جملهHave a nice weekend را چاپ نماید واگر هم روز کنونی جمعه نبود ، جملهHave a nice day را چاپ نماید

در حالت فوق ، هر دو شرط ارزیابی می شوند ، اما اگر بخواهید تنها یک عامل انجام شود و شرط دوم بر مبنای شرط اول بررسی شود از یکelseif استفاده می شود:

```
if (شرط شما)
{
;اگر شرط اول صحیح باشد این کد ارزیابی می شود
}
elseif (شرط دوم که در صورتی اجرا می شود که شرط اول صحیح نباشد)
{
;کد مربوط به شرط دوم
}
else
{
;کد پیش فرض در صورتی که هیچکدام صحیح نباشند
}
```

به مثال زیر توجه نمائید:

```
<html>
<body>

<?php
$d=date("D");
if ($d=="Fri")
{
echo "Have a nice weekend!";
}
elseif ($d=="Sun")
{
echo "Have a nice Sunday!";
}
else
{
echo "Have a nice day!";
}
?>

</body>
</html>
```

در قطعه کد بالا ، اگر روز کنونی ما Fri بود ، جمله Have a nice weekend چاپ می شود ، در صورتی که جمعه نبود شرط دوم چک می شود ، اگر روز کنونی برابر با Sun بود جمله Have a nice Sunday چاپ می شود و در غیر اینصورت جمله Have a nice day چاپ می شود.

نکته : دستور elseif را می توانید به صورت else if نیز تایپ نمائید

دستور شرطی: Switch

دستور switch که یکی دیگر از دستورات شرطی می‌باشد را می توانید یک حالت دیگر از if-else در نظر بگیرید که با استفاده از این دستور می توان یک متغیر را با شرط ها بسیار زیادی مقایسه نمود ،دستور switch اینگونه نوشته می شوند:

```
<?php
switch (n)
{
case label1:
code to be executed if n=label1;
break;
case label2:
code to be executed if n=label2;
break;
default:
code to be executed if n is different from both label1 and label2;
}
?>
```

همانطور که قطعه کد بالا را مشاهده نمودید ، ما یک متغیر را با دستور switch مورد شرط قرار می دهیم (n) سپس با استفاده از دستور **case** یک مقدار جایگذاری می کنیم که اگر n برابر با **label1** بود دستورات ما اجرا شود و سپس با استفاده از دستور **break** از شرط خارج می شود و به همین صورت می توانید دستورات شرطی خود را با استفاده از case قرار دهید و در آخر یک گزینه پیش فرض (**Default**) می توانید قرار دهید که اگر متغیر شما برابر با هیچکدام از شرط ها نبود ، اجرا شود

به مثال زیر توجه فرمائید:

```
<html>
<body>

<?php
$x=1;
switch ($x)
{
case 1:
echo "Number 1";
break;
case 2:
echo "Number 2";
break;
case 3:
echo "Number 3";
break;
default:
echo "No number between 1 and 3";
}
?>

</body>
</html>
```

در قطعه کد فوق ، یک متغیر با نامx و مقدار یک قرار داده ایم ، سپس متغیر را با دستورswitch مورد سرط قرار داده ایم ،سپس متغیر را با مقادیر 1 و 2 و 3 مقایسه نموده ایم و در دستورdefault نیز یک مقدار پیش فرض قرار داده ایم ، هم اکنون اگر شما کد فوق را در xampp اجرا نمائید ، در خروجی Number 1 چاپ می شود زیرا متغیر ما برابر با یک بود.

نکته :

- عبارت بعد از کلمه کلیدی case حتما باید یک عدد یا یک رشته متنی باشد
- درون کلمه کلیدی case از عملگرها نمی توانید استفاده نمائید مثلا200<100 case
- اگر هیچ کدام از شروط صحیح نباشد ، دستورdefault اجرا می شود و اگر defaultنوشته نشده باشد ، دستورswitch خیلی آروم و با آرامش خارج می شود

حلقه ها در زبان php

یک حلقه یا Loop قسمتی از یک کد می باشد که بارها و بارها اجرا میشود ، اگر بخواهم برای این بخشمثالی بزنم ، از یک سایت مثال خواهم زد:

شما حتما سایت ها را مشاهده نموده اید و یااینکه شاید خودتان مدیر یک سایت هستید که زمانی که یک مطلب به سایت اضافه می شود ، مطلب جدید در بالای مطالب قبلی قرار می گیرد و پشت سر هم این امرتکرار می شود ، این عمل یک حلقه می باشد که پیاپی ادامه پیدا می کند و شمامی توانید مطالب جدیدی به سایت خود اضافه نمائید

تمامی حلقه ها دارای یک شمارنده هستند ، تازمانی که حلقه مورد نظر ما به شمارنده نرسیده باشد ، یعنی از شمارنده کوچکتر باشد ، حلقه ما اجرا می شودهمچنین برای خواندن مقادیر یک آرایه ازحلقه ها استفاده می نمائید

نکته : معمولا در کد نویسی ، ترکیبی از شرط هاو حلقه ها استفاده می شود که این کار باعث انعطاف پذیری برنامه ما می شود .

حلقه: while

حلقه while ساده ترین نوع حلقه می باشد (تا زمانی که) وطریقه نوشتن این حلقه بدین صورت می باشد

```
<?php
while (تا وقتی شرط صحیح است)
{
;دستورات حلقه
}
?>
```

: زیر توجه نمائید آشنا شوید به مثال while برای اینکه با کربرد

```
<html>
<body>

<?php
$i=1;
while($i<=5)
{
echo "The number is " . $i . "<br />";
$i++;
}
?>

</body>
</html>
```

در مثال فوق ، ما یک متغیر i با مقدار 1 تعریف نموده ایم ، سپس در قسمت while شرط گذاشته ایم که تا زمانی ادامه پیدا کند که متغیر ما (که برابر با یک می باشد) کوچکتر مساوی 5 شود ، یعنی حلقه ما 5 بار تکرار می شود ، سپس مقدار متغیر i چاپ شود و پس از چاپ یک مقدار به متغیر اضافه شود مثلا بعد از اینکه 1 چاپ شد ، متغیر با ++ یک مقدار اضافه می شود سپس مجددا شرط برقرار می شود.

نکته : در قسمت های گذشته آموزش ، ما تاکید نمودیم که از نام های متغیر مفهموی استفاده نمائید که اگر دو هفته دیگر کد خود را مشاهده نمودید ف بدانید که متغیر مورد نظر شما برای چه کاری استفاده می شود ولی متغیر هایی همچون i و j و k به صورت قرار دادی بین برنامه نویسان برای شمارنده حلقه ها استفاده می‌شوند .
نکته : همانطور که کد فوق را مشاهده نمودید ، ما بعد از چاپ هر مقدار متغیر ، یکمقدار به متغیر اضافه نموده ایم ، اگر این کار انجام نمی دادیم ، شرط تا بی نهایت تکرار می شد .

حلقه: do...while

یک حالت دیگر حلقه while ، حلقه do...while می باشد ، که به جای اینکه شرط حلقه را در ابتدای حلقه بررسی کند ، در انتهای حبقه بررسی می نمائید:

```
<?php
do
{
کدی که باید اجرا شود;
}
) while شرطی که باید بررسی شود(
?>
```

نکته : شاید از خودتان بپرسید که چرا اصلا از این دستور استفاده می شود ، چه فرقی می کند که شرط در ابتدای حلقه باشد و یا در انتهای حلقه ؟؟؟ در پاسخ به این سوال باید گفت که دقیقا تفوات این دو در این ابتدا و انتها می باشد ، دردستور while چون

شرط در ابتدای حلقه ایجاد می گردد ، ممکن ست حلقه یک بارهم اجرا نشود ولی درحلقهdo...while ، حلقه ما حداق یک بار اجرا می شود حتی اگر شرط ما هیچ وقت صحیح نباشد

به مثال زیر توجه نمائید:

```
<html>
<body>

<?php
$i=1000;
do
{
echo "$i<br>";
$i++;

}

while ($i<=100);
?>

</body>
</html>
```

در قطعه کد فوق ، متغیر ما برابر با1000 می باشد ، ولی شرط ما گفته است که تا زمانی که متغیر ما کوچکتر مساوی100 باشد ، با این حال متغیر ما بسیار بزرگتر از شرط می باشد ولی شرط ما یک باراجرا می گردد ، می توانید امتحان نمائید

حلقه: for

این حلقه پر کاربرد ترین نوع حلقه ها می باشد ، حلقهfor دارای تعداد تکرار مشخصی می باشد ، ساختار این حلقه بدین گونه می باشد:

```
<?php
for ( شروع حلقه; شرط; کدی که بعد از هر بار تکرار ، اجرا می شود)
{
کدی که باید اجرا شود;
}
?>
```

به مثال زیر توجه نمائید:

```
<html>
<body>

<?php
for ($i=1; $i<=5; $i++)
{
echo "The number is " . $i . "<br />";
}
?>

</body>
</html>
```

در قطعه کد فوق ،نقطه شروع رو برابر با1 قرار داده ایم ، و شرط گذاشته ایم تا زمانی حلقه تکرار شوید که متغیر ماکوچکتر مساوی 5 باشد و به ازای هر بار تکرار حلقه ، یک مقدار به متغیرi اضافه شود . (++). خروجی را حتما درxampp اجرا نمائید.

حلقه روی آرایه ها: foreach

نوع آخر حلقه ها در زبانphp ، حلقه روی آرایه ها می باشد ، حلقهforeach به دو صورت کاربرد دارد

کاربرد اول :اگر تنها می خواهید هر عنصر آرایه را بخوانید از این نوع استفاده می شود

```
<?php
foreach ($array as $value)
{
code to be executed;
}
?>
```

در قسمت فوق ، arrayنام آرایه وvalue نام متغیر موقت می باشد

به مثال زیر توجه نمائید:

```
<html>
<body>

<?php
$x=array("one","two","three");
foreach ($x as $value)
{
echo $value . "<br />";
}
?>

</body>
</html>
```

در قطعه کد فوق ، روی آرایهx یک حلقه را اجرا نموده ایم و نام هر عنصر در یک خط جداگانه برای ما به نمایش در خواهد آمد

کاربرد دوم : ایجاد حلقه بر روی شاخص گذاری رشته ای

از این نوع حلقه زمانی استفاده می شود که شمادارای یک آرایه همراه با شاخص رشته ای باشد زیرا به شما اجازه دسترسی به کلید و هم مقدار عنصر آرایه را می دهد:

```
<?php
foreach (array_name as key_variable => value_variable){
دستورات شما
}
?>
```

به مثال زیر توجه نمائید:

```php
<?php
$book['title'] = '7Learn .com with learn PHP';
$book['author'] = 'loghman avand , Milad Heydari ';
$book['publisher'] = '7Learn .com ';
$book['ISBN'] = '7123=1-1234-1234-0';
foreach($book as $key => $value){
echo "the value of $key is $value<br>";
}
?>
```

در قطعه کد فوق ، یک حلقه بر روی آرایهbook قرار داده ایم و مقدارkey و value آن را چاپ می نمائیم.

نکته : کلمه کلیدیforeach باید سرهم نوشته شود وfor each اشتباه می باشد .