آموزش PHP

ارائه شده به : دکتر حسن حقیقی

توسط : سیده زهرا حسینی و سیده الهه جلمبادانی

تیر ۹۰

فهرست

PHP چیست؟

یک زبان برنامه نویسی با کد باز (Open-Source) است که اگرچه در ابتدا صرفا جهت برنامه نویسی تحت وب تولید شد، امروزه کارآییهای فراوانی پیدا کرده است PHP .یک زبان تحت سرور است(Server-Side)که طبیعتا برای اجرا نیاز به یک برنامه سرویس دهنده وب (Web Server) دارد .

PHP در سال ۱۹۹۵ توسط راسموس لردوف به وجود آمده است . راسموس یک پیاده سازی از PHP را توسط زبان C ایجاد کرد و آن را در اختیار عموم قرار داد .

PHP مخفف عبارت Hypertext Preprocessor یا پیش پردازنده ابرمتن می باشد .

هم اکنون از PHP به طور گسترده برای ایجاد وب سایت ها استفاده می شود.

PHP یک زبان اسکریپت نویسی سمت سرویس دهنده است که برروی ویندوز و لینوکس عمل می نماید و برای ایجاد صفحات وب پویا به کار می رود .

این زبان اسکریپت نویسی همراه HTML به کار می رود و برخلاف آن دارای قابلیت پردازش داده می باشد .

صفحات وب ثابت :

تنها با زبان HTML ایجاد می شوند . لذا محتوا و ساختار کلی این صفحات ثابت بوده و کاربر در هر بار مراجعه به ای صفحات ، همان صفحه مشاهده شده در مراجعه قبلی را میبینند .

صفحات وب پویا :

کاربر در هربار مراجعه به صفحه متفاوتی را مشاهده می کند و ساختار و محتوای صفحات تغییر می یابد . این صفحات به کمک زبان های متعددی مانند ASP و PHP و ۰۰۰ به وجود می آیند .

تفاوت و شباهت ASP و Php

شباهت :

در هر دو زبان هم می توان کدها را به تنهایی نوشت و هم می توان آنها را همراه کدهای HTML به کار برد .

تفاوت :

ASP یک محصول تجاری است ولی PHP رایگان است و کدهای آن در دسترس عموم می باشد .

دلایل استفاده

سرعت بالایی دارد .

رایگان است و سورس کدهای آن در دسترس همگان قرار دارد .

یادگیری و برنامه نویسی آن راحت می باشد .

PHPقابلیت حمل بالایی دارد و بر روی سیستم عامل های مختلف کار می کند .

امنیت بالایی دارد .

قابلیت انعطاف بالایی دارد و برنامه نویسان می توانند براساس نیازهای خود آن را تنظیم و پیکربندی نمایند .

قابلیت های php

ارتباط متقابل با فرم هایHTML : با PHP می توان یک فرم HTML را ایجاد کرد سپس داده های ارسالی آن را پردازش نمود .

ارتباط با بانک اطلاعاتی :PHP قابلیت کار با بانک های اطلاعاتی نظیر Oracle ، Sql و MySql را دارا می باشد .

ایجاد صفحات وب به صورت امن :PHP برای برنامه نویسان امکان ایجاد امنیت برای صفحات وب را فراهم می کند . در این صفحات باید کاربران قبل از مشاهده صفحات وب نام کاربری و رمز عبور خود را وارد نمایند .

کاربرد های php

سمت سرویس گیرنده جاوا اسکریپت

زبان های اسکریپت نویسی

سمت سرویس دهنده PHP

PHP یک زبان اسکریپت نویسی عام منظوره می باشد .

یک اسکریپت PHP شامل دستوراتی به زبان PHP است که به کامپیوتر می گوید چه عملیاتی را انجام دهد .

به عنوان مثال یک اسکریپت می تواند یک پیام را در صفحه نمایش نشان دهد ، داده خاصی را در یک بانک اطلاعاتی ذخیره سازد و یا اطلاعاتی را از یک فایل بخواند .

نرم افزار های مورد نیاز

سیستم عاملی همانند ویندوز ۲۰۰۰ سرور ، XP و یا لینوکس .

سرویس دهنده سازگار با PHP نظیر آپاچی ، IIS ، XAMPP، WAMP یا EASY PHP .

PHP نسخه ۵ و یا بالاتر .

یک مرورگر وب همانند اکسپلورر ، نت اسکیپ ، موزیلا و ...

عملیات نمایش یک صفحه وب ثابت

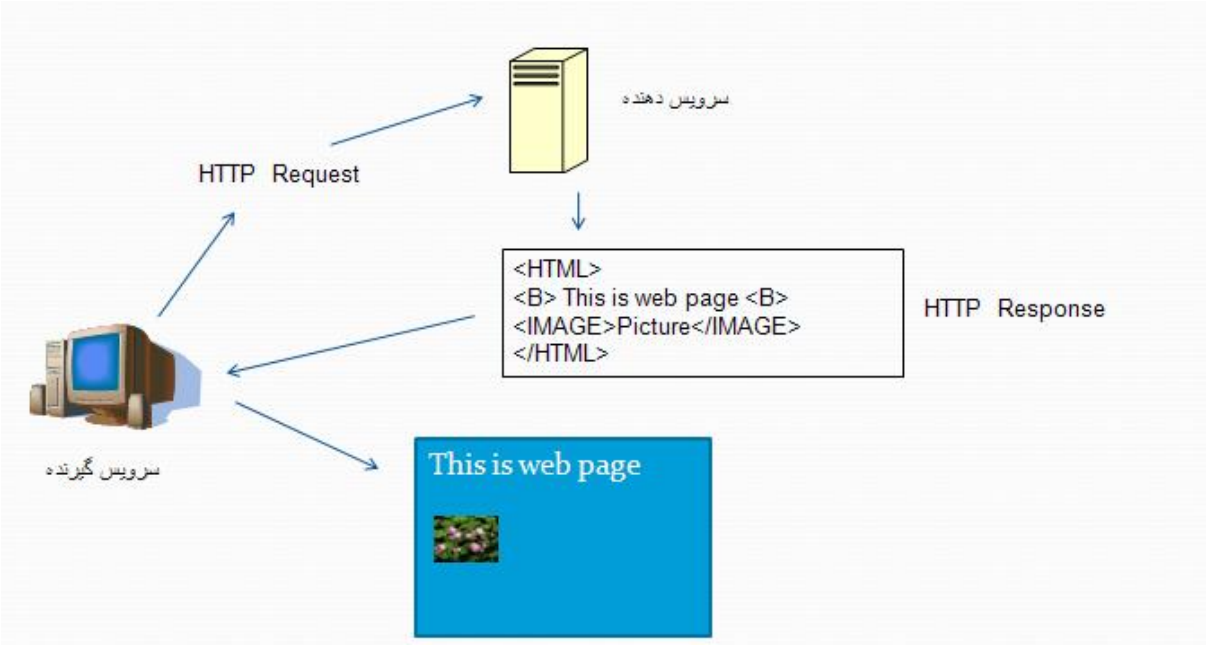
زمانیکه کاربری می خواهد یک صفحه وب ثابت را مشاهده نماید :

۱. کاربر در قسمت آدرس یک مرورگر وب آدرس یک وب سایت را می نویسد و کلید اینتر را می زند .

۲. با این کار یک پیام HTTP Request به سرویس دهنده مورد نظر ارسال می گردد .

۳. سرویس دهنده وب نیز صفحه درخواست شده مرورگر وب را از هارددیسک بازیابی کرده و آن را در یک پیام HTTP Response به برنامه مرورگر کاربر مورد نظر ارسال می کند .

۴. مرورگر وب با دریافت صفحه وب آن را برای کاربر نمایش می دهد .



عملیات نمایش یک صفحه وب php

زمانیکه کاربری می خواهد یک صفحه وب نوشته شده به زبان php را مشاهده نماید :

۱. کاربر در قسمت آدرس یک مرورگر وب آدرس یک وب سایت را می نویسد و کلید اینتر را می زند .

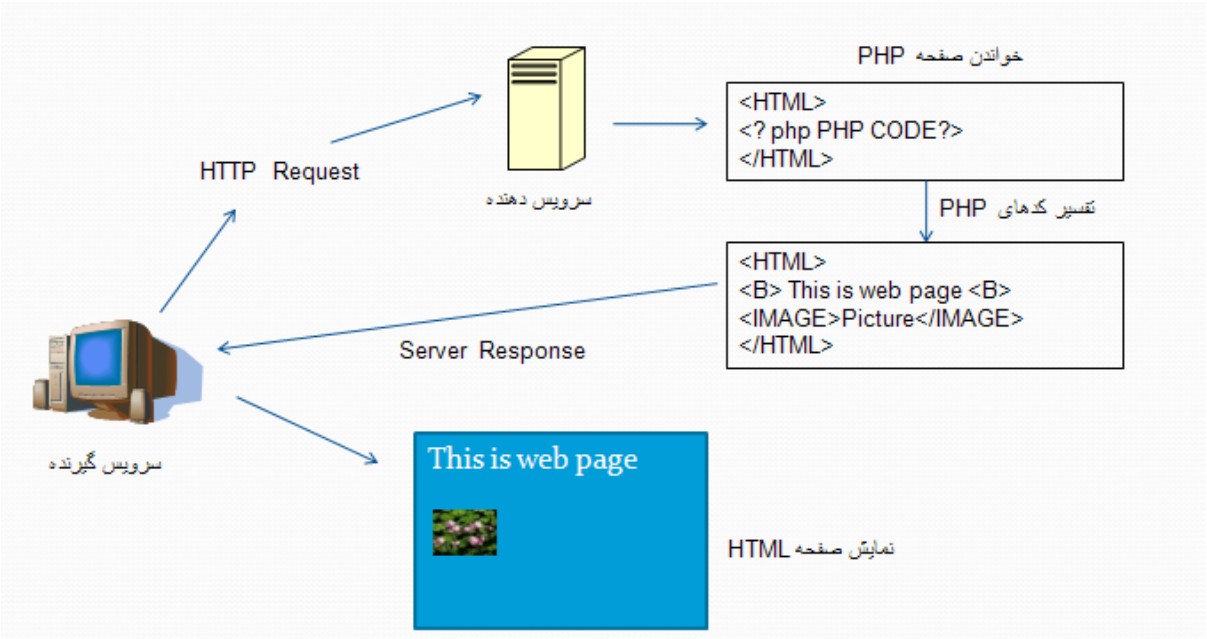
۲. با این کار یک پیام HTTP Request به سرویس دهنده مورد نظر ارسال می گردد .

۳. سرویس دهنده وب نیز صفحه درخواست شده مرورگر وب را از هارددیسک بازیابی می کند.

۴. صفحه وب بازیابی شده توسط موتور PHP مورد پردازش قرار می گیرد .

۵. نتیجه این پردازش به صورت کدهای HTML است و در یک پیام HTTP Response به برنامه مرورگر کاربر مورد نظر ارسال می کند .

۶. مرورگر وب با دریافت صفحه وب HTML آن را برای کاربر نمایش می دهد .



مقدمات برنامه نویسی در php

نوشتن کد های php

کدهای PHP را به سه طریق می توان نوشت :

۱. <? کدهای PHP ?>

۲. <? Php کدهای PHP ?>

۳. <script language="php"> کدهای PHP </script>

نوشتن کد های php درمیان کد های html

در مثال زیر هر کد توسط تگ های مربوط به آن مشخص شده است .

<h۱> This is going to be ignored. </h۱>

<?php echo 'While this is going to be parsed.'; ?>

 This will also be ignored.

در php در پایان دستورات از کاراکتر ; استفاده می کنیم ولی در اخرین دستور نیازی به گذاشتن این کاراکتر نیست

توضیحات در php

در php از کاراکتر های // و یا # برای نوشتن توضیحات تک خطی استفاده می شود . با کاراکترهای /* و */ هم می توانیم توضیحات چند خطی را مشخص سازیم .

```
<?php
    echo ‘This is a test’ ;   //This is one line
    /*This is a multi line comment
      yet another line of comment*/
    echo ‘one final test’ # this is a shell-style comment
?>
```

متغیر ها در php

در php نام متغیرها با $ شروع می شوند .

بعد از کاراکتر $ باید از یک حرف یا کاراکتر _ استفاده کرد و نباید رقم نوشت . بقیه کاراکترها می توانند حروف ، ارقام و_ باشند .

به حروف بزرگ و کوچک حساس است . بنابراین $a و $A دو متغیر متفاوت می باشند .

```
<?php
    $name = ‘Bob’;
    $Name = ‘Joe’;
    echo   $name;
?>
```

زبان php از نوع زبان های loosely typed می باشد که در ان نیازی به تعریف متغییر و تعیین نوع داده ان نیست در این گونه زبان ها تغییر نوع داده ها به صورت خودکار و برحسب نیاز توسط خود زبان برنامه سازی انجام می شود

انواع داده ها درphp

PHP از ۸ نوع داده اصلی استفاده می کند :

boolean

integer

float

string

array

object

resource

NULL

تابع var_dump()

در PHP نوع داده یک متغیر توسط برنامه نویس تعیین نمی شود بلکه در زمان اجرا توسط PHP مشخص می شود . در صورتیکه بخواهید مقدار و نیز نوع داده یک متغیر را بررسی کنید از این تابع استفاده نمایید .

```
<?php
    $str=“foo”;
    $int=۱۲;
   echo var_dump($int);
   echo var_dump($str);
    }
?>
```

خروجی

int(۱۲)

String(۳) “foo”

نوع داده Boolean

یک متغیر از این نوع داده ، تنها یکی از دو مقدار True و False را می تواند بگیرد . این دو نوع مقدار می تواند بصورت حروف بزرگ ، کوچک و یا به هر صورت دلخواه نوشته شود .

```
<?php
    $foo=True;
?>
```

PHP بطور خودکار مقادیر زیر را False در نظر می گیرد :

عدد صفر

رشته تهی

یک آرایه با صفر عنصر

مقدار ویژه NULL

PHP بجز مقادیر فوق ، تمامی مقادیر دیگر را True در نظر می گیرد .

تابع is bool

این تابع بررسی می کند یک متغیر از نوع boolean هست یا نه ؟ اگر متغیر مورد نظر boolean باشد آنگاه مقدار True برمی گرداند . در غیراین صورت مقدار بازگشتی False خواهد بود .

```
<?php
    $a=false;
```

```
$b=۰;
if (is_bool($a))  {
   echo “Yes, a is a boolean”;
}
if (is_bool($b))  {
   echo “Yes, b is a boolean”;
}
?>
```

خروجی

Yes, a is a boolean

نوع داده صحیح

PHP از نوع داده صحیح پشتیبانی می کند و می توان اعداد صحیح را بصورت دسیمال (مبنای ۱۰) ، هگزادسیمال (مبنای ۱۶) و اوکتال (مبنای ۸) به کار برد .

```
<?php
   $a=۱۲۳۴;
   $a=-۱۲۳۴;
?>
```

برای عدد در مبنای ۸ باید در ابتدای آن ۰ قرار دهید و عدد درمبنای ۱۶ در ابتدای آن ۰x قرار دهید .

```
<?php
   $a=۰۱۲;
   $a=۰xA;
```

```
?>
```

زمانیکه یک عدد در مبنای ۸ را استفاده می کنیم ، تمامی ارقام این عدد باید مابین ۰ و ۷ باشند . در صورتیکه که در یکی از ارقام عدد مبنای ۸ این خاصیت را رعایت نکنیم آنگاه تمامی ارقام بعدی صرف نظر خواهند شد .

```
<?php
    var_dump(۰۱۰۹۰);
?>
```

خروجی

int(۸)

نوع داده اعشاری

در PHP اعداد اعشاری را با نوع داده float مشخص می کنند که می توان آنها را به روش های متفاوتی نوشت . در زیر سه روش آن در مثال بیان شده ست :

```
<?php
    $a=۱.۲۳۴;
    $b=۱.۲e۳;
    $c=۷E-۱۰;
?>
```

اگر در هر نوع متغیری بخواهیم بیشتر از اندازه مورد نظر آن داده قرار دهیم سرریز رخ می دهد . بطور کلی اگرعدد بزرگتری را در متغیر صحیح قرار دهیم آن متغیر به نوع float تبدیل خواهد شد .

```
<?php
```

```
$a=۲۱۴۷۴۸۳۶۴۷;
var_dump($a);
$b=۲۱۴۷۴۸۳۶۴۸;
var_dump($b);
?>
```

خروجی

```
Int(۲۱۴۷۴۸۳۶۴۷)
Float(۲۱۴۷۴۸۳۶۴۸)
```

در PHP عملگر تقسیمی وجود ندارد که مقدار صحیح برگرداند . به طور مثال ½ مقدار ۰.۵ برمی گرداند که از نوع float می باشد . به همین منظور شما می توانید نتایج بدست آمده از نتایج تقسیم را با توابعی همانند ()round گرد کنید . همچنین می توانید آن را به نوع integer تبدیل کنید به این منظور قبل از عبارت و یا متغیر از عبارات (int) یا (integer) استفاده نمایید .

```
<?php
var_dump(۲۵/۷);
var_dump(round(۲۵/۷));
var_dump((int)  (۲۵/۷));
var_dump(round(۲۵/۷));
?>
```

خروجی

```
Float(۳.۵۷۱۴۲۸۵۷۱۴۲۸۶)  float(۴)  int(۳)  int(۳)
```

نوع داده رشته

یک رشته مجموعه ای از کاراکترها می باشد . PHP از رشته های طولانی پشتیبانی می کند و هیچ محدودیتی برای طول رشته ندارد . در PHP ، سه روش متفاوت برای ایجاد رشته ها وجود دارد :

۱. با کاراکترهای نقل قول تکی
۲. با کاراکترهای نقل قول جفتی
۳. به روش heredoc

با کاراکترهای نقل قول تکی :

ساده ترین راه برای ایجاد یک رشته می باشد . در این روش اگر بخواهیم کاراکتر نقل قول تکی را در رشته به کار ببریم باید قبل از آن یک کاراکتر \ را بنویسیم . و برای چاپ کاراکتر \ در جمله باید از دو کاراکتر \ پشت سرهم استفاده نماییم (\\)

```
<?php
    echo ‘this is a simple string’;
    echo ‘this is a
    simple string’;
    echo ‘salam \’ : ‘
    echo ‘salam :\\ ‘
?>
```

خروجی

This is a simple string this is a simple string salam ‘ : salam :\

لازم به ذکر است که در این روش ما نمی توانیم در رشته مورد نظر محتویات متغیر را نمایش دهیم .

```
<?php
    $i=۱۰;
```

```
    echo ‘Index is $i’;
?>
```

خروجی

Index is $i

با کاراکترهای نقل قول جفتی :

در این حالت از کدهای ویژه ای برای تولید کاراکترهای خاص می توان استفاده نمود که در جدول زیر آمده است :

رشته	توضیح
\n	کاراکتر ۱۰ اسکی را تولید می کند یعنی یک خط پایین می رود
\r	کاراکتر ۱۳ اسی را تولید می کند یعنی به ابتدای سطر می رود
\t	معادل کاراکتر تب افقی می باشد
\\	کاراکتر \ را ایجاد می کند
\$	کاراکتر $ را ایجاد می کند
\”	کاراکتر “ را ایجاد می کند

مثال

```
<?php
    echo  “My name is :\n Mohammad”;
    echo ‘\t Masdari’;
```

```
?>
```

در این حالت می توان محتویات داخل متغیر ها را نمایش داد .

```
<?php
    $name="\n\t\t Mohammad";
    $i=۱۰;
    echo  "My name is : $name";
    echo "Index is $i";
?>
My name is :
        Mohammad index is ۱۰
```

به روش heredoc :

برای نوشتن رشته های طولانی می باشد . برای نوشتن یک رشته باید مراحل زیر را طی نماییم :

۱ - ابتدا کاراکترهای <<< را نوشته و یک شناسه دلخواه را ذکر نمایید .

۲ - رشته مورد نظر را بنویسید .

۳ - شناسه استفاده شده در ابتدای رشته را نیز بنویسید .

شناسه ای که در ابتدا و انتهای رشته بکار می رود اختیاری است . هر نامی می تواند باشد فقط نباید در داخل خود رشته بکار رفته باشد . نامگذاری شناسه ها مانند متغیرها می باشد . اولین کاراکتر آن باید حرف یا کاراکتر _ باشد و در شناسه های دیگر می توان از حروف ، ارقام و _ استفاده نمود .

زمانیکه با شناسه دلخواه انتهای رشته را مشخص می سازید این شناسه را باید در ستون اول سطر مورد نظر بنویسید و قبل از شناسه از کاراکترهای فاصله استفاده نکنید و بعد از شناسه نیز تنها باید کاراکتر ; قرار دهیم .

مثاال

```
<?php
    $str=<<<EOD
     Example of string
     spanning multiple lines
     using heredoc syntax.
     EOD;
     echo $str;
?>
```

خروجی

```
Example of string
spanning multiple lines
using heredoc syntax.
```

در این حالت می توان محتویات متغیرها را نیز در رشته نمایش داد .

```
<?php
    $a=۱۰;
    $b=۲۰;
    $c=($a+$b)/۲;
     echo<<<EOD
     a is=$a
     b is=$b
     avrage is=$c
     EOD;
?>
```

خروجی

a is=۱۰ b is=۲۰ avrage is=۱۵

در PHP برای اتصال رشته از کاراکتر نقطه . استفاده می شود .

```
<?php
    $a=“Hello”;
    $b=$a . “world”;
    echo $b;
    $a .= “world”;
    echo $a;
?>
```

خروجی

Hello world Hello world

تبدیل رشته به عدد

عملیات تبدیل رشته به عدد براساس قوانین زیر انجام می گیرد :

۱ - در صورتیکه در رشته یکی از کاراکترهای . ، e یا E وجود داشته باشد رشته یک عدد float در غیراینصورت یک عدد integer در نظر گرفته می شود .

```
<?php
    $a=۱+“۱۰”;
    var_dump($a);
```

```
$a=۱+"۱۰.۵";

echo $a;
?>
```

خروجی

Hello world Hello world

عملگر ها در php

عملیات ریاضی

Operator	Description	Example	Result
+	Addition	x=2 x+2	4
-	Subtraction	x=2 5-x	3
*	Multiplication	x=4 x*5	20
/	Division	15/5 5/2	3 2.5
%	Modulus (division remainder)	5%2 10%8 10%2	1 2 0
++	Increment	x=5 x++	x=6
--	Decrement	x=5 x--	x=4

عملیات انتساب

Operator	Example	Is The Same As
=	x=y	x=y
+=	x+=y	x=x+y
-=	x-=y	x=x-y
*=	x*=y	x=x*y
/=	x/=y	x=x/y
.=	x.=y	x=x.y
%=	x%=y	x=x%y

عملیات مقایسه ای

Operator	Description	Example
==	is equal to	5==8 returns false
!=	is not equal	5!=8 returns true
<>	is not equal	5<>8 returns true
>	is greater than	5>8 returns false
<	is less than	5<8 returns true
>=	is greater than or equal to	5>=8 returns false
<=	is less than or equal to	5<=8 returns true

عملیات منطقی

Operator	Description	Example
&&	and	x=6 y=3 (x < 10 && y > 1) returns true
||	or	x=6 y=3 (x==5 || y==5) returns false
!	not	x=6 y=3 !(x==y) returns true

عبارات شرطی

گاهی نیاز داریم که عملکرد های مختلفی برای تصمیم هایمان داشته باشیم

در php از عبارت های شرطی زیر استفاده می شود

If statemenet

If… else statement

If… elseif … else statement

Switch statement

عبارت شرطی if statement

Syntax

if (condition) code to be executed if condition is true;

به عنوان مثال

```
<html>
<body>

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

</body>
</html>
```

عبارت شرطی if…else statement

Syntax

```
if (condition)
code to be executed if condition is true;
else
code to be executed if condition is false;
```

مثال

```
<html>
<body>

<?php
```

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

</body>
</html>
```

عبارت شرطی if.. else if … else

Syntax

```
if (condition)
  code to be executed if condition is true;
elseif (condition)
  code to be executed if condition is true;
else
  code to be executed if condition is false;
```

مثال

```
<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>
```

عبارت شرطی switch

Syntax

```
switch (n)
{
case label ۱:
  code to be executed if n=label ۱;
  break;
case label ۲:
  code to be executed if n=label ۲;
  break;
default:
  code to be executed if n is different from both label ۱ and label ۲;
}
```

مثال

```
<html>
<body>

<?php
switch ($x)
{
case ۱:
  echo "Number ۱";
  break;
case ۲:
  echo "Number ۲";
  break;
case ۳:
  echo "Number ۳";
  break;
default:
  echo "No number between ۱ and ۳";
```

```
}
?>

</body>
</html>
```

آرایه ها در php

آرایه ها انواع خاصی از متغیرها به حساب می آیند که می توانند چندین داده را در قالب یک نام ذخیره کنند.

در زبان php روش های مختلفی برای ایجاد آرایه وجود دارد. ساده ترین روش برای ساخت لیست اسامی دانشجویان مثال قبل به صورت زیر است:

```
$stdNames = array("ali", "hamid", "moeen", "amin", "mina");
```

روش دیگر ایجاد یکی یکی عناصر این آرایه است:

```
$stdNames[۰] = "ali";
$stdNames[۱] = "hamid";
$stdNames[۲] = "moeen";
$stdNames[۳] = "amin";
$stdNames[۴] = "mina";
```

اگر شماره گذاری اندیس ها برای شما مشکل ساز است می توانید به صورت زیر عمل کنید:

```
$stdNames[] = "ali";
$stdNames[] = "hamid";
$stdNames[] = "moeen";
$stdNames[] = "amin";
$stdNames[] = "mina";
```

این روش نیز مانند روش قبلی است با این تفاوت که شماره اندیس عناصر آرایه را ذکر نکردیم و php خود برای این عناصر شماره های ۰ الی ۴ را در نظر خواهد گرفت. این عمل را در آینده auto-incremented

keys می نامیم. اینکه از کدام روش برای ایجاد آرایه استفاده نمایید هیچ تاثیری در نتیجه کار نخواهد داشت.

برای نمایش مقدار یک عنصر از آرایه کافی است که نام آرایه و بعد شماره اندیس آرایه را داخل کروشه ذکر کنیم:

```
echo $stdNames[۱];
```

در مثال قبل php نام hamid را در مرورگر نمایش خواهد داد. می توانید از یک حلقه برای نمایش همه عناصر آرایه استفاده نمایید.

```
for($i=۰;$i<=۴;$i++)
  echo $stdNames[$i];
```

روش قبل روش خوبی برای نمایش همه عناصر یک آرایه است ولی بهترین روش نیست چراکه تعداد عناصر آرایه hard-code شده است. روش بهتر استفاده از حلقه های foreach به شکل زیر است:

```
foreach($stdNames as $name){
  echo $name;
}
```

در صورتیکه برای اشغال زدایی نیاز دارید همه عناصر آرایه را نمایش دهید می توانید از دو توابع داخلی php به نام های print_r و var_dump استفاده نمایید:

```
$print_r($stdNames);
echo "<br />";
$var_dump($stdNames);
```

آرایه های انجمنی (Associative arrays)

اگر بخواهیم به زبان ساده تعریفی از مکانیزم آرایه ها در php داشته باشیم باید از نگاشت یاد کنیم. به عبارت دیگر آرایه php یک نگاشت از کلید (key) به مقدار (value) است. در مثال اسامی دانشجویان کلید ۰ نگاشتی از مقدار "ali" و کلید ۱ نگاشتی از مقدار "hamid" است. php به ما اجازه می دهد که

کلیدهای با معنی تری داشته باشیم. associative array به این معنی است که php به شما اجازه می دهد برای مقادیر آرایه کلیدهای دلخواه در نظر بگیرید. گاهی اوقات به آرایه های associative دیکشنری نیز گفته می شود. لیست اسامی دانشجویان به صورت آرایه associative به شکل زیر تعریف می شود:

```
$stdNames = array(
    "۸۴۲۱۲۲۳۴" => "ali",
    "۸۶۲۸۷۶۳۸" => "hamid",
    "۸۳۸۶۲۸۳۷" => "moeen",
    "۷۹۸۲۹۲۷۳" => "amin",
    "۸۹۷۳۶۲۸۶" => "mina"
);
```

در مثال بالا شماره دانشجویی را به عنوان کلید در نظر گرفتیم. این نکته را حتما به خاطر بسپارید که کلیدها case-sensitive ولی type-insensitive هستند.

case-sensitive: یعنی بین حروف کوچک و بزرگ باید تفاوت قائل شویم برای مثال Air با aiR تفاوت خواهند داشت.
type-insensitive: به این معناست که نوع داده ای کلید برای ما تفاوتی ندارد مثلا ۱ که از نوع int است با '۱' که از نوع رشته ای است با هم تفاوتی نخواهند داشت.

آرایه های associative در زبان php به صورت زیر نیز قابل تعریف اند:

```
$stdNames["۸۴۲۱۲۲۳۴"] = "ali";
$stdNames["۸۶۲۸۷۶۳۸"] = "hamid";
$stdNames["۸۳۸۶۲۸۳۷"] = "moeen";
$stdNames["۷۹۸۲۹۲۷۳"] = "amin";
$stdNames["۸۹۷۳۶۲۸۶"] = "mina";
```

برای نمایش و استفاده از مقادیر این نوع آرایه ها نیز می توانید مثل قبل عمل کنید با این تفاوت که باید از کلیدهای جدید برای دسترسی به عناصر آرایه استفاده نمایید.

```
$echo 'Name ------- Student No'.'<br />';
$echo stdNames["۸۴۲۱۲۲۳۴"].'-۸۴۲۱۲۲۳۴';
```

آرایه های چند بعدی (Multidimensional)

هریک از عناصر آرایه در php می توانند از هر نوعی باشند پس می توانیم آرایه را نیز به عنوان عضو عناصر در نظر بگیریم. بنابراین به زبان ساده تر می توانیم یک آرایه داخل آرایه دیگر تعریف کنیم و داخل آن نیز یک آرایه دیگر و داخل آن نیز و به همین ترتیب. تعریف آرایه چندبعدی در زبان php به سادگی آرایه های معمولی است. در مثال زیر سعی می کنیم لیست داشنجویان را بر اساس رشته تحصیلی در داخل آرایه چندبعدی ذخیره نماییم:

```
$students["Physics"] = array(
    "۸۴۲۱۲۲۳۴" => "ali",
    "۸۶۲۸۷۶۳۸" => "hamid",
    "۸۳۸۶۲۸۳۷" => "moeen",
    "۷۹۸۲۹۲۷۳" => "amin",
    "۸۹۷۳۶۲۸۶" => "mina"
);

$students["Computer"] = array (
    "۸۴۵۳۴۲۴۳" => "samira",
    "۸۵۴۳۲۲۵۵" => "payam",
    "۸۲۳۵۵۲۳۴" => "hossain",
    "۸۸۵۵۳۲۸۹" => "mehrdad",
    "۸۳۸۲۶۲۶۴" => "fatemeh"
);
```

برای دسترسی به عناصر آرایه چندبعدی php کافی است به شکل زیر عمل نماییم:

```
echo $students["Computer"]["۸۸۵۵۳۲۸۹"];
```

در مثال بالا مفسر php نام mehrdad را در مرورگر به نمایش خواهد گذاشت.

حلقه های تکرار در php

معمولا وقتی کدی نوشته می شود که در ان قسمتی از کد بارها تکرار می شود به جای اینکه قطعه کد را بارها در کد تکرار کنیم انرا در حلقه می گذاریم

در php چهار نوع حلقه داریم

While

Do while

For

For each

حلقه while تا زمانی که شرطش درست باشد اجرا می شود

```
while (condition)
  {
  code to be executed;
  }
```

به عنوان مثال کد زیر را در نظر بگیرید

```
<html>
<body>

<?php
$i=۱;
while($i<=۵)
  {
  echo "The number is " . $i . "<br />";
  $i++;
  }
```

```
?>

</body>
</html>
```

خروجی :

The number is ۱

The number is ۲

The number is ۳

The number is ۴

The number is ۵

حلقه do while همیشه یکبار قطعه کد درون انرا اجرا می کند شرط را بررسی می کند و در صورتی که شرط درست بود حلقه را تکرار می کند

```
do
  {
  code to be executed;
  }
while (condition);
```

به عنوان مثال

```
<html>
<body>

<?php
$i=۱;
do
  {
  $i++;
  echo "The number is " . $i . "<br />";
  }
while ($i<=۵);
?>
```

```
</body>
</html>
```

خروجی

The number is ۲

The number is ۳

The number is ۴

The number is ۵

The number is ۶

حلقه for را وقتی می دانیم قطعه کد درونش چند بار باید اجرا شود بکار می بریم

```
for (init; condition; increment)
  {
  code to be executed;
  }
```

init معمولا برای تنظیم شمارنده بکار می رود

Condition برای هر تکرار چک می شود اگر درست بود حلقه اجرا می شود و اگر درست نبود حلقه پایان می یابد.

Increment معمولا برای افزایش شمارنده بکار می رود

هر کدام از پارامتر های بالا می تواند خالی باشدیا اینکه شامل چند عبارت باشد.

مثال

```
<html>
<body>

<?php
for ($i=۱; $i<=۵; $i++)
  {
  echo "The number is " . $i . "<br />";
  }
?>
```

```
</body>
</html>
```

خروجی

The number is ۱

The number is ۲

The number is ۳

The number is ۴

The number is ۵

حلقه foreach برای حلقه درون ارایه ها بکار می رود

```
foreach ($array as $value)
  {
  code to be executed;
  }
```

مثال

```
<html>
<body>

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

</body>
</html>
```

خروجی

one
two
three

توابع (Functions)

توابع قلب یک کد درست طراحی شده است و باعث می شوند کدها خوانا تر شوند و بتوان دوباره از آنها استفاده نمود. هیچ پروژه بزرگی بدون استفاده از تابع نمی تواند انجام شود.

فراخوانی تابع

دو مدل تابع وجود دارد. اولی توابعی هستند که درون خود php هستند و دیگری توابعی است که شما می نویسید .

یکی از ابتدایی ترین توابعی که در خود php هستند تابع print است .

PHP Code:

```
print("Hello Web");
```

در جلو تمامی توابع حتما باید () پرانتزها باشند ، البته print یک استثنا است که بدون پرانتز هم کار می کند

PHP Code:

```
print(("Hello Web");
and
print "Hello Web";
```

هر دو دستور بالا یک خروجی را می دهند ولی این مورد فقط در دستور print عملی است.

در مثال بالا ما تابع print را صدا کردیم و مقدار "hello word" رو برای اون فرستادیم. حالا تابع وارد عمل می شود و این جمله را چاپ می کند. تابع شامل دو بخش است. اولی نام تابع Print در اینجا و دیگری مقادیری که برای تابع می فرستیم argument همان که در داخل پرانتز جلوی تابع آمده است.

برخی توابع نیاز به چند Argument دارند که آنها را با کاما ، جدا می کنیم. مثلا:

PHP Code:

```
some_function( $an_argument, $another_argument );
```

بسیاری از توابع اطلاعاتی برای شما بر می گرداند در راستای عملی که انجام می دهند. مثلا در صورت درست بودن یا نبودن True یاFalse بر می گردانند .

ABS()مثلا ، یک عدد را می گیرد و قدر مطلق آن را بر می گرداند .

PHP Code:

```
۱: <html>
۲: <head>
۳: <title>Listing ۶.۱</title>
۴: </head>
۵: <body>
۶: <?php
۷: $num = − ۳۲۱;
۸: $newnum = abs( $num );
۹: print $newnum;
۱۰: // prints "۳۲۱"
۱۱: ?>
۱۲: </body>
۱۳: </html>
```

در این مثال ما عدد -۳۲۱ را به $num دادیم. این مقدار را به تابع abs فرستادیم در آنجا محاسبات لازم انجام شد و جواب برگردانده شد که ما آنرا در داخل $newnum ریختیم و آن را چاپ کردیم. البته ما می توانستیم کد را کمی جمع و جور تر بنویسیم و مستقیما عدد را به abs بدهیم و همانجا چاپ کنیم.

PHP Code:

```
print( abs( − ۳۲۱ ) );
```

این یک خط کد همان کاری را می کند که در مثال قبل انجام دادیم.

قوانین استفاده از توابعی که خودمان می نویسیم هم به همین شکل است .

تعریف یک تابع

شما می توانید تابع را با استفاده از دستور Function تعریف نمایید.

PHP Code:

```
function some_function( $argument۱, $argument۲ )
{
// function code here
}
```

نام تابع درست بعد از دستور Function می آید و بلافاصله بعد از آن پرانتزها قرار می گیرند. اگر تابع شما Argumentاحتیاج دارد ، شما باید متغیرهای مورد نیاز را (که به وسیله کاما از هم جدا شده اند) را داخل پرانتز بنویسید. اگر تابع شما به Argument احتیاجی ندارد داخل پرانتز چیزی ننویسید.

PHP Code:

```
۱: <html>
۲: <head>
۳: <title>Listing ۶.۲</title>
۴: </head>
۵: <body>
۶: <?php
۷: function bighello()
۸: {
۹: print "<h۱>HELLO!</h۱>";
۱۰: }
۱۱: bighello();
۱۲: ?>
۱۳: </body>
۱۴: </html>
```

در خط ۷ کد بالا ما تابع bighello را تعریف کردیم. مشخص است عملیاتی که این تابع انجام می دهد این است که کلمه !Hello را بین کدهای H۱ چاپ خواهد نمود. ما تابع bighello را بدون Argument تعریف کردیم ، به همین دلیل داخل پرانتز چیزی ننوشتیم .

PHP Code:

```
۱: <html>
۲: <head>
۳: <title>Listing ۶.۳</title>
۴: </head>
۵: <body>
۶: <?php
۷: function printBR( $txt )
۸: {
۹: print ("$txt<br>\n");
۱۰: }
۱۱: printBR("This is a line");
۱۲: printBR("This is a new line");
۱۳: printBR("This is yet another line");
۱۴: ?>
۱۵: </body>
۱۶: </html>
```

در مثال بالا ما تابع printBR را با Argument تعریف می کنیم. حالا در خطوط ۱۱ و۱۲ و ۱۳ سه مقدار متفاوت را به تابع می فرستیم و مثلا سه خط چاپ شده در خروجی خواهیم داشت $txt .همانطور که می بینید در خط ۷ تعریف شده است. موقعی که در خطوط ۱۱ و ۱۲ و۱۳ ما تابع را صدا می کنیم $txt هر دفعه مقداری که برایش فرستاده شده است را به خود می گیرد و در خط ۹ آن را چاپ می کند.

هر مقدار که بخواهیم می توانیم این تابع را اجرا کنیم و خروجی بگیریم.

توجه داشته باشید که اگر تابعی Argument نیاز داشته باشد ، موقع صدا کردن تابع باید حتما مقدار برای آن بفرستیم .

نکته در صورتیکه تابع را به این صورت تعریف کنید :

PHP Code:

```
function printBR( $txt = "nothing")
```

در این حالت $txt به صورت default مقدار "nothing" را دارد. یعنی اگر ما موقع صدا کردن تابع مقداری برای Argument نفرستیم تابع خودش مقدار Defaultرا به $txt می دهد ولی اگر ما مقدار

بفرستیم $txt برابر با مقدار فرستاده شده می باشد.

مثلا :

[php]
PrintBR();
PrintBr("Hello");

PHP Code:

در خط اول کد بالا چاپ خواهد شد Nothing و در خط بعد چاپ می شود . Hello مقدار Default در ب رخی توابع بسیار کارا هستند .

[b]بازگرداندن مقادیر از توابع تعریف شده توسط کاربر

شما می توانید از داخل تابع با استفاده از دستور Return مقداری را برگردانید. دستور return عملیات تابع را متوقف می نماید و مقدار گفته شده را بر می گرداند.

PHP Code:

```
۱: <html>
۲: <head>
۳: <title>Listing ۶.۴</title>
۴: </head>
۵: <body>
۶: <?php
۷: function addNums( $firstnum, $secondnum;
۸: {
۹: $result = $firstnum + $secondnum )
۱۰: return $result;
۱۱: }
۱۲: print addNums(۳,۵);
۱۳: // will print "۸"
۱۴: ?>
```

۱۵: </body>

۱۶: </html>

کد بالا عدد ۸ را در خروجی چاپ می کند. عددهای ۳ و ۵ در $firstnum and $secondnum ذخیره شدهند و بعد با هم جمع شدند. و جواب آنها در $result ذخیره شد و سپس در خط ۱۰ آن مقدار برگردانده شده است. با دستور Return شما می توانید هر چیزی را برگردانید مثلا :

PHP Code:

```
function addNums( $firstnum, $secondnum )
{
return ( $firstnum + $secondnum );
}
```

تابع بالا نیز دقیقا همان کاری را می کند که تابع قبلی می نمود .
حتی می توانید به این صورت نیز از return استفاده کنید.

PHP Code:

```
return ۴;
```

می توانید نتیجه یک عملیات را برگردانید:

PHP Code:

```
return ( $a/$b );
```

یا حتی مقداری از یک تابع دیگر را برگردانید:

PHP Code:

```
return ( another_function( $an_argument ) );
```

صدا کردن یک Function به صورت داینامیک

این امکان وجود داره که شما اسم تابع یک String یا یک متغیر بگذارید. و برای صدا کردنش از اون استفاده کنید. مثلا:

PHP Code:

```
<html>
<head>
<title>Listing ۶.۵</title>
</head>
<body>
<?php
function sayHello()
{
print "hello<br>";
}
$function_holder = "sayHello";
$function_holder();
?>
</body>
</html>
```

در مثال بالا در خط ۷ تابع با اسم Sayhello تعریف شده و در خط ۱۱ function_holderیک متغیری تعریف شده با مقدار sayHello حالا می توان از function_holderبا اضافه پرانتزها برای صدا کردن تابع استفاده کرد.

شاید این سوال پیش بیاد که چرا ما باید همچین چیزی رو لازم داشته باشیم. در مثال فوق عملا ما فقط کار خودمون رو زیادتر کردیم. ولی در واقع در برخی مواقع لازم داریم که جریان کد رو با توجه به مولفه های داخل url یا شرایط برنامه عوض کنیم.
یعنی مثلا در شرایطی یک function اجرا شود و در شرایط دیگه function دیگری.

متغیر ها در داخل تابع

مهم : متغیرهایی که داخل یک تابع تعریف می شوند ، فقط در داخل همون تابع قابل دسترسی هستند.
یعنی اون متغیر ها بیرون تابع یا در تابع های دیگر در دسترس نیستند. در پروژهای بزرگ این امکان خیلی به شما کمک می کند

چون شما می توانید از اسم های تکراری برای متغیرهایتان در تابع های مختلف استفاده کنید بدون اینکه دخالتی در یکدیگر داشته باشند.

PHP Code:

```
۱: <html>
۲: <head>
۳: <title>Listing ۶.۶</title>
۴: </head>
۵: <body>
۶: <?php
۷: function test()
۸: {
۹: $testvariable = "this is a test variable";
۱۰: }
۱۱: print "test variable: $testvariable<br>";
۱۲: ?>
۱۳: </body>
۱۴: </html>
```

در مثال بالا خروجی چیزی رو چاپ نخواهد نمود. چون $testvariable که در خط ۱۱ برای چاپ خوانده می شود قبلا تعریف نشده است و $testvariable خط ۹ فقط در داخل function قابل دسترسی هستند.

استفاده از متغیر به صورتGlobal یعنی داخل و خارج تابع ها

به صورت Default متغیرهای تعریف شده بیرون یک تابع ،داخل تابع در دسترس نیست.
مثلا در مثال زیر

PHP Code:

```
۱: <html>
۲: <head>
```

```
۳: <title>Listing ۶.۷</title>
۴: </head>
۵: <body>
۶: <?php
۷: $life = ۴۲;
۸: function meaningOfLife()
۹: {
۱۰: print "The meaning of life is $life<br>";
۱۱: }
۱۲: meaningOfLife();
۱۳: ?>
۱۴: </body>
۱۵: </html>
```

خروجی خالی چاپ می شود و مقدار $life را چاپ نخواهد کرد. در برخی موارد ما نیاز به استفاده از متغیرهای بیرون تابع داخل یک تابع داریم. برای این کار کافیه که از دستور Global استفاده کنیم. به طور مثال می توانید برای این منظور کد بالا را به صورت زیر بازنویسی کنیدر :

[php]

```
۱: <html>
۲: <head>
۳: <title>Listing ۶.۸</title>
۴: </head>
۵: <body>
۶: <?php
۷: $life=۴۲;
۸: function meaningOfLife()
۹: {
۱۰: global $life;
```

```
۱۱: print "The meaning of life is $life<br>";
۱۲: }
۱۳: meaningOfLife();
۱۴: ?>
۱۵: </body>
۱۶: </html>
```

در خط ۱۰ از دستور global $life; استفاده کردیم. در این حالت مقدار $life که بیرون تابع و در خط ۷ تعریف شده در داخل تابع قابل دسترس می شود و خروجی این کد ۴۲ The meaning of life is چاپ خواهد شد.

شما باید برای هر متغیری که می خواهید ازش در تابع استفاده کنید از این دستور استفاده کنید. و همچنین در هر تابعی که می خواهید از متغیری خارج از آن تابع استفاده کند باید از این دستور استفاده شود.

مهم : دقت کنید که اگه $life داخل تابع تغییر دهید مقدار $life در کل برنامه عوض می شود.

مدیریت فرم هابا php

یک فرم html با دو فیلد ورودی و یک دکمه در نظر بگیرید

```
<html>
<body>

<form action="welcome.php" method="post">
Name: <input type="text" name="fname" />
Age: <input type="text" name="age" />
<input type="submit" />
</form>

</body>
</html>
```

وقتی کاربر دکمه را فشار می دهد اطلاعات به صفحه welcome.php می رود

در صفحه welcome.php می نویسیم

```
<html>
<body>

Welcome <?php echo $_POST["fname"]; ?>!<br />
You are <?php echo $_POST["age"]; ?> years old.

</body>
</html>
```

خروجی

Welcome John!
You are ۲۸ years old.

توابع $_GET و $_post در php

وقتی فرم در روش get ارسال می شود اطلاعات برای همه قابل مشاهده است و داری محدودیت در میزان اطلاعات فرستاده شده دارد . در حالی که در روش post اطلاعات قابل مشاهده نیست و دارای محدودیت در میزان اطلاعات نیز نمی باشد.

از تابع $_GETو $_Post برای جمع اوری داده ها استفاده می شود.

Date در php

از تابع date برای فرمت بندی زمان وتاریخ استفاده می کنیم.
این تابع با استفاده از یک timestamp زمان و یا تاریخ را فرمت بندی میکند تا بیشتر قابل خواندن باشند.

date(format, timestamp)

اولین پارامتر که پارامتر الزامی محسوب می شود پارامتر format است که فرمت timestamp را مشخص می کند. و دومین پارامتر هم timestamp را مشخص که یک پارامتر اختیاری است .

فرمت بندی تاریخ

اولین پارامتر در تابع date فرمت تاریخ و زمان را تعیین می کند .در اینجا برخی از کاراکترهایی که می توان استفاده کرد برای شما مشخص کرده ایم:

- d - : چندمین روز ماه را مشخص می کند ۰۱ تا ۳۱
- m - : شماره ماه را مشخص میکند۰۱تا ۱۲
- Y - : سال را در چهار رقم نشان می دهد .

در مرجعی که در آینده برای شما تهیه خواهیم کرد می توانید تمام این کاراکترها را مشاهده کنید.
کاراکتر های دیگر مثل "/" و "."و یا "-" میتوانند بین حروف برای فرمت بندی استفاده شوند:

```
<?php
echo date("Y/m/d")."<br>" ;
echo date("Y-m-d")."<br>";
echo date("y.m.d")."<br>";
?>
```

خروجی دستورات فوق که تاریخ جاری را بر میگرداند می تواند مشابه زیر باشد:

۲۰۱۱/۰۷/۰۱

۲۰۱۱-۰۷-۰۱

۲۰۱۱.۰۷.۰۱

اضافه کردن یک Timestamp

قبلا اشاره کردیم پارامتر اختیاری که در تابع date استفاده می کردیم مشخص کننده یک timestamp بود . در صورت استفاده نکردن از این پارامتر از تاریخ جاری استفاده می شود.ما از تابع mktime برای برگرداندن یک unix timestamp استفاده می کنیم .

```
mktime(hour,minute,second,month,day,years,is_dst);
```

درمثال زیر ما نحوه استفاده از این موضوع را مشخص کردیم این مثال برای مشخص کردن تاریخ فردا استفاده میشود:

```
<?php
$farda = mktime(۰,۰,۰,date("m") ,date("d")+۱,date("Y"));
echo "farda is ".date("Y/m/d", $farda);
?>
```

خروجی :

۲۰۱۱-۰۷-۰۲

کار با فایل ها در php

تابع ;Include() این امکان رو به شما میده و باعث سهولت و کم حجم شدن صفحات میشه.
این تابع فقط به یک آرگومان نیاز داره و اون مسیر فایل پی اچ پی هست که میخوایم به صفحمون پیوند بدیم
مثال : فرض کنید ما در فایل a.php یک کد داریم که یک پیغام رو چاپ میکنه
حالا ما می خواهیم همین دستور در فایل اول رو در فایل b.php بدون نوشتن دوباره دستور و با دستور include بنویسیم

PHP Code:

```
// file name is b.php
<?php
include("a.php");
?>
//will Print message in a.php
```

یک مثال :

PHP Code:

```
<?php
//this file name is a.php
$ret=(۴+۴);
```

```
return $ret;
?>
```

_____

PHP Code:

```
<?php
//this file name is b.php
$flag=true;
if($flag) {
$resualt=include("a.php");
print " The Sum Of (۴+۴) Is $resualt";
}
?>
```

می توانیم include را به گونه ای تعریف کنیم که اگر شرط درست نبود اصلا دستور include اجرا نشود و در خط بعدی مقدار فایل a.php رو توی یه متغییر دیگه مینوسیسم و چاپ میکنیم.

کار با فایل ها

مثال:

PHP Code:

```
<?php
if (file_exists("a.php"))
print "The File Exists";
?>
```

حتی میتونید با تابع دیگری بفهمید مسیر داده شده یک فایل هست یا یک دایرکتوری

PHP Code:

```
//Check if it's a file
<?php
if(is_file("a.php")
print"yes this is file";
?>
/*----------------------------*/
//Check if Current Path is a dir
<?php
```

```
if(is_dir("/tmp"))
print"/tmp is valid";
?>
```

تابع دیگری که وجود داره توابع is_executeable , is_writeable , is_readable هستن که چک میکنن ببینن فایل مورد نظر قابل خواندن و یا نوشتن ویا اجرا شدن هست یا مسیر داده شده معتبر هست یا نه و یک مقدار از نوع بولین برمیگردونه .

تابع دیگری نیز وجود دارد که سایز یک فایل رو برمیگردونه خیلی ساده

```
Print filesize("a.php");
```

این تابع سایز فایل شما رو برحسب بایت نمایش میده

تابع دیگری که میخوایم بررسی کنیم تابع ;()fileatime میباشد که اخرین باری که یک فایل دسترسی پیدا کرد رو به ما بر میگردونه ما در مثال زیر میخوایم بدونیم فایل a.php در چه تاریخ و زمانی برای اخرین بار دسترسی پیدا کرده است :

PHP Code:

```
<?php
$lasttime=fileatime("a.php");
print "The File last time accessed  in ".date("D d M Y  g:i A",$lasttime).".";
// Will Print Sat ۱۴ jan ۲۰۰۶ ۱۰:۳۰ Pm
?>
```

تابع ;()filemtime نیز مشابه ;()fileatime هستش با این تفاوت که تاریخ و زمان اخرین باری که فایل ویرایش شد رو برمیگردونه . تابع ;()filectimeنیز وجود داره که در سیستم های یونیک تاریخ تغییر یا ویرایش فایل رو برمیگردونه ولی در پلت فرم های دیگه تاریخ بوجود آمدن فایل رو برمیگردونه

حالا میرسیم به توابع کاربردی تر :

تابع

```
touch("file-path.txt");
```

در صورتی که فایلی با این نام وجود نداشته باشد این فایل رو ایجاد میکنه ولی اگه وجود داشته باشه کاری نمیکنه و فقط تاریخ ویرایش فایل تغییر پیدا میکنه و فایل از بین نمیره

با تابع

```
unlink("file-path.txt");
```

میتونید یک فایل رو پاک کنید

نکته : در سیستم های یونیکس برای اینکه یک فایل را پاک یا ویرایش یا دست یابی پیدا کنیم لازم است که دسترسی به فایل رو داده باشید.

بازکردن فایل قبل از خواندن و نوشتن :

قبل از اینکه بتونید یک فایل رو بخونید یا محتوایش رو عوض کنید به این احتیاج دارید که اون فایل رو باز کنید

شما با این دستور میتونید یک فایل رو برای خواندن اماده کنید

$f=fopen("file.txt",'r');

و با این دستور میتونید فایل رو برای نوشتن اماده کنید

$f=fopen("file.txt",'w');

و برای اضافه کردن اطلاعات به یک فایل باید از این دستور استفاده کنید (Append)

$f=fopen("file.txt",'a');

بهتره قبل از اینکه اقدام به ویرایش یا باز کردن یک فایل کنید اون رو امتحان کنید ببینید اجازه باز شدن یا ویرایش شدن رو داره ؟

PHP Code:

```
If ($fp=fopen("file.txt",'w'))
{
// codehaie marboot be viraiesh file
}
```

یا میتونید بجای کد بالا اینگونه عمل کنید

PHP Code:

($fp=fopen("file.txt",'w')) or die("Could Not open file");

اگه دستور فوق مقدار درست رو برگردونه پیغام Could Not open file نشون داده نمیشه در غیر اینصورت نشون داده میشه .

همون طور که متوجه شدید هر عملیاتی که بخوایم بر سر فایل اجرا کنیم باید داخل :

Fopen();

// Code

Fclose();

انجام بدیم .

پی اچ پی امکانات زیادی رو برای خوندن یک فایل در اختیار ما میزاره بعنوان مثال شما میونید یک فایل رو برحسب بایت یا برحسب لاین یا برحسب کاراکتر بخونید .

به عنوان مثال:

PHP Code:

```
<?php
$filename = "test.txt";
$fp = fopen( $filename, "r" ) or die("Couldn't open $filename");
while ( ! feof( $fp ) )
 {
 $line = fgets( $fp, ۱۰۲۴ );
 print "$line<br>";
 }
 ?>
```

PHP Code:

```
<?php
$filename = "test.txt";
$fp = fopen( $filename, "r" ) or die("Couldn't open $filename");
while ( ! feof( $fp ) )
{
$chunk = fread( $fp, ۱۶ );
print "$chunk<br>";
}
?>
```

PHP Code:

```
<?php
$filename = "test.txt";
$fp = fopen( $filename, "r" ) or die("Couldn't open $filename");
```

```
$fsize = filesize($filename);
$halfway = (int)( $fsize / ۲ );
fseek( $fp, $halfway );
$chunk = fread( $fp, ($fsize − $halfway) );
print $chunk;
?>
```

در کد بالا ما نیمه دوم یک فایل رو چاپ میکنیم .

دستور ;()fgetc مثل دستور ()fgets میباشد که اگر در کد بالا که خط به خط یک فایل رو اجرا میکرد کاراکتر به کاراکتر فایل رو نشان میده‌د .

برای نوشتن یا اضافه کردن مقدار به یک فایل باید ابتدا فایل رو بصورت

Foper("file.txt",'w');

Or

Fopen("file.txt",'a');

شما میتونید با تابع ;()fwrite داخل یک فایل مقداری رو قرار بدید ، که در اینصورت محتوای فایل قبلی پاک میشه و میتونید با تابع ;()fputs یک مقدار رو به فایل مورد نظر اضافه کنید .

مثال :

PHP Code:

```
<?php
 $filename = "test.txt";
 $fp = fopen( $filename, "w" ) or die("Couldn't open $filename");
 fwrite( $fp, "Hello world\n" );
 fclose( $fp );
 print "Appending to $filename<br>";
 $fp = fopen( $filename, "a" ) or die("Couldn't open $filename");
 fputs( $fp, "And Hello To You\n" );
 fclose( $fp );
 ?>
```

تابع تعیین دسترسی

فایل شما میتونید با دستور ;flock() برای یک فایل دسترسی های متفاوتی داشته باشد.

PHP Code:

◊ Sharing ---- ۱اجازه خواندن میده ولی نوشتن خیر

◊ Exclusive ---- ۲اجازه خواندن و نوشتن نمیدهد

◊ Release ---- ۳دسترسی های بالا را ازاد میکند

کار با پوشه ها :

شما میتونید با دستور mkdir() , rmdir() پوشه ای ایجاد یا پاک کنید

مثال برنامه ای که فایل های داخل یک پوشه رو نمایش بدهد

PHP Code:

```
<?php
$dirname = "testdir";
$dh = opendir( $dirname );
while ( gettype( $file = readdir( $dh )) != boolean )
{
if ( is_dir( "$dirname/$file" ) )
print "(D)";
print "$file<br>";
}
closedir( $dh );
?>
```

کلاس ها در php

ابجکت چیست ؟ مجموعه ای از متغییرها و توابع است که از یک الگوی خاص به نام کلاس ساخته شده اند .

شکل کلی یک کلاس

PHP Code:

```
Class First_class
{
 //این شکل کلی از یک کلاس هستش
}
```

پروپرتی ها:

آبجکتها به متغییر های خاصی دسترسی دارند که به آنها پروپرتی میگویند این پروپرتی ها میتونند در هرجای بدنه کلاس باشند اما برای اینکه کد مون مرتب باشه بهتره که در بالای کلاس تعریف بشن.مثال :

PHP Code:

```
Class f_class {
var $name="php_۱ ";
}
$obj۱=new f_class();
$obj۲=new f_class();
$obj۱->name="php_۲";
print "$obj۱->name<br>";
print"$obj۲->name<br>";
```

دیدیم که برای اختصاص دادن یک کلاس به یک متغییر اینگونه عمل کردیم

PHP Code:

```
$obj۱=new f_class();
```

علامت <- به شما اجازه میده تا به متغییر های درون یک کلاس دسترسی داشته باشید و اونها رو تغییر بدید همونطور که در کد میبینید ما در خط ششم متغییر name در ابجکت یک رو مساوی علی قرار دادیم که باعث عوض شدن متغییر میشه همچنین برای چاپ خروجی نیز به همین صورت عمل کردیم ولی بدون علامت مساوی

PHP Code:

```
Print  " $obj۱->name ";
```

متد ها :

متدها در واقع توابعی هستند که داخل یک کلاس وجود دارند بزارید با یک مثال واضح تر بیان کنم :

PHP Code:

```
class f_class
    {
```

```
var $name;
function sayHello()
 {
Print "Hello World";
}
    }
$obj۱=new f_class();
$obj۱->sayHello();
 چاپ میشود// Hello World
```

همونطور که میبینید یک متد خیلی شبیه به تابع معمولی هستش با این تفاوت که متد همیشه داخل کلاس تعریف میشه در ضمن شما میتونید با علامت <- یک متد ابجکت را صدا بزنید

مهمتر اینکه متدها به اعضای متغییر های یک کلاس دسترسی دارند

شما همین الان دیدید که چطوری به یک پروپرتی از خارج یک آبجکت دسترسی پیدا کنیم اما چطوری یک آبجکت میتونه خودشو به اصطلاح Return کنه :

PHP Code:

```
class f_class
    {
var $name="php_۱";
function sayHello()
 {
Print "Hello My names $this->name<br>";
}
    }
$obj۱=new f_class();
$obj۱->sayHello();
 چاپ میکنه// Hello My names php_۱
```

یک عبارت مخصوص رو بکار بردیم به اسم $this تا کلاس به آبجکت کنونی Return بشه شما با ترکیب این عبارت با علامت <- میتونید داخل یک کلاس به هر پروپرتی و متدی که بخواهید دسترسی داشته باشید حالا اگه بخواهیم به پروپرتی name در همه آبجکت های کلاسمون مقدار خاصی بدیم میتونیم به این صورت عمل کنیم :

PHP Code:

```
class f_class     {
var $name="php_١";
function setName($n){
$this->name=$n;      }
function sayHello()
 {
Print "Hello My names $this->name<br>";
}
    }
$obj١=new f_class();
$obj١->setName("php_٢");
$obj١->sayHello();
 چاپ میکنه// Hello My names php_٢
```

کد بالا رو می توانیم بصورت ساده تر وکمی پیچیده تر هم بنویسیم :

PHP Code:

```
class first_class {
 var $name;
 function first_class( $n="php_١" ) {
 $this->name = $n;
 }
 function sayHello() {
 print "hello my name is $this->name<BR>";
 }
 }
 $obj١ = new first_class("php_٢");
 $obj٢ = new first_class("php_٣");
 $obj١->sayHello();
 چاپ میکنه // hello my name is php_٢
 $obj٢->sayHello();
 چاپ میکنه // hello my name is php_٣
```

extend

مثال :

PHP Code:

```
 class first_class{
 var $name = "php_١";
 function first_class( $n ) {
 $this->name = $n;
 }
 function sayHello(){
print "Hello my name is $this->name<br>";
}
}
 class second_class extends first_class {
 }
 $test = new second_class("son of php_١");
$test->sayHello();
 // outputs "Hello my name is son of php_١"
```

PHP Code:

```
class first_class {
 var $name = "harry";
 function first_class( $n ) {
 $this->name = $n;
 }
 function sayHello() {
print "Hello my name is $this->name<br>";
 }
 }
 class second_class extends first_class  {
 function sayHello() {
 print "I'm not going to tell you my name -- ";
 first_class::sayHello();
 }
```

```
}
$test = new second_class("son of harry");
$test->sayHello();
// "I'm not going to tell you my name --
چاپ میکنه Hello my name is son of harry"
```

همونطور که میبینید دستور (متد:: وارث) ما میتونیم هر متدی رو که ما تغییرش دادیم دوباره صدا بزنیم چون در کلاس دو ما متد sayHello رو تغییر دادیم با این دستور اونو دوباره برگردوندیم .

بررسی cookies ها در php

کوکی ها فایل های کوچکی هستند که وب سایت های مختلف از آن برای ذخیره و بازیابی اطلاعات محدودی بر روی کامپیوتر بازدیدکنندگان خود استفاده می کنند.

ایجاد اولین کوکی

کار با کوکی ها در PHP بسیار راحت است. برای ایجاد کوکی ها از تابع setcookieاستفاده می شود که دارای چندین پارامتر می باشد، ساختار دستور را در زیر مشاهده می فرمایید:

```
setcookie(name, value, expire, path, domain, secure);
```

پارامتر name نام کوکی درخواستی برای ایجاد است و از این نام برای مراجعات بعدی به این کوکی استفاده می شود.

پارامتر value حاوی مقداری که می خواهیم در این کوکی با نام name ذخیره شود می باشد.

پارامتر expire تاریخی را مشخص می کند که کوکی تا آن تاریخ باید بر روی کامپیوتر بازدیدکننده نگهداری شود. اگر این مقدار را مشخص نسازید کوکی با بستن مرورگر بازدیدکننده پاک می شود.

پارامتر path مسیری را در سرور مشخص می کند که کوکی برای آن مسیر و تمام زیر شاخه های آن قابل دسترسی است. برای مثال اگر این پارامتر را با کاراکتر '/' مقدار دهی نماییم کوکی برای کل domain یا دامنه قابل دسترسی خواهد بود. اگر آن را برابر '/test/' قرار دهیم کوکی برای دایرکتوری test و تمام زیر شاخه های آن قابل دسترسی خواهد بود و مثلا اسکریپتی در شاخه '/test/subdir' هم می تواند به کوکی مورد نظر دسترسی داشته باشد و اگر اصلا این پارامتر را مقدار دهی نکنیم کوکی فقط برای مسیر جاری و

تمام زیر شاخه های آن که اسکریپت در آن اجرا می شود قابل دسترسی خواهد بود.

پارامتر domain هم نام دامنه ای را مشخص می کند که کوکی باید برای آن قابل دسترس باشد. اگر این پارامتر مقدار دهی نشود کوکی تنها برای domain یا subdomain فعلی که اسکریپت در آن اجرا می شود قابل دسترسی خواهد بود. اگر بخواهیم کوکی برای کل دامنه و تمام subdomainها و یا زیر دامنه ها هم قابل دسترسی باشد باید مقدار آن را برابر '.example.com'قرار دهید که example نام سایت شما خواهد بود. اگر مقدار این پارامتر را 'www.example.com' قرار دهید کوکی ها فقط برای زیردامنه wwwقابل دسترسی خواهند بود.

و آخرین پارامتر هم secure است که مقداری برابر ۰ یا ۱ خواهد داشت. مقدار پیش فرض آن ۰ است و اگر آنرا ۱ قرار دهیم کوکی ها فقط در صورت وجود اتصال secure یا HTTPSبه سرور ارسال می شوند.

معمولا موارد کمی پیش می آید که باید از پارامترهای path و domain و secure استفاده کنیم.

در زیر مثال های مختلف برای آشنایی بهتر شما با این تابع آورده شده است:

با استفاده از مثال پایین یک کوکی با نام size و مقدار ۳ که با بستن مرورگر کاربر پاک میشود ساخته می شود:

```
setcookie ("size", "۳");
```

با استفاده از دستور پایین هم یک کوکی با نام language و مقدار fa که به مدت ۳۶۰۰ ثانیه یا یک ساعت در کامپیوتر بازدیدکننده باقی میماند ساخته می شود:

```
setcookie ("language", "fa", time()+۳۶۰۰);
```

با استفاده از دستور زیر یک کوکی با نام test و مقدار hello ساخته می شود که به مدت یکسال در کامپیوتر کلاینت باقی می ماند و در کل دامنه example.comو تمام زیردایرکتوری های آن قابل دسترسی است:

```
setcookie ("test", "hello", time()+۳۱۵۳۶۰۰۰, "/", ".example.com", ۱);
```

اگر بخواهیم مقدار یک کوکی را که قبلا ساخته ایم تغییر دهیم هم از همین تابع setcookie استفاده می کنیم، کافیست با استفاده از نام کوکی مورد نظر و مقدار جدید تابع را اجرا کنیم و مقدار جدید بطور خودکار جای مقدار قدیمی را می گیرد.

نکات مهم برای استفاده از تابعsetcookie برای کار با کوکی ها:

.۱اجرای تابع setcookie به تعداد دلخواه باید قبل از ارسال header ها و یا بطور کلی هر خروجی (شامل تگ ها، فضاهای خالی و حتی یک خط خالی) به کامپیوتر کلاینت یا بازدیدکننده مورد استفاده قرار گیرد در غیراینصورت با پیغام خطای Cannot modify header information - headers already sentمواجه می شوید.

.۲بعد از ارسال کوکی ها صفحه باید یکبار بروز رسانی شود تا کوکی ها قابل دسترسی توسط سرور باشند.

.۳همانطور که می دانید کاربران اینترنت به راحتی می توانند کوکی ها را غیرفعال سازند هر چند که با اینکار بسیاری از سایتهای نیازمند کوکی ها مانند Yahoo یا Gmail دیگر بدرستی کار نمی کنند، با این حال اگر سایت شما نیازمند کار با کوکی هاست شما باید بعد از ارسال کوکی و به روز رسانی صفحه از دریافت کوکی توسط کاربر مطمئن شوید، اینکار در قسمت بعدی تشریح شده است.

.۴از ذخیره اطلاعات مهم مانند رمز عبور کاربران بصورت ساده و رمز گذاری نشده در کوکی ها خودداری فرمایید چون به راحتی قابل بازیابی توسط دیگر استفاده کنندگان از یک کامپیوتر مشترک می باشد.

دسترسی به مقادیر کوکی های از قبل ذخیره شده

فرض کنید قبلا یک کوکی با نام name که حاوی نام بازدیدکننده است با تاریخ انقضای یکسال برای کاربران ایجاد کردیم و کاربر در این یکسال دوباره به سایت ما وارد می شود، حال می خواهیم بینیم که نام این کاربر چیست:

```
print \$_COOKIE["name"];
```

متغیر \$_COOKIE\$آرایه ای شامل تمام کوکی هایی است که ما قبلا برای یک بازدیدکننده ارسال کرده ایم و ما با استفاده از نام هایی که قبلا کوکی ها را ایجاد کرده ایم می توانیم به مقادیر تک تک کوکی ها دسترسی داشته باشیم.

اگر می خواهید تمامی کوکیهای یک بازدیدکننده را با هم ببینید و یا اگر نام های نسبت داده شده به کوکیهای بازدیدکنندگان سایت خود را فراموش کرده اید می توانید با استفاده از دستور زیر تمامی کوکی های یک بازدیدکننده را نمایش دهید:

```
print_r(\$_COOKIE);
```

حذف یک کوکی

برای حذف کوکی ها هم از همان تابع setcookie استفاده می کنیم فقط با این تفاوت که باید تاریخ انقضای کوکی یا پارامتر expire را یک مقدار در گذشته نسبت دهیم. برای مثال دستور زیر کوکی با نام testرا که قبلا ایجاد شده است حذف می کند:

```
setcookie ("test", "", time() - ۳۶۰۰);
```

در پایان می خواهیم یک مثال کامل از یک نمونه استفاده از کوکی ها را بیان کنیم. فرض کنید سایت شما می خواهد تعداد بازدید از صفحات سایت را توسط هر کاربر برای یک آمارگیری بشمارد:

```
if (isset(\$_COOKIE["visits"])) {
$visits = \$_COOKIE["visits"] + ۱;
setcookie ("visits", $visits, time()+۳۱۵۳۶۰۰۰);
}
else {
$visits = ۱;
setcookie ("visits", $visits, time()+۳۱۵۳۶۰۰۰);
}
print "You have visited this site for ".$visits." time(s)" ;
```

session ها در php

برخلاف کوکی در سرویس دهنده (Server) ذخیره می شود مکان ذخیره Session را میتوان تغییر داد ولی به صورت پیشفرض در حافظه ذخیره می شود برای هم به سرعت قابل دسترسی است ، در هنگام

ساخت یک جلسه (Session) یک کوکی هم در سرویس گیرنده با مقدار آیدی Session ایجاد می گردد ، این به این دلیل است که مقادیر Session عمومی نیست و باید به اضای هر کاربر ایجاد گردد.

نکته مهم این است که به این دلیل که Session در حافظه سرور ذخیره می شود و حافظه از منابع بسیار مهم سرور به حساب می آید استفاده نا بجا از آن می تواند به سرور لطمه وارد کند ، البته معمولا سرویس دهندگان میزبانی زمانی را جهت TimeOut شدن Session قرار میدهند تا مشکلی از نظر مدیریت حافظه پیش نیاید . بنابراین شما قادر نخواهید بود مقادیری را برای مدت طولانی در حافظه ذخیره کنید مگر اینکه محل ذخیره Session را تغییر دهید.

سشن برای ارسال یک متغییر استفاده میشود. برای ساخت یک سشن باید از دستور زیر استفاده کرد:

```
$_SESSION['YOUR SESSION NAME'] =سشن  محتوای
شما ;
```

در کد بالا باید برای سشن خود یک نام انتخاب کنید و محتوای آنرا نیز مشخص گردانید.

برای استفاده از سشن در صفحه باید سشن را پس از ایجاد شدن حفظ نماییم . برای این کار کد زیر را در تمامی صفحاتتان قرر دهید:

```
session_start();
```

برای چاپ کردن سشن از دستور زیر استفاده نمایید:

```
print $_SESSION['YOUR SESSION NAME'] ;
```

برای از بین بردن سشن از دستور زیر استفاده نمایید:

```
if(isset($_SESSION['YOUR SESSION NAME'] )){
session_destroy();
}
```

پایان