

معرفی (زبان اسکریپت نویسی) PHP

نکته: در طول آموزش منظور از زبان PHP همان اسکریپت نویسی به زبان PHP است.

مؤلف: میثم ولی اللهی . بروجرد آبان ۱۳۸۹

WWW.COMBOOK.MIHANBLOG.COM

قسمت اول

مقدمه:

PHP یا Personal Home Page (Hypertext Preprocessor) زبان اسکریپت نویسی توکار HTML است، به عبارتی زبانست برای تکامل و استفاده و همچنین پردازش صفحات HTML به منظور ایجاد و تبدیل آنها به صفحات پویا.

Syntax یا املاء این زبان شباهت زیادی به زبان C و Java دارد یا بهتر است بگوییم پی اچ پی قواعدش را از perl و C و Java به ارث برده، به کمک این زبان برنامه نویس قادر خواهد بود در زمان کمی به ایجاد صفحات پویا برای استفاده در محیط وب اقدام کند. بد نیست بدانید پی اچ پی تنها زبانی برای پردازش کد های HTML نیست و از PHP میتوان برای ایجاد فایل های فلش، تصاویر و فایل های PDF استفاده کرد.

از مزایای PHP علاوه بر رایگان و کدباز بودن آن می توان به قدرت بالا در پشتیبانی از بسیاری از پایگاه های داده اشاره کرد همچنین پی اچ پی از بسیاری از پروتکل های موجود پشتیبانی می کند.

تاریخچه PHP

در سال ۱۹۹۴ برای اولین بار شخصی به نام راسموس لردورف "Rasmus Lerdorf" نسخه اولیه PHP را به منظور دریافت و ثبت بازدید کننده های سایتش بصورت خیلی ابتدایی به کار برد که یک سال بعد نیز نسخه ۲ پی اچ پی را ارائه کرد و این روند تا نسخه ۳ ادامه داشت که در این نسخه PHP به صورت کدباز درآمد و توسط تیمی از برنامه نویسان کدباز توسعه پیدا کرد. در حال حاضر جدید ترین نسخه PHP تا زمان نگارش این مقاله PHP 6 Beta میباشد.

مقایسه PHP بر ASP

اگر بخواهیم در بیان ساده مزیت های PHP را بیان کنیم اولین پارامتر متمایز کننده PHP سرعت بالای اجرای صفحات PHP نسبت به ASP میباشد که در پارامتر دوم میتوان به قابلیت پرتابل بودن PHP یعنی اجرای PHP بر روی انواع سیستم عامل ها اشاره کرد که در مقابل asp از سازگاری بیشتری برخوردار است و در پارامتر بعدی به سینتکس PHP اشاره کرد که برای کسانی که قبلا با زبانهایی مثل C و JAVA کار کرده اند یادگیری PHP بسیار سریعتر و آسانتر است.

در مقابل وقتی برنامه نویس غیر حرفه ای مبادرت به نوشتن برنامه ای به زبان PHP می کند باید اطلاعات وسیعی در زمینه امنیت وب داشته باشد چون در حالت پیش فرض PHP راهکاری برای اصلاح خودکار مشکلات امنیتی ناشی از دانش کم برنامه نویس ارائه نمی کند و برنامه نویس خودش باید مراقب باشد باگ در برنامه اش به حداقل برسد که در مقابل ASP به صورت هوشمند به رفع این مشکلات می پردازد.

در ادامه میتوان اشاره کرد در نسخه تحت عنوان ASP.NET می توان برنامه ها و سینتکس را بر اساس زبان های VB.Net, C#, JAVA, Delphi نوشت که این امر در PHP امکان پذیر نیست.

هنگامی که در ASP مبادرت به نوشتن برنامه های تحت وب می کنید چون شی گزایی در سطح بالاییست در مقایسه با نوشتن برنامه های تحت ویندوز تفاوت زیادی احساس نمیکنید که این امر در PHP کم رنگتر جلوه دارد.

باز در آخر میتوان به کم هزینه تر بودن راه اندازی وب سایت با زبان PHP اشاره کرد که به دلیل ارزان بودن هاست های تحت لینوکس و پشتیبانی PHP از سیستم عامل های مختلف، در مقایسه با ASP هزینه کمتری را دارد. قابل توجه است که سایت های بزرگی مثل یاهو، فیس بوک و ... از PHP به عنوان زبان استفاده شده در سایت هایشان استفاده کرده اند.

نحوه ی اجرای صفحات PHP در کامپیوتر Local (خودمان)

نکته: برای اجرای برنامه های PHP نیاز به فضای هاست با امکان پشتیبانی از PHP دارید که معمولاً سرویس های هاست به صورت پیش فرض دارای نرم افزار ها و ابزار های مورد نیاز برای اجرای صفحات PHP مثل وب سرور و پایگاه داده تحت وب هستند. حال در زیر نحوه ی اجرای صفحات PHP را در کامپیوتر خودمان به منظور تست برنامه یاد می گیریم.

برای اجرای صفحات PHP در کامپیوتر Local نرم افزار های رایگان و مختلفی وجود دارد که در زیر نام چندین نرم افزار مشهور شبیه ساز وب سرور PHP وجود دارد از جمله:

- Easy PHP که اکنون در نسخه ۳ قرار دارد.
- XAMPP که نسخه کنونی آن ۱.۷ است.
- AppServ با نسخه ۲.۶
- WampServer با نسخه ۲
- و ...

توصیه می شود که از نرم افزار Wamp Server برای این کار استفاده کنید که علاوه بر حجم کم نسبت به نرم افزار های مشابه دارای ابزار های :

· Apache

· MySQL

· PHP

· phpMyAdmin

· SQLiteManager

به صورت پکیج و یکجا می باشد.

بد نیست بدانید Apache نام وب سرور صفحات PHP ست و MySQL نسخه رایگان و تحت وب نرم افزار پایگاه داده مشهور و قدرتمند SQL است. از phpMyAdmin برای ساخت پایگاه های داده تحت وب استفاده می شود که محیط ویژال برای ساخت و ویرایش بانک های اطلاعاتی را فراهم می کند.

کار با تمام وب سرورها سازوکار یکسانی دارد که کفایت پس از نصب صفحات PHP (که در ادامه نحوه ی ساخت آن را یاد خواهید گرفت) را در پوشه ای معمولاً با نام www در مسیر نصب نرم افزار (مثلاً C:\wamp\www) قرار دارد کپی کنید سپس در مرورگر خود با تایپ آدرس <http://localhost> یا استفاده از آدرس آی پی <http://127.0.0.1> مبادرت به اجرا و نمایش صفحات موجود در این پوشه کنید.

برای دانلود وب سرور مورد نظر به آدرس زیر مراجعه کنید.

www.wampserver.com

نحوه ی کار با PHP و آزمایش صفحات

بعد از نصب وب سرور نیاز به یک ویرایشگر برای نوشتن کد های پی ایچ پی دارید که میتوانید از Notepad ویندوز برای اینکار استفاده کنید.

پسوند فایل های پی ایچ پی *.php است، بعد از نوشتن دستورات و ذخیره آنها با این پسوند میتوانید آنها را در پوشه وب سرور کپی، و با استفاده از مرورگری نظیر IE، با وارد کردن آدرس کامپیوتر لوکال مثلاً: <http://localhost/test/test1.php> فایل پی ایچ پی مورد نظرتان را اجرا و مشاهده کنید.

قالب برنامه (اسکرپت) های PHP

کد های PHP به صورت استاندارد در بین علامت `<?php` و `>?` قرار میگیرند، میتوان آنها در به شیوه های دیگر مثل `<?>` همچنین بصورت اسکرپتی

`<script language="php">` و `</script>` به کار برد اما بهترین حالت استفاده از روش استاندارد به صورت زیر است:

```
<?php
```

```
دستورات و کد های پی اچ پی
```

```
?>
```

برای مثال به تکه کد زیر توجه کنید.

```
<?php
```

```
Print("Meysam Valiolahi");
```

```
?>
```

خروجی

Meysam Valiolahi

به جای دستور `Print` میتوان از دستور `echo` استفاده کرد.

نکته: اگر بجای `" "` از `' '` استفاده کنیم باعث میشود هرچه میان آن قرار می گیرد چاپ شود، زمانی که لازم است شما مقادیر متغیر ها را به همراه متنی در خروجی چاپ کنید بهتر است از `" "` استفاده کنید به مثال های زیر توجه کنید.

```
<?php
```

```
$user="combook.mihanblog.com";
```

```
Print ("Welcome $user");
```

```
?>
```

خروجی:

Welcome combook.mihanblog.com

```
<?php
```

```
$user="combook.mihanblog.com";  
  
Print ('Welcome $user');  
  
?>
```

خروجی:

Welcome \$user

انواع داده در PHP :

پی اچ پی از انواع داده مختلفی مانده سایر زبان های پشتیبانی می کند با این تفاوت که کمی کار را برای برنامه نویس راحت تر کرده و نیاز به معرفی نوع داده، در زمان تعریف متغیر ها، نیست کافیسست متغیری را تعریف کنید، در لحظه ای که مقداری در متغیر می ریزید به صورت خودکار نوع آن مشخص می شود به این نوع داده نوع داده پویا میگویند که با تغییر مقدار متغیر نوع آن نیز تغییر میکند. php متغیرها را بصورت تصادفی مقدار اولیه میدهد.

نحوه ی تعریف متغیر

```
[مقدار]= نام متغیر $
```

همانطور که در بالا مشاهده میکنید به سادگی میتوانید با نوشتن علامت \$ قبل از نام متغیر متغیری را تعریف کنید که با علامت انتساب = می توانید مقداری را درون متغیر بریزید. قابل توجه است مانند سایر زبان های نام متغیر باید با حروف شروع شود و در انتهای دستورات علامت ; را قرار داد.

مثال:

در زیر چند شکل صحیح از نحوه ی تعریف متغیرها را مشاهده میکنید:

```
$Meysam ;  
  
$meysam=10;  
  
$meysam_10;  
  
$m1;
```

همانطور که مشاهده کردید برای مراجعه به یک متغیر یا تغییر مقدار آن باید همیشه \$ را قبل از نام متغیر بنویسید و در صورت عدم رعایت این نکته زمان اجرای برنامه با خطا روبرو میشوید.

-چندین متغیر نادرست:

```
$1389;  
$1389shamsi;  
$12_a;  
$_meysam;
```

نکته : PHP مانند زبان C خاصیت Case-Sensitive دارد، یعنی به کوچک و بزرگی حروف حساس است یعنی متغیر \$M با \$m برابر نیست و دو متغیر جدا هستند .

توابع کار با نوع داده

همانطور که دیدید php بصورت پیش فرض نوع داده ی متغیرها را تعیین میکند، این موضوع گاهی باعث افزایش فضای رم و گاهی باعث ایجاد خطاهایی در زمان استفاده از متغیر های میشود (هر چند تبدیل و کار با عملگر ها بصورت هوشمند عمل میکند) برای همین در php تابعی داریم که نوع متغیر را برمیگرداند و برعکس تابعی داریم که نوع یک متغیر را به نوع دیگر تبدیل میکند.

```
<?php  
$x="combook.mihanblog.com"  
  
Print gettype($a);  
?>
```

خروجی

String

```
<?php
$Num=12.9;

Print gettype($Num) . "<br>";

Settype($Num,integer);

Print gettype($Num) . "<br>";

Print "$Num" . "<br>";

?>
```

خروجی

```
double
integer
12
```

علاوه بر این چندین تابع داریم که با خروجی بولی نوع متغیر های ما را مشخص می کنند.

این توابع شامل:

```
is_numeric(),is_null(),is_int(),is_string(),is_array(),is_object,is_bool(),is_float()
```

میباشند، نحوه ی کار بسیار ساده است کفایت متغیری را به عنوان ورودی به تابع ارسال کنید در صورتی که متغیر با تابع مورد نظر هم نوع باشد خروجی تابع بصورت True میباشد.

برای بررسی یک متغیر که مقدار اولیه دارد یا نه،تابعی به نام `isset()` با خروجی بولی وجود دارد.

نحوه ی تعریف ثابت

ثابت نوعی متغیر است که مقدار آن در طول برنامه تغییر نمیکند و توسط یک نام با مقدار خاص در ابتدای برنامه یا هر جا که نیاز باشد تعریف میشود سپس با مراجعه و نوشتن نام آن از به مقدارش دسترسی پیدا میکنیم.

گاهی در طول اجرای برنامه نیاز میشود از داده ای که کاربر نتواند مقدار آنرا تغییر دهد استفاده کنید مثلا اگر بخواهید برنامه ای بنویسید که مالیاتی را از حقوق کارمندان یک بانک کم کند کفایت میزان مالیات مثلا ۱۰ هزار تومان را درون یک ثابت بریزید و در طول برنامه ثابت را از حقوق کارمندان کم کنید حال فرض کنید سال بعد مالیات به ۱۲ هزار تومان تغییر پیدا مینکند در این حالت

به اهمیت ثابت پی میبیریم که کافیت فقط مقدار ثابت را تغییر دهیم تا تاثیر آن برای سایر کارمندان اعمال شود که اگر این کار به وسیله ی متغیر های معمولی انجام میشود بایستی مقدار تمام متغیر ها را جداگانه با مقدار جدید جایگزین میکردیم.

ثابت در php به صورت زیر تعریف میشود.

```
<?php
```

```
define('مقدار ثابت','نام ثابت');
```

```
?>
```

عملگر ها

در جدول زیر انواع عملگر ها را مشاهده میکنید.

ریاضی	/ تقسیم، + جمع، - تفریق، * ضرب، % باقیمانده
افزایشی و کاهش	-- کاهش، ++ افزایش
انتساب	=
منطقی	And ، or ، ! نقیض
مقایسه ای	< کوچکتر، > بزرگتر، == مساوی، != نامساوی

دستورات شرطی

برای کنترل بر اجرای برنامه از دستورات و عبارات شرطی استفاده می کنیم مثلا زمانی که می خواهیم گزرواژه کاربر را بررسی کنیم که در صورت صحت به صفحه ای دستیابی داشته باشد و یا عکس این عمل نیاز به دستورات شرطی داریم. در زیر دستورات شرطی موجود در php قابل مشاهده است.

دستور if

if (شرط)

دستور ۱

else

دستور ۲

مثال:

```
<?php
$user="meysam";
$password=1389;
If ($user="meysam" and $password=1389)
    {
        Print ("Welcome");
    }
else
    {
        Print("Error!");
    }
?>
```

همانند زبان C میتوان در صورت تک دستوری بود جواب شرط از قرار دادن آن در {} صرف نظر کرد. همچنین میتواند از بررسی شرط در صورت عدم برقرای نیز صرف نظر و تکه کد else را ننویسیم. همچنین امکان استفاده از if بصورت تو در تو نیز وجود دارد.

دستور Switch

زمانی که نیاز به بررسی حالت های مختلف یک شرط داشته باشیم استفاده از تسور if به صورت تودرتو کمی پیچیده می شود از این دستور برای بررسی چندین حالت شرط استفاده می شود در زیر ساختار دستور را می توانید مشاهده کنید.

```
<?php
```

```
Switch ()
```

```
{
```

```
Case 1 مقدار :
```

```
    دستور ۱
```

```
break;
```

```
Case 2 مقدار:
```

```
    دستور ۲
```

```
break;
```

```
    .
```

```
    .
```

```
    .
```

```
default:
```

```
    دستور
```

```
}
```

```
?>
```

همانطور که مشاهده کردید بعد از هر case نیاز به نوشتن break برای خروج از دستور switch نیاز است که در نهایت اگر هیچ یک از case ها اجرا نشود دستور بعد از default اجرا میشود.

حلقه ها

در تمام زبان ها از حلقه ها برای تکرار دستورات استفاده می شود در زیر ساختار حلقه های for، while را میتوانید مشاهده کنید.

حلقه ی for

از این حلقه زمانی استفاده میشود که تعداد دفعات تکرار را اطلاع داریم مثلاً زمانی که میخواهیم کلمه ی combook.mihanblog.com را ۱۰۰ بار چاپ کنیم.

(گام حلقه؛ شرط پایان حلقه؛ شمارنده \$) For

```
{  
  
    دستورات;  
  
}
```

برای خروج زود هنگام از حلقه میتوان از دستور break استفاده کرد.

برای ادامه حلقه از دستور continue استفاده می شود.

حلقه ی while

زمانی که از تعداد دفعات تکرار اطلاع نداریم میتوان با استفاده از این حلقه تا زمان برقراری شرط به تکرار دستورات پردازیم.

(شرط) While

```
{  
  
    دستورات;  
  
}
```

توابع در php

از توابع برای جلوگیری از تکرار بی مورد کد های استفاده کنیم مثلا تابعی می نویسیم که اطلاعات و مشخصات کاربر را چاپ کند یا تابعی مینویسیم که به کاربر ایمیلی ارسال کند. در این حالت تابع شامل دستورات ارسال ایمیل میشود که نام کاربر را به عنوان ورودی میگیرد و براساس دستورات موجود نامه ای به او ارسال میکند. (پارامتر های ورودی تابع با ، از هم جدا میشوند)

Function (پارامتر های ورودی) نام تابع

```
{  
دستورات;  
}
```

برای فراخوانی تابع کافیه نام تابع را بصورت (); function name بنویسید.

نکته: متغیر هایی که درون تابع تعریف میشوند در خارج از تابع قابل دسترسی نیستند و برعکس، زمانی که میخواهیم به متغیر های خارج از تابع دسترسی داشته باشیم باید قبل از نام متغیر از کلمه ی کلیدی global استفاده کنیم تا تابع به متغیر خارج از محدوده یعنی متغیر های سراسری دسترسی داشته باشد.

```
<?php  
$MeySam="www.combook.mihanblog.com";  
  
Function a()  
{  
    Print (" $meysam");  
}  
?>
```

```
<?php
$Meysam="www.combook.mihanblog.com";

Function b()
{
    Global $Meysam;
    Print ("{$Meysam}");
}
?>
```

www.combook.mihanblog.com

مقدار بازگشتی تابع

در واقع شکل قبلی بیشتر به روال ها شبیه بود تا یک تابع، کار اصلی تابع بدین شکل است که پس از اجرای یک سری کد خروجی را به کاربر بدهد مثلاً پس از فراخوانی تابع جستجو، جستجو در بانک وب سایت انجام شده و نتیجه جستجو را برای ما برگرداند.

برای ارسال جواب تابع به خروجی از دستور زیر استفاده می کنیم.

```
Function نام تابع ()
{
    دستورات تابع;
    Return نام متغیر خروجی;
}
```

مستند سازی

Php نیز به رسم سایر زبان ها به نوشتن توضیحات درون کد های اصلی اجازه میدهد، این کار باعث میشود که بعدا زمانی که برنامه ی شما بزرگتر شده از سردرگمی دور شوید و به راحتی با خواندن توضیحاتی که از قبل نوشته اید از عملکرد متغییر ها و توابع و ... با خبر شوید.

```
<?php
// توضیحات یک خطی
# توضیحات یک خطی
/*
توضیحات چند خطی
*/
?>
```

آرایه ها

آرایه را میتوان یک سطر از خانه یا محل های هم نوع به حساب آورد که هر خانه دارای آدرسی به نام اندیس است اندیس آرایه در php از صفر شروع می شود و هم میتوان بجای اندیس از حروف استفاده کرد، مجموع این خانه ها در یک محل از حافظه و فقط با یک نام قابل دسترسی و فراخوانیست.

Php قاعده ی کلی آرایه ها که در سایر زبان ها به یک شکل است را بطور جالبی تغییر داده مثلا یک آرایه میتواند دارای عناصر غیر هم نوع باشد، همچنین نیاز به تعریف نوع آرایه نیست، همچنین طول آرایه را نیاز نیست تعریف کنیم و میتوان در هر زمان با استفاده از دستوراتی که گفته خواهد شد عنصری را به آرایه اضافه کنیم.

آرایه یک بعدی را در php اگر به صورت رکورد در نظر بگیریم بهتر میتوانیم آن را درک کنیم چون فیلد های این رکورد یا همان عناصر آن بر خلاف سایر زبان های میتواند دارای مقادیر با نوع های مختلف باشد.

نحوه ی تعریف آرایه

```
<?php
$info=array("meysm ","valiolahi ",1389);
Print "$info[2]";
?>
```

خروجی

1389

همانطور که در ابتدا گفته شد آرایه یک بعدی در php حکم رکورد را دارد حال میتوانید بجای اندیس عددی رکورد فرضیتان را با حروف قابل دسترسی کنید، مثلا با نوشتن نام مقدار فیلد نام یا همان عنصر نام را چاپ کرد.

```
<?php
$info=array(fname=>"meysm ",lname=>"valiolahi ",year=>1389);
Print "$info[fname]";
?>
```

خروجی

meysam

در زیر مثالی کامل را میبینید که آرایه ای را ایجاد میکند و بعد از مقدار دهی آن را چاپ میکند. توجه داشته باشید که اگر مقدار اندیس آرایه مانند مثال زیر وارد نشود php اندیس را از صفر شروع کردن و بصورت خود کار یک واحد اضافه تر برای عنصر بعدی در نظر میگیرد، همانطور که مشاهده خواهید کرد برای افزودن عنصر به یک آرایه کافیت از دستور `$ArrayName[]=New Value;` استفاده کنید که نیاز نیست اندیس به صورت دستی وارد شود و php هوشمندانه عنصر جدید را اندیس گذاری میکند. همچنین در بدنه ی حلقه دستور `count` برای به دست آوردن طول آرایه به کار برده شده:

```

<?php
$test=array();//تعریف آرایه
$test[]="meysam";//مقدار دهی به آرایه
$test[]="valiolahi";
$test[]=10;
For ($i=0 ;$i<=$test[count($test)-1];$i++)// پیمایش آرایه
{
    Print $test[$i] . " ";// چاپ آرایه
}
?>

```

خروجی

meysam valiolahi 10

در زیر نیز برنامه ای را مشاهده می کنید که آرایه ای را با مقادیر ۱ تا ۱۰ مقدار دهی می کند سپس مبادرت به چاپ آرایه میکند.

```

<?php
$ar=array();
for($j=0;$j<=10;$j++)
{
    $ar[$j]=$j;
}
for ($k=0;$k<=$ar[count($ar)-1];$k++)
{

```



```
print $ar[$k] . " ";  
}  
?>
```

برای پیمایش عناصر آرایه از حلقه ی foreach نیز میتوان استفاده کرد.

```
(متغیر موقت جهت نگداری عناصر آرایه $ as نام آرایه)$foreach  
{  
}
```

اگر آرایه با اندیس الفبایی تعریف شده باشد ساختار حلقه foreach به صورت زیر است.

```
(متغیر موقت نگداری عناصر $ => متغیر نگداری متغیر الفبایی اندیس $ as نام آرایه)$foreach  
{  
}
```

چاپ مقادیر آرایه مثال قبل با حلقه ی foreach :

```
<?php  
foreach($ar as $arS)  
{  
    print $arS . " " ;  
}  
?>
```

برای دسترسی به آخرین عنصر آرایه زمانی که شماره ی اندیس ها به ترتیب نباشد توسط تابع زیر امکان پذیر است.

```
<?php
```

```
Print end($ArrayName);
```

```
?>
```

توابع پر کاربرد در زمینه کار با آرایه ها

اولین عنصر آرایه را حذف و به عنوان خروجی تابع برمیگرداند.	Array_shift()
یک مقدار را به انتهای آرایه اضافه میکند.	Array_push()
یک مقدار را از انتهای آرایه حذف می کند.	Array_pop()
آرایه دوم را به انتهای آرایه اول اضافه میکند.	Array_merge()
عناصر یک آرایه را در آرایه دیگری میریزد.	Ayyay_value()
بررسی خالی یا پر بودن آرایه.	In_array()
ترتیب عناصر را برعکس می کند.	Array_reverse()
آرایه را مرتب میکند.	Sory()
آرایه با اندیس الفبایی را مرتب میکند.	Ksort()
آرایه را بصورت معکوس مرتب میکند.	Rsort()
نوع دیگری از مرتب سازی آرایه	Usort()
نوع دیگری از مرتب سازی آرایه	Hatsort()
مقداری را در آرایه جستجو میکند و در صورت پیدا شدن اندیس آن را برمیگرداند	Array_search()
اندیس های آرایه را برمیگرداند	Array_keys()