

php

پی اچ پی

مقدمه



- زبان برنامه نویسی پی اچ پی در سال 1994 توسط Rasmus Lerdorf ابداع شد
- پی اچ پی رایگان است
- آخرین نسخه پی اچ پی 7.3 است
- طبق آخرین آمارها بیش از 78 درصد سایتها از زبان برنامه نویسی پی اچ پی استفاده می کنند

	2018 1 Feb	2018 1 Mar	2018 1 Apr	2018 1 May	2018 1 Jun	2018 1 Jul	2018 1 Aug	2018 1 Sep	2018 1 Oct	2018 1 Nov	2018 1 Dec	2019 1 Jan	2019 1 Feb	2019 26 Feb
PHP	80.1%	79.9%	79.9%	79.7%	79.6%	79.4%	79.3%	79.1%	79.0%	78.9%	78.9%	78.9%	78.9%	78.9%
ASP.NET	13.5%	13.3%	13.0%	12.7%	12.6%	12.5%	12.3%	12.2%	12.1%	12.0%	11.9%	11.8%	11.6%	11.5%
Java	3.4%	3.5%	3.5%	3.6%	3.7%	3.7%	3.8%	3.8%	3.9%	3.9%	3.9%	4.0%	4.0%	4.0%
Ruby	1.6%	1.7%	1.8%	1.8%	1.8%	1.9%	2.0%	2.1%	2.1%	2.2%	2.3%	2.4%	2.4%	2.5%
static files	1.6%	1.7%	1.7%	1.8%	1.8%	1.8%	1.9%	2.1%	2.2%	2.2%	2.1%	2.1%	2.1%	2.1%
Scala	0.5%	0.6%	0.7%	0.8%	0.9%	0.9%	1.0%	1.1%	1.1%	1.2%	1.2%	1.2%	1.3%	1.3%
Python	1.3%	1.3%	1.3%	1.2%	1.2%	1.2%	1.2%	1.2%	1.2%	1.2%	1.1%	1.1%	1.1%	1.1%
JavaScript	0.4%	0.4%	0.4%	0.5%	0.5%	0.6%	0.6%	0.6%	0.6%	0.6%	0.6%	0.7%	0.7%	0.7%
ColdFusion	0.6%	0.6%	0.6%	0.6%	0.6%	0.5%	0.5%	0.5%	0.5%	0.5%	0.5%	0.5%	0.5%	0.5%
Perl	0.3%	0.3%	0.3%	0.4%	0.4%	0.4%	0.3%	0.3%	0.3%	0.3%	0.3%	0.3%	0.3%	0.3%
Erlang	0.1%	0.1%	0.1%	0.1%	0.1%	0.1%	0.1%	0.1%	0.1%	0.1%	0.1%	0.1%	0.1%	0.1%



منبع آمار: https://w3techs.com/technologies/history_overview/programming_language

مقدمه

- زبان پی اچ پی یک زبان سمت سرور یا server side محسوب می شود
- زبان پی اچ پی یک زبان چند سکویی یا cross platform است (بر روی تمامی سیستمهای عامل قابل اجرا است)
- براحتی با زبان html ادغام میشود یا html embedded است
- اکثر سیستم های مدیریت محتوای (cms) مشهور همانند جوملا، وردپرس ، دروپال و ... با این زبان نوشته شده اند
- سایتهای مشهوری مانند

[Facebook.com](https://www.facebook.com)

[Wikipedia.org](https://www.wikipedia.org)

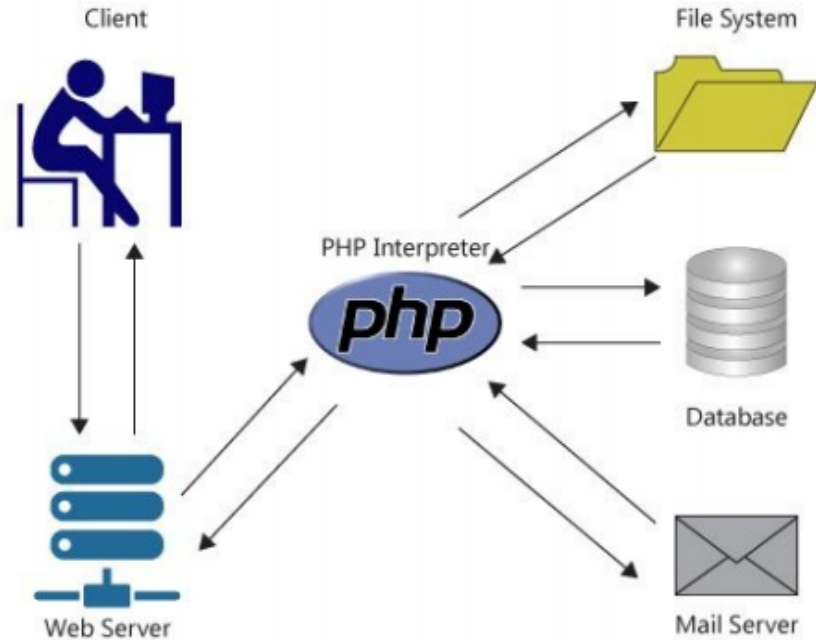
[Twitter.com](https://www.twitter.com)

با زبان پی اچ پی نوشته شده اند

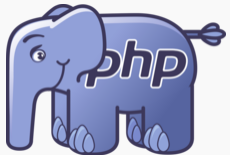


معرفی

- نقش مفسر پی اچ پی و وب سرور

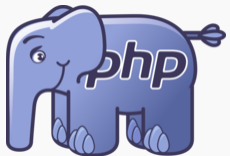
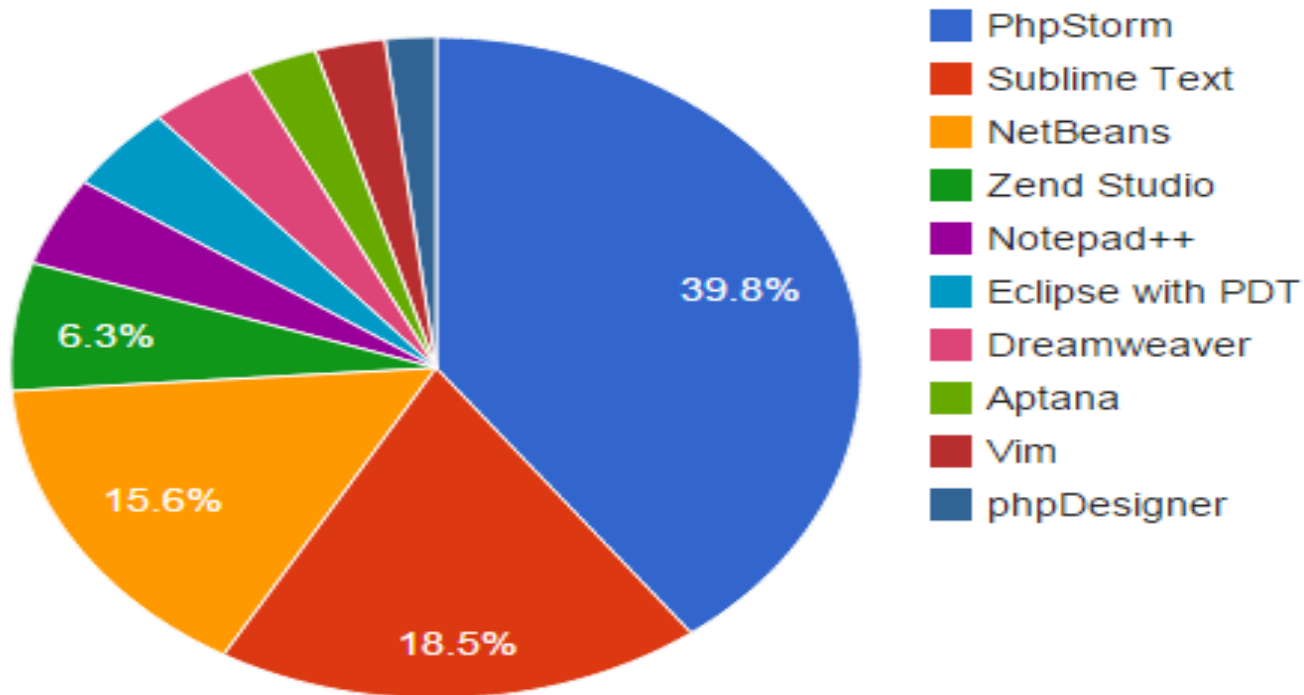


Working of PHP



phpstorm

- محبوبترین محیط برنامه نویسی پی اچ پی در بین برنامه نویسان phpstorm است :



wamp

- ومپ یا WAMP مخفف کلمات PHP ، MySQL ، Apache ، Windows می باشد و بستری فراهم می کند که یکباره و بدون نیاز به انجام تنظیمات جانبی ، تمامی سرویس های ذکر شده را بر روی سیستم عامل ویندوز بصورت یکجا نصب و راه اندازی کنید .
- برای سیستم عاملهای (لینوکس و مک) نرم افزارهای مشابه lamp , mamp را داریم



LAMP



MAMP



WAMP

مقدمات

```
<?php  
echo "hello world";  
?>
```

• بلاک شروع کد پی اچ پی <? php statements ?>

• دستور چاپ:

• خاتمه دستور با semicolon (;) مشخص می شود .

• نحوه کامنت کردن متن برای مستند سازی کد :

• برای یک خط

```
// این یک کامنت است  
# این شیوه متفاوت دیگری برای ایجاد کامنت است
```

• برای چندین خط



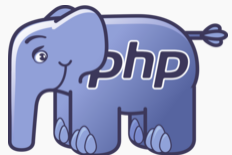
```
/*  
برای استفاده از  
چند خط کامنت برای  
مستند سازی از این شیوه استفاده می شود  
*/
```

عملگرها

- عملگرهای ریاضی

عملگر	نام	مثال	نتیجه مثال
+	جمع	10+3	13
-	منها	10-3	7
*	ضرب	10*3	30
/	تقسیم	10/3	3.33333333333333
%	باقیمانده	10%3	1
**	توان	10**3	1000

- عملگر الحاق



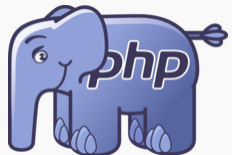
```
echo "hello". " world";
```



```
hello world
```


عملگرهای ترکیبی

عملگر	مثال	معادل
+=	<code>\$x += 5;</code>	<code>\$x = \$x + 5;</code>
-=	<code>\$x -= 5;</code>	<code>\$x = \$x - 5;</code>
*=	<code>\$x *= 5;</code>	<code>\$x = \$x * 5;</code>
/=	<code>\$x /= 5;</code>	<code>\$x = \$x / 5;</code>
%=	<code>\$x %= 5;</code>	<code>\$x = \$x % 5;</code>
**=	<code>\$x **= 5;</code>	<code>\$x = \$x ** 5;</code>
.=	<code>\$x .= "test";</code>	<code>\$x = \$x . "test";</code>



عملگرهای مقایسه ای

عملگر	نام	جواب true خواهد بود اگر...	مثال : \$x = 4;	نتیجه
=	تساوی	طرفین عملگر با هم از نظر مقدار مساوی باشند	\$x==5	false
!=	نامساوی	طرفین عملگر با هم از نظر مقدار مساوی نباشند	\$x != 5	true
===	معادل	طرفین عملگر هم از نظر نوع و هم از نظر مقدار باید با یکدیگر مساوی باشند	\$x === "4"	false
>	بزرگتر	طرف سمت چپ عملگر از طرف سمت راست عمگر بزرگتر باشد	\$x > 4	false
<	کوچکتر	طرف سمت چپ عملگر از سمت راست عملگر کوچکتر باشد	\$x < 4	false
>=	بزرگتر یا مساوی	طرف سمت چپ عملگر از طرف سمت راست عمگر بزرگتر یا مساوی باشد	\$x >= 4	true
<=	کوچکتر یا مساوی	طرف سمت چپ عملگر از سمت راست عملگر کوچکتر باشد	\$x <= 4	true



عملگرهای منطقی

عملگر	جواب true خواهد بود اگر ...	مثال	نتیجه
or	اگر حداقل یکی از طرفین عملگر true باشد	True or false	true
	اگر حداقل یکی از طرفین عملگر true باشد	True false	true
xor	اگر فقط و فقط یکی از طرفین عملگر true باشد	True xor true	false
and	اگر هر دو طرف عملگر true باشد	False and true	false
&&	اگر هر دو طرف عملگر true باشد	True && false	false
!	فقط بر روی یک عملوند اعمال می‌شود در صورتی که آن عملوند false باشد	! false	true

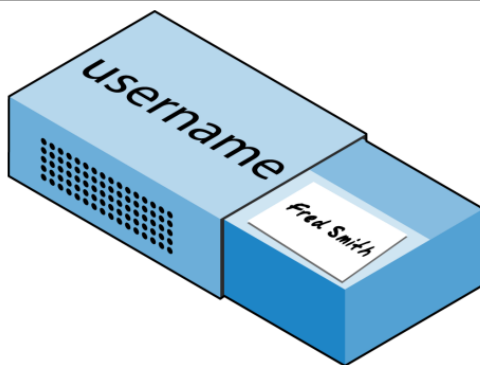


متغیرها

- متغیرها با علامت \$ مشخص می شوند مانند \$x
- در نام گذاری متغیرها از حروف الفبا، اعداد، خط زیرین (_) باید استفاده کرد و نباید از فاصله در نام گذاری استفاده نمود ضمناً حرف اول آن باید از حروف الفبا و یا خط زیرین باشد.
- در پی اچ پی نیازی به تعریف نوع متغیر نیست و بعد از اولین مقدار دهی نوع آن مشخص می شود
- دستور انتساب با علامت = مشخص می شود مثال :

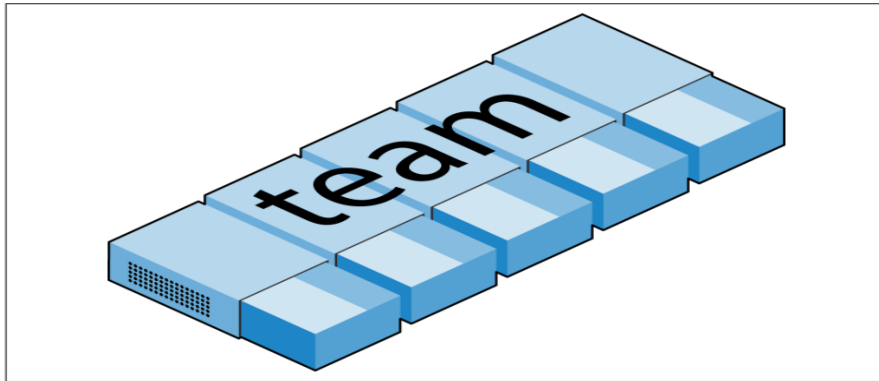
```
$username = "fred smith";
```

```
$username = "Fred Smith";
```



نوع	مثال
Boolean	True
integer	5
Float /double	3.14
string	"hello"

آرایه

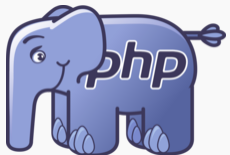


- آرایه متغیری است که چندین مقدار را در خود جای می‌دهد
- متغیر تیم آرایه ای است که از پنج بازیکن تشکیل شده:
رونالدو ، مسی، دایی، عابدزاده، عزیزی

```
$team = ["ronaldo", "messi", "daeii", "abedzadeh", "azizi"];
```

شیوه قدیمی تر تعریف آرایه:

```
$daysOfWeek =  
array("sat", "sun", "mon", "tue", "wed", "thu", "fri");
```



آرایه

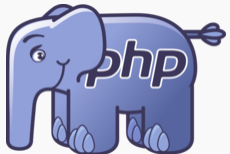
- دسترسی به عناصر آرایه با اندیس ها : در آرایه روزهای هفته ما هفت عنصر داریم که اولین آن یعنی شنبه در مکان شماره صفر و جمعه در مکان شماره شش قرار دارد

```
echo $daysOfWeek[2];
```

```
$team[] = "bagheri";
```

```
echo count($daysOfWeek);
```

- با دستور مقابل دوشنبه چاپ می شود
- با دستور مقابل یکی به اعضای آرایه تیم اضافه می شود و شماره اندیس 5 به آن تعلق میگیرد
- با تابع از پیش تعریف شده count تعداد عناصر یک آرایه مشخص می شود در دستور مقابل عدد 7 چاپ خواهد شد



آرایه

• عناصر یک آرایه می توانند نوع های `$person = ["ali", "javadi", 21, 13.5];`

متفاوتی داشته باشند در آرایه `person` عنصر اول و دوم که نام و نام خاوادگی یک شخص را مشخص می کند از نوع رشته عنصر سوم که سنش را مشخص می کند از نوع عدد صحیح و عنصر آخر که معدلش را مشخص میکند از نوع اعشار می باشد .

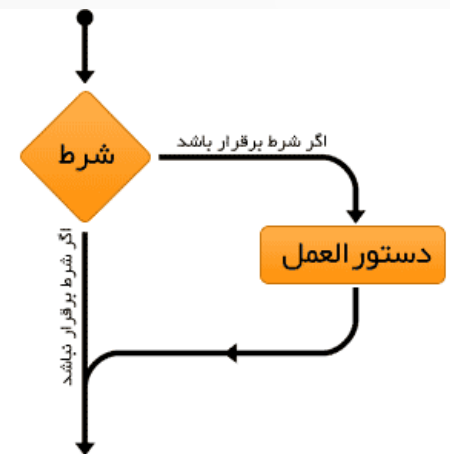


دستورات شرطی

- دستورات شرطی و حلقه ها جزو عناصر اصلی و جدایی ناپذیر هر زبان برنامه نویسی ساختار یافته هستند و تقریبا در تمامی برنامه ها کاربرد دارند. دستورات شرطی با برقراری شرط خاصی، مجموعه ای از دستورات عملی را اجرا میکنند.

- **دستور شرطی if**

- نحوه ی استفاده از دستور if به این شکل است که ابتدا شرط را بیان میکنیم و سپس با استفاده از آکولاد باز " { " و آکولاد بسته " } " دستورات عملی را بیان میکنیم. به ساختار زیر توجه کنید:



```
<?php
if (شرط) {
    دستور العمل
}
?>
```


مثال برای دستور شرطی if

- با استفاده از تابع از پیش تعریف شده rand نمره یک دانشجو را بین 0 تا 20 بصورت تصادفی تعیین می کنیم و در صورتی که بزرگتر یا مساوی 10 بود تبریک قبولی می گوییم

```
<?php
$avg = rand(0,20);
echo " شما از امتحان نمره " . $avg . " گرفتی <br>";
if($avg >=10 )
    echo " شما قبول شدی - تبریک ";
?>
```



شما از امتحان نمره 11 گرفتی
شما قبول شدی - تبریک



شما از امتحان نمره 5 گرفتی

خروجی وقتی که شرط برقرار است

خروجی وقتی که شرط برقرار نیست



دستور if ..else

- فرق این دستور با دستور if در این است که شما حالت دومی را نیز برای شرط در نظر دارید، یعنی اگر شرط اول برقرار بود، دستوالعمل مناسب اجرا میشود ولی اگر شرط برقرار نبود دستوالعمل جایگزین اجرا میشود (بر خلاف دستور if که اگر شرط اول برقرار نبود هیچ دستوالعملی اجرا نمیشد).

```
<?php
if(شرط){
    دستوالعمل
}
else{
    دستوالعمل
}
?>
```

```
<?php
$avg = rand(0,20);
echo "گرفتی " . $avg . " شما از امتحان نمره ";
if($avg >=10 ){
    echo "شما قبول شدی - تبریک ";
}
else{
    echo "شما این درس را افتادی - متاسفم ";
}
?>
```



شما از امتحان نمره 16 گرفتی
شما قبول شدی - تبریک

خروجی وقتی که شرط برقرار است



شما از امتحان نمره 7 گرفتی
شما این درس را افتادی - متاسفم

خروجی وقتی که شرط برقرار نیست



دستور elseif

- این دستور این امکان را فراهم میکند تا در صورت عدم برقراری شرط دستور if، شرطهای دیگری را نیز بررسی کنیم.

```
<?php
$abcd;
$avg = rand(0,20);
echo "<br>گرفتی " . $avg . " شما از امتحان نمره ";
if($avg < 5 ){
    $abcd = "D";
}
elseif($avg >=5 && $avg <10){
    $abcd = "C";
}
elseif($avg >=10 && $avg <15){
    $abcd = "B";
}
elseif($avg >=15 ){
    $abcd = "A";
}
echo $abcd;
?>
```

- مثال : نمره دریافتی بر اساس سیستم 0 تا 20 را به سیستم نمره دهی ABCD تبدیل می کنیم



شما از امتحان نمره 2 گرفتی
D



شما از امتحان نمره 11 گرفتی
B



دستور switch

- می توان بجای شرطهای متعدد elseif از دستور switch استفاده کرد:
- اجزای دستور switch عبارتند از :
 - switch** : متغیر یا عبارتی که باید مقادیر مختلف آن بررسی شود
 - case** : مقادیر مختلف و مرتبط با متغیر مقابل عبارت switch تعیین می شود .
 - default** : در صورت عدم برقراری تمام مقادیر موجود در case ها دستورات مقابل default انجام می گیرد
 - break** : برای جلوگیری از چک کردن case بعدی باید از دستور break برای خروج استفاده نمود



مثال switch

```
$avg = rand(0, 20);  
echo " شما از امتحان نمره " . $avg . " گرفتی <br>";  
if ($avg < 5) {  
    $abcd = "D";  
} elseif ($avg >= 5 && $avg < 10) {  
    $abcd = "C";  
} elseif ($avg >= 10 && $avg < 15) {  
    $abcd = "B";  
} elseif ($avg >= 15) {  
    $abcd = "A";  
}  
echo $abcd;  
switch ($abcd) {  
    case "A":  
        echo "نمره عالی";  
        break;  
    case "B":  
        echo "نمره خوب";  
        break;  
    case "C":  
        echo "نمره ضعیف";  
        break;  
    default:  
        echo "نمره بد";  
}  
}
```



- مثال قبلی را با یک switch تکمیل کردیم



شما از امتحان نمره 11 گرفتی
نمره خوب B

شما از امتحان نمره 16 گرفتی
نمره عالی A

شما از امتحان نمره 2 گرفتی
نمره بد D

شما از امتحان نمره 6 گرفتی
نمره ضعیف C

دستورات حلقه- while

- در بسیاری از مواقع ما نیاز به تکرار اجرای یک یا چند دستور داریم برای این منظور از حلقه ها در برنامه نویسی استفاده می نماییم .

while(شرط)

{

دستورات برای تکرار //

}

- دستور while :

- حلقه while با کلمه کلیدی while شروع شده و در مقابل آن در داخل پرانتز شرطی که برای اجرای حلقه در نظر داریم را می نویسیم.

- حلقه while تا زمانی اجرا می شود که شرط داخل پرانتز مقدار true برگرداند.

- دستورات مرتبط با حلقه که باید تکرار شوند را داخل {} می نویسیم

- دستوراتی که داخل {} نوشته می شوند، می توانند یک یا چندین دستور باشند.

- دستورات مربوط به حلقه while در صورت برقرار نبودن شرط در ابتدای حلقه اجرا نخواهند شد.



دستورات حلقه - while

```
$x=1;  
while ($x<=10)  
    echo($x++);
```



- یک مثال ساده برای چاپ اعداد 1 تا 10 با استفاده از حلقه while

12345678910

- با استفاده از دستورات حلقه می توان عناصر یک آرایه را چاپ نمود :

```
$index = 0;  
while($index <count($daysOfWeek))  
{  
    echo $daysOfWeek[$index]. " ";  
    $index++;  
}
```



sat sun mon tue wed thu fri



دستورات حلقه do while-

- در حلقه while، شرط اجرای حلقه قبل از رسیدن به دستورات بررسی می شود، اما در ساختار do..while شرط در انتهای حلقه بررسی می شود، به همین خاطر و بر خلاف دستور while، دستورات حلقه حداقل یکبار اجرا می شوند. در زیر ساختار کلی حلقه do..while را مشاهده می کنید:

```
do
{
    // دستورات برای تکرار
} while(شرط)
```

```
$x=1;
do{
    echo $x++;
}while($x<=10);
```

- چاپ 1-10 با استفاده از حلقه do..while

```
do{
    echo "سلام به همه";
}while(false);
```

- در کد مقابل متن سلام به همه چاپ شده و به شرط حلقه که میرسد چون شرط برقرار نیست از حلقه خارج می



حلقه for

```
for(init; condition; increment)
{
    //تكرار
}
```

- ساختار کلی دستور for به صورت مقابل است:

- همانطور که مشاهده می کنید این حلقه با کلمه کلیدی for آغاز شده و داخل پرانتز و مقابل کلمه کلیدی for سه بخش داریم که با کاراکتر ; از هم جدا شده اند.

- بخش init: متغیری که داخل حلقه استفاده می شود را تعریف و مقدار دهی اولیه می کنیم

- بخش condition: شرطی را مشخص می کند که حلقه بر اساس آن اجرا خواهد شد

- بخش increment: مشخص می کنید با هر بار تکرار، متغیری که در بخش init تعریف شده است چه افزایشی پیدا می کند .

- چاپ اعداد 1-10 با استفاده از حلقه for

```
for($i=1;$i<=10;$i++)
    echo $i;
```



حلقه foreach

```
foreach (array_expression as $value)
{
    statement
}
```

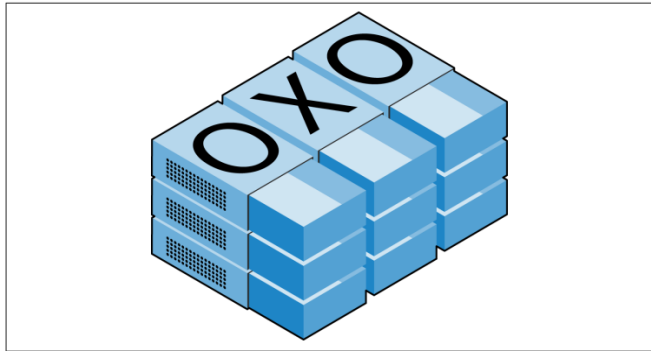
- این دستور بهترین و راحتترین دستور برای کار کردن با آرایه ها است:

- حلقه foreach همونطور که از اسمش پیدااست به ازای هر آیتm در یک آرایه، کدهای داخل بلاک رو اجرا میکنه. یعنی اگر پنج ایتm در یک آرایه داشته باشیم حلقه ما ۵ بار تکرار میشه. معمولا زمانی که نمیدونیم حلقه ما باید چند بار اجرا بشه یا از تعداد ایتm ها خبر نداریم از foreach استفاده میکنیم.

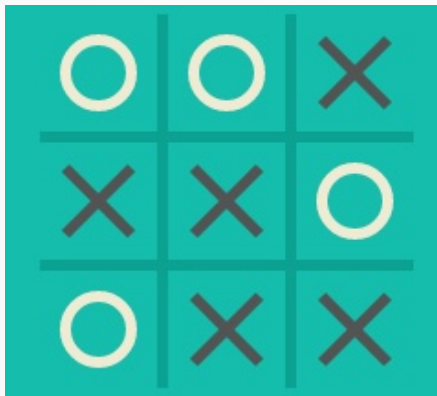
```
foreach($daysOfWeek as $day)
echo $day." ";
```



آرایه چند بعدی



- آرایه ای که مقدار هر عنصرش خود یک آرایه دیگر است
- مثال بازی دوز آرایه `xox` وضعیت بازی را نشان میدهد که خب آنچه که مشخص است بازی مساوی تمام شده است



```
$oxo = [  
    ['o', 'o', 'x'],  
    ['x', 'x', 'o'],  
    ['o', 'x', 'x']  
];
```



آرایه چند بعدی

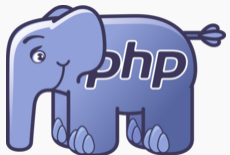
مثال: آرایه ای که مشخصات سه دانشجو را نشان می‌دهد که هر دانشجو آرایه ای است از مقادیری که نشان دهنده نام، سن، معدل می‌باشد.

```
$students = [ ["ardeshir", 23, 16.35],  
["abolfazl", 20, 14.53], ["mehdi", 21, 18.50] ];
```

برای دسترسی و تغییر دادن مقادیر آرایه دو بعدی باید از دو اندیس استفاده کنیم

در این مثال معدل ابوالفضل را به 17.25 تغییر داده‌ایم.

```
$students[1][2]=17.25;
```



استفاده از foreach برای آرایه دو بعدی

```
foreach($oxo as $value)
    echo "<br>".$value[0]." ".$value[1]." ".$value[2];
```



```
o o x
x x o
o x x
```

```
$students = [
    ["ardeshir", 23, 16.35],
    ["abolfazl", 20, 14.53],
    ["mehdi", 21, 18.50]
];
foreach($students as $student)
    echo "<br> name = ".$student[0].", age = ".
        $student[1].", avg = ".$student[2];
```



```
name = ardeshir, age = 23, avg = 16.35
name = abolfazl, age = 20, avg = 14.53
name = mehdi, age = 21, avg = 18.5
```

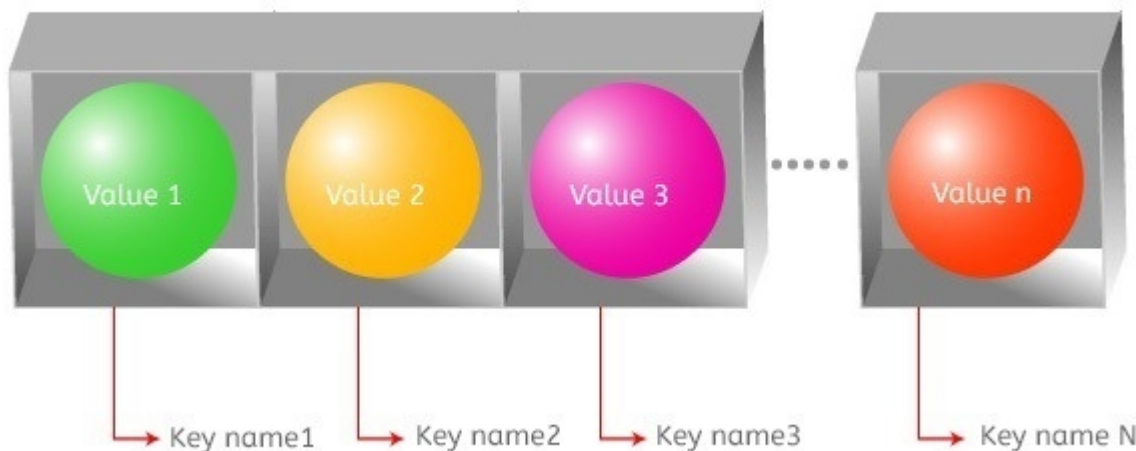


آرایه های انجمنی associative array

- آرایه انجمنی یا (Associative Arrays) آرایه هایی هستند که با یک نام کلیدی خاص خود فراخوان می شود. در شکل زیر هر یک از دایره ها عضوهای مقابل آرایه است که هر کدام با نام کلیدی خاص خود (key name) فراخوان می شود.

```
$student=["name"=>"sajjad",  
         "family"=>"habibzadeh",  
         "age"=> 22  
];
```

- برای دسترسی به عناصر آرایه انجمنی بجای شماره اندیس از برچسب و یا کلیدشان استفاده می کنیم



```
echo $student["age"];
```

عدد ۲۲ چاپ خواهد شد



مقایسه آرایه انجمنی با آرایه اندیس دار

Before



After



A teachable
TEACHER

دسترسی به مقادیر و کلیدها در حلقه foreach

```
$student=["name"=>"sajjad",  
         "family"=>"habibzadeh",  
         "age"=> 22  
];
```

```
foreach($student as $key=>$value)  
    echo "<br>".$key ." = " . $value;
```



```
name = sajjad  
family = habibzadeh  
age = 22
```

```
foreach ($student as $value)  
    echo "<br>".$value;
```



```
sajjad  
habibzadeh  
22
```

- در مثال مقابل در حلقه foreach هم مقادیر و هم کلیدها را در اختیار می‌گیریم

- اگر نیازی به دسترسی به کلیدها نداشته باشیم میتوانیم به همان شیوه قبلی فقط به مقادیر عناصر آرایه دسترسی پیدا کنیم در حلقه foreach

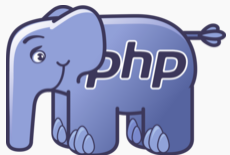


آرایه دو بعدی انجمنی

```
$students=[  
    ["name"=>"sajjad","family"=>"habibzadeh","age"=> 22],  
    ["name"=>"amin","family"=>"hajbagheri","age"=>23],  
    ["name"=>"milad","family"=>"molaei","age"=>20]  
];
```

دسترسی به عناصر آرایه دو بعدی انجمنی: در مثال مقابل سن
میلااد ملایی به 21 سال افزایش پیدا کرد

```
$students[2]["age"]=21;
```



چاپ عناصر آرایه دو بعدی انجمنی با استفاده از حلقه foreach

```
foreach ($students as $student) {  
    foreach ($student as $key => $value) {  
        echo " ".$key ." = " . $value;  
    }  
    echo "<br>";  
}
```

شیوه اول با استفاده از دو حلقه foreach :



```
name = sajjad family = habibzadeh age = 22  
name = amin family = hajbagheri age = 23  
name = milad family = molaei age = 21
```

شیوه دوم با استفاده از یک حلقه :

```
foreach ($students as $student) {  
    echo "<br> name = ".$student["name"]." family = ".$student["family"]  
        ." age = ".$student["age"];  
}
```



```
name = sajjad family = habibzadeh age = 22  
name = amin family = hajbagheri age = 23  
name = milad family = molaei age = 21
```

عناصر form در html

```
<form>  
input elements  
</form>
```

- فرم های HTML برای انتقال داده ها به یک سرور استفاده می شود.
 - یک فرم شامل سه بخش مهم می باشد:
- (1) تگ form که شامل URL صفحه ای که اطلاعات فرم را پردازش می کند.
 - (2) عناصر فرم، مانند کادرهای متنی، کادرهای علامت، لیست ها و ...
 - (3) دکمه Submit که داده هایی را که از کاربر گرفته را برای صفحه پردازشگر رویداد سرور میفرستد.



عناصر فرم

- مهم ترین عنصر فرم، عنصر `<input>` است.
- عنصر `<input>` برای انتخاب اطلاعات کاربر مورد استفاده قرار می گیرد.
- یک عنصر `<input>` می تواند بسته به صفت نوع ، در بسیاری جهات متفاوت باشد . یک عنصر `<input>` می تواند نوعی از فیلد متن، چک باکس، رمز عبور، دکمه های رادیویی، دکمه ارسال و... باشد.
- **فیلد متنی:**

```
<form>  
First name :<br>  
<input type="text" name="firstname"> <br>  
Last name :<br>  
<input type="text" name="lastname">  
</form>
```



First name:

Last name:



عناصر فرم

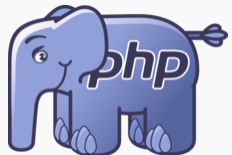
- **دکمه های رادیویی:** دکمه های رادیویی به کاربر این امکان را میدهد تا تنها یکی از تعداد محدود گزینه ها را انتخاب کند : در مثال زیر چون برای دکمه رادیویی اول از خصوصیت checked استفاده شده است بصورت پیشفرض انتخاب شده می باشد

```
<form >
  <input type="radio" name="sex" value="male" checked>Male <br>
  <input type="radio" name="sex" value="female">Female <br>
</form>
```



☒ Male
☐ Female

- **چک باکس:** چک باکس ها را علامت بزند. چون امکان ارسال چند مقدار وجود دارد نام عناصر را یکسان و بصورت ارایه ای می نویسیم و بمانند عناصر رادیویی با خصوصیت checked بصورت پیشفرض انتخاب شده می باشد



```
<form>
<input type="checkbox" name="favorites[]" value="music" checked> music <br>
<input type="checkbox" name="favorites[]" value="movie"> movie <br>
</form>
```



☒ music
☐ movie

عناصر فرم

```
<form>
  <select name="country">
    <option value="iraq"> iraq</option>
    <option value="iran" selected> iran</option>
    <option value="usa"> usa</option>
    <option value="china"> china</option>
  </select>
</form>
```



لیست کشویی:

لیست کشویی

بطور پیش فرض بگونه ای است که کاربر حق انتخاب یک گزینه را دارد اما با اضافه کردن خصوصیت multiple می توان از لیست کشویی چندین گزینه را انتخاب نمود، در ضمن عناصری که بیش از یک مقدار ارسال می کنند نامشان را بصورت آرایه ای می نویسیم مثلا در اینجا در مقابل خصوصیت name نوشته ایم countries[]

```
<form>
<select name="countries[]" multiple>
  <option value="iraq"> iraq</option>
  <option value="iran" selected> iran</option>
  <option value="usa"> usa</option>
  <option value="china"> china</option>
</select>
</form>
```

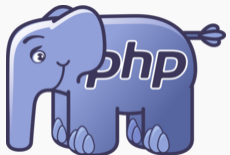


عناصر فرم

- **Textarea** شبیه فیلد متنی می باشد فقط فضای بیشتری را در اختیار کاربر می گذارد تا بتواند متن بیشتری وارد کند

```
<form>  
  <textarea name="msg" rows="8" cols="80" placeholder="put your message here">  
  </textarea>  
</form>
```

put your message here



عناصر فرم

- **دکمه ارسال :** در واقع این نوع دکمه ها وظیفه ارسال فرم به صفحه پردازشگر را دارند.

```
<form action="form_process.php" method="get">  
First name :<br>  
<input type="text" name="firstname"> <br>  
Last name :<br>  
<input type="text" name="lastname"> <br>  
<input type="submit" value="send">  
</form>
```



First name :

Last name :

خصوصیت action در فرم : فرم ها بعد از اینکه توسط کاربر پر می شوند می بایست برای یک صفحه وب برای پردازش که معمولا به یکی از زبان های تحت سرور همانند php , asp.net , jsp ... نوشته می شوند. فرستاده شوند. ما باید در این خصوصیت آدرس صفحه پردازشگر که قرار است فرم به آن ارسال شود را مشخص نماییم.



عناصر فرم

- **خصوصیت method در فرم:** همانطور که از اسم این صفت پیداست روش ارسال فرم را مشخص می کند. خصوصیت متد دو مقدار زیر را می پذیرد:

get (1)

post (2)

- این دو روش فرق هایی با هم دارند :
- (1) متد get از URL (آدرس) صفحه برای ارسال داده ها استفاده می کند و آن ها را در قالب جفت های name=value به آدرس صفحه پردازشگر متصل می کند. اما متد post به صورت مخفی (در واقع از طریق Http Request ها) برای این کار استفاده می کند.



عناصر فرم

```
<form action="form_process.php" method="get">  
First name :<br>  
<input type="text" name="firstname"> <br>  
Last name :<br>  
<input type="text" name="lastname"> <br>  
<input type="submit" value="send">  
</form>
```



First name :

Last name :

• اگر کاربر مقادیر بالا را در فرم وارد کند و دکمه ارسال را کلیک کند به آدرس زیر فرستاده می شود

از آن جا که متد get از آدرس برای ارسال

داده ها استفاده می کند، امکان نمایش محتویات فیلدها (حتی فیلدهای رمزی) در نوار آدرس وجود دارد که

'form_process.php?firstname=mahdi&lastname=asadi'

این نشان دهنده عدم امنیت کافی در ارسال داده از این طریق است. اما متد post به دلیل ارسال مخفی

داده ها، امنیت کافی دارد.



عناصر فرم

- (3) تفاوت سوم در محدودیت حجم داده های ارسالی از طریق URL توسط مرورگرها معمولا در ارسال به این طریق محدودیت قائل می شوند که البته این مورد از مرورگر تا مرورگر متفاوت است. اما در روش ارسال داده ها از طریق متد post هیچ محدودیتی در حجم داده های ارسالی وجود ندارد.
- (4) یکی دیگر از تفاوت های این دو روش در سرعت ارسال داده هاست که معمولا سرعت انجام این کار از طریق get کمی بیشتر از post است.



متغیرهای فوق سراسری - superglobal

- متغیرهای فوق سراسری (superglobal) در PHP ، متغیرهایی هستند که از پیش تعریف شده اند و همیشه در دسترس می باشند و همه جا می توانید به آنها دسترسی داشته باشید.
- برای پردازش اطلاعات فرمهای ارسالی ما از دو متغیر فوق سراسری زیر استفاده می کنیم :

`$_GET`

از این متغیر در صفحه پردازش فرم زمانی استفاده می کنیم که متد فرم ارسال شده get باشد

`$_POST`

از این متغیر در صفحه پردازش فرم زمانی استفاده می کنم که متد فرم ارسال شده post باشد



متغیرهای فوق سراسری - superglobal

- در صفحه پردازش فرم متغیرهای فوق سراسری `$_GET` و `$_POST` یک آرایه انجمنی هستند که برچسب (کلید) هایشان نام عناصر فرم و مقادیرشان مقدار وارد یا انتخاب شده توسط کاربر می باشد .
- مثال برای فرمی که اطلاعات را با متد `get` ارسال می کند

```
<form action="form_process.php" method="get">  
  First name : <input type="text" name="firstname"> <br>  
  Last name : <input type="text" name="lastname"> <br>  
  <input type="submit" value="send">  
</form>
```

صفحه form.html

First name :

Last name :



```
<?php  
var_dump($_GET);  
?>
```

صفحه form_process.php

```
array (size=2)  
  'firstname' => string 'abolfazl' (length=8)  
  'lastname' => string 'ayyoubi' (length=7)
```

متغیر فوق سراسری

- مثال برای فرمی که اطلاعات را با متد post ارسال می کند

```
<form action="form_process.php" method="post">  
  First name : <input type="text" name="firstname"> <br>  
  Last name : <input type="text" name="lastname"> <br>  
  <input type="submit" value="send">  
</form>
```



First name :

Last name :

صفحه form.html

```
<?php  
var_dump($_POST);  
?>
```



```
array (size=2)  
  'firstname' => string 'mehdi' (length=5)  
  'lastname' => string 'sanji' (length=5)
```

صفحه form_process.php



لازم به ذکر است تابع var_dump نوع و اندازه متغیری که بعنوان پارمتر ورودی میگیرد را مشخص می کند

یک فرم ساده که اطلاعاتش از طریق متد get فرستاده می شود

```
<form action="formprocess.php" method="get">
  <table align="center" border="1">
    <tr>
      <td> First name :</td><td><input type="text" name="firstname"> </td>
    </tr>
    <tr>
      <td>Last name :</td><td><input type="text" name="lastname"></td>
    </tr>
    <tr>
      <td colspan="2" align="right"><input type="submit" value="send"> </td>
    </tr>
  </table>
</form>
```



صفحه form.html



First name :	<input type="text" value="karim"/>
Last name :	<input type="text" value="bagheri"/>
<input type="submit" value="send"/>	

پردازش فرم که از طریق متد get فرستاده شده بود

```
if(!empty($_GET["firstname"]) && !empty($_GET["lastname"]))  
{echo "<ul>";  
    echo "<li> First name = " . $_GET["firstname"] . "</li>";  
    echo "<li> Last name = " . $_GET["lastname"] . "</li>";  
    echo "</ul>";  
}  
else  
    header("location:form.html");
```

صفحه formprocess.php



test/formprocess.php?firstname=karim&lastname=bagheri

- First name = karim
- Last name = bagheri



- همانطور که می بینید چون اطلاعات فرم از طریق متد get فرستاده شده است داده های وارد شده در فرم در url قابل مشاهده اند
- با استفاده تابع empty چک می شود که آیا یک متغیر دارای مقدار هست یا خیر
- تابع header ما رابه آدرس مقابل ذکر شده می برد

اتصال به پایگاه داده mysql

- برای ایجاد ارتباط با سرور MySQL از تابع `mysqli_connect()` استفاده میشود.
- پارامترهای ورودی این تابع بدین شرح می باشند :

```
mysqli_connect(host,username,password,dbname);
```

نام پارمتر	توضیح
host	آی پی و یا نام دامنه سرور mysql که می خواهیم متصل شویم
username	نام کاربری در mysql
password	رمز عبور
dbname	نام پایگاه داده ای که می خواهیم بدان متصل شویم



```
$link = mysqli_connect("localhost","root","","university");
```

مثال :

بستن ارتباط

- با استفاده از تابع `mysqli_close` ارتباط با پایگاه داده `mysql` را قطع می نماییم

```
$link = mysqli_connect("localhost","root","","university");  
mysqli_close($link);
```

- در مثال بالا ابتدا با تابع `mysqli_connect` به دیتابیس `university` متصل شدیم و این اتصال را در متغیر `link` ذخیره کردیم و در خط بعد اتصال قطع شده است



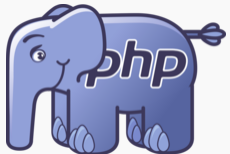
اجرای دستورات sql

- بعد از اتصال به دیتابیس با استفاده از تابع `mysqli_query` می توانیم دستورات sql را بر روی پایگاه داده خود اجرا نماییم (ایجاد جدول - افزودن رکورد به یک یا چند جدول و ...)
- پارامتر اول تابع ، اتصال به پایگاه داده را مشخص می کند و پارامتر دوم مشخص می کند چه کوئری یا دستور sql باید بر روی پایگاه داده اجرا شود.
- مثال :
- اگر در Mysql پایگاه داده ای بنام `university` و جدولی بنام `student` ایجاد کرده باشیم :

```
mysqli_query($link,$query);
```

Field	Type	Null	Key	Default	Extra
sid	int(11)	NO	PRI	NULL	auto_increment
sname	varchar(40)	NO		NULL	
sfamily	varchar(50)	NO		NULL	
savg	float	YES		NULL	

• جدول student



mysqli_query

- کد زیر دانشجویی بنام اردشیر سلطانی را به جدول دانشجو اضافه می کند.

```
$link = mysqli_connect("localhost","root","","university");  
$query = "insert into student (sname,sfamily,savg) values ('ardeshir','soltani',18.45)";  
mysqli_query($link,$query);  
mysqli_close($link);
```

```
mysql> select * from student;
```

sid	sname	sfamily	savg
1	ardeshir	soltani	18.45

1 row in set (0.00 sec)

رکوردهای جدول
دانشجو در کنسول

sid	sname	sfamily	savg
1	ardeshir	soltani	18.45

جدول دانشجو در
phpmyadmin



mysqli_set_charset

- اگر بخواهیم داده هایی که از حروف فارسی تشکیل شده اند به پایگاه داده بفرستیم بدرستی ذخیره و نمایش داده شوند بایستی علاوه بر اینکه یونیکد پایگاه داده خود را utf8 قرار می دهیم ، ارتباط با پایگاه داده را هم از طریق تابع mysqli_set_charset به utf8 تبدیل کنیم که تمام کاراکترهای زبانهای مختلف دنیا را پشتیبانی میکند (از جمله فارسی) در غیر اینصورت با کاراکترهای ناخوانا مواجه خواهیم شد

```
$link = mysqli_connect("localhost","root","","university");  
mysqli_set_charset($link,'utf8');  
$query = "insert into student (sname,sfamily,savg) values ('اردشیر','soltani',18.45)";  
mysqli_query($link,$query);  
mysqli_close($link);
```



sid	sname	sfamily	savg
1	ardeshir	soltani	18.45
2	اردشیر	soltani	18.45



mysqli_fetch_all

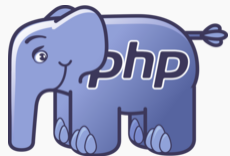
- برای واکشی رکوردهای یک جدول ابتدا از طریق تابع `mysqli_query` یک کوئری (`select...`) می فرستیم نتیجه این کوئری را در یک متغیر ذخیره نموده و از طریق تابع `mysqli_fetch_all` رکوردهای بازگشتی را در یک آرایه دو بعدی ذخیره می نماییم که هر رکورد جدول یک سطر آرایه را تشکیل می دهد

```
$link = mysqli_connect("localhost","root","","university");  
mysqli_set_charset($link,'utf8');  
$query = "select * from student ";  
$result=mysqli_query($link,$query);  
$rows=mysqli_fetch_all($result);  
var_dump($rows);
```

تابع `var_dump` نوع و اندازه یک متغیر را برای ما مشخص می کند همانطور که در خروجی مشخص شده یک آرایه دو عنصری داریم که هر عنصرش خود یک آرایه 4 عنصری است که برابر با مقادیر رکوردهای جدول دانشجو است



```
array (size=2)  
  0 =>  
    array (size=4)  
      0 => string '1' (length=1)  
      1 => string 'ardeshir' (length=8)  
      2 => string 'soltani' (length=7)  
      3 => string '18.45' (length=5)  
  1 =>  
    array (size=4)  
      0 => string '2' (length=1)  
      1 => string 'اردشیر' (length=12)  
      2 => string 'soltani' (length=7)  
      3 => string '18.45' (length=5)
```



mysqli_fetch_all

- شکل عمومی استفاده از تابع mysqli_fetch_all بشرح زیر است :

```
mysqli_fetch_all( result , resulttype ) ;
```

پارامتر	توضیح
result	این پارامتر خروجی کوئری mysqli_query زمانی که بخواهیم تعدادی رکورد را واکنشی کنیم استفاده می شود - اجباری
resulttype	این پارامتر تعیین می کند که تابع اطلاعات خود را در قالب چه نوع آرایه ای برگرداند و می تواند یکی از سه حالت زیر را داشته باشد : MYSQLI_ASSOC آرایه دو بعدی شامل جفت کلید و مقدار MYSQLI_NUM آرایه اندیس دار MYSQLI_BOTH هر دو استفاده از این پارامتر اختیاری است .



mysqli_fetch_all

- مثال برای زمانی که مقدار پارامتر resulttype برابر با MYSQLI_ASSOC باشد

```
$link = mysqli_connect("localhost","root","","university");
mysqli_set_charset($link,'utf8');
$query = "select * from student ";
$result=mysqli_query($link,$query);
$rows=mysqli_fetch_all($result,MYSQLI_ASSOC);
var_dump($rows);
```



```
array (size=2)
  0 =>
    array (size=4)
      'sid' => string '1' (length=1)
      'sname' => string 'ardeshir' (length=8)
      'sfamily' => string 'soltani' (length=7)
      'savg' => string '18.45' (length=5)
  1 =>
    array (size=4)
      'sid' => string '2' (length=1)
      'sname' => string 'اردشیر' (length=12)
      'sfamily' => string 'soltani' (length=7)
      'savg' => string '18.45' (length=5)
```

همانطور که از خروجی پیداست زمانی که از
MYSQLI_ASSOC استفاده می‌کنیم نتیجه یک آرایه دو بعدی
انجمنی می‌باشد که برای هر سطر کلیدهای آرایه نام فیلد
جدول مورد نظر می‌باشد



• ساختار جدول
student



Field	Type	Null	Key	Default	Extra
sid	int(11)	NO	PRI	NULL	auto_increment
sname	varchar(40)	NO		NULL	
sfamily	varchar(50)	NO		NULL	
savg	float	YES		NULL	