

# فصل ۴

## توابع

هدف‌های رفتاری:

پس از آموزش این فصل هنرجو می‌تواند:

۱- مهم‌ترین انواع توابع رشته‌ای مطرح شده را توضیح داده و هر

یک را به کار ببرد.

۲- در برنامه از توابع تاریخ و زمان استفاده کند.

۳- از توابع مهم ریاضی در برنامه PHP استفاده کند.

۴- با توجه به نیاز برنامه، توابع جدیدی را تعریف کند.

در PHP حدود ۷۰۰ تابع به صورت درون ساخته وجود دارد که برنامه نویسان به راحتی می توانند در برنامه های خود از آنها استفاده کنند. برخی از این توابع مربوط به متغیرهای رشته ای هستند و به منظور کار با رشته ها طراحی شده اند. در ادامه با مهمترین توابع کار با رشته ها و چند نمونه تابع پر کاربرد دیگر آشنا خواهید شد.

## ۴-۱- توابع رشته ای

### ۴-۱-۱- تابع strlen

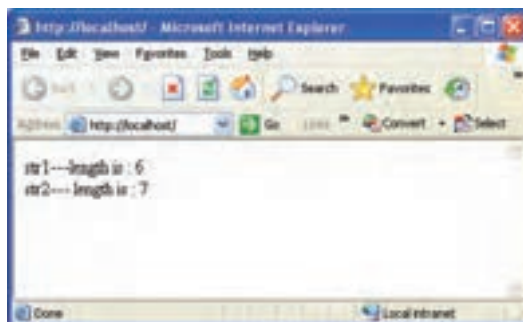
این تابع طول رشته ای را که به عنوان پارامتر ورودی به آن داده می شود برمی گرداند. در نتیجه مقدار برگشت داده شده آن یک عدد صحیح است. به عبارت دیگر کاراکترهای تشکیل دهنده یک رشته را شمرده و تعداد آنها را برمی گرداند. ساختار این تابع به صورت زیر است :

strlen ("پارامتر ورودی");

### مثال :

```
<?php
$str1 = "abcdef";
echo "$str1". "----length is : ".strlen($str1). "<br/>";
$str2 = " ab cd ";
echo "$str2". "----length is : ".strlen($str2);
?>
```

خروجی :



شکل ۴-۱- نتیجه اجرای تابع strlen

## ۲-۱-۴- تابع () strtoupper

این تابع، رشته‌ای را که به عنوان پارامتر ورودی می‌گیرد، به حروف الفبای بزرگ انگلیسی تبدیل می‌کند.

### مثال:

```
<?php
$str1 = "aBc def";
echo "$str1". "----strtoupper is : ".strtoupper($str1). "<br/>";
?>
```

خروجی :

aBc def ---- strtoupper is: ABC DEF

## ۳-۱-۴- تابع () strtolower

این تابع عکس strtoupper عمل می‌کند و تمام حروف پارامترهای ورودی را به حروف کوچک الفبای انگلیسی تبدیل می‌کند.

### مثال:

```
<?php
$str1 = "aBc def";
echo "$str1". "----strtolower is : ".strtolower($str1). "<br/>";
?>
```

خروجی :

aBc def ---- strtolower is: abc def

## ۲-۴- توابع تاریخ و زمان

### ۱-۲-۴- تابع () date

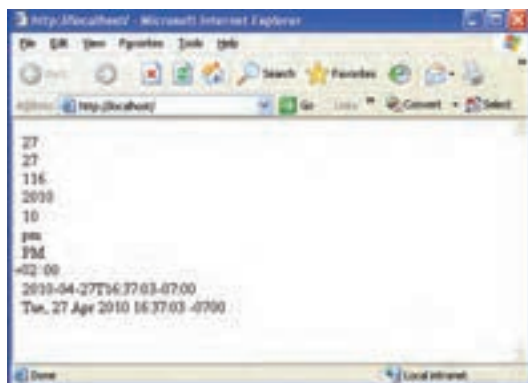
یک پارامتر را به عنوان ورودی دریافت می‌کند و سپس یک رشته را به عنوان خروجی برمی‌گرداند. پارامتر ورودی یک قالب درخواستی برای برگرداندن تاریخ است. هر کدام از قالب‌ها یک شناسه استاندارد دارند که در جدول ۴-۱ چند نمونه از این شناسه‌ها بیان شده‌اند.

## جدول ۴-۱

شناسه	عملکرد	مثال (نمونه رشته خروجی)
d	شماره روز ماه به صورت عدد (در صورتی که شماره تک رقمی باشد به ابتدای آن صفر را اضافه می‌کند)	01 تا 31
j	شماره روز ماه به صورت عدد (بدون درج صفر)	1 تا 31
z	تعداد روزهای گذشته از سال	0 تا 365
Y	شماره سال را به صورت یک عدد ۴ رقمی برمی‌گرداند.	1999 یا 2003
y	شماره سال را به صورت یک عدد ۲ رقمی برمی‌گرداند.	99 یا 03
a	فقط تعیین می‌کند که قبل از ظهر است یا بعد از ظهر (با حروف الفبای کوچک انگلیسی)	am or pm
A	فقط تعیین می‌کند که قبل از ظهر است یا بعد از ظهر (با حروف الفبای بزرگ انگلیسی)	AM or PM
P	تفاوت زمانی منطقه جاری را با زمان به وقت گرینویچ نشان می‌دهد.	+02: 00
c	سال - روز - ماه - ساعت - دقیقه - ثانیه	2010_02_12T15: 19: 21+00: 00
r	سال - روز - ماه - ساعت - دقیقه - ثانیه	Thu, 21 Dec 2000 16: 01: 07 +0200

### مثال:

```
<?php
echo date("d")."<br/>";
echo date("j")."<br/>";
echo date("z")."<br/>";
echo date("Y")."<br/>";
echo date("y")."<br/>";
echo date("a")."<br/>";
echo date("A")."<br/>";
echo date("P")."<br/>";
echo date("c")."<br/>";
echo date("r")."<br/>";
?>
```



شکل ۴-۲

### ۴-۲-۲- تابع time()

تابع time() در PHP تعداد ثانیه‌های گذشته از اولین روز سال ۱۹۷۰ میلادی تا لحظه جاری را نشان می‌دهد.

اگر کد زیر را اجرا کنید، شکل خروجی این تابع را مشاهده خواهید کرد.

**مثال:**

```
<?php
$Now = time();
echo $Now;
?>
```

خروجی:

۱۲۷۲۴۱۲۰۶۷

تابع time() معمولاً در محاسبهٔ زمان انقضای یک عملیات خاص مثلاً در آزمون‌های online و یا همان‌طور که در فصل‌های آینده خواهید دید در بحث cookie ها و جلسه‌ها کاربرد دارد.

### ۴-۳- توابع ریاضی

#### ۴-۳-۱- تابع abs()

این تابع به منظور محاسبه قدر مطلق یک پارامتر به کار می‌رود.

**مثال:**

```
<?php
$num1 = -243;
```

```
echo (" $num1 " . "<br/>");
$num2 = abs($num1);
echo "$num2" . "<br/>";
?>
```

خروجی :

```
-۲۴۳
۲۴۳
```

### ۲-۳-۴- تابع floor ()

این تابع عددی را که به عنوان پارامتر ورودی دریافت می کند به عدد صحیح کوچک تر تبدیل می نماید و مقدار اعشار آن را از بین می برد.  
مثال زیر با استفاده از پارامترهای مختلف، رفتار این تابع را به خوبی نشان می دهد.

**مثال :**

```
<?php
echo(floor(0.60) . "<br />");
echo(floor(0.40) . "<br />");
echo(floor(5) . "<br />");
echo(floor(5.1) . "<br />");
echo(floor(-5.1) . "<br />");
echo(floor(-5.9))
?>
```

خروجی :

```
0
0
5
5
-6
-6
```

### ۳-۳-۴- تابع fmod ()

این تابع دو یا چند پارامتر را دریافت و باقیمانده حاصل تقسیم پارامتر اول بر پارامتر دوم

را برمی گرداند.

### مثال:

```
<?php
$r = fmod(15,2);
echo $r
?>
```

خروجی :

1

### ۴-۳-۴- تابع max ()

این تابع دو یا چند پارامتر را به عنوان ورودی دریافت می کند، سپس عدد بزرگ تر را به عنوان خروجی برمی گرداند.

### مثال:

```
<?php
echo(max(5,7,2,0) . "<br />");
echo(max(_3,5,0) . "<br />");
echo(max(_3,_5) . "<br />");
echo(max(7.25,7.30))
?>
```

خروجی :

7  
5  
-3  
7.3

### ۵-۳-۴- تابع min ()

این تابع دو یا چند پارامتر را به عنوان ورودی دریافت می کند، سپس عدد کوچک تر را به عنوان خروجی برمی گرداند.

### مثال:

```
<?php
echo(min(5,7,2,0) . "<br />");
echo(min(_3,5,0) . "<br />");
```

```
echo(min(-3,-5) . "<br />");
echo(min(7.25,7.30))
?>
```

خروجی :

```
0
-3
-5
7.25
```

### ۶-۳-۴- تابع rand ()

از تابع rand () برای تولید اعداد صحیح تصادفی استفاده می‌شود. اگر این تابع بدون پارامتر ورودی فراخوانی شود، عددی که تولید می‌شود مقداری بین صفر و بیشترین عدد صحیح روی سیستم خواهد داشت. اما چنانچه بخواهید محدوده مشخصی برای تولید اعداد تصادفی در نظر بگیرید باید دو پارامتر ابتدا و انتهای محدوده را به عنوان ورودی تابع قرار دهید.

به طور مثال حاصل فراخوانی تابع به صورت rand() عدد تصادفی صحیح و بزرگ‌تر از صفر خواهد بود. اما با فراخوانی تابع به شکل rand(۱۰۰,۱۰۰۰) عددی تصادفی در محدوده بین ۱۰۰ و ۱۰۰۰ تولید خواهد کرد.

### مثال:

```
<?php
echo(rand() . "<br />");
echo(rand() . "<br />");
echo(rand(10,100)) . "<br />");
echo(rand(100,300))
?>
```

خروجی :

```
2776
37974
35
186
```



### ۷-۳-۴- تابع (round)

این تابع اعداد اعشاری را به نزدیک ترین اعداد صحیح تبدیل می کند.

**مثال:**

```
<?php
echo(round(0.60) . "<br />");
echo(round(0.50) . "<br />");
echo(round(0.49) . "<br />");
echo(round(-4.40) . "<br />");
echo(round(-4.60))
?>
```

خروجی:

```
1
1
0
-4
-5
```

### ۴-۴- تعریف توابع جدید

همان طور که گفته شد در زبان PHP توابع متعددی در نظر گرفته شده اند که در روند برنامه نویسی بسیاری از مسائل را تسهیل می کنند. اما ممکن است یک برنامه نویس در نظر داشته باشد یک سری عملیات مشخص را به دفعات در برنامه خود استفاده کند. در این گونه مواقع می توان بدون نیاز به درج کدهای تکراری، یک تابع طراحی نمود و هر جای برنامه که لازم باشد، از نام تابع برای فراخوانی آن استفاده نمود.

ساختار کلی تعریف تابع جدید به شکل زیر است:

```
function نام تابع ( پارامترهای ورودی)
{
    بدنه تابع
}
```

**نکته:** اگر تابع پارامتر ورودی نداشته باشد، باید پرانتزهای جلوی نام آن را خالی بگذارید. برنامه زیر یک تابع ساده با نام hello می‌سازد که عبارت hello, welcome! را در خروجی چاپ می‌کند.

```
function hello()  
{  
    echo ("hello , welcome!");  
}
```

سعی کنید در انتخاب نام تابع دقت کنید، و نام آن را از روی عملکردش انتخاب نمایید. پس از نوشتن تابع، در برنامه و هر جایی که نیاز به تابع داشته باشید کافی است نام تابع را بنویسید. به این عمل فراخوانی تابع گفته می‌شود.

```
<?php  
function hello()  
{  
    echo ("hello , welcome!");  
}  
  
hello();  
?>
```

خروجی:

hello, welcome!

هر تابع می‌تواند یک یا چند پارامتر ورودی داشته باشد. فرض کنید قصد داریم برنامه‌ای طراحی کنیم که در صفحه مرورگر کاربر، با استفاده از یک تابع پیغام خوشامدگویی به همراه نام او را نشان دهد. در این صورت نام کاربر به عنوان پارامتر ورودی در اختیار تابع قرار می‌گیرد.

**مثال:**

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://  
www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">  
<html><head>
```

```
<meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>
</head><body>
<?php
function hello($name)
{
echo ("hello , welcome! " . $name);
}
$myname="mina";
hello($myname);
?>
</body>
</html>
```

hello , welcome! mina

خروجی

در برنامه فوق، پارامتر نام به عنوان یک مقدار ثابت در نظر گرفته شده و از قبل مقداردهی شده است اما شما می‌توانید هنگام باز شدن صفحه مرورگر، با استفاده از زبان‌هایی مانند جاوااسکریپت از کاربر بخواهید اسم خود را وارد کند.

ممکن است توابع طوری پیاده‌سازی شوند که مقدار مشخصی را محاسبه کنند و در برنامه از این مقدار به عنوان پارامتر خروجی تابع در برنامه استفاده شود. ساختار کلی پیاده‌سازی چنین توابعی به صورت زیر است :

```
function نام تابع ( پارامترهای ورودی)
{
بدنه تابع
return "متغیری که باید برگشت داده شود." ;
}
```

### مثال :

می‌خواهیم تابعی را پیاده‌سازی کنیم که دو عدد را به عنوان پارامتر ورودی دریافت کرده، سپس حاصل ضرب آنها را محاسبه و به عنوان پارامتر خروجی برمی‌گرداند.

```
<?php
```

```
function calc($n,$m)
{
    $p = $n*$m;
    return $p;
}
$n1=10;
$n2=43;
$n3 = calc($n1,$n2);
echo $n3;
?>
```

۴۳.

خروجی :

در کد فوق دو متغیر با نام‌های  $n1$  و  $n2$  تعریف شده است که به ترتیب با اعداد ۱۰ و ۴۳ مقداردهی شده‌اند، سپس تابع  $calc()$  با دو پارامتر ورودی (۴۳ و ۱۰) فراخوانی می‌شود. درون این تابع، حاصل ضرب پارامترهای ورودی محاسبه شده، توسط دستور  $return$  برگردانده می‌شود.

درون برنامه اصلی، مقدار برگشت داده شده توسط دستور  $return$  در متغیر  $$n3$  قرار داده می‌شود و در سطر آخر برنامه توسط دستور  $echo$  مقدار این متغیر در خروجی چاپ می‌شود.

**تمرین:** تابعی برای محاسبه مجذور یک عدد دلخواه ایجاد کنید.

**نکته:** تابع را می‌توان در هر جایی از برنامه  $php$  تعریف کرد ولی بهتر است کلیه توابع را در ابتدای برنامه تعریف کنیم و سپس در صورت لزوم در ادامه برنامه فراخوانی نماییم.

**نکته:** در هنگام تعریف توابع جدید باید به این نکته توجه داشت که متغیر در زبان  $PHP$  محدوده کاملاً محلی دارند و فقط درون تابعی که تعریف شده‌اند قابل دسترسی هستند. به همین دلیل اغلب برنامه‌نویسان متغیرهای مهم را که ممکن است در جاهای مختلف برنامه مورد استفاده قرار گیرند، درون اسکریپت اصلی می‌نویسند.

روش دیگری به منظور تعریف حوزه سراسری برای متغیرها، استفاده از آرایه از پیش تعریف شده [ ] \$GLOBALS است.

استفاده از این آرایه به این صورت است که نام متغیر مورد نظر را به عنوان اندیس آرایه درج می‌کنید.

به این ترتیب با استفاده از آرایه مخصوص [ ] \$GLOBALS می‌توان برنامه فوق را به شکل زیر بازنویسی کرد :

\$PI=3.14;

\$r=5;

\$circum ference=\$ GLOBALS ["PI"]\*۲\*\$ GLOBALS ["r"];

تابع `strlen` طول رشته‌ای را که به عنوان پارامتر ورودی به آن داده می‌شود برمی‌گرداند.  
تابع `strtoupper` رشته‌ای را که به عنوان پارامتر ورودی می‌گیرد به حروف الفبای بزرگ انگلیسی تبدیل می‌کند.

تابع `strtolower` تمام حروف پارامترهای ورودی را به حروف کوچک الفبای انگلیسی تبدیل می‌کند.

تابع `date()` یک پارامتر را به منظور قالب درخواستی برای برگرداندن تاریخ به عنوان ورودی دریافت می‌کند، سپس یک رشته را به عنوان خروجی برمی‌گرداند.

تابع `time()` در PHP تعداد ثانیه‌های گذشته از اولین روز سال ۱۹۷۰ میلادی تا لحظه جاری را نشان می‌دهد.

تابع `abs()` به منظور محاسبه قدر مطلق یک پارامتر به کار می‌رود.

تابع `floor()` عددی را که به عنوان پارامتر ورودی دریافت می‌کند به عدد صحیح کوچک‌تر تبدیل می‌نماید و مقدار اعشار آن را از بین می‌برد.

تابع `fmod()` دو پارامتر را دریافت و باقیمانده حاصل تقسیم پارامتر اول بر پارامتر دوم را برمی‌گرداند.

تابع `max()` دو یا چند پارامتر را به عنوان ورودی دریافت می‌کند، سپس عدد بزرگ‌تر را به عنوان خروجی برمی‌گرداند.

تابع `min()` دو یا چند پارامتر را به عنوان ورودی دریافت می‌کند، سپس عدد کوچک‌تر را به عنوان خروجی برمی‌گرداند.

از تابع `rand()` برای تولید اعداد صحیح تصادفی استفاده می‌شود.

تابع `round()` اعداد اعشاری را به نزدیک‌ترین اعداد صحیح تبدیل می‌کند.

برنامه‌نویس می‌تواند تابع جدیدی را برای عملیات موردنظر خود ایجاد نماید. برای فراخوانی از نام تابع استفاده می‌شود.

اگر در تابع محاسبه خاصی انجام شود مقدار خروجی این محاسبه توسط دستور `return` به برنامه اصلی بازگردانده می‌شود.

- ۱- برنامه‌ای بنویسید که نام و نام خانوادگی شما را به عنوان پارامتر ورودی دریافت نموده، سپس تعداد حروف آن را چاپ کند.
  - ۲- یک تابع برای محاسبه مساحت مستطیلی به طول ۵ و عرض ۴ پیاده‌سازی کنید.
  - ۳- برنامه‌ای بنویسید که در آن بدون استفاده از تابع `date()` تاریخ جاری را به صورت سال، ماه و روز نمایش دهد.
-