

توابع

فصل چهارم

هدف‌های رفتاری:

پس از آموزش این فصل هنرجو می‌تواند:

- ۱- انواع توابع رشته‌ای و کاربرد هر یک را بیان کند.
- ۲- اصول کار با توابع تاریخ و زمان را تعریف کند.
- ۳- از توابع مهم ریاضی در برنامه PHP استفاده کند.
- ۴- با توجه به نیاز برنامه، توابع جدیدی را تعریف کند.

مقدمه

در PHP حدود ۷۰۰ تابع به صورت درون ساخته وجود دارد که برنامه نویسان به راحتی می‌توانند در برنامه‌های خود از آن‌ها استفاده کنند، برخی از این توابع مربوط به متغیرهای رشته‌ای هستند و به منظور کار با رشته‌ها طراحی شده‌اند، در ادامه با مهمترین توابع کار با رشته‌ها و چند نمونه تابع پرکاربرد دیگر آشنا خواهید شد.

۴-۱ توابع رشته‌ای

۴-۱-۱ عملگر الحاق یا چسباندن (Concatenation)

همان‌طور که در درس‌های قبل اشاره شد، عملگر (.) برای چسباندن رشته‌ها به هم‌دیگر به کار می‌رود.

مثال زیر دو رشته \$str1 و \$str2 را بوسیله این عملگر به هم‌دیگر چسبانده، در متغیر رشته‌ای \$str3 ذخیره، و سپس در خروجی چاپ می‌کند.



```
<?php
$str1="Hello Word!";
$str2="What a nice day!";
$str3=$str1." ".$str2;
echo $str3;
?>
```

خروجی:

Hello Word! What a nice day!

در کد فوق از دو عملگر الحاق استفاده شده است، به دلیل اینکه هدف این بوده که دو رشته \$str1 و \$str2 در خروجی با فاصله چاپ شوند و رشته " " بین این دو رشته قرار داده شده است.

۴-۱-۲ تابع strlen

این تابع طول رشته‌ای را که به عنوان پارامتر ورودی به آن داده می‌شود برمی‌گرداند، در نتیجه مقدار برگشت داده شده آن یک عدد صحیح است. 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);
?>
```



شکل ۴-۱ نتیجه اجرای تابع strlen

۴-۱-۳ تابع strtoupper ()

این تابع رشته‌های 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/>";
?>
```

۴-۱-۴ تابع strtolower ()

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



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

۲-۴ توابع تاریخ و زمان

۱-۲-۴ تابع date()

یک پارامتر را به عنوان ورودی دریافت می‌کند، 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

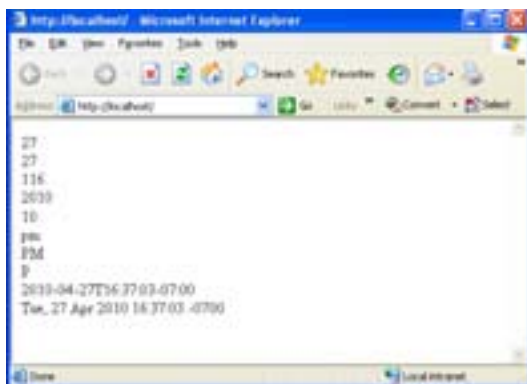
مثال(نمونه رشته خروجی)	عملکرد	شناسه
01 تا 31	شماره روز و ماه به صورت عدد(در صورتیکه شماره تک رقمی باشد به ابتدای آن صفر را اضافه می‌کند)	d
1 تا 31	شماره روز و ماه به صورت عدد(بدون درج صفر)	j
0 تا 365	تعداد روزهای گذشته از سال	z
1999 تا 2003	شماره سال را به صورت یک عدد ۴ رقمی بر می‌گرداند.	Y
99 تا 03	شماره سال را به صورت یک عدد ۲ رقمی بر می‌گرداند.	y
am or pm	فقط تعیین می‌کند که قبل از ظهر است یا بعد از ظهر (با حروف الفبای کوچک انگلیسی)	a
AM or PM	فقط تعیین می‌کند که قبل از ظهر است یا بعد از ظهر (با حروف الفبای بزرگ انگلیسی)	A
+02:00	ساعت و دقیقه را به وقت گرینویچ تعیین می‌کند.	P
2010-02-12T15:19:21+00:00	سال-روز-ماه-ساعت-دقیقه-ثانیه-صدم ثانیه	c
Thu, 21 Dec 2000 16:01:07 +0200	سال-روز-ماه-ساعت-دقیقه-ثانیه	r

```

<?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/>";
?>

```

مثال:



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()` به تنهایی کمتر به کار می‌رود اما می‌تواند در توابع دیگری به منظور تبدیل ساعت نواحی مختلف جغرافیایی و بر اساس اختلاف زمانی که با گرینویچ دارند، مورد استفاده قرار گیرد. و کاربرد مهم آن در محاسبه زمان و ورودی تابع `date()` می‌باشد

۳-۴ توابع ریاضی

۱-۳-۴ 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 می‌کند به عدد صحیح کوچک‌تر تبدیل می‌نماید و مقدار اعشار آن را از بین می‌برد. مثال زیر با استفاده از پارامترهای مختلف، رفتار این تابع را به خوبی نشان می‌دهد.

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 بزرگتر را به عنوان خروجی برمی گرداند.

```
<?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 می‌کند، سپس عدد کوچک‌تر را به عنوان خروجی برمی‌گرداند.



```
<?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:

```
5
-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(۱۰,۱۰۰) عددی تصادفی در محدوده بین ۱۰ و ۱۰۰ تولید خواهد کرد.



```
<?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 نزدیکترین اعداد صحیح تبدیل می‌کند.



```
<?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

همان‌طور که گفته شد در زبان PHP توابع متعددی در نظر گرفته شده‌اند که در روند برنامه نویسی بسیاری از مسائل را تسهیل می‌کنند، 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 می‌توان 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
}
```

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 آنرا 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();
?>
```

هر تابع می‌تواند یک یا چند پارامتر ورودی داشته باشد.

فرض کنید قصد داریم برنامه‌ای طراحی کنیم که در صفحه مرورگر 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 کاربر 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>
```

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:

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

مثال:



می‌خواهیم تابعی را پیاده‌سازی کنیم که دو عدد را به عنوان پارامتر ورودی دریافت می‌کند، 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;
?>
```

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.



چکیده فصل

عملگر (.) برای چسباندن(الحاق) رشته‌ها به همدیگر به کار می‌رود.
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 لحظه جاری را به صورت یک عدد صحیح شامل ثانیه، دقیقه، ساعت، روز، ماه و سال به وقت گرینویچ 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 پارامتر را به عنوان ورودی دریافت می‌کند، سپس عدد بزرگ‌تر را به عنوان خروجی برمی‌گرداند.

تابع min() دو 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.

خودآزمایی

۱. برنامه ای 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 پیاده سازی کنید.
۳. برنامه ای بنویسید که در آن بدون استفاده از تابع `date()` تاریخ جاری را به صورت سال ، ماه و روز نمایش دهد.

