

کار با توابع

واحد کار نهم

پس از مطالعه این واحد کار از فراگیر انتظار می‌رود که:

۱ - اصول کار با توابع رشته‌ای زیر را بداند:

• strlen

• strtolower

• strtoupper

• ucfirst

• ucwords

• addslashes

۲ - اصول کار با توابع تاریخ و زمان را بشناسد.

۳ - بتواند تابع جدیدی ایجاد کند.

۴ - بتواند اعداد و متغیرها را در توابع به‌کار ببرد.

زمان (ساعت)	
عملی	نظری
۱۰	۵

مقدمه

در طراحی یک سایت پویا اطلاعات ورودی کاربران نقش اصلی روی تصمیمات سرویس دهنده دارند، لذا زبان‌هایی که به منظور طراحی سایت‌های پویا به کار برده می‌شوند باید بتوانند پردازش‌های لازم را روی اطلاعات ورودی انجام دهند، اغلب پردازش‌ها توسط توابع از پیش تعریف شده‌ای قابل انجام هستند. زبان PHP نیز دستورات متعددی برای کار با رشته‌ها و دیگر متغیرها دارد که در ادامه این مبحث با مهم‌ترین آن‌ها آشنا خواهیم شد.

۹-۱ توابع رشته‌ای

۹-۱-۱ 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»)

```
<?php
```

```
$str1 = 'abcdef';
```

```
echo "str1."---length is : ".strlen($str1)."<br>";
```

```
$str2 = ' ab cd ';
```

```
echo "str2."---length is : ".strlen($str2)."<br>";
```

```
?>
```



شکل ۹-۱

همان‌طور که در نتیجه اجرای کد قبل ملاحظه می‌کنید تابع `strlen` توانایی شمارش کاراکتر Space را نیز دارد، به همین دلیل رشته 'ab cd' که شامل ۴ حرف و ۳ فضای خالی است به عنوان پارامتر ورودی تابع، نتیجه ۷ را نشان می‌دهد.

۹-۱-۲ تابع `strtoupper()`

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

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



شکل ۹-۲

در کد فوق توسط این تابع تمام حروف مربوط به پارامتر ورودی "aBc def" به حروف بزرگ تبدیل شده‌اند. فاصله‌ها (Space) در خروجی این تابع تأثیر ندارند و باقی می‌مانند

۹-۱-۳ تابع `strtolower()`

این تابع عکس تابع `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>";  
?>
```



شکل ۳-۹

۴-۱-۹ تابع `ucfirst()`

این تابع حرف اول پارامتر ورودی را به حرف الفبایی بزرگ انگلیسی تبدیل می‌کند. این تابع جز اولین حرف از رشته، هیچ‌کدام از حروف دیگر را تحت تأثیر قرار نمی‌دهد و بدون تغییر برمی‌گرداند.

مثال:

```
<?php  
$foo = 'hello world!';  
echo "$foo"."---ucfirst is : ".ucfirst($foo)."<br>";  
$bar = 'HELLO WORLD!';  
echo "$bar"."---ucfirst is : ".ucfirst($bar)."<br>";  
  
echo "$bar"."---ucfirst(strtolower) is : ".ucfirst(strtolower($bar))."<br>";  
?>
```



شکل ۴-۹

۵-۱-۹ تابع ()ucword

این تابع حرف اول تمام کلمات موجود در پارامتر ورودی را به حرف الفبای بزرگ انگلیسی تبدیل می‌کند. این تابع از کاراکتر Space برای تشخیص کلمه بعدی استفاده می‌کند، یعنی هر جا به این کاراکتر برسد، کلمه قبلی را خاتمه یافته فرض می‌کند و حرف بعد از کاراکتر Space را اولین حرف از کلمه بعدی می‌داند.

```
<?php
$foo = 'hello world!';
echo "$foo". "---ucword is : ".ucwords($foo). "<br>";

$bar = 'HELLO WORLD!';
echo "$bar". "---ucword is : ".ucwords($bar). "<br>";
echo "$bar". "---ucword is : ".ucwords(strtolower($bar)). "<br>";
?>
```



شکل ۵-۹

تمرین:

برنامه‌ای بنویسید که در آن یک متغیر با نام str1 تعریف شده است که با مقادیر نام، نام خانوادگی و نام مدرسه شما به دنبال یکدیگر (با رعایت فاصله) مقداردهی شده است، حال متغیر str1 را طوری چاپ کنید که حرف اول، نام، نام خانوادگی و نام مدرسه به صورت حروف بزرگ الفبای انگلیسی و سایر حروف این کلمات با حروف کوچک الفبا، چاپ شود.

۹-۱-۶ تابع addslashes ()

همان طور که می دانید کاراکتر (') برای متغیرهای رشته‌ای استفاده می شود و وارد کردن این کاراکتر در پایگاه داده‌ها مشکل آفرین می شود. اما اگر در بعضی مواقع این کاراکتر جزئی از رشته باشد باید به نوعی بتوان آن را در پایگاه داده درج کرد. به عنوان مثال اسم (o'reilly) را در نظر بگیرید، کاراکتر (') جزئی از اسم است و تنها راه چاره برای درج آن در پایگاه داده استفاده از کاراکتر (\) است. تابع addslashes () این کار را به طور خودکار انجام می دهد.

```
<?php
```

```
$str = "Is your name O'reilly?";
```

```
echo addslashes($str);
```

```
?>
```



شکل ۹-۶

۹-۲ توابع تاریخ و زمان

۹-۲-۱ تابع date ()

یک پارامتر را به عنوان ورودی دریافت می کند و سپس یک رشته را به عنوان خروجی برمی گرداند. پارامتر ورودی یک قالب درخواستی برای برگرداندن تاریخ است. هر کدام از قالب‌ها یک شناسه استاندارد دارند که در جدول ۹-۱ چند نمونه از این شناسه‌ها بیان شده است.

جدول ۹-۱

شناسه	توضیحات	مثال (نمونه رشته خروجی)
d	شماره روز و ماه به صورت عدد (در صورتی که شماره تک رقمی باشد به ابتدای آن صفر را اضافه می‌کند).	01 to 31
j	شماره روز و ماه به صورت عدد (بدون درج صفر)	1 to 31
z	تعداد روزهای گذشته از سال	0 through 365
Y	شماره سال را به صورت یک عدد ۴ رقمی برمی‌گرداند.	1999 or 2003
y	شماره سال را به صورت یک عدد ۲ رقمی برمی‌گرداند.	99 or 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	ساعت و دقیقه را به وقت گرینویچ تعیین می‌کند.	+02:00
c	سال- روز - ماه - ساعت - دقیقه - ثانیه - صدم ثانیه	2010-02-12T15:19:21+00:00
r	سال- روز - ماه - ساعت - دقیقه - ثانیه	Thu, 21 Dec 2000 16:01:07 +0200

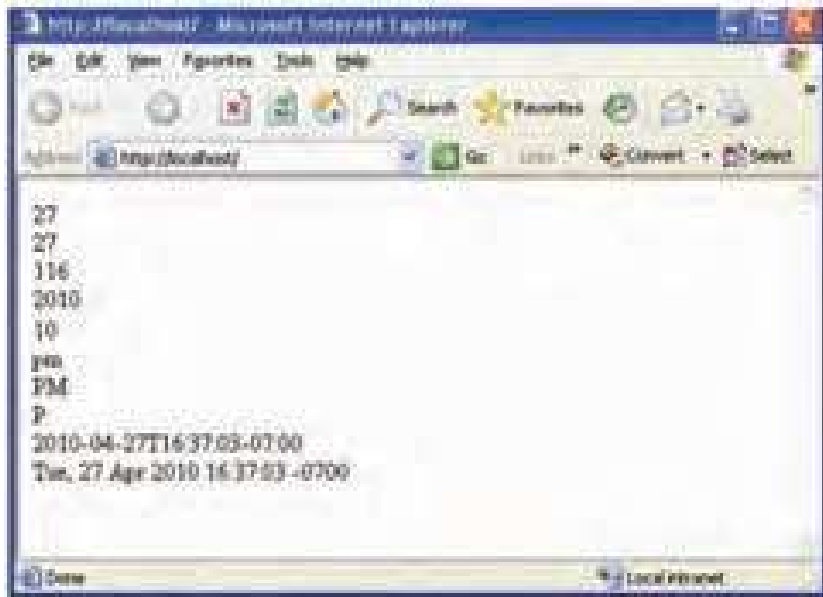
<?php

```

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 ۷-۹

تمرین:

برنامه‌ای بنویسید که زمان جاری سیستم را بر حسب ساعت و دقیقه به وقت گرینویچ، چاپ کند.

۹-۲-۲ تابع time

این تابع در PHP لحظه جاری را به صورت یک عدد صحیح شامل ثانیه، دقیقه، ساعت، روز، ماه و سال به وقت گرینویچ نشان می‌دهد.
کد زیر را اجرا کنید:

```
<?php
```

```
$Now = time();
```

```
echo $Now;
```

```
?>
```



شکل ۸-۹

تابع `time()` به تنهایی کاربرد کمی دارد اما می‌تواند با توابع دیگر به منظور تبدیل ساعت نواحی مختلف براساس اختلاف زمانی که با گرینویچ دارند، مورد استفاده قرار بگیرد.

۳-۹ توابع عددی

۱-۳-۹ تابع `abs()`

این تابع قدر مطلق پارامتر ورودی خود را محاسبه می‌کند.

```
<?php  
$num1 = -243;  
echo ("$num1."<br>");  
$num2 = abs($num1);  
echo "$num2."<br>";  
?>
```



شکل ۹-۹

۲-۳-۹ تابع `floor()`

این تابع عددی را که به عنوان پارامتر ورودی دریافت می‌کند به عدد صحیح کوچک‌تر تبدیل کرده 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
```

۲-۳-۹ تابع fmod()

این تابع دو پارامتر را دریافت می‌کند و باقیمانده حاصل تقسیم پارامتر اول بر پارامتر دوم را بر می‌گرداند.

مثال:

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

خروجی:

```
1
```

۴-۳-۹ تابع max()

این تابع دو پارامتر را به عنوان ورودی دریافت می‌کند، سپس عدد بزرگ‌تر را به عنوان خروجی بر می‌گرداند.

مثال:

```
<?php
echo(max(5,7) . "<br />");
echo(max(-3,5) . "<br />");
echo(max(-3,-5) . "<br />");
```

```
echo(max(7.25,7.30))
```

```
?>
```

خروجی:

7

5

-3

7.3

۵-۳-۹ تابع min()

این تابع دو پارامتر را به عنوان ورودی دریافت می‌کند، سپس عدد کوچک‌تر را به عنوان خروجی برمی‌گرداند.

۶-۳-۹ 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 را 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(10,100) عددی تصادفی در محدوده بین ۱۰ و ۱۰۰ 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

۷-۳-۹ تابع round()

این تابع اعداد اعشاری را به نزدیک‌ترین اعداد صحیح تبدیل می‌کند.
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

۴-۹ تعریف توابع جدید

همان‌طور که ملاحظه کردید در زبان PHP توابعی وجود دارند که در روند برنامه‌نویسی به ما کمک می‌کنند، بسیاری از این توابع جزء ساختار اولیه زبان PHP نیستند و به دلیل متن باز بودن به ساختار آن اضافه شده‌اند.

فرض کنید می‌خواهید یک سری عملیات مشخص را به دفعات در برنامه خود استفاده کنید، می‌توانید بدون نیاز به درج کدهای تکراری برای خود یک تابع بسازید که عملیات کدنویسی را برای شما راحت‌تر کند.
ساختار کلی تعریف تابع جدید به شکل زیر است:

```
function نام تابع (پارامترهای ورودی)
```

```
}
```

```
بدنه تابع
```

```
{
```

اگر تابع پارامتر ورودی نداشته باشد می‌توانید پرانتزهای جلوی نام تابع 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!");
```

```
}
```

پس از نوشتن تابع، در برنامه هر جایی که نیاز به تابع داشتید کافی است نام تابع را بنویسید، به این عمل فراخوانی تابع گفته می‌شود.

```
<?php
```

```
function hello()
```

```
{
```

```
echo ("hello , welcome!");
```

```
}
```

```
hello();
```

```
?>
```

همان‌طور که اشاره شد تابع می‌تواند پارامتری یا پارامترهایی را به عنوان ورودی داشته باشد.
فرض کنید می‌خواهیم در صفحه مرورگر با استفاده از یک تابع پیغام خوشامدگویی

به کاربر به همراه نام او ظاهر شود، در این صورت نام کاربر به عنوان پارامتر ورودی در اختیار تابع قرار می‌گیرد:

```
<?php
function hello($name)
{
echo ("hello , welcome!". "$name");
}
$myname="mina";
hello($myname);
?>
```

برخی از توابع هستند که عبارت خاصی را چاپ نمی‌کنند و فقط یک مقدار را به عنوان خروجی برمی‌گردانند. برای پیاده‌سازی این‌گونه توابع از دستور return استفاده می‌شود. ساختار کلی این‌گونه توابع به صورت زیر است:

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

می‌خواهیم یک تابع پیاده‌سازی کنیم به طوری که دو عدد را دریافت کند و حاصل ضرب آن‌ها را برگرداند، سپس حاصل را در خارج از تابع چاپ کند:

```
<?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 تابع $calc()$ حاصلضرب این دو پارامتر ورودی محاسبه شده، توسط دستور $return$ برگردانده می‌شود.

درون برنامه اصلی مقدار برگردانده 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.

واژه‌نامه

Absolute	قدر مطلق
Length	طول
Lowercase	حروف کوچک
Uppercase	حروف بزرگ

خلاصه مطالب

- به دلیل این که اغلب اطلاعات ورودی کاربران در سایت‌های پویا به صورت متنی وارد می‌شود، PHP باید بتواند پردازش‌های لازم را روی آن‌ها انجام دهد، اغلب این پردازش‌ها توسط توابع از پیش تعریف شده انجام می‌شوند.
- تابع رشته‌ای `strlen` به منظور محاسبه طول رشته‌ای به کار می‌رود که به عنوان پارامتر ورودی به تابع داده می‌شود.
- تابع `strtoupper` رشته‌ای را که به عنوان پارامتر ورودی می‌گیرد، به حروف بزرگ تبدیل می‌کند.
- تابع `strtolower` رشته‌ای را که به عنوان پارامتر ورودی می‌گیرد، به حروف کوچک تبدیل می‌کند.
- تابع `ucfirst()` حرف اول عبارتی را که به عنوان پارامتر ورودی می‌گیرد به حرف بزرگ تبدیل می‌کند.
- تابع `ucword()` حرف اول تمام کلمات عبارت ورودی را به حرف بزرگ تبدیل می‌کند.
- تابع `date()` یکی از توابع پرکاربرد در PHP است، این تابع با استفاده از پارامترهای دریافتی نحوه نمایش تاریخ و زمان را تعیین می‌کند.