

Web Application Development (WAD)

Vth Sem – BBAITM (Unit – 3)

By: Binit Patel

Number Functions:

1) abs()

It is the most basic function and returns the absolute value of the parameter passed to it. i.e. the function leaves positive values untouched, while converts negative values into positive values

Syntax: abs(number value)

e.g.:

```
<?php
```

```
echo abs(-3.1);
```

```
echo "<br>";
```

```
echo abs(47.1);
```

```
?>
```

Output:

3.1

47.1

2) ceil()

It accepts one parameter, a number and returns the round to the nearest integer greater than the number.

Syntax: ceil(number value)

e.g.:

```
<?php
```

```
echo ceil(15.85);
```

```
echo "<br>";
```

```
echo ceil(47.1);
```

```
?>
```

Output:

16

48

3) floor()

It accepts one parameter, a number and returns the round to the nearest integer smaller than the number.

Syntax: floor(number value)

e.g.:

```
<?php
```

```
echo floor(15.85);
```

```
echo "<br>";
```

```
echo floor(47.1);
```

```
?>
```

Output:

15

47

4) round()

It accepts two parameters, a number and returns the round to the nearest integer smaller than the number.

Syntax: round(number value [,int precision])

e.g.:

```
<?php
```

```
echo round(15.85);
```

```
echo "<br>";
```

```
echo round(15.85,2);
```

```
?>
```

Output:

16

15.90

5) sqrt()

It is short name of square root and takes just one parameter i.e. the value whose square root needs to be calculated.

Syntax: sqrt(number value)

e.g.:

```
<?php
```

```
echo sqrt(16);
```

```
echo "<br>";
```

```
echo sqrt(9);
```

```
?>
```

Output:

4

3

6) pow()

It takes two parameters i.e. a base and a power to raise it by. The value in the first parameter is multiplied with itself, one time less than the value entered as the second parameter. The pow() function is capable of handling negative powers for the second parameter.

Syntax: pow(number base_value, number exponent)

e.g.:

```
<?php
```

```
echo pow(3,2);
```

```
echo "<br>";
```

```
echo pow(4,3);
```

```
?>
```

Output:

9

64

String Functions:

1) chr()

The chr() function is used to convert ASCII to textual characters. This function accepts an ASCII value as its single parameter and returns the text equivalent.

Syntax: chr(ASCII value)

e.g.:

```
<?php  
echo chr(105);  
echo "<br>";  
echo chr(74);  
?>
```

Output:

```
i  
J
```

2) ord()

The ord() function is reverse of chr() function which converts textual characters to ASCII. This function accepts a string and returns the equivalent ASCII value.

Syntax: ord(string value)

e.g.:

```
<?php  
echo ord(i);  
echo "<br>";  
echo ord(J);  
?>
```

Output:

```
105  
74
```

3) strlen()

Strlen() is the easiest way to measure a string. strlen() counts the number of characters in the string. it takes just one parameter and returns the number of characters in it.

Syntax: strlen(string)

e.g.:

```
<?php
echo strlen("SEMCOM");
echo "<br>";
echo strlen("WAD");
?>
```

Output:

```
6
3
```

4) trim()

It removes whitespaces from either sides of the string passed as the first parameter.

Syntax: trim(string)

e.g.:

```
<?php
echo trim(" SEMCOM ");
echo "<br>";
echo trim(" First Name");
?>
```

Output:

SEMCOM

FIRST Name

5) strtolower()

It takes a string parameter as an argument and returns that string entirely in lower case.

Syntax: strtolower(string)

e.g.:

```
<?php
```

```
echo strtolower("SEMCOM");
```

```
echo "<br>";
```

```
echo strtolower("Wad-TYitm");
```

```
?>
```

Output:

semcom

wad-tyitm

6) strtoupper()

It takes a string parameter as an argument and returns that string entirely in upper case.

Syntax: strtoupper(string)

e.g.:

```
<?php
```

```
echo strtoupper("semcom");
```

```
echo "<br>";
```

```
echo strtoupper("wad-TYitm");
```

```
?>
```

Output:

```
SEMCOM
```

```
WAD-TYITM
```

7) substr()

The substr() function allows reading just part of a string and takes a minimum of two parameters that is the string to work with and where to start reading from. There is an optional third parameter to allow specifying how many characters are to be read.

Syntax: substr(string, int start position, int length)

e.g.:

```
<?php
```

```
echo substr("semcom",1,3);
```

```
echo "<br>";
```

```
echo substr("wad-TYitm",3,4);
```

```
?>
```

Output:

```
EMC
```

```
-TYi
```

Date & Time Functions:

1) date()

The function date() is used to format either time or a date. This function can take the current date and extract specific information from it.

To do so, call `date()` and provide a string as the first parameter for format.

Syntax: `string(format, int timestamp)`

e.g.:

```
<?php
```

```
echo date("d-m-y");
```

```
echo "<br>";
```

```
echo date("d:M:Y");
```

```
?>
```

Output:

01-09-14

01:Sep:2014

2) `time()`

PHP has a basic function `time()` to get the current time in epoch format. `Time()` takes no parameter and returns the current timestamp representing the current time.

`Time()` value can be either directly printed or it can be stored in a variable and then the contents of the variable printed. But in either case the result is identical.

Syntax: `time()`

e.g.:

```
<?php
```

```
echo time();
```

```
?>
```

Output:

1254681000

3) strtotime()

PHP has a built in function to help convert strings to a timestamp. The function strtotime() takes two parameters i.e. the string time to convert and a second optional parameter that can be relative timestamp.

The function returns a timestamp on success, FALSE otherwise.

Syntax: strtotime(string time, int now)

e.g.:

```
<?php
```

```
echo strtotime("28 October 2005");
```

```
echo "<br>";
```

```
echo strtotime("2005/10/28");
```

```
?>
```

Output:

```
1254681000
```

```
1254681000
```

DISCLAIMER :

This study material is prepared by **Mr. Binit Patel**. The objective of this material is to supplement teaching and discussion in the class room in the subject. Students are required to go for extra reading in the subject through library book.