

PHP Functions

Primary source: <https://www.w3schools.com/php/>

Internet Programming 2, Lesson 4

April 2024

PHP Functions

PHP Functions

The real power of PHP comes from its functions; it has more than 1000 built-in functions.

PHP User Defined Functions

Besides the built-in PHP functions, we can create our own functions.

A function is a block of statements that can be used repeatedly in a program.

A function will not execute immediately when a page loads.

A function will be executed by a call to the function.

Create a User Defined Function in PHP

A user defined function declaration starts with the word "function":

Syntax

```
function functionName() {  
    // code to be executed;  
}
```

Note: A function name can start with a letter or underscore (not a number).

Tip: Give the function a name that reflects what the function does!

Function names are NOT case-sensitive.

In the example below, we create a function named "writeMsg()". The opening curly brace ({) indicates the beginning of the function code and the closing curly brace (}) indicates the end of the function. The function outputs "Hello world!". To call the function, just write its name:

```
<!DOCTYPE html>
<html>
<body>

<?php
function writeMsg() {
    echo "Hello world!";
}

writeMsg();
?>

</body>
</html>
```

Output

Hello world!

PHP Function Arguments

Information can be passed to functions through arguments. An argument is just like a variable. Arguments are specified after the function name, inside the parentheses. You can add as many arguments as you want, just separate them with a comma.

The following example has a function with one argument (\$fname). When the familyName() function is called, we also pass along a name (e.g. Jani), and the name is used inside the function, which outputs several different first names, but an equal last name:

```
<?php
function familyName($fname) {
    echo "$fname Ramdeyal.<br>";
}
```

```
familyName("Pradesh");
familyName("Kesarie");
familyName("Jushen");
familyName("Kimmeshen");
familyName("Nelly");
?>
```

Output

```
Pradesh Ramdeyal.
Kesarie Ramdeyal.
Jushen Ramdeyal.
Kimmeshen Ramdeyal.
Nelly Ramdeyal.
```

The following example has a function with two arguments (\$fname and \$year):

```
<?php
function familyName($fname, $year) {
    echo "$fname Ramdeyal. Born in $year <br>";
}

familyName("Pradesh", "1975");
familyName("Kesarie", "1978");
familyName("Nelly");
?>
```

Output

Pradesh Ramdeyal. Born in 1975
Kesarie Ramdeyal. Born in 1978
Nelly Ramdeyal. Born in 1983

PHP Default Argument Value

The following example shows how to use a default parameter. If we call the function `setHeight()` without arguments it takes the default value as argument:

```
<?php
function setHeight($minheight = 50) {
    echo "The height is : $minheight <br>";
}
```

```
setHeight(350);
setHeight();
setHeight(135);
setHeight(80);
?>
```

Output

```
The height is : 350
The height is : 50
The height is : 135
The height is : 80
```

PHP Functions - Returning values

To let a function return a value, use the return statement:

```
<?php
function sum($x, $y) {
    $z = $x + $y;
    return $z;
}

echo "5 + 10 = " . sum(5,10) . "<br>";
echo "7 + 13 = " . sum(7,13) . "<br>";
echo "2 + 4 = " . sum(2,4);
?
```

Output

5 + 10 = 15

7 + 13 = 20

2 + 4 = 6

Review the section on PHP Variables Scope, including static variables from the previous section .

Exercise

Create a new file and save it as **sequences.php**. Write the following functions:

1. `displaySquares(X)` – that will display the first X terms of the sequence: 1, 4, 9, 16, 25, ...
2. `displayPrimes(X)` – that will display the first X terms of the sequence 1, 2, 3, 5, 7, 11, 13, ...
3. `displayFibo(X)` – that will display the first X terms of the sequence: 1,1,2,3,5,8,13,21,...

Test the above functions by calling them with different test values.