



F. Y. BCA

Web Development Using PHP

Chapter 3 – Functions and Strings

Dr. Savita Mohurle

INTRODUCTION TO FUNCTIONS

- A function is a block of code written in a program to perform some specific task.
- A function is a block of statements that can be used repeatedly in a program.
- A function will not execute automatically when a page loads.
- A function will be executed by a call to the function.
- They take information as parameter, executes a block of statements or perform operations on this parameters and returns the result.




WHY SHOULD WE USE FUNCTIONS?

- **Reusability:**

- A common code can be use at various parts of a program.
- Reduces the time and effort of repetition of a single code.

- **Easier error detection:** Since, our code is divided into functions, we can easily detect in which function, the error could lie and fix them fast and easily.

- **Easily maintained:**

- If anything or any line of code needs to be changed, can be easily changed it inside the function and the change will be reflected everywhere, where the function is called. Hence, easy to maintain.
- 

MAJOR TYPES OF FUNCTIONS

PHP provides us with two major types of functions:

- **Built-in functions :**

- PHP provides us with huge collection of built-in library functions.
- Functions are already coded and stored in form of functions.
- To use those we just need to call them as per our requirement
- Examples - `var_dump`, `fopen()`, `print_r()`, `gettype()` and so on.

- **User Defined Functions :**

- PHP allows us to create our own customised functions called the user-defined functions.

Note - Using this we can create our own packages of code and use it wherever necessary by simply calling it.



CREATING A FUNCTION – THINGS TO KEEP IN MIND

- Any name ending with an open and closed parenthesis is a function.
- A function name always begins with the keyword *function*.
- To call a function we just need to write its name followed by the parenthesis.
- A function name cannot start with a number. It can start with an alphabet or underscore.
- A function name is not case-sensitive.



FUNCTION DEFINITION

syntax-

```
function function_name()  
{  
    executable code;  
}
```

Example-

```
<?php  
    function Hello() //function definition  
    {  
        echo "Hello How are you?";  
    }  
    Hello(); // function call  
?>
```




EXAMPLE

```
<?php
$a=10;$b=20;
function add($x,$y) // function declaration
{
    $sum=$x+$y;
    echo "Sum is : ".$sum;
}
add($a,$b); //function call
?>
```

OUTPUT- Sum is : 30



FUNCTION WITH PARAMETERS OR ARGUMENTS

- The information or variable, within the function's parenthesis, are called parameters.
 - These are used to hold the values executable during runtime.
 - A user is free to take in as many parameters as he wants, separated with a comma(,) operator.
 - These parameters are used to accept inputs during runtime.
 - While passing the values like during a function call, they are called arguments.
 - An argument is a value passed to a function and a parameter is used to hold those arguments.
 - In common term, both parameter and argument mean the same.
 - We need to keep in mind that for every parameter, we need to pass its corresponding argument.
- 

FUNCTION WITH PARAMETERS OR ARGUMENTS

Syntax-

```
function function_name($first_parameter, $second_parameter)
{
    executable code;
}
```

Example-

<?php

// function along with three
parameters

```
function product($num1, $num2,
$num3)
{
    $p = $num1 * $num2 * $num3;
    echo "The product is $p";
}
```

// Calling the function
// Passing three arguments
product(2, 3, 5);

?>

Output- The product is 30



SETTING DEFAULT VALUES FOR FUNCTION PARAMETER

PHP allows us to set default argument values for function parameters. If we do not pass any argument for a parameter with default value then PHP will use the default set value for this parameter in the function call.



<?php

DEFAULT PARAMETER

```
// function with default parameter
function person($str, $num=10)
{
    echo "$str is $num years old \n";
}
```

```
// Calling the function
person("Rahul", 25);
```

```
// In this call, the default value 10 will be considered
person("Saksham");
```

?>

Note - The parameter \$num has a default value 12, if we do not pass any value for this parameter in a function call then this default value 12 will be considered. Also the parameter \$str has no default value, so it is compulsory.

PROGRAM FOR ADDITION OF TWO NUMBERS USING FUNCTION

```
<html>
```

```
<form action="numadd.php" method="GET"/>
```

numadd.html

Enter first number

```
<input type="text" name="t1"/><br><br>
```

Enter second number

```
<input type="text" name="t2"/><br>
```

```
<input type="submit" value="OK"/>
```

```
</html>
```

```
<?php
```

```
    $n1=$_GET["t1"];
```

```
    $n2=$_GET["t2"];
```

```
    function add($n1,$n2)
```

```
    {
```

```
        $c=$n1+$n2;
```

```
        echo "ADDITION OF TWO NUMBERS IS = $c";
```

```
    }
```

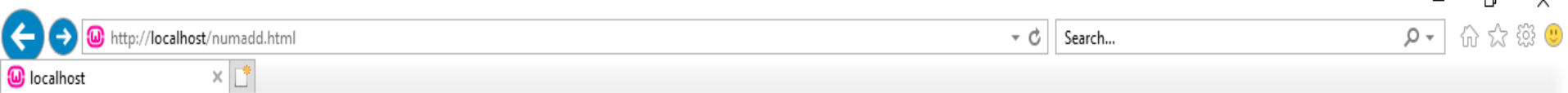
```
    add($n1,$n2);
```

```
?>
```

numadd.php



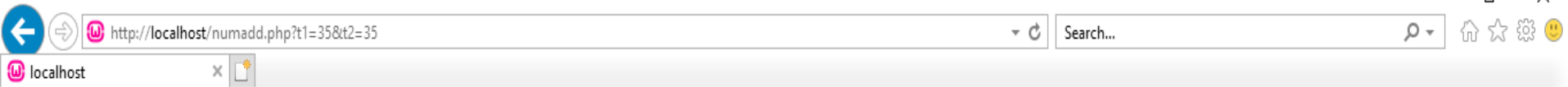
Output of HTML FILE numadd.html



Enter first number

Enter second number

Output of php FILE numadd.php



ADDITION OF TWO NUMBERS IS = 70

EXERCISE 1

```
<?php
msg("Hello"); // calling a function
function msg($a) // defining a function
{
    echo $a;
}
?>
```



EXERCISE 2

```
<?php
function display($greeting, $message="Good Day")
{
    echo $greeting;
    echo "<br>";
    echo $message;
}
display("Hello");
?>
```



EXERCISE 3

```
<?php
echo "Passing 3 arg. to xconcat <br>";
echo "Result is ....";
xconcat(" How "," are "," you?");
function xconcat()
{
    $ans = "";
    $arg = func_get_args();
    for ($i=0; $i<func_num_args(); $i++ )
    {
        $ans .= $arg[$i]."";
    }
    echo $ans;
}
?>
```



EXERCISE 4

```
<?php
function makecoffee ($type = "Nescafe")
{
return "Making a cup of $type<br>";
}
echo makecoffee ();
echo makecoffee ("espresso");
?>
```



EXERCISE 5

```
<?php
function make ($type = "acidophilus", $flavour)
{
return "Making a bowl of $type $flavour<br>";
}
echo make ("raspberry"); //
won't work
?>
```



EXERCISE 6

```
<?php
function make ($flavour, $type ="acidophilus")
{
return "Making a bowl of $type $flavour<br>";
}
echo make ("raspberry"); //it works
?>
```



ANONYMOUS FUNCTIONS

- The function that does not possess any name are called anonymous functions. Such functions are created using *`create_function()` built-in function.*
- *Anonymous* functions are also called as lambda functions.

Example :

```
<?php
```

```
$fname=create_function('$a,$b', '$c = $a + $b; return $c;');
```

```
echo $fname(10,20);
```

```
?>
```



Thank You

