



# **Ch3 - Lesson 5: Functions in PHP**

# PHP FUNCTIONS

## ■ Functions in PHP

- Must be defined before they called.
- Defined as:

```
function functionName ($arg1, arg2, ...$argn) {  
    // statements  
}
```

- Function names are not case sensitive unlike variables

# PHP FUNCTIONS

## ■ Functions in PHP

- Function statements do the actual work of the function and must be contained within the function braces.

```
{ ..... }
```

# PHP FUNCTIONS

## ■ Creating function

```
function addFunction ($num1, $num2) {  
    $sum = $num1 + $num2;  
    echo "Sum of the two numbers is : $sum";  
}  
addFunction (10, 20);
```

# PHP FUNCTIONS

## ■ Passing arguments by reference

```
function addFive (&$num) {  
    $num += 5; }  
function addSix (&$num) {  
    $num += 6;  
}  
$orignum = 10;  
addFive ($orignum );  
echo "Original Value is $orignum<br />";  
addSix ($orignum );  
echo "Original Value is $orignum<br />";
```

# PHP FUNCTIONS

- **Function's return value:**
  - A function can return a value using the **return** statement in conjunction with a **value** or **object**.
  - Return stops the execution of the function and sends the value back to the calling code.
  - It is possible to return more than one value from a function using **return array(1,2,3,4)**.

# PHP FUNCTIONS

## ■ Example:

```
function addFunction($num1, $num2) {  
    $sum = $num1 + $num2;  
    return $sum;}  
$return_value = addFunction(10, 20);  
echo "Returned value from the function : $return_value";
```

- We can return multiple values from a function, by placing them all in an array and returning the array

# PHP FUNCTIONS

```
function add_sub ($num1, $num2) {  
    $add=$num1+$num2;  
    $sub=$num1-$num2;  
    return array ($add, $sub) ;  
}  
$result_array=add_sub (15,20) ;  
echo "Add:". $result_array[0]. "<br>";  
echo "Sub:". $result_array[1]. "<br>";
```

# PHP FUNCTIONS

## Setting Default Values for Function Parameters

```
function studinfo ($department="CS", $year="3rd") {  
    return "You are $year year $department  
student<br>";  
}  
echo studinfo ();  
echo studinfo ("Software Engineering", "1st");  
echo studinfo ("Electrical", Null);  
echo studinfo ("2nd");
```

### Output

```
You are 3rd year CS student  
You are 1st year Software Engineering student  
You are year Electrical student  
You are 3rd year 2nd student
```

# PHP FUNCTIONS

## ■ Dynamic Function Calls:

- It is possible to assign function names as strings to variables and then treat these variables exactly as the function name itself.

```
function sayHello() {  
    echo "Hello<br />";  
}$function_holder = "sayHello";  
$function_holder();
```



**thankYou();**