

Lecture 07

PHP – Part 2

IT1100 Internet and Web technologies

Content

- Arrays
- Functions
- Superglobal variables and Form handling

Arrays

What is an Array?

- An array is a special variable, which can hold more than one value at a time.

There are two types of arrays in PHP

1. Numeric (indexed) Arrays
2. Associative Arrays

Create an Array in PHP

- In PHP, the array() function is used to create an array:

array();

Numeric (indexed) Arrays

- //Declare array

`$n=array();`

- //Assign values

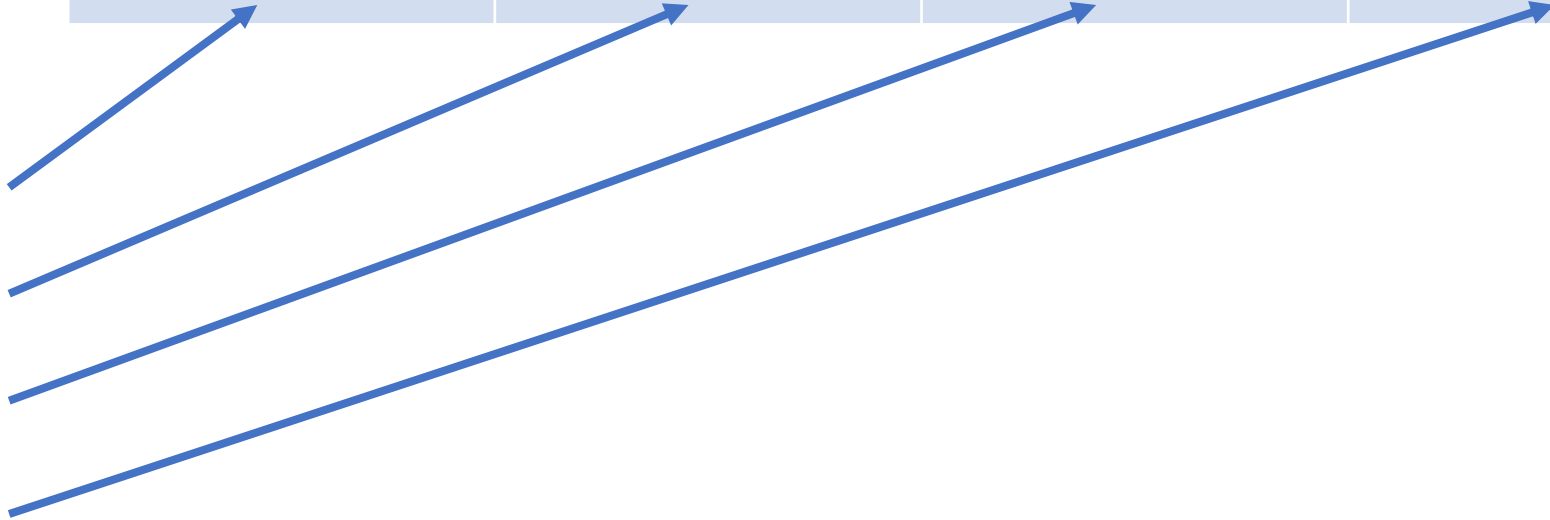
`$n[0] = 5;`

`$n[1] = 2;`

`$n[2] = 8;`

`$n[3] = 1;`

5	2	8	1
0	1	2	3



Numeric (indexed) Arrays

//Declare and initialize array in a single statement

```
$n=array(2,5,8,5);
```

Value	→	2	5	8	5
Index	→	0	1	2	3

- Can read the values of array using the index

```
echo $n[2] + $n[3] ;
```

2	5	8	5
0	1	2	3

Loop Through an Indexed Array

```
<!DOCTYPE html>
```

```
<html>
```

```
<body>
```

```
<?php
```

```
    $colors = array("red","green","blue","yellow");
```

```
    foreach ($colors as $value) {
```

```
        echo "$value <br>";
```

```
    }
```

```
?>
```

```
</body>
```

```
</html>
```

Declare the array

Loop
through the
Array

Output

red
green
blue
yellow

Loop Through an Indexed Array

Use a for loop to loop through and print all the values of an indexed array

```
<!DOCTYPE html>
<html>
<body>
<?php
    $subjects=array("ITA", "DBMS", "ST");
    $arrrlength=count($subjects);

    for($x=0;$x<$arrrlength;$x++) {
        echo $subjects [$x];
        echo "<br>";
    }

?>
</body>
</html>
```

Declare the array

Get the array length

Loop through the Array

Output

ITA
DBMS
ST

Associative Arrays

- For the index
 - Instead of numerical values
 - we can give string IDs for each and every item

```
$marks = array("Maths"=>32, "Web"=>30);  
echo $marks["Maths"];  
echo $marks["Web"];
```


Associative Arrays

```
$marks = array();
```

```
$marks["Maths"]=65;
```

```
$marks["Web"]=23;
```

```
$marks["JAVA"]=76;
```

65	23	76
Maths	Web	JAVA

← Value
← Key/ID

```
echo "Maths: " . $marks["Maths"]. "<br/>"
```

```
"Web: " . $marks["Web"]. "<br/>"
```

```
"JAVA: " . $marks["JAVA"]. "<br/>";
```

Loop Through an Associative Array

To loop through and print all the values of an associative array, you could use a foreach loop, like this:

```
<!DOCTYPE html>
<html>
<body>
<?php
    $age=array("Peter"=>"35", "Ben"=>"37", "Joe"=>"43");
    foreach($age as $x=>$x_value) {
        echo "Key=" . $x . ", Value=" . $x_value;
        echo "<br>";
    }
?>
</body>
</html>
```

Output ?

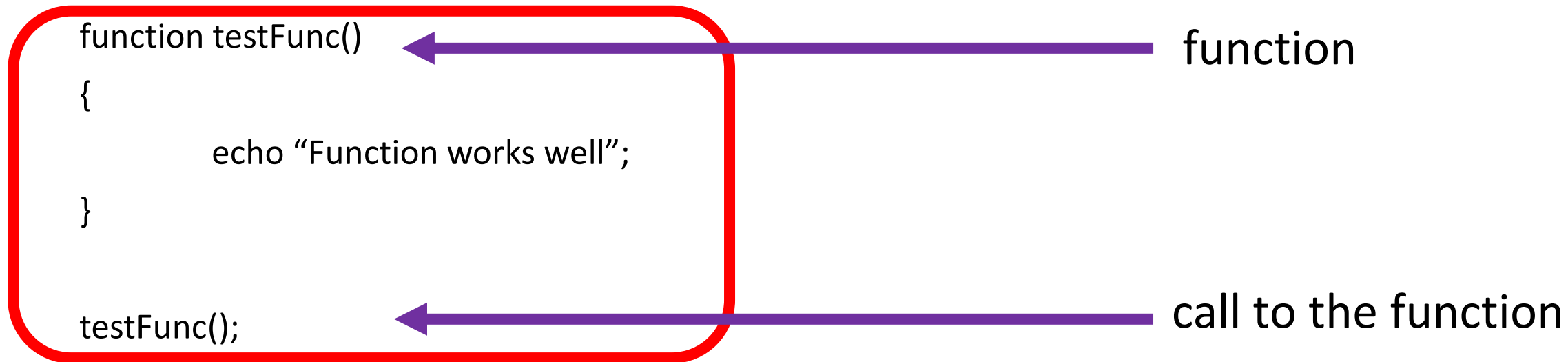
```
Key=Peter, Value=35
Key=Ben, Value=37
Key=Joe, Value=43
```

Functions

- Function is a block of statements that can be used repeatedly in a program.
- PHP Built-in Functions
 - PHP has over 1000 built-in functions that can be called directly, from within a script, to perform a specific task.
- PHP User Defined Functions
 - Besides the built-in PHP functions, it is possible to create your own functions.

User Defined Functions

- 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.



User Defined Functions

```
<!DOCTYPE html>
```

```
<html>
```

```
<body>
```

```
<?php
```

```
function writeMsg() {
```

```
    echo "Hello world!";
```

```
}
```

```
writeMsg();
```

```
?>
```

```
</body>
```

```
</html>
```

function

call the function

Global variables

- Subject to the context in which a variable is declared it will have a scope that is either local or global.
- If it is declared within the constraints of a function, it will have local scope and will not be recognized outside of its constraints.
- If you wish to declare a variable with scope beyond its constraints use the term **global**.

Global variables

```
$name = "Kamal";  
$age = 22;  
  
function testFunc()  
{  
    global $name, $age;  
    $greet = "Hello ";  
    echo $greet . "Name: " . $name . " - Age: " . $age;  
}  
  
testFunc();
```

Output

Hello Name: Kamal - Age: 22

User Defined Functions with Parameters

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

<?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);
?>

</body>
</html>
```

Output

5 + 10 = 15

7 + 13 = 20

2 + 4 = 6

Superglobal variables

- There are some built-in variables in PHP environment, which provides some special features, they are called **superglobal** variables.
- They are basically associative arrays
- You will learn different types of supergloabal variables throughout the module

Reading forms with Superglobal variables

- HTML form on a web page allows a user to enter data that is sent to a server for processing.

`$_GET["index"]`

- An array of variables passed to the current script via the URL parameters.

`$_POST["index"]`

- An array of variables passed to the current script via the HTTP POST method.

Reading forms

Methods to read forms

1. HTML form (frond end) + PHP backend to read form inputs
2. PHP form + PHP backend to read form inputs

Reading forms (HTML + PHP)

index.html

```
<form method="get" action="process.php">
  Name: <input type="text" name="txtName">
  <br>
  <input type="submit">
</form>
```

process.php

```
<?php
    echo "Hello ". $_GET["txtName"];
?>
```

Reading forms (HTML + PHP)

process.php


```
<!DOCTYPE html>
<html>
<body>
<form method="post" action="<?php echo htmlspecialchars($_SERVER["PHP_SELF"]);?>">
  Name: <input type="text" name="txtName">
  <br>
  <input type="submit" value="Submit" name="btnSubmit">
</form>
```

```
<?php
    if(isset($_POST["btnSubmit"])){
        echo "<h1> Hi " . $_POST["txtName"];
    }
?>
```

- **htmlspecialchars** — Convert special characters to HTML entities
- **\$_SERVER["PHP_SELF"]** - a super global variable that returns the filename of the currently executing script

Reading forms - textbox

```
<!DOCTYPE html>
<html>
<body>
<form method="post" action="php-html-form-textbox-example.php">
Name: <input type="text" name="fname">
      <input type="submit">
</form>
</body>
</html>
```



```
<?php
$name = $_REQUEST['fname'];
echo $name;
?>
```

Reading forms – textarea

```
<html>
<h2>PHP HTML Form textarea Example</h2>
<form name="php-html-form-textarea-example" method="POST" action="php-html-form-textarea-example.php">
  Enter Your Full Name :
  <input name="Fullname" type="text" value=""><br/><br/>
  Enter Your Address :
  <textarea name="UserAddress" rows="3" cols="20"></textarea>
  <br/><br/>
  <input name="BtnSubmit" type="submit" value="Submit">
</form>
</html>
```

- **isset** — Determine if a variable is declared and is different than NULL

```
<?php
if(isset($_POST['BtnSubmit']))
{
  echo "<h2>Bellow is the form data</h2>";
  echo "<hr>";
  echo "</br>Your Name :".$_POST['Fullname'];
  echo "</br>Your Address :".$_POST['UserAddress'];
  echo "<hr>";
}
?>
```

Output

PHP HTML Form textarea Example

Enter Your Full Name :

Enter Your Address :

Bellow is the form data

Your Name :Saman Gamage
Your Address :New Kandy Road, Malabe, Sri Lanka

Reading forms - radio

```
<html>
<h3>PHP HTML Form radio button Example</h3>
<form name="UserInformationForm" method="POST" action="php-html-form-radio-button-example.php">
  Enter Your Full Name :
  <input name="FullName" type="text" value=""><br/><br/>
  You are :
  <input name="YourGender" type="radio" value="male"> Male
  <input name="YourGender" type="radio" value="female" checked=checked"> Female<br/><br/>
  <input name="BtnSubmit" type="submit" value="Submit">
</form>
</html>
```

Output

PHP HTML Form radio button Example

Enter Your Full Name :

You are : ☒ Male ☐ Female

Your form data as bellow

Your Name :Saman Gamage
Your are a:Male

```
<?php
if(isset($_POST['BtnSubmit']))
{
  echo "<h3>Your form data as bellow</h3>";
  echo "</br>Your Name :{$_POST['FullName']}";
  echo "</br>Your are :{$_POST['YourGender']}";
  echo "<hr>";
}
?>
```


Reading forms – checkbox

```
<html>
Check box test
  <form action="cbk.php" method="post">
    <input type="checkbox" name="check_list[]" value="one">One<BR>
    <input type="checkbox" name="check_list[]" value="two">Two<BR>
    <input type="checkbox" name="check_list[]" value="three">Three<BR>
    <button type="submit">submit</button>
  </form>
</html>
```



```
<?php
echo "Check box test<hr>";
if(!empty($_POST['check_list'])) {
    foreach($_POST['check_list'] as $check) {
        echo "check=$check<br>";
    }
}else{
    echo "Check list is empty";
}
?>
```

Output

- empty — Determine whether a variable is empty

Check box test

☐ One

☐ Two

☐ Three



Check box test

Check list is empty

Check box test

☒ One

☐ Two

☒ Three



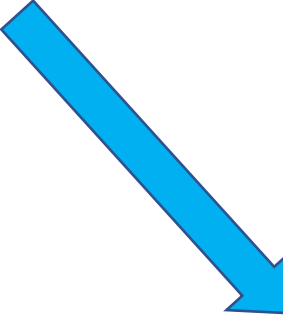
Check box test

check=one

check=three

Reading forms - Select

```
<html>
<h3>PHP HTML Form select box Example</h3>
<form name="UserInformationForm" method="POST" action="php-html-form-select-box-example.php">
    Enter Your Full Name :
    <input name="FullName" type="text" value=""><br/><br/>
    I like:
    <select name="I_like">
        <option value="java">JAVA</option>
        <option value="php">PHP</option>
        <option value="asp">ASP .Net</option>
    </select>
    <br/><br/>
    <input name="BtnSubmit" type="submit" value="Submit">
</form>
</html>
```



```
<?php
if(isset($_POST['BtnSubmit']))
{
    echo "<h3>Your submitted form data as bellow</h3>";
    echo "Your Name : {$_POST['FullName']}</br>";
    echo "I like : <h2>{$_POST['I_like']} </h2> Programming...!</br>";
    echo "<hr>";
}
?>
```

Output

PHP HTML Form select box Example	Your submitted form data as bellow
Enter Your Full Name : <input type="text" value="Saman Gamage"/>	Your Name : Saman Gamage
I like: <input type="text" value="ASP .Net"/>	I like :
<input type="submit" value="Submit"/>	asp
	Programming...!

Summary

- Arrays
- Functions
- Superglobal variables and Form handling