

Functions, Arrays and Date in PHP



Lecture 2

2021-2022

Lecturer: Dezheen Hussein Abdulazeez

E-mail: dezheen.abdulazeez@uod.ac

Computer science Dept_UOD

OUTLINES

- **Functions**
- **Arrays**
 - ✓ **Numeric Array**
 - ✓ **Associative Arrays**
 - ✓ **Multidimensional Arrays**
- **Manipulating Array**
- **PHP Date**

PHP FUNCTIONS

- The real power of PHP comes from its functions.
- In PHP, there are more than 1000 built-in available.
- Besides the built-in PHP functions, we can create our own functions.
- Function names are NOT case-sensitive.
- Syntax

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

FUNCTION

Example1: Function

```
<?php
```

```
function writeMsg(){
```

```
    echo "You are really a nice person, Have a nice time!";
```

```
}
```

```
writeMsg();
```

```
?>
```

FUNCTION WITH PARAMETERS

- Example2: Function with Parameters

```
<?php
```

```
function add($num1, $num2)
```

```
{
```

```
    $sum = $num1 + $num2;
```

```
    echo "Sum of the two numbers is : $sum";
```

```
}
```

```
    add(10, 20);
```

```
?>
```

Output: Sum of the two numbers is : 30

FUNCTION WITH RETURN VALUES

Example3: Function with Return Values

```
<?php
```

```
function add($num1, $num2)
{
    $sum = $num1 + $num2;
    return $sum;
}

echo " 7+ 8= " . add(7,8) ;
```

```
?>
```

Output: 7+ 8= 15

PHP ARRAYS

There are three different kind of arrays:

- **Indexed array** - An array with a numeric index.
- **Associative array** - An array where each ID key is associated with a value.
- **Multidimensional array** - An array containing one or more arrays.

PHP NUMERIC ARRAYS

- These arrays can store numbers, strings but their index will be represented by numbers.
- By default, the array index starts from zero.
- Example4: Numeric Arrays

```
<?php
```

```
$numbers = array( 1, 2, 3);
```

```
echo "Numbers: " . $numbers[0] . ", " . $numbers[1] . " and " .
```

```
$numbers[2] . ".";
```

```
?>
```

Output:Numbers: 1, 2 and 3.

METHODS FOR CREATING INDEXED ARRAY

1. <?php

/* First method to create array. */

\$numbers = array(1, 2, 3, 4, 5);

?>

2. <?php

/* Second method to create array. */

\$numbers[0] = "one";

\$numbers[1] = "two";

\$numbers[2] = "three";

\$numbers[3] = "four";

\$numbers[4] = "five";

?>

LOOP THROUGH AN INDEXED ARRAY

Example5: Loop Through Numeric Array

<?php

// For Loop through an numeric array

\$numbers = array(1, 2, 3, 4, 5);

\$arrlength = count(\$numbers);

for (\$i = 0; \$i <\$arrlength; \$i++) {

 echo \$numbers[\$i];

 echo "
";

}

// Foreach Loop through an numeric array

foreach(\$numbers as \$value)

{ echo "Value is \$value
"; }

?>

PHP ASSOCIATIVE ARRAYS

- Associative array will have their index as string so that you can establish a strong association between key and values.
- There are two ways to create an associative array:
 - 1) `$age = array("Peter"=>"35", "Ben"=>"37", "Joe"=>"43");`
or
 - 2) `$age['Peter'] = "35";`
`$age['Ben'] = "37";`
`$age['Joe'] = "43";`

ASSOCIATIVE ARRAYS

Example6: Associative Array

```
<?php
```

```
$age = array("Peter"=>"35", "Ben"=>"37", "Joe"=>"43");
```

```
echo "Peter is " . $age['Peter'] . " years old.";
```

```
?>
```

Output: Peter is 35 years old.

LOOP THROUGH AN ASSOCIATIVE ARRAY

Example7: Associative Array Looping

```
<?php
```

```
$age = array("Peter"=>"35", "Ben"=>"37", "Joe"=>"43");
```

```
foreach($age as $x => $x_value){
```

```
    echo $x."s age is " . $x_value;
```

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

```
}
```

```
?>
```

Output: Peter's age is 35
Ben's age is 37
Joe's age is 43

EXERCISE

Q8: Store the salaries of employees in an array and displayed them with foreach loop.

Employee Name	Salary
Dana	750000
Zana	800000
Snur	900000

Answer: `$salaries = array(" Dana " => 750000, " Zana " => 800000,"
Snur" => 900000);`

```
foreach($ salaries as $name=> $salary) {  
    echo "Salary of " . $name. " is " . $salary;  
    echo "<br>";}
```

MULTIDIMENSIONAL ARRAYS

- An array containing one or more arrays and values are accessed using multiple index.

Books	Stock	Sold
HTML	20	15
CSS	12	17
JavaScript	9	8
PHP	17	15

- We can store the data from the table above in a two-dimensional array, like this:

TWO-DIMENSIONAL ARRAYS

Example9: `<?php`

```
$books = array(  
    array("HTML",20,15),  
    array("CSS",12,17),  
    array("JavaScript",9,8),  
    array("PHP",17,15)  
);
```

```
for ($i = 0; $i < 4; $i++) {  
    echo "<p><b>Row number $i</b></p>";  
    echo "<ul>";  
    for ($j = 0; $j < 3; $j++) {  
        echo "<li>".$books[$i][$j]."</li>";  
    }  
    echo "</ul>";  
}
```

`?>`

OUTPUT

Row number 0

- HTML
- 20
- 15

Row number 1

- CSS
- 12
- 17

Row number 2

- JavaScript
- 9
- 8

Row number 3

- PHP
- 17
- 15

PHP SORTING ARRAYS

- `sort()` - sort arrays in ascending order.
- `rsort()` - sort arrays in descending order.
- `asort()` - sort associative arrays in ascending order, according to the value.
- `ksort()` - sort associative arrays in ascending order, according to the key.
- `arsort()` - sort associative arrays in descending order, according to the value.
- `krsort()` - sort associative arrays in descending order, according to the key.

SORTING ARRAY

Example10: Sorting Array

```
<?php
```

```
$books = array("HTML", "CSS", "JavaScript", "PHP");
```

```
sort($books);
```

```
foreach($books as $value)
```

```
{
```

```
    echo $value;
```

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

```
}
```

```
?>
```

DESCENDING ASSOCIATIVE ARRAY

Example11: Descending Associative Array

```
<?php
```

```
$age = array("Peter"=>"45", "Ben"=>"37", "Joe"=>"43");
```

```
arsort($age);
```

```
foreach($age as $x => $x_value){
```

```
    echo $x."s age is " . $x_value;
```

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

```
?>
```

MANIPULATING ARRAYS

- **ARRAY_PUSH():** Adds one or more elements to the end of an existing array.
- Syntax: `array_push($existingArray, 1 , 2 , 3);`
- Example12: Array Push

```
<?php
```

```
    $first=array( "a", "b", "c" );
```

```
        array_push( $first, 1 , 2 , 3);
```

```
Foreach ( $first as $val) {
```

```
    echo"$val <br>" ;    }
```

```
?>
```

MANIPULATING ARRAYS

- **ARRAY_POP():** Removes and returns the value of the last element of an array.
- Syntax: `$last_element=array_pop($existingArray);`
- Example13: Array Pop

<?php

```
$existingArray=array( "a", "b", "c" );  
  
$last_element=array_pop($existingArray);  
    while ($last_element!=NULL) {  
        echo"$last_element<br>" ;  
  
        $last_element=array_pop($existingArray); }  

```

?>

MANIPULATING ARRAYS

- **ARRAY_SHIFT():** Removes the first element of an array of an existing array.
- `$first_element=array_shift($existingArray);`
- Example14: Array Shift

`<?php`

```
$existingArray=array( "a", "b", "c" );  
$first_element=array_shift($existingArray);  
while ($first_element!=NULL) {  
    echo"$first_element<br>" ;  
    $first_element=array_shift($existingArray);  
}
```

`?>`

MANIPULATING ARRAYS

- **ARRAY_UNSHIFT():** Add one or more elements to the beginning of an array.
- Syntax: `array_unshift($existingArray, 1 , 2 , 3);`
- Example15: Array Unshift

`<?php`

```
$ExistingArray = array("three","four","five");
```

```
array_unshift($ExistingArray , "one","two");
```

```
Foreach ( $ExistingArray as $val) {
```

```
    echo"$val <br>" ;  }
```

`?>`

MANIPULATING ARRAYS

- `Array_Merge()` : Merges the elements of one or more arrays together so that the values of one are appended to the end of the previous one.
- Example16: Array Merge

`<?php`

```
$first=array( "a", "b" , "c");
```

```
$second=array(1,2,3);
```

```
$third= array_merge( $first , $second ) ;
```

```
foreach( $third as $val) {
```

```
    echo "$val<br>";}
```

`?>`

PHP DATE

- The PHP date() function is used to format a date and/or a time.
- Here are some characters that are commonly used for dates:
 - ✓ d - The day of the month (01-31).
 - ✓ m – The current month, as a number (01-12).
 - ✓ Y – The current year in four digits.
 - ✓ l (lowercase 'L') - The day of the week.
- Other characters, like "/", ".", or "-" can also be inserted between the characters to add additional formatting.

GET A SIMPLE DATE WITH DATE() FUNCTION

Example17: Formats today's date in three different ways.

```
<?php
    echo "Today is " . date("Y/m/d") . "<br>";
    echo "Today is " . date("Y.m.d") . "<br>";
    echo "Today is " . date("Y-m-d") . "<br>";
    echo "Today is " . date("l");
?>
```

Output: Today is 2022/01/27
 Today is 2022.01.27
 Today is 2022-01-27
 Today is Thursday

EXERCISE

- Q19:How to automatically update the copyright year on a website?
- Answer:

© 2010-<?php echo date("Y")?>

- Output: © 2010-2022