German Jordanian University الجامعة الألمانية الأردنية

CS355 Web Technologies

Dr. Ismail Hababeh

German-Jordanian University

Lecture 11 – PHP 4

PHP Loops

- Loops are used to repeat executing the same block of code as long as a certain condition is true or a specific number of times.
- PHP loop types:
  - while
  - do...while
  - For
  - foreach

PHP While Loop

The while loop executes a block of code as long as the specified condition is true.

- Example: Display odd numbers from 1 to 10

```
<?php
   $number = 1;
   while($number <= 10)
   {
      echo "The number is: $number <br>";
      $number += 2;
   }
?>
```

PHP Do .. While Loop

The do .. while loop executes a block of code as long as the specified condition is true.

- Example: What is the output of the following program?

```
<?php
 $x = 11;
 do
      {
      echo "The number is: $x <br>";
      $x++;
   }
while($x <= 10);
?>
```

PHP For Loop

The for loop iterates through a block of code a specific number of times.

Example: What will be the output of the following program?

```
<?php
 for ($x = 10; $x >= 0; $x--)
 {
    echo "The number is: $x <br>";
 }
?>
```

PHP Foreach Loop

The foreach loop works only on arrays. The foreach loop iterates through a block of code for each element in an array.

- Syntax:

```
foreach ($arrayname as $value)
{
  code to be executed;
}
```

PHP Foreach Loop

- Example: The following foreach loop displays the values of the given array ($languages):

```
<?php
  $languages = array("C++", "Java", "Python", "C#");
  foreach ($languages as $value) {
  echo "$value <br>";
}
?>
```

PHP Foreach Loop

- Example: The following PHP code displays both the keys and the values of the associative array ($grade):

```
<?php
$grade = array("Omar"=>"95", "Ahmad"=>"77", "Zaid"=>"83");
foreach($grade as $key => $value)
  {
    echo " $key . ": " . $value  <br>";
}
?>
```

PHP Break Statement

- The break statement is used to quit from a loop.
- Example:

```
<?php
for ($counter = 0; $counter < 10; $counter++) {
  if ($counter == 5) {
    break;
  }
  echo "The counter is: $counter <br>";
}
?>
```

PHP Continue Statement

- The continue statement is used to break one iteration in a loop.
- Example:

```
<?php
for ($counter = 0; $counter < 10; $counter++) {
  if ($counter == 5) {
    continue;
  }
  echo "The counter is: $counter <br>";
}
?>
```

PHP Functions

- PHP has large number of built-in functions.
- PHP supports user-defined functions.

- Syntax:

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

PHP Functions without Arguments

- Example 1:

```
<?php
function welcome()  // function definition
  {
    echo "Hello Students!";
  }

welcome(); // function call
?>
```

PHP Functions with Arguments

- In the following example, we use function arguments without data type.
- PHP automatically associates a data type to the variable, depending on its value.

```
<?php
function studentGPA($fname, $GPA)
{
  echo "$fname. "GPA : ". $GPA <br>";
}
studentGPA(“Laith", "2.81");
studentGPA("Mohammad", "3.97");
studentGPA(“Zaid", "3.55");
?>
```

PHP Functions Throwing Exceptions

- In the following example, we declare strict to throw exception.

```
<?php
declare(strict_types=1);
function totalGrade(int $grade, int $bonus)
 {
   return $grade + $bonus;
 }
echo totalGrade(85, "5 points");
?>
```

Note: since strict is enabled, then "5 points" is not an int, and the program throws error exception.

PHP Function Arguments with Data Type

- Example:

```
<?php
// missing declare strict
function totalGrade(int $grade, int $bonus)
 {
   return $grade + $bonus;
 }
echo totalGrade(85, "5 points");
?>
```

Note: since strict is not enabled, then "5 points" is converted to int(5), and the function will return 90

PHP Function Returned Data Type

- Use data type casting to specify a different return type than the argument types.
- Example:

```
 <?php
function sum(float $a, float $b) : int
{
  return ($a + $b);
}
echo sum(10.5, 20.2);
?>
```

Passing PHP Function Arguments by Reference

- Use a pass-by-reference argument to update a function variable.
- The & operator is used to change a function argument into a reference.
- Example:

```
<?php
function add(&$value) {
  $value += 2;
}
$x = 2;
add($x);
echo $x;
?>
```

PHP Date and Time

- The date() function is used to format a date.
  - d - Represents the day of the month (1 – 31)
  - m - Represents a month (1 - 12)
  - Y - Represents a year (4 digits)
  - l (lowercase 'L') - Represents the day of the week

PHP Date and Time

- The date() function is used to format a time.
  - H - 24-hour format (00 - 23)
  - h - 12-hour format (01 - 12)
  - i - Minutes (00 - 59)
  - s - Seconds (00 - 59)
  - a - Lowercase (am or pm)

PHP Date – Example

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

Output:
Today is 06/04/2025
Today is 06.04.2025
Today is 06-04-2025
Today is Sunday

PHP Time – Example

```
<?php
echo "The time now is " . date("h:i:sa");
?>
```

Output:

The time now is 11:40:33am

PHP Date and Time – Example 1

The mktime() make time function returns the Unix timestamp for a date.

syntax: mktime(hour, minute, second, month, day, year)

```
<?php
$md=mktime(11, 40, 33, 04, 06, 2025);
echo " Today is " . date("d-m-Y h:i:sa", $md);
?>
```

Output:
Today is 06-04-2025 11:40:33am

PHP Date and Time – Example 2

The strtotime() string to time function is used to convert a string date into a Unix timestamp.

```
<?php
$md=strtotime("11:50am April 06 2025");
echo " New date is " . date("d-m-Y h:i:sa", $md);
?>
```

Output:

My date is 06-04-2025 11:50:00am

PHP Future Dates and Time – Example 1

```
<?php
$md=strtotime("tomorrow");
echo date("d-m-Y h:i:sa", $md) . "<br>";
$md=strtotime("next Sunday");
echo date("d-m-Y h:i:sa", $md) . "<br>";
?>
```

Output:

07-04-2025 12:00:00am
13-04-2025 12:00:00am

PHP Text File Functions – Example 1

The readfile() function reads a file and writes it to the output buffer.

Assume the following text file "CScourses.txt", stored in the server:

```
CS355 = Web Technologies
CS342 = Software Engineering
CS263 = Database Management Systems
```

PHP Text File Functions – Example 1

Write a PHP code that allows to write the contents of the previous file CScourses.txt to the output buffer:

```
<?php
echo readfile("CScourses.txt") . "<br>";
?>
```

Output:

CS355 = Web Technologies

CS342 = Software Engineering

CS263 = Database Management Systems

PHP Text File Functions – Handling Errors

The fopen() function gives more options in reading a file.

Example:

```
<?php
$myfile = fopen("CScourses.txt", "r") or die("Unable to open file!");
/*you can handle errors by calling or die to exit when you encounter certain errors. It used to print message and exit from the current php script. It is equivalent to exit() function in PHP. */
?>
```

PHP Text File Functions – Example 2

The fread() is used to read some data from a file.

```
<?php
$myfile = fopen("CScourses.txt", "r") or die("Unable to open file!");
echo fread($myfile,filesize("CScourses.txt"));
fclose($myfile);
?>
```

Notes:

"r" parameter means that the file is opened for read only.

fread() function reads from an opened file.

fclose() function is used to close an open file.

PHP Text File Functions – Example 3

The fgets() is used to read a single line from a file.

```
<?php
$myfile = fopen("CScourses.txt", "r") or die("Unable to open file!");
echo fgets($myfile);
fclose($myfile);
?>
```

Output:

CS355 = Web Technologies

PHP Text File Functions – Example 4

The fwrite() is used to write to an open file. It writes a specified number of bytes from the memory address specified and places them into the file.

```
<?php
$myfile = fopen("newfile.txt", "w") or die("Unable to open file!");
$x = "hello Students\n";
fwrite($myfile, $x);
$x = "Have a nice day!\n";
fwrite($myfile, $x);
fclose($myfile);
?>
```

The "newfile.txt" file would consists of two lines:
Hello Students
Have a nice day!

PHP File Upload Steps

To upload files to the server, perform the following steps:

Ensure that PHP is configured to allow file uploads:

1. Configure The "php.ini" File
   – in "php.ini" file, set the file_uploads directive to on.

PHP File Upload Steps

2. Create the HTML Form that allow user to choose the file for upload.

```
<form action="upload.php" method="post"
enctype="multipart/form-data">
<input type="file" name="fileToUpload" id="fileToUpload">
<input type="submit" value="Upload file" name="submit">
</form>
```

PHP File Upload Steps

3. Rules on the HTML form:

- Make sure that the form uses method="post“

//specifies that the form data will be sent to the server by storing it in HTTP request body. This method is used to transfer data securely using HTTP headers.

- The form also needs the following attribute:

  enctype="multipart/form-data"

// form data divides into multiple parts and send to server

PHP File Upload Steps

3. Rules on the HTML form:

   – The type="file" attribute of the <input>

   //let the user choose one or more files from their device storage.

   This tag shows the input field as a file-select control, with a

   "Browse" button next to the input control.

- The created form sends data to a file "upload.php".

PHP File Upload

```
<?php
$target_dir = "uploads/";
$target_file = $target_dir .
basename($_FILES["fileToUpload"]["name"]);
$uploadOk = 1;
$FileType =
strtolower(pathinfo($target_file,PATHINFO_EXTENSION));
?>
```

PHP File Upload

Notes:

- $target_dir = "uploads/"

// specifies the directory where the file is going to be placed

- $target_file // specifies the path of the file to be uploaded
- $uploadOk=1 //is not used yet (will be used later)

  ($uploadOk = 0; // File already exists).
- $FileType //returns information about a file path and

  holds the file extension (in lower case)