Web Technology II (BIT301)

Instructor: Prakash Neupane

Introduction:Flow Control Statements

- Control Statements:
  - PHP supports a number of traditional programming constructs for controlling the flow of execution of a program.
  - Conditional statements, such as if/else and switch, allow a program to execute different pieces of code, or none at all, depending on some condition.
  - Loops, such as while and for, support the repeated execution of particular segments of code.

Introduction:Flow Control Statements

if Statement

- The ***if*** statement checks the truthfulness of an expression,
- if the expression is true, evaluates a statement.
- An if statement looks like:

  if (expression)statement
- To specify an alternative statement to execute when the expression is false, use the ***else*** keyword:

  if (expression)

  statement

  else statement

- Example:

```
<?php
$t = date("H");

if ($t < "20") {
  echo $t;
}
?>
```

Introduction:Flow Control Statements

if...else Statement

- The ***if...else*** statement executes some code if a condition is true and another code if that condition is false.
- Syntax:

```
 if (expression) {
  //code to be executed if condition is true;
} else {
  //code to be executed if condition is false;
}
```

- Example:

```
<?php
$t = date("H");

if ($t < "10") {
  echo "If branch is executed!";
} else {
  echo "Else branch is executed!";
}
?>
```

Introduction:Flow Control Statements

if...elseif...else Statement

- The if...elseif...else statement executes different codes for more than two conditions.
- Syntax:

```
 if (expression) {
  //code to be executed if condition is true;
} elseif(expression) {
  //code to be executed if condition is false and this condition is true;
} else {
//code to be executed if all conditions are false;
}
```

- Example:

```
<?php
$t = date("H");
echo "<p>The hour of the class is " . $t;
echo ", and will give the following message:</p>";

if ($t < "10") {
  echo "we are missing those who are absent”;
} elseif ($t < "20") {
  echo "Have a good day!";
} else {
  echo "Have Fun!";
}
?>
```

Introduction:Flow Control Statements

switch Statement

- The **switch** statement is used to perform different actions based on different conditions.
- Syntax:

```
switch (n) {
  case label1:
   // code to be executed if n=label1;
    break;
  case label2:
 //   code to be executed if n=label2;
    break;
  case label3:
   // code to be executed if n=label3;
    break;
    ...
  default:
  //  code to be executed if n is different from all labels;
}
```

- Example:

```
<?php
$favcolor = "red";

switch ($favcolor) {
  case "red":
    echo "Your favorite color is red!";
    break;
  case "blue":
    echo "Your favorite color is blue!";
    break;
  case "green":
    echo "Your favorite color is green!";
    break;
  default:
    echo "Your favorite color is neither red, blue, nor green!";
}
?>
```

Introduction:Flow Control Statements

Quick Exercise: Conditional Statements

- Write a PHP program that takes a student's score as input and displays their grade based on the following grading system:
  Score 90-100: A

  Score 80-89: B

  Score 70-79: C

  Score 60-69: D

  Score below 60: F
- Write a PHP program that checks whether a given year is a leap year or not. A leap year is divisible by 4, except for years that are divisible by 100 but not divisible by 400.

Introduction:Flow Control Statements

Quick Exercise: Conditional Statements

- Write a PHP program that converts a given temperature in Fahrenheit to Celsius using the formula Celsius = (Fahrenheit - 32) * 5/9. Additionally, create a conditional statement to print funny messages based on the temperature values. Use the following temperature ranges for your messages:

  If the temperature in Celsius is less than 0, print "Brrr! It's freezing!"

  If the temperature in Celsius is between 0 and 15 (inclusive), print "It's a bit chilly, isn't it?"

  If the temperature in Celsius is between 15 and 25 (inclusive), print "It's a pleasant temperature."

  If the temperature in Celsius is between 25 and 35 (inclusive), print "It's getting warm!"

  If the temperature in Celsius is greater than 35, print "It's scorching hot!"

Introduction:Flow Control Statements

Quick Exercise: Conditional Statements

- Write a PHP program that simulates a traffic light. The program should take a color code as input (1 for red, 2 for yellow, 3 for green) and use a switch state0000000ment to display a message indicating the meaning of the traffic light color ("Stop," "Slow down," "Go").

Introduction:Flow Control Statements

Break, Continue and Exit

Introduction:Flow Control Statements

Loops

- Frequently, when you're writing code, there's a need to execute a specific set of instructions repeatedly.
- Rather than duplicating nearly identical lines of code in your script, loops provide an elegant solution.
- Loops allow you to repeatedly run the same block of code as long as a particular condition remains true.
- They provide a way to efficiently automate repetitive tasks and streamline your code by ensuring that you don't have to write similar code segments multiple times.

- In PHP, we have the following loop types:
  - **while** - loops through a block of code as long as the specified condition is true
  - **do...while** - loops through a block of code once, and then repeats the loop as long as the specified condition is true
  - **for** - loops through a block of code a specified number of times
  - **foreach** - loops through a block of code for each element in an array

Introduction:Flow Control Statements

Loops: while

- The ***while*** loop - Loops through a block of code as long as the specified condition is true.
- Syntax:

```
while (condition is true) {
  //code to be executed;
}
```

- Example:

```
<?php
$x = 0;
while($x <= 100) {
  echo "The number is: $x \n";
  $x+=10;
}
?>
```

Introduction:Flow Control Statements

Loops: do...while

- The **do...while** loop - Loops through a block of code once, and then repeats the loop as long as the specified condition is true.
- Syntax:

```
 do {
  //code to be executed;
} while (condition is true);
}
```

- Example:

```
<?php
$x = 0;

do {
  echo "The number is: $x \n";
  $x++;
} while ($x <= 5);
?>
```

Introduction:Flow Control Statements

Loops: for

- The **for** loop - Loops through a block of code a specified number of times.
- Syntax:

for (init counter; test counter; increment counter) {

//code to be executed for each iteration;

}

- Example:

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

Introduction:Flow Control Statements

Loops: for

- The **foreach** loop - Loops through a block of code for each element in an array.
- The **foreach** loop works only on arrays, and is used to loop through each key/value pair in an array.
- Syntax:

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

- *Example:*

```
<?php
$phone =
array("s1"=>"9812345670",
"s2"=>"9842512345",
"s3"=>"9852012378");

foreach($phone as $key => $value) {
  echo "$key = $value \n";
}
?>
```

Introduction:Flow Control Statements

Quick Exercise: Loops

- *You're writing code for a smart coffee machine. Write a PHP program that asks the user how many cups of coffee they want to brew. Then, using a loop, display a message for each cup, such as "Brewing cup number X." After all the cups are brewed, display "Coffee is ready!" and celebrate with a virtual coffee break.*

Introduction:Flow Control Statements

Including PHP Code and Embedding PHP in Web Pages