

Intro to PHP

Front End Development

- Front End Development manages everything that users visually see first in their browser or application. Front End Developers are responsible for the look and feel of a site.
- Front end languages include HTML, CSS, and Javascript.
- Sites created by Front End Developers won't interact with information stored on a database in order to be functional.
- The content will be “static”, meaning that large pieces of new data will not be constantly uploaded.
- Small business owners and restaurants usually have great examples of static sites.

Back End Development

- Back end development refers to the server side of an application and everything that communicates between the database and the browser.
- Many back end developers know front end languages such as HTML and CSS but need to use languages such as Java, PHP, Ruby on Rails, Python, and .Net to get the back end job done.
- These languages are used to create dynamic sites which are different from static sites in that these types of websites store database information.
- Content on the site is constantly changing and updating. Examples of dynamic sites include Facebook, Twitter, and Google Maps.

What is PHP?

- PHP is a server scripting language, and a powerful tool for making dynamic and interactive Web pages.

What is PHP?

- PHP is a server scripting language, and a powerful tool for making dynamic and interactive Web pages.
- PHP is an acronym for “PHP: Hypertext Preprocessor”
- PHP is a widely-used, open source scripting language
- PHP scripts are executed on the server
- PHP is free to download and use.

What is PHP?

- It is powerful enough to be at the core of the biggest blogging system on the web: Wordpress
- It is deep enough to run the largest social network: Facebook
- It is also easy enough to be a beginner's first server side language.

What is a PHP File?

- PHP files can contain text, HTML, CSS, JavaScript, and PHP Code
- PHP files have extension “.php”

What can PHP do?

- PHP can generate dynamic page content
- PHP can create, open, read, write, delete, and close files on the server
- PHP can collect form data
- PHP can send and receive cookies
- PHP can add, delete, modify data in your database
- PHP can be used to control user-access
- PHP can encrypt data

Basic PHP Syntax

- A PHP script can be placed anywhere in the document.
- A PHP script starts with “<?php” and end with “?>”

```
<?php  
// PHP code goes here  
?>
```

Basic PHP Syntax

- Below, we have an example of a simple PHP file, with a PHP script that uses a built-in PHP function "echo" to output the text "Hello World!" on a web page:

```
<body>

    <h1>My first PHP page</h1>

    <?php
    echo "Hello World!";
    ?>

</body>
```

Declaring PHP Variables

- In PHP, a variable starts with the “\$” sign, followed by the name of the variable:

```
<?php

$txt = "Hello world!";
$x = 5;
$y = 10.5;

?>
```

Rules for PHP variables

- A variable starts with the \$ sign, followed by the name of the variable
- A variable name must start with a letter or the underscore character
- A variable name cannot start with a number
- A variable name can only contain alpha-numeric characters and underscores (A-z, 0-9, and _)
- Variable names are case-sensitive (“\$age” and “\$AGE” are two different variables)

PHP echo statement

- With PHP, you can use “echo” to get output to your page.
- The following example shows how to output text with the echo command (notice that the text can contain HTML markup):

```
<?php|  
  
echo "<h2>PHP is Fun!</h2>";  
echo "Hello world!<br>";  
echo "I'm about to learn PHP!<br>";  
echo "This ", "string ", "was ", "made ", "with multiple parameters.";   
  
?>
```

PHP echo statement

- The following example shows how to output text and variables with the “echo” statement.

```
$txt1 = "Learn PHP";  
$txt2 = "Jax Code Academy";  
$x = 5;  
$y = 4;  
  
echo "<h2>" . $txt1 . "</h2>";  
echo "Study PHP at " . $txt2 . "<br>";  
echo $x + $y;
```

PHP echo statement

- The following example shows how to output text and variables with the “echo” statement.

```
$txt1 = "Learn PHP";  
$txt2 = "Jax Code Academy";  
$x = 5;  
$y = 4;  
  
echo "<h2>" . $txt1 . "</h2>";  
echo "Study PHP at " . $txt2 . "<br>";  
echo $x + $y;
```

PHP Comparison Operators

- PHP uses the same comparison operators as JavaScript. For example, `>`, `<`, `==`, `===`, `!==`, etc.

PHP Logical Operators

- PHP can use the same logical operators as JavaScript, but can also use the words for the symbols as well.
 - Can use “&&” or “AND”
 - Can use “|” or “OR”
- Also comes with a new logical operator: XOR
 - True if either condition is true, but not both.

PHP if/else/else if

- Like JavaScript, PHP also utilizes conditionals. Here is the syntax for an if statement.

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

```
$x = 20;  
  
if ($x > 10) {  
    echo "It's true!";  
}
```

PHP if/else/else if

- Like JavaScript, PHP also utilizes conditionals. Here is the syntax for an if/else statement.

```
if (condition) {  
    //code to execute if true  
} else {  
    //code to execute if false  
}  
  
?>
```

```
$x = 20;  
  
if ($x > 10) {  
    echo "It's true!";  
} else {  
    echo "It's false!";  
}
```

PHP if/else/elseif

- Like JavaScript, PHP also utilizes conditionals. Here is the syntax for an if/elseif/else statement.

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

```
$x = 20;  
  
if ($x > 10) {  
    echo "It's greater than 10!";  
} elseif ($x > 5) {  
    echo "It's less than 10 but greater than 5!";  
} else {  
    echo "It's less than 5!";  
}
```

PHP while loop

- Like JavaScript, PHP also utilizes while loops. Here is the syntax for a while loop.

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

```
$x = 1;  
  
while ($x <= 10) {  
    echo $x;  
    $x++;  
}
```

PHP for loop

- Like JavaScript, PHP also utilizes while loops. Here is the syntax for a for loop.

```
for (init counter; test counter; increment counter) {  
    //code to be executed;  
}
```

```
for ($x = 0; $x <= 10; $x++) {  
    echo $x;  
}
```

PHP functions

- Like JavaScript, PHP also utilizes functions. They work exactly the same as they do in JavaScript.

```
function sum($x, $y) {  
    echo $x + $y;  
}
```

```
writeMsg(5, 10); //would return 15
```

PHP arrays

- Like JavaScript, PHP also utilizes arrays. They are very similar to JavaScript arrays except a little difference in the syntax. To declare and access an array:

```
$cars = array("Volvo", "BMW", "Toyota");  
echo "I like " . $cars[0] . ", " . $cars[1] . " and " . $cars[2] . " .";
```

- Arrays are indexed the same way as they are in JavaScript, starting at "0".

PHP foreach loop

- Like JavaScript, PHP also utilizes foreach loops to loop through an array. The syntax is different though. The syntax for a foreach loop in PHP looks like this:

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

```
$colors = array("red", "green", "blue", "yellow");  
  
foreach ($colors as $color) {  
    echo $color;  
}
```

Assignment 15.1: PHP Problem Set

- Create a function that takes a number as an argument, increments the number by +1 and returns the result.
- Given two strings, “firstName” and “lastName”, return a single string in the format “last, first”.
- Create a function that takes a number as its only argument and returns “true” if it’s less than or equal to zero, otherwise return “false”.
- Create a function that takes an array of numbers. Return the largest number in the array.
- Create a function that takes an array of numbers and returns the smallest number in the set.