Hypertext Preprocessor

PHP

- PHP (recursive acronym for PHP: Hypertext Preprocessor) is a widely-used open source general-purpose scripting language that is especially suited for web development and can be embedded into HTML.

The History of PHP

- PHP, which stands for "PHP: Hypertext Preprocessor," was created by Rasmus Lerdorf in 1994. Originally PHP was known as "Personal Home Page Tools" and was created as a series of scripts to manage Lerdorf's personal website responsibilities; however, it has since evolved into a more powerful and adaptable programming language.

- In 1995, Lerdorf released the source code for PHP, and it gained attention from the developer community. PHP underwent a major overhaul with the assistance of Andi Gutmans and Zeev Suraski and was released as PHP/FI (Personal Home Page/Forms Interpreter) version 2.0. This rework included a C-written parser, which considerably improved efficiency and enabled PHP to be utilized for more complex web applications.

What is a PHP file?

- PHP files can contain text, HTML, CSS, JavaScript, and PHP code
- PHP code is executed on the server, and the result is returned to the browser as plain HTML
- PHP files have extension ".php"

What is a PHP file?

```
<!DOCTYPE html>
<html>
	<head>
		<title>Example</title>
	</head>
	<body>

	<?php
		echo "Hi, I'm a PHP script!";
	?>

	</body>
</html>
```

Comments in PHP

A comment in PHP code is a line that is not executed as a part of the program. Its only purpose is to be read by someone who is looking at the code.

- Comments can be used to:
- Let others understand your code
- Remind yourself of what you did - Most programmers have experienced coming back to their own work a year or two later and having to re-figure out what they did. Comments can remind you of what you were thinking when you wrote the code
- Leave out some parts of your code

Comments in PHP

- // This is a single-line comment
- # This is also a single-line comment
- /* This is a multi-line comment */

PHP echo

```php
echo "Hello";
//same as:
 echo("Hello");

$txt1 = "Learn PHP";
 $txt2 = "W3Schools.com";
echo "<h2>$txt1</h2>";
echo "<p>Study PHP at $txt2</p>";
```

PHP echo

```
$txt1 = "Learn PHP";
 $txt2 = "W3Schools.com";
echo '<h2>' . $txt1 . '</h2>’;
echo '<p>Study PHP at ' . $txt2 . '</p>';
```

PHP Variables

Variables are "containers" for storing information.

```php
<?php
        $x = 5;
        $y = "John";
?>
```

PHP Variables Data Types

String

Integer

Float (floating point numbers - also called double)

Boolean

Array

Object

NULL

Resource

PHP if Statements

- if statement - executes some code if one condition is true
- if...else statement - executes some code if a condition is true and another code if that condition is false
- if...elseif...else statement - executes different codes for more than two conditions
- switch statement - selects one of many blocks of code to be executed

PHP Shorthand if Statements

```
$a = 5;

if ($a < 10) $b = "Hello";

echo $b
```

Short Hand If...Else

```
$a = 13;

$b = $a < 10 ? "Hello" : "Good Bye";

echo $b;
```

PHP switch Statement

```
switch (expression) {
        case label1:
        //code block
        break;

        case label2:
        //code block;
        break;

        case label3:
        //code block
        break;

        default:
        //code block
}
```

PHP Loops

Loops are used to execute the same block of code again and again, as long as a certain condition is true.

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

PHP while Loop

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

```
$i = 1;

        while ($i < 6) {

        echo $i;

        $i++;

}
```

PHP do while Loop

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

```
$i = 1;
do {
        echo $i;
        $i++;
}
while ($i < 6);
```

PHP for Loop

The for loop is used when you know how many times the script should run.

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

        echo "The number is: $x <br>";

}
```

PHP foreach Loop

The most common use of the foreach loop, is to loop through the items of an array.

```
$colors = array("red", "green", "blue", "yellow");
 foreach ($colors as $x) {
            echo "$x <br>";
 }
```

PHP Break

The break statement can be used to jump out of different kind of loops.

```
for ($x = 0; $x < 10; $x++) {
        if ($x == 4) {
                break;
        }
        echo "The number is: $x <br>";
}
```

PHP Continue

The continue statement can be used to jump out of the current iteration of a loop, and continue with the next.

```
for ($x = 0; $x < 10; $x++) {
        if ($x == 4) {
                continue;
        }
        echo "The number is: $x <br>";
}
```

PHP Functions

A user-defined function declaration starts with the keyword function, followed by the name of the function:

```
function myMessage() {

        echo "Hello world!";

}

//To call the function, just write its name followed by parentheses ():

myMessage();
```

PHP Arrays

An array is a special variable that can hold many values under a single name, and you can access the values by referring to an index number or name.

```
$cars = array("Volvo", "BMW", "Toyota");

// Same as the code below

$cars = ["Volvo", "BMW", "Toyota"];
```

PHP Indexed Arrays

In indexed arrays each item has an index number.
By default, the first item has index 0, the second item has item 1, etc.

```
$cars = array("Volvo", "BMW", "Toyota");

echo $cars[0];
```

PHP Associative Arrays

Associative arrays are arrays that use named keys that you assign to them.

```php
$car = array("brand"=>"Ford", "model"=>"Mustang", "year"=>1964);

echo $car["model"];

$car["year"] = 2024;
```

PHP Add Array Items

```php
$fruits = array("Apple", "Banana", "Cherry");

$fruits[] = "Orange";

//For Associative arrays

$cars = array("brand" => "Ford", "model" => "Mustang");

$cars["color"] = "Red";
```

PHP Add Array Items

```php
$fruits = array("Apple", "Banana", "Cherry");

array_push($fruits, "Orange", "Kiwi", "Lemon");


//Add Multiple Items to Associative Arrays

$cars = array("brand" => "Ford", "model" => "Mustang");

 $cars += ["color" => "red", "year" => 1964];
```

PHP Delete Array Items

To remove an existing item from an array, you can use the array_splice() function.

With the array_splice() function you specify the index (where to start) and how many items you want to delete.

```
$cars = array("Volvo", "BMW", "Toyota");
array_splice($cars, 1, 1);
//Removes the BMW
```

PHP Delete Array Items

You can also use the unset() function to delete existing array items.

```
$cars = array("Volvo", "BMW", "Toyota");

unset($cars[1]);
```

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

PHP Sorting Arrays

```php
$cars = array("Volvo", "BMW", "Toyota");
sort($cars);

$numbers = array(4, 6, 2, 22, 11);
sort($numbers);
```

PHP Multidimensional Arrays

A multidimensional array is an array containing one or more arrays.

```
$cars = array (
            array("Volvo",22,18),
            array("BMW",15,13),
            array("Saab",5,2),
            array("Land Rover",17,15)
 );
```

PHP Multidimensional Arrays

Hands-on 2 Activity

Thank you

Lemuel R. Francisco

Source:
https://www.w3schools.com/php