PHP

Intro, Syntax, Variables, Echo, Data Types

Introduction

PHP scripts are executed on the server.

What You Should Already Know

Before you continue you should have a basic understanding of the following:

* HTML
* CSS
* JavaScript

Introduction

What is PHP?

* 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 costs nothing, it is free to download and use

What is a PHP File?

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

Introduction

What Can PHP Do?

* PHP can generate dynamic page content
* PHP can create, open, read, write, 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 restrict users to access some pages on your website
* PHP can encrypt data
* With PHP you are not limited to output HTML. You can output images, PDF files, and even Flash movies. You can also output any text, such as XHTML and XML.

Introduction

Why PHP?

* PHP runs on various platforms (Windows, Linux, Unix, Mac OS X, etc.)
* PHP is compatible with almost all servers used today (Apache, IIS, etc.)
* PHP supports a wide range of databases
* PHP is free. Download it from the official PHP resource: www.php.net
* PHP is easy to learn and runs efficiently on the server side

Installation

What Do I Need?

* To start using PHP, you can:
* Find a web host with PHP and MySQL support
* Install a web server on your own PC, and then install PHP and MySQL

Installation

Use a Web Host With PHP Support

If your server has activated support for PHP you do not need to do anything.

Just create some .php files, place them in your web directory, and the server will automatically parse them for you.

You do not need to compile anything or install any extra tools.

Because PHP is free, most web hosts offer PHP support.

Installation

Set Up PHP on Your Own PC

However, if your server does not support PHP, you must:

* install a web server
* install PHP
* install a database, such as MySQL

Syntax

A PHP script can be placed anywhere in the document.

A PHP script starts with **<?php** and ends with **?>** :

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

The default file extension for PHP files is ".php".

A PHP file normally contains HTML tags, and some PHP scripting code.

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:

Syntax

Example

```
<!DOCTYPE html>
<html>
<body>

<h1>My first PHP page</h1>

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

</body>
</html>
```

Note: PHP statements are terminated by semicolon (;). The closing tag of a block of PHP code also automatically implies a semicolon (so you do not have to have a semicolon terminating the last line of a PHP block).

Syntax

Comments in PHP

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

Comments are useful for:

* To let others understand what you are doing - Comments let other programmers understand what you were doing in each step (if you work in a group)
* To remind yourself 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

Syntax

PHP supports three ways of commenting:

Example

```
<!DOCTYPE html>
<html>
<body>

<?php
// This is a single line comment

# This is also a single line comment

/*
This is a multiple lines comment block
that spans over more than
one line
*/
?>

</body>
</html>
```

Syntax

PHP Case Sensitivity

In PHP, all user-defined functions, classes, and keywords (e.g. if, else, while, echo, etc.) are case-insensitive.

In the example below, all three echo statements below are legal (and equal):

Syntax

Example

```
<!DOCTYPE html>
<html>
<body>

<?php
ECHO "Hello World!<br>";
echo "Hello World!<br>";
EcHo "Hello World!<br>";
?>

</body>
</html>
```

Syntax

However; in PHP, all variables are case-sensitive.

In the example below, only the first statement will display the value of the $color variable (this is because $color, $COLOR, and $coLOR are treated as three different variables):

Example

```
<!DOCTYPE html>
<html>
<body>

<?php
$color="red";
echo "My car is " . $color . "<br>";
echo "My house is " . $COLOR . "<br>";
echo "My boat is " . $coLOR . "<br>";
?>

</body>
</html>
```

Variables

Variables are "containers" for storing information:

Example

```
<?php
$x=5;
$y=6;
$z=$x+$y;
echo $z;
?>
```

Variables

Much Like Algebra

x=5

y=6

z=x+y

In algebra we use letters (like x) to hold values (like 5).

From the expression z=x+y above, we can calculate the value of z to be 11.

In PHP these letters are called **variables.**

Variables

PHP Variables

As with algebra, PHP variables can be used to hold values (x=5) or expressions (z=x+y).

A variable can have a short name (like x and y) or a more descriptive name (age, carname, total_volume).

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 ($y and $Y are two different variables)

Variables

Creating (Declaring) PHP Variables

PHP has no command for declaring a variable.

A variable is created the moment you first assign a value to it:

Example

```
<?php
$txt="Hello world!";
$x=5;
$y=10.5;
?>
```

After the execution of the statements above, the variable **txt** will hold the value **Hello world!**, the variable **x** will hold the value **5**, and the variable **y** will hold the value **10.5**.

Note: When you assign a text value to a variable, put quotes around the value.

Variables

PHP is a Loosely Type Language

In the example above, notice that we did not have to tell PHP which data type the variable is.

PHP automatically converts the variable to the correct data type, depending on its value.

In other languages such as C, C++, and Java, the programmer must declare the name and type of the variable before using it.

Variables

PHP Variables Scope

In PHP, variables can be declared anywhere in the script.

The scope of a variable is the part of the script where the variable can be referenced/used.

PHP has three different variable scopes:

* local
* global
* static

echo and print Statements

In PHP there is two basic ways to get output: echo and print.

In this tutorial we use echo (and print) in almost every example. So, this chapter contains a little more info about those two output statements.

echo and print Statements

PHP echo and print Statements

There are some differences between echo and print:

* echo - can output one or more strings
* print - can only output one string, and returns always 1

Tip: echo is marginally faster compared to print as echo does not return any value.

echo and print Statements

The PHP echo Statement

echo is a language construct, and can be used with or without parentheses: echo or echo().

Display Strings

The following example shows how to display different strings with the echo command (also notice that the strings 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.";
?>
```

echo and print Statements

Display Variables

The following example shows how to display strings and variables with the echo command:

Example

```
<?php
$txt1="Learn PHP";
$txt2="W3Schools.com";
$cars=array("Volvo","BMW","Toyota");

echo $txt1;
echo "<br>";
echo "Study PHP at $txt2";
echo "My car is a {$cars[0]}";
?>
```

echo and print Statements

The PHP print Statement

print is also a language construct, and can be used with or without parentheses: print or print().

Display Strings

The following example shows how to display different strings with the print command (also notice that the strings can contain HTML markup):

Example

```
<?php
print "<h2>PHP is fun!</h2>";
print "Hello world!<br>";
print "I'm about to learn PHP!";
?>
```

echo and print Statements

Display Variables

The following example shows how to display strings and variables with the print command:

Example

```
<?php
$txt1="Learn PHP";
$txt2="W3Schools.com";
$cars=array("Volvo","BMW","Toyota");

print $txt1;
print "<br>";
print "Study PHP at $txt2";
print "My car is a {$cars[0]}";
?>
```

Data Types

String, Integer, Floating point numbers, Boolean, Array, Object, NULL.

PHP Strings

A string is a sequence of characters, like "Hello world!".

A string can be any text inside quotes. You can use single or double quotes:

Example

```
<?php
$x = "Hello world!";
echo $x;
echo "<br>";
$x = 'Hello world!';
echo $x;
?>
```

Data Types

PHP Integers

An integer is a number without decimals.

Rules for integers:

* An integer must have at least one digit (0-9)
* An integer cannot contain comma or blanks
* An integer must not have a decimal point
* An integer can be either positive or negative
* Integers can be specified in three formats: decimal (10-based), hexadecimal (16-based - prefixed with 0x) or octal (8-based - prefixed with 0)

Data Types

In the following example we will test different numbers. The PHP var_dump() function returns the data type and value of variables:

Example

```
<?php
$x = 5985;
var_dump($x);
echo "<br>";
$x = -345; // negative number
var_dump($x);
echo "<br>";
$x = 0x8C; // hexadecimal number
var_dump($x);
echo "<br>";
$x = 047; // octal number
var_dump($x);
?>
```

Data Types

PHP Floating Point Numbers

A floating point number is a number with a decimal point or a number in exponential form.

In the following example we will test different numbers. The PHP var_dump() function returns the data type and value of variables:

Example

```
<?php
$x = 10.365;
var_dump($x);
echo "<br>";
$x = 2.4e3;
var_dump($x);
echo "<br>";
$x = 8E-5;
var_dump($x);
?>
```

Data Types

PHP Booleans

Booleans can be either TRUE or FALSE.

```
$x=true;
$y=false;
```

Booleans are often used in conditional testing. You will learn more about conditional testing in a later chapter of this tutorial.

PHP Arrays

An array stores multiple values in one single variable.

In the following example we create an array, and then use the PHP var_dump() function to return the data type and value of the array:

Example

```
<?php
$cars=array("Volvo","BMW","Toyota");
var_dump($cars);
?>
```

Data Types

PHP Objects

An object is a data type which stores data and information on how to process that data.

In PHP, an object must be explicitly declared.

First we must declare a class of object. For this, we use the class keyword. A class is a structure that can contain properties and methods.

Data Types

We then define the data type in the object class, and then we use the data type in instances of that class:

Example

```
<?php
class Car
{
  var $color;
  function Car($color="green")
  {
    $this->color = $color;
  }
  function what_color()
  {
    return $this->color;
  }
}
?>
```

Data Types

PHP NULL Value

The special NULL value represents that a variable has no value. NULL is the only possible value of data type NULL.

The NULL value identifies whether a variable is empty or not. Also useful to differentiate between the empty string and null values of databases.

Variables can be emptied by setting the value to NULL:

Example

```
<?php
$x="Hello world!";
$x=null;
var_dump($x);
?>
```