WEB TECHNOLOGIES 2

Variables & Regular Expression

Lec2

Mohammed Sultan

Outlines

- Variables& variables scope in php
- Data Types
- Constants in php
- Printing in php
- Operators in php
- Conditional statements
- Loops

Variables

- Variables are "containers" for storing information.
- Variable in php starts with the $ sign, followed by the name of the variable
- Ex:

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

Naming Rules

- 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)

Output Variables

- The PHP echo statement is often used to output data to the screen.

```
$txt = “PHP";
echo "I love $txt!";
```

- The following example will produce the same output as the example above:

```
$txt = “PHP";
echo "I love " . $txt . "!";
```

PHP is a Loosely Typed Language

- PHP automatically associates a data type to the variable, depending on its value. Since the data types are not set in a strict sense, you can do things like adding a string to an integer without causing an error. (strict in Functions)

- Ex:

```
<?php
$x = 5;
$y = 4;
echo $x + $y;
?>
```

Note :

In PHP 7, type declarations were added.

PHP Data Types

- Variables can store data of different types, and different data types can do different things.

- PHP supports the following data types:
  - String
  - Integer
  - Float (floating point numbers - also called double)
  - Boolean
  - Array
  - Object
  - NULL
  - Resource

String

- A string is a sequence of characters, like "Hello world!".
- A string can be any text inside quotes.
- Ex:
  - <?php
  - $x = "String example!";
  - echo $x;
  - ?>

You can assign the same value to multiple variables in one line:

$x = $y = $z = "var";

Numbers

- There are three main numeric types in PHP:
  - Integer
  - Float
  - Number Strings

```
$a = 5;
$b = 5.34;
$c = "25";
```

- In addition, PHP has two more data types used for numbers:
  - Infinity
  - NaN

Numbers

- An integer data type is a non-decimal number between -2,147,483,648 and 2,147,483,647.
- Ex:
  - <?php
  - $x = 5985;
  - var_dump($x); //
  - var_dump(is_int($x));// it is a functions to check if the type of a variable is integer
  - ?>

- A float (floating point number) is a number with a decimal point or a number in exponential form.
- Ex:
  - <?php
  - $x = 10.365;
  - var_dump($x); //
  - var_dump(is_float($x));//It is a functions to check if the type of a variable is float
  - ?>

PHP Numerical Strings

- The PHP is_numeric() function can be used to find whether a variable is numeric. The function returns true if the variable is a number or a numeric string, false otherwise.

```
$x = "5985";
var_dump(is_numeric($x));
$x = "59.85" + 100;
var_dump(is_numeric($x));
```

```
$x = "10a";
$y = $x * 5;
echo $y;     //50
```

Boolean & NULL Value

- A Boolean represents two possible states: TRUE or FALSE :
  - $x = true;
  - $y = false;

- Null is a special data type which can have only one value: NULL.
- Null has no value assigned to it.
- Variables can also be emptied by setting the value to NULL.
- Ex:
  - <?php
  - $x = null;
  - var_dump($x);
  - ?>

Array, object & Resource

- An array stores multiple values in one single variable.(will be discussed in Array lecture)

- Classes and objects are the two main aspects of object-oriented programming .(will be discussed in OOP lecture)

- The special resource type is not an actual data type. It is the storing of a reference to functions and resources external to PHP.(it is an advanced topic)

PHP Constants

- The value of constant cannot be changed during the script.
- A valid constant name starts with a letter or underscore (no $ sign before the constant name).
- define() function used to create constant
  - define(name, value, case-insensitive)

```
<?php
define("GREETING", "Welcome! ", true);
echo greeting;
?>
```

Output is:

Welcome!

Note:Constants are automatically global

PHP const Keyword

- You can also create a constant by using the const keyword.

- const vs. define()
  - Const:
    - are always case-sensitive
    - const cannot be created inside another block scope, like inside a function or inside an if statement.
  - define():
    - has a case-insensitive option.
    - can be created inside another block scope.

Note: See PHP Magic Constants like (--LINE --) for return The current line number.

PHP Casting

- Sometimes you need to cast a numerical value into another data type.
- The (int) , (integer) , and intval() functions are often used to convert a value to an integer.
- Cast float and string to integer:

```
// Cast float to int
$x = 23465.768;
$int_cast = (int)$x;
echo $int_cast;
```

```
// Cast string to int
$x = "23465.768";
$int_cast = (int)$x;
echo $int_cast;
```

```
$x = 5;
$x = (string) $x;
var_dump($x);
```

Variables scope

- Global
- Local
- Static

Printing statements

- Echo
- Print
- Printf
- Print_r
- Var_dump

Echo & Print

- There are two basic ways to get output.
- echo and print are more or less the same.
  - They are both used to output data to the screen.

- They are both used with or without parentheses
  - echo or echo()
    - echo "<h2>PHP is Fun!</h2>";
    - echo "<h2>" . $txt1 . "</h2>";
  - print or print()
    - print "<h2>PHP is Fun!</h2>";
    - print "<h2>" . $txt1 . "</h2>";

```
<?php
$txt1 = "Learn PHP";
?>
```

Echo & Print differences

- echo is marginally faster than print.
- echo has no return value while print has a return value of 1.
  - Print can be used in expressions
  - If(print(ok)) {
    Code..... }

- echo can take multiple parameters (although such usage is rare) while print can take one argument

- The short form of echo is <?=$x?>

Printf, print_r & var_dump

- Printf output a formatted string
  - printf(string $format, mixed ...$values): int
  - Printf("this is %dth day of the week", 4) ;//Outputs this is 4th day of the week
- Print_r
  - Printing array elements
  - More readable than var_dump
- var_dump()
  - function returns the data type and value.
  - Print with data type.

PHP Operators

- Operators are used to perform operations on variables and values.

- Arithmetic operators
- Assignment operators
- Comparison operators
- Increment/Decrement operators
- Logical operators
- String operators
- Array operators
- Conditional assignment operators

PHP Operators

- Arithmetic operators
  - Addition, subtraction, multiplication.....
  - $x + $y,$x - $y,$x * $y,$x ÷ $y
- Assignment Operators
  - The basic assignment operator in PHP is "=".
- Comparison Operators
  - The PHP comparison operators are used to compare two values (number or string)
  - == equal
  - === Identical
  - != Not equal

PHP Operators

- Increment / Decrement
  - increment operators are used to increment a variable's value.
    - ++$x Pre-increment, $x++ Post-increment
  - decrement operators are used to decrement a variable's value.
    - --$x Pre-decrement,$x-- Post-decrement
- Logical Operators
  - The PHP logical operators are used to combine conditional statements.
  - && And, || Or and ! Not
    - $x && $y

PHP Operators

- String Operators
  - PHP has two operators that are specially designed for strings.
    - . Concatenation
    - .= Concatenation assignment
- Conditional assignment operators are used to set a value depending on conditions
  - ?: Ternary
  - $x = expr1 ? expr2 : expr3
  - ?? Null coalescing
  - echo $user = $_GET["user"] ?? "anonymous";

PHP if...else...elseif Statements

- Conditional statements are used to perform different actions based on different conditions.
- The if statement executes some code if one condition is true.
  - if (condition) {
  - code to be executed if condition is true;
  - }

PHP if...else...elseif Statements

- If.. Else Syntax :
  - if (condition) {
  - code to be executed if condition is true;
  - } else {
  - code to be executed if condition is false;
  - }

switch

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

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

PHP Loops

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

- 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 Regular Expressions

- A regular expression is a sequence of characters that forms a search pattern.
- A regular expression can be a single character, or a more complicated pattern.
- regular expressions are strings composed of delimiters, a pattern and optional modifiers.
  - $exp = "/php/i";

Regular Expression Functions

- preg_match()
  - Returns 1 if the pattern was found in the string and 0 if not

  - <?php
  - $str = "PHP is Fun ";
  - $pattern = "/PHP/i";
  - echo preg_match($pattern, $str); // Outputs 1
  - ?>

Regular Expression Functions

- preg_match_all()
  - Returns the number of times the pattern was found in the string, which may also be 0

  - <?php
  - $str = "The rain in SPAIN falls mainly on the plains.";
  - $pattern = "/ain/i";
  - echo preg_match_all($pattern, $str); // Outputs 4
  - ?>

Regular Expression Functions

- preg_replace()
  - Returns a new string where matched patterns have been replaced with another string

```
<?php
$str = "Visit Microsoft!";
$pattern = "/microsoft/i";
echo preg_replace($pattern, "PHP", $str); // Outputs "Visit PHP!"
?>
```

Any Questions?