Introduction to PHP

Objectives

- History of PHP
- Characteristics of PHP
- Variables in PHP
- Syntax of PHP
- Data types in PHP
- Conditional statements
  - If .. Else statement
  - Switch statement

Introduction to PHP

- PHP started out as a small open source project that evolved as more and more people found out how useful it was. Rasmus Lerdorf unleashed the first version of PHP way back in 1994.
- PHP is a recursive acronym for "PHP: Hypertext Preprocessor".
- PHP is a server side scripting language that is embedded in HTML. It is used to manage dynamic content, databases, session tracking, even build entire e-commerce sites.
- It is integrated with a number of popular databases, including MySQL, PostgreSQL, Oracle, Sybase, Informix, and Microsoft SQL Server.
- PHP is pleasingly zippy in its execution, especially when compiled as an Apache module on the Unix side. The MySQL server, once started, executes even very complex queries with huge result sets in record-setting time.
- PHP Syntax is C-Like.

Common uses of PHP

- PHP performs system functions, i.e. from files on a system it can create, open, read, write, and close them.
- PHP can handle forms, i.e. gather data from files, save data to a file, through email you can send data, return data to the user.
- You add, delete, modify elements within your database through PHP.
- Access cookies variables and set cookies.
- Using PHP, you can restrict users to access some pages of your website.
- It can encrypt data.

Characteristics of PHP

- Open source, free to download and use
- Cross platform
- Compatible with any server
- Supports popular databases
- Easy to learn

"Hello World" Script in PHP

- Example

```
<html>

   <head>
      <title>Hello World</title>
   </head>

   <body>
      <?php echo "Hello, World!";?>
   </body>

</html>
```

- Output:

```
Hello, World!
```

Data types supported by PHP

- Following data types are supported by PHP:
  - Integer: It is stored as signed integers with 32 bits, with a range of -2,147,483,648 and 2,147,483,647
  - Float: It is stored as IEEE floating point number with 64 bits.
  - String: It is a sequence of 8-bit characters.
  - Boolean: It has two literal values: 'true' and 'false'
  - Array: It store an ordered map of pairs of keys and values.
  - Object: It stores an instance of class.
  - Resource: It is for storing file handling, or database connection.
  - Null: It has only value null.

Commenting PHP Code

- A *comment* is the portion of a program that exists only for the human reader and stripped out before displaying the programs result. There are two commenting formats in PHP
  - **Single-line comments** − They are generally used for short explanations or notes relevant to the local code.

```
// This is a comment too. Each style comments only
print "An example with single line comments";
```

  - **Multi-lines comments** − They are generally used to provide pseudocode algorithms and more detailed explanations when necessary. The multiline style of commenting is the same as in C.

```
<?
   /* This is a comment with multiline
      Author : Mohammad Mohtashim
      Purpose: Multiline Comments Demo
      Subject: PHP
   */

   print "An example with multi line comments";
?>
```

Syntax of PHP

- Every code in PHP starts with **<?php** and ends with **?>**
- PHP is whitespace insensitive
- PHP is case sensitive
- Statements are expressions terminated by semicolons (;)
- The PHP echo statement is often used to output data to the screen.
- PHP is a Loosely Typed Language.
  - PHP automatically converts the variable to the correct data type, depending on its value.

PHP variables

- In PHP, a variable starts with the $ sign, followed by the name of the variable.
- When you assign a text value to a variable, put quotes around the value.
- PHP variable names are case-sensitive.
- 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 variables

- 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 constants

- A constant is a name or an identifier for a simple value.
- A constant value cannot change during the execution of the script. By default, a constant is case-sensitive.
- By convention, constant identifiers are always uppercase.
- A constant name starts with a letter or underscore, followed by any number of letters, numbers, or underscores. If you have defined a constant, it can never be changed or undefined.
- To define a constant you have to use define() function and to retrieve the value of a constant, you have to simply specifying its name. Unlike with variables, you do not need to have a constant with a $.

PHP constants

- Differences between constants and variables:
  - There is no need to write a dollar sign ($) before a constant, where as in Variable one has to write a dollar sign.
  - Constants cannot be defined by simple assignment, they may only be defined using the define() function.
  - Constants may be defined and accessed anywhere without regard to variable scoping rules.
  - Once the Constants have been set, may not be redefined or undefined.

```
// Valid constant names
define("ONE",     "first thing");
define("TWO2",    "second thing");
define("THREE_3", "third thing");

// Invalid constant names
define("2TWO",    "second thing");
define("__THREE__", "third value");
```

Demo

- Teacher demo code about variable and creating PHP file.

Conditional Statements

- Very often when you write code, you want to perform different actions for different conditions. You can use conditional statements in your code to do this.
- In PHP we have the following conditional 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

If statement

- The if statement executes some code if one condition is true.

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

- Example:

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

if ($t < "20") {
    echo "Have a good day!";
}
?>
```

If … else statement

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

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

- Example:

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

if ($t < "20") {
    echo "Have a good day!";
} else {
    echo "Have a good night!";
}
?>
```

if...elseif....else Statement

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

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

- Example:

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

if ($t < "10") {
    echo "Have a good morning!";
} elseif ($t < "20") {
    echo "Have a good day!";
} else {
    echo "Have a good night!";
}
?>
```

Switch Statement

- The switch statement is used to perform different actions ba
- Th
  - that
  - es for
  - ase is
  - ase

```
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;
}
```

Switch Statement

- Example

```php
<?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!";
}
?>
```

Demo

- Teacher demo code about if..else and switch statements for students

Summary

- History of PHP
- Characteristics of PHP
- Variables in PHP
- Syntax of PHP
- Data types in PHP
- Conditional statements
  - If .. Else statement
  - Switch statement