Presentation 1:
PHP Language

PHP 6 fast & easy web development

Objectives

- Explain the difference between client-side programming and server-side programming.
- Describe How all Web-programming pieces work together.
- Explain how all software works for parsing PHP documents
- Introduction to PHP Programming

Client-side vs. Server-side programming.

- Client side programming
  - Code executed by Client Web Browser, such as Java script, Java Applet.
- Server side programming
  - PHP code executed by PHP engine installed within a Web server and the result is returned to Web server.

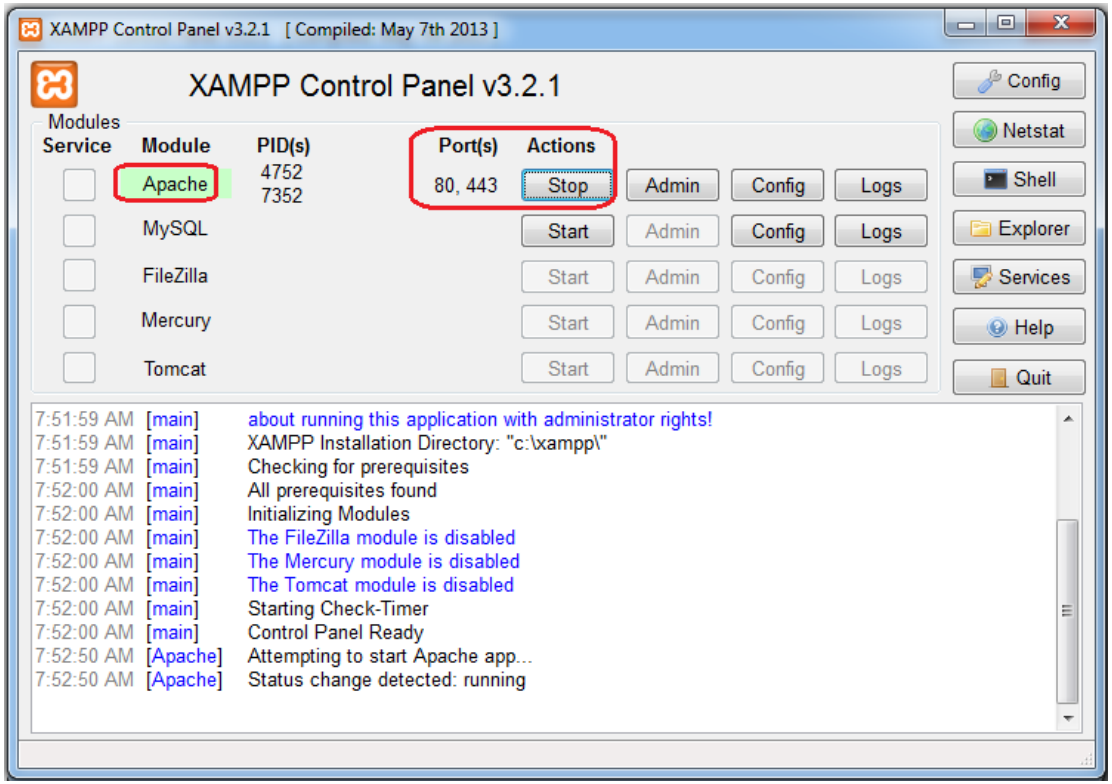
How all Web-programming pieces work

- PHP software for parsing PHP code
- IIS Web Server / or Apache Server for handling user requests
- Deployment of your PHP file within a given root folder.

Software for parsing PHP files

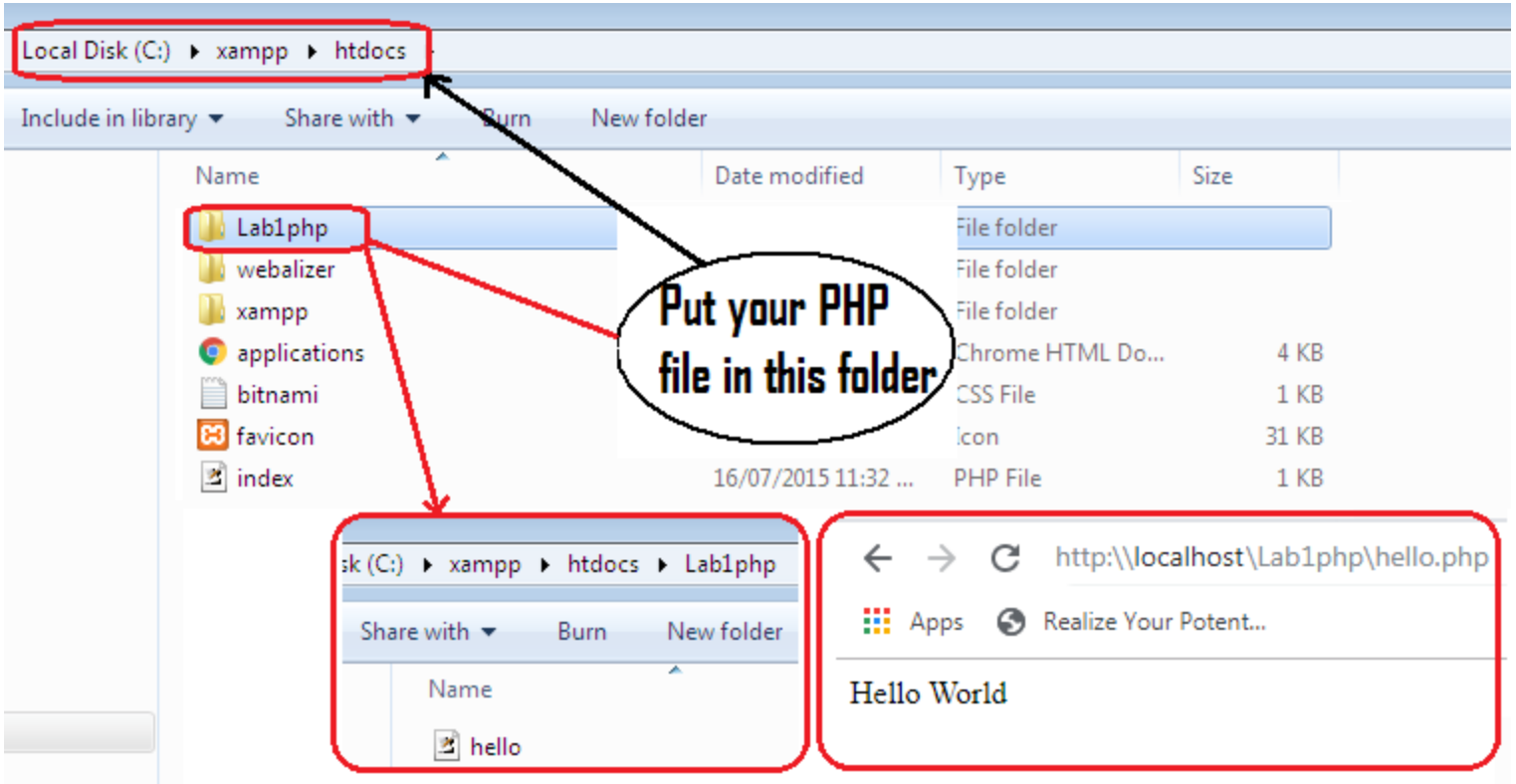
Start Apache Server for handling user request

Use XAMPP Control Panel, Start Apache Server.



Software for parsing PHP files

Deployment of your PHP file. Create any folders under the "HTDocs" folder for ex a folder named “Lab1php“. Store a php file named hello.php in it.



Introduction to PHP Programming

- How PHP is Parsed:
  - The Web browser requests a document with .php extension
  - The Web server look for a *PHP engine* to parse the PHP document, to see if there is no error
  - The output sent back to Web server
  - The web server sent back the final document to Web browser for final display.

PHP Start and End Tags

- PHP Start and End Tags:

  1)<?php ?>

  2)<? ?>

  3)<script language=" php"> </script>

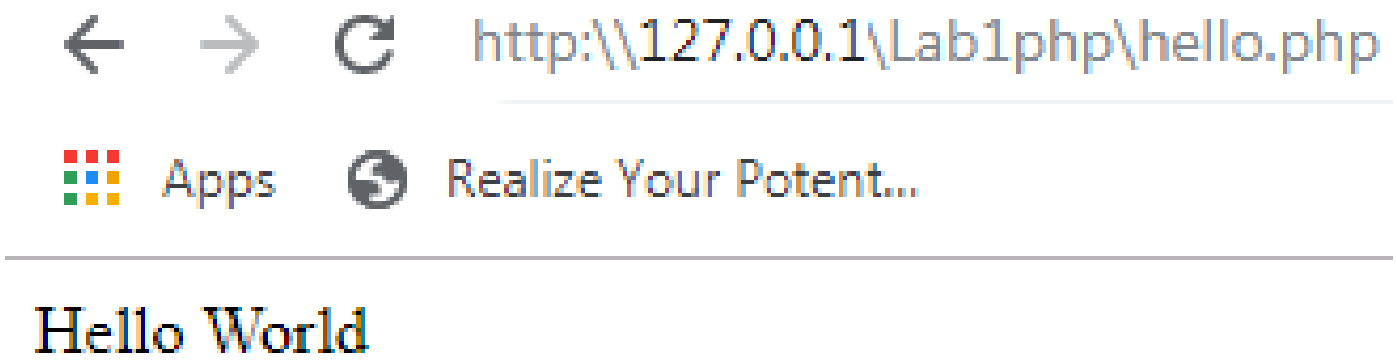
- With Notepad++, type :

```
<?php
echo "<p>Hello World</p>";
?>
```

PHP Execution test

- With Notepad++, type :

```
<?php
echo "<p>Hello World</p>";
?>
```



Code Cohabitation with HTML

- Mixing PHP code with HTML text
  - Type :

```
<html>
 <head>
 <title>PHP Test</title>
 </head>
 <body>
 <?php echo "<p>Hello World</p>"; ?>
 </body>
</html>
```

PHP Code

- The importance of the Instruction Terminator

- Escaping your code: use (\) backslash

```
<?php
    echo "<p> I think that PHP programming is
    \"cool\"</p>";
?>
```

Commenting your Code

- Commenting your Code
  - <!– This is a HTML comment, --> Table

```
<?
// This is a simple PHP comment
/* this a C-style comments
 long
*/
# Use this kind of comment
?>
```

Declaring Variables

- Variable are always prefaced by

```
$intVar = "954678";
$floatVar = "1453,54";
$stringVar = "This is a string.";
$bigintVar = "9546789999999999";
```

- To output it on the screen:

```
echo "<p>integer: $intVar</p>";
```

Operators

Assignment operator :

- $a +=3;
- $a -=3;
- $a .= Vanier;

Arithmetic operators :

- $c = $a + $b; $c = $a - $b;
- $c = $a * $b; $c = $a / $b;
- $c = $a % $b;

Operators (continue)

Comparison operator :

- == Equal to
- != Not equal to
- > Greater than
- < Less than
- >= Greater than or equal to
- <= Less than or equal to

Logical operator : ||

Getting variable from form

- Submit a form :

```
<FORM METHOD="post" ACTION="calculate.php">

<p> Value 1: <INPUT TYPE="text" NAME="val1" SIZE=10></p>
<p> Value 2: <INPUT TYPE="text" NAME="val2" SIZE=10></p>
<INPUT TYPE="radio" NAME="calc" VALUE="add"> add<br>
```

- You will have three variables referenced as:

$_POST["val1"]

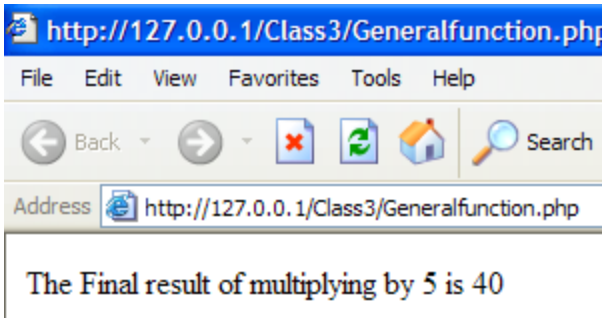
$_POST["val2"]

$_POST["calc"]

PHP functions

Functions have a specific structure

```
<?php

 function multiplier ($num)
  {
    $answer = $num * 5;
    return $answer;
  }
 echo "The Final result of multiplying by 5 is " .multiplier(8);
?>
```



PHP functions (continue)

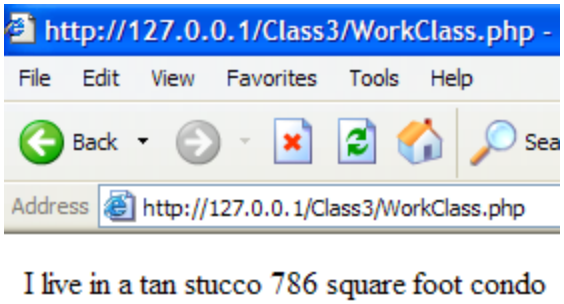
- A *return* statement can be anywhere in a function, but when used, it ends the execution of the function.

- You can use the *return* statement to get multiple values, but only if they are part of an array or an object.

Working with Objects

– Use the *var* keyword to declare your data properties.

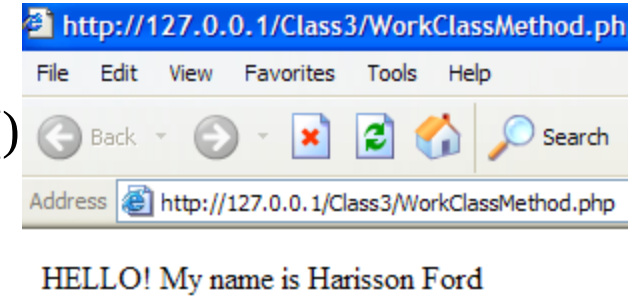
```
<?php
 class House
 {
      public $type ;
      public $sqfootage ;
      public $color ;
 }
 $house = new myHouse();
$myhouse -> type = "condo“; // continue coding to display
 echo "I live in a ".$house -> color." ".$house -> sqfootage."
    square foot ".$house -> type;
 ?>
```



Working with Objects (continue)

– Use functions as methods of your own classes.

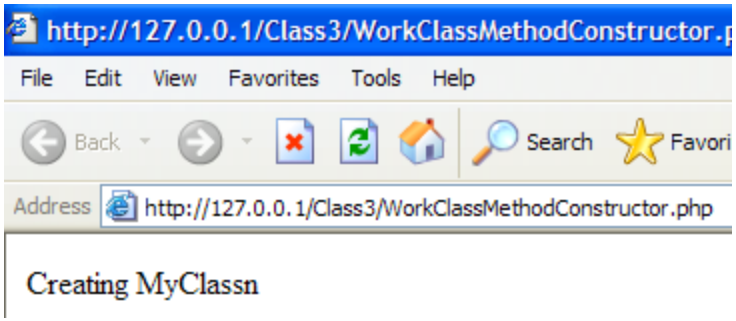
```
<?php
 class Person
 {
      public $fname ;
      public $lname ;

  function sayHello()
  {
   echo "HELLO! My name is ".$this->fname." ".$this->lname;
  }
}
$ myPreferedActor = new Person()
$ myPreferedActor -> sayHello();
?>
```



Constructors functions (continue)

- Constructor is a function, which is automatically called when a new instance of the class is created using *new classname*.

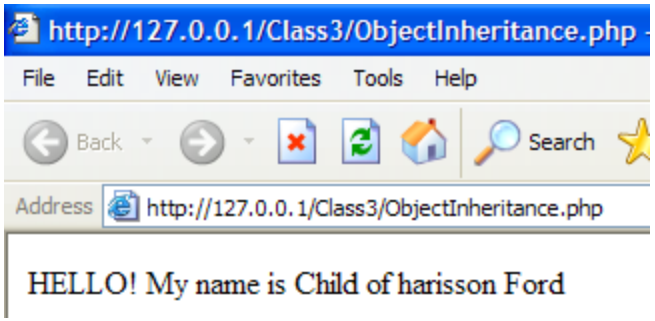
```
<?php
 class MyClass
 {
   function __construct()
   {
     print "Creating MyClassn";
   }
 }
 $my_obj = new MyClass();
 ?>
```



Object Inheritance

- A class inherits the functionality from its parent class.

```
<?php
 class Person
 {
        public $fname;
        public $lname;
  function Person($n)
  {
    $this->fname = $n ;
  }
  function sayHello()
  {
    echo "HELLO! My name is ".$this->fname." ".$this->lname;
  }
 }
 class Employee extends Person
 {
   // any code here
 }
 $my_obj = new employee ("Child of Person ");
 $my_obj -> sayHello();
 ?>
```



Summary

- Difference between client-side programming and server-side programming.
- Web-programming pieces all together.
- The software for running PHP documents
- Starting with PHP Programming
- Learn how to use and call PHP functions
- Learn how to write Classes in PHP
- Learn how to use Object Inheritance in PHP