

Class Note 2
PHP Oracle Development
(Updated 6/2/2015)

The “class note” is the typical material I would prepare for my face-to-face class. I am sharing the notes with everyone assuming you are in the class.

After you installed and set up your Zend server, you are ready to learn the basics of PHP programming. Of course, reading Chapter 3 is very essential. My suggestion is that you “write” and test each code in the textbook. Before you know it, you will become a skilled programmer. If you simply copy the code, you may get by for this class. But, you may not gain any permanent skill. The choice is all yours.

In order for you to test and especially to debug your code, you will need to turn on the error handling feature in your Zend Server. Here is what you can do:

1. Login Zend Server: <http://localhost/ZendServer/#1273675755405>

Please note that your Zend Server login may be different from above depending upon your version.

2. Under “Server Setup,” select “Directives” and expand “Error Handling and Logging.”
3. Select “On” under “display-errors.”
4. Under “error_reporting,” input E_ALL.

The above will enable the Zend to display error messages. To test it, you may use any php program with intentionally created error. For example, let us use TestEcho.php on page 51 in your textbook. The code looks as follows:

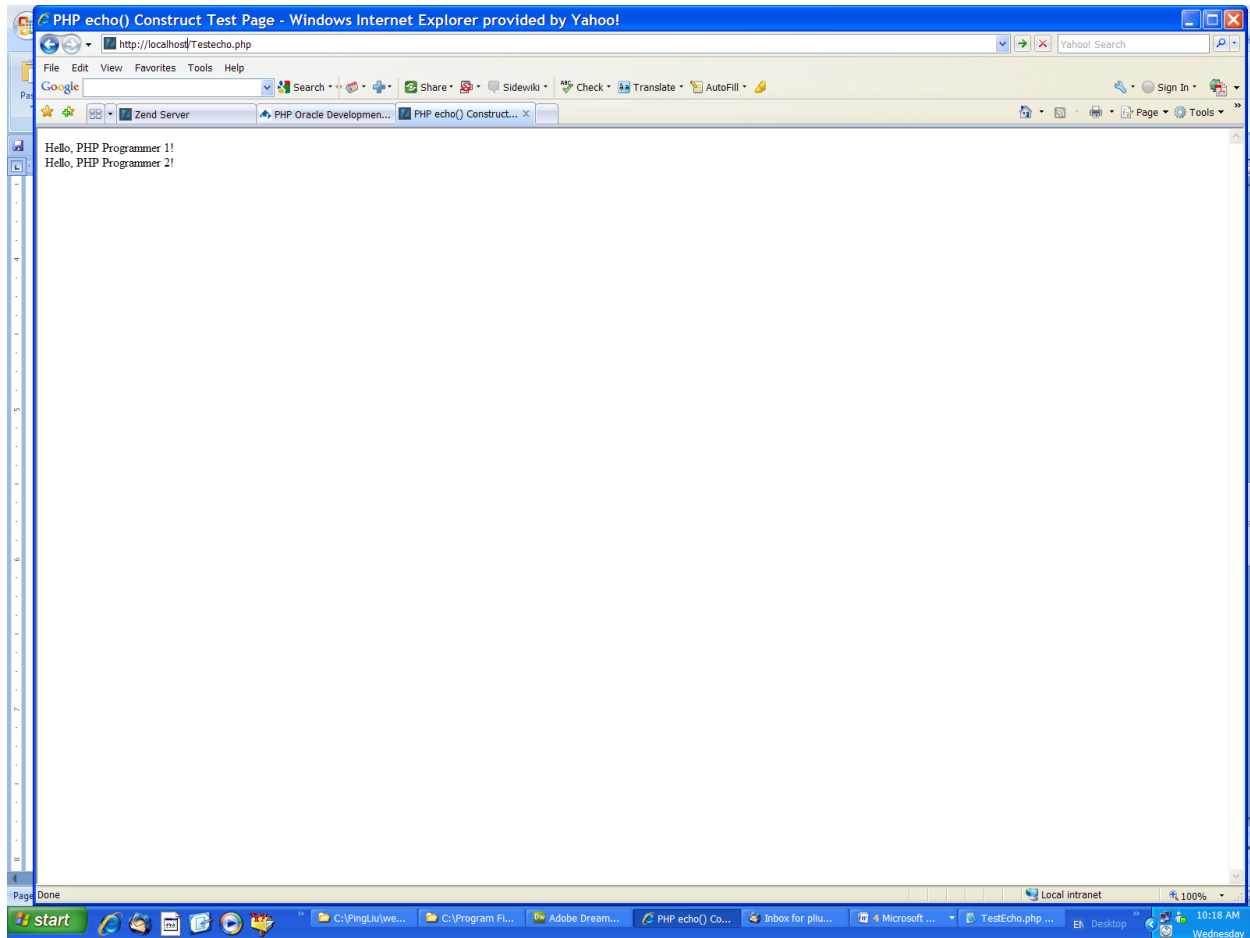
```
<html>
<head>
<title>PHP echo() Construct Test Page</title>
</head>
<body>
<?php
echo("Hello, PHP Programmer 1!<br>");
echo("Hello, PHP Programmer 2!<br>");

?>
</body>
</html>
```

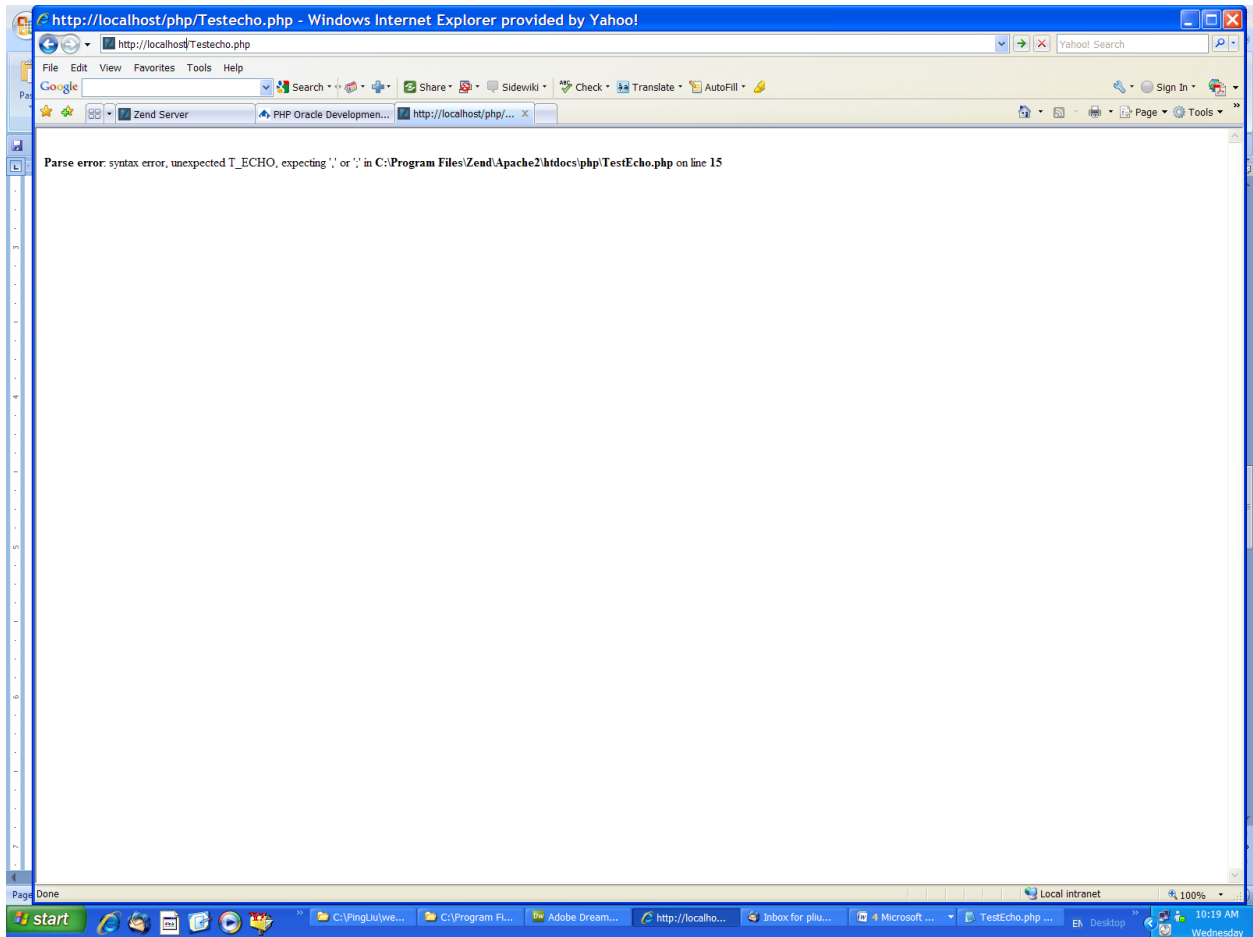
Save the above code in your Document Root directory. If you do not recall from the week activities, my is at: C:\Program Files\Zend\Apache2\htdocs.

Then, you can go to your browser, type <http://localhost/TestEcho.php>.

If there is no typo in your program, you will see the output.



Now, if you remove the semicolon “;” at the end of the first echo statement “echo("Hello, PHP Programmer 1!
");” and save your file. You will see an error, similar to the following:



Please note the above error handling feature is convenient when you try to develop and debug your program. However, in actual production environment, you need to disable this feature since the error-displaying function is a security hazard that may enable a hacker to get in your server. For the time being, since you are working at your own computer, it is not a problem.

Now, we are ready to learn some basic PHP concepts:

1. Delimit PHP program units/segments:

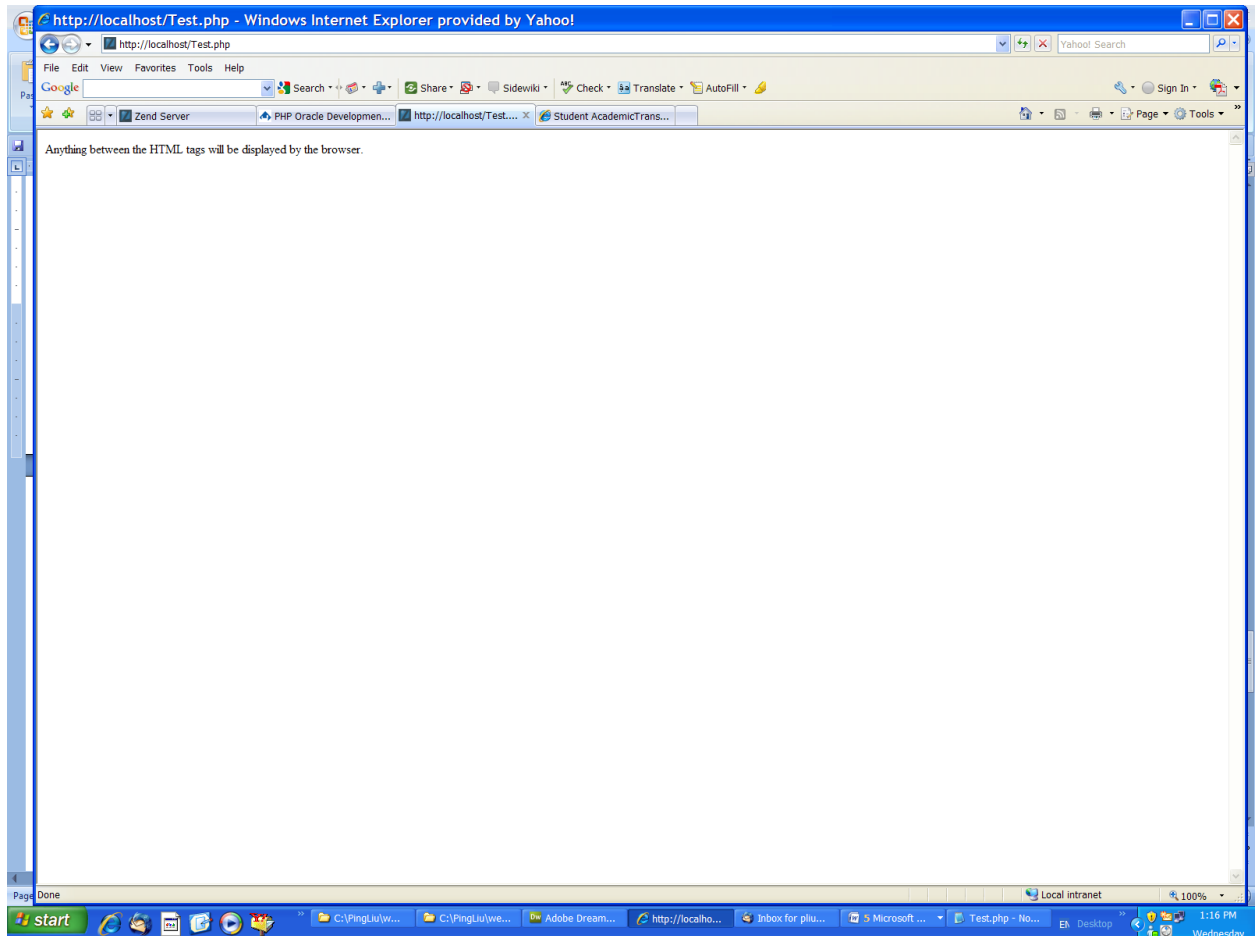
We can include PHP code in HTML file or HTML in PHP program. This is a very convenient feature in actual programming. The only thing we have to do is that we need to tell your browser what you want to do. To let your browser understand what you want, we usually use tags such as "<" and ">."

For example, if we use the following tags:

```
<html> Anything between the HTML tags will be displayed by the browser.</html>
```

Where `<html>` is called HTML tag and `</html>` ends a HTML tag. The browser will display anything between the tags.

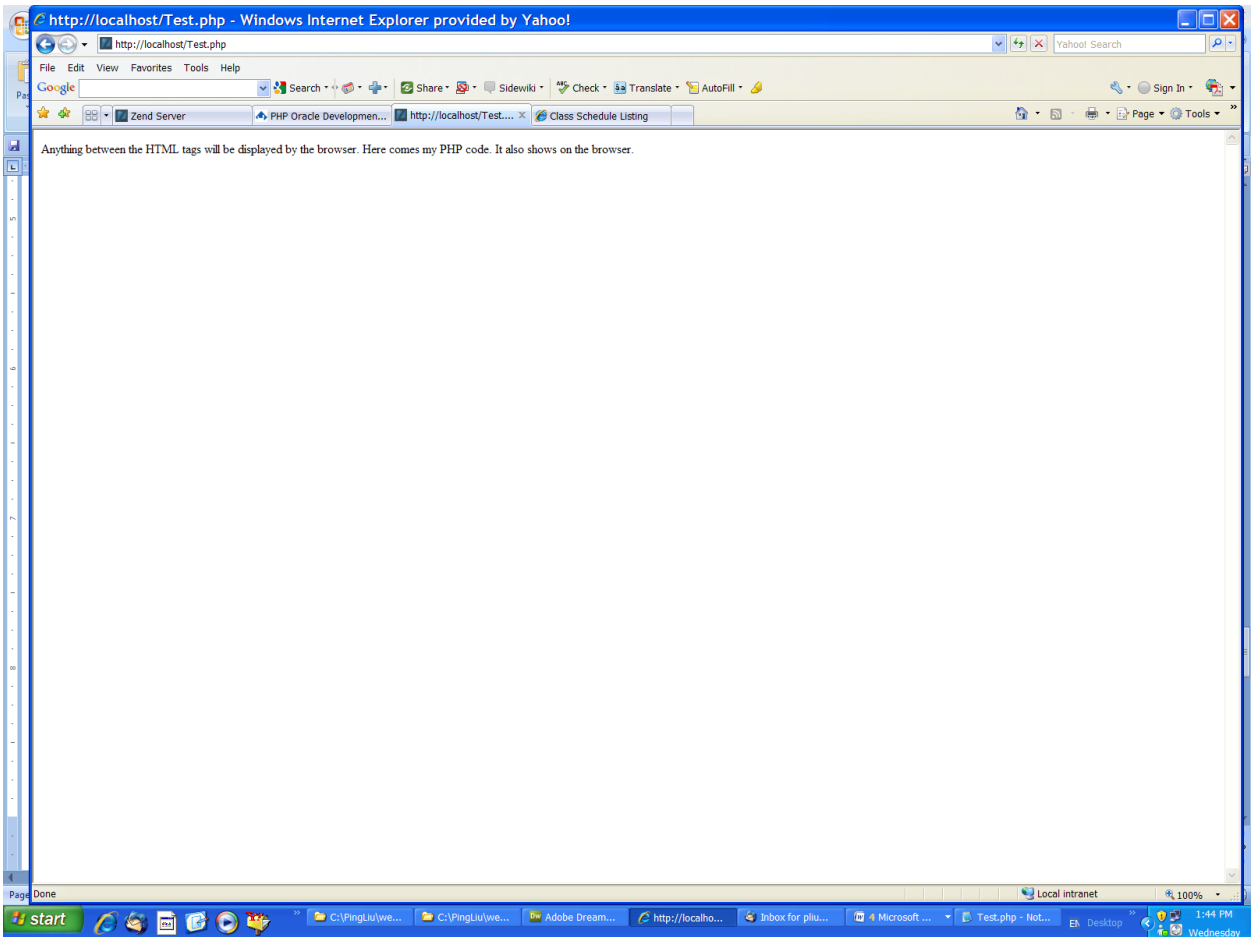
If you do not believe me, please put the above statement in file called `test.php`. Place the file in your Document Root directory. You can then test it using your browser.



Now, let us add some PHP code within the HTML tags.

```
<html> Anything between the HTML tags will be displayed by the browser.  
  
<?php  
echo "Here comes my PHP code. It also shows on the browser";  
?>  
</html>
```

Warning: I have encountered problems with the PHP if I copied from Microsoft Word and pasted it in Notepad. The Word has something extra we normally do not see. But, those stuff will cause problems in your PHP code.



The point is that you have to tell the browser you are using PHP code even if you name your file as Test.php. To define your PHP code, you have three possible ways as illustrated in your textbook. Now it is time to read your textbook.

- a. `<?php`
`?>`
- b. `<script language=php>`
`</script>`
- c. `<?`
`?>`

All the above three ways work. But, I personally like the first one the best. That is the one I use on daily basis.

2. Displaying Your Output.

You may use:

- a. echo
- b. print
- c. printf

You just have to test the codes on pages 51-55.

- 3. Putting Comments in PHP: see pages 56-57 in your textbook.

Now, let us discuss the basic building blocks (like bricks for a building) in PHP. Those concepts are almost the same with other computer languages except some specifics (or details). If you know anything about any computer language, this chapter will be pretty easy. However, those concepts are really important.

- 4. Variables: Please refer to your textbook, pages 60-63.
- 5. Operators: I am going to ask everyone to read pages 63-73.
- 6. Data types: Refer to pages 73-79 of your textbook.

Those above pages seem to be a lot. If this is your first computer language course, you may not be able to remember everything. That is fine. You may refer to this chapter back and forth when you do the actual program. After you complete the project for this week, you will feel more ease at the content.