Lecture #9

NEWM N510: Web-Database Concepts

PHP (1)

kharrazi@iupui.edu
http://www.info510.com

Review Last Lecture

- XML
- XSL
- XSLT
- XSL-FO
- XPath
- XQuery
- XLink/XPointer
- DTD
- Schema (XSD)
- XML DOM
- XForms
- SOAP
- WSDL
- RDF
- RSS
- WAP

PHP in a Nutshell

1. PHP Intro
2. PHP Syntax
3. PHP *echo*
4. PHP Comment
5. PHP Variables
6. PHP String/Array Manipulation
7. PHP Conditions
8. PHP Loops
9. PHP Functions
10. PHP Cookies/Sessions
11. PHP SSI
12. PHP Forms
13. PHP/MySQL Integration

Lecture in a Nutshell

1. PHP Intro
2. PHP Syntax
3. PHP *echo*
4. PHP Commenting
5. PHP Variables

1. PHP Introduction

- A PHP file may contain text, HTML tags and scripts. Scripts in a PHP file are executed on the server.

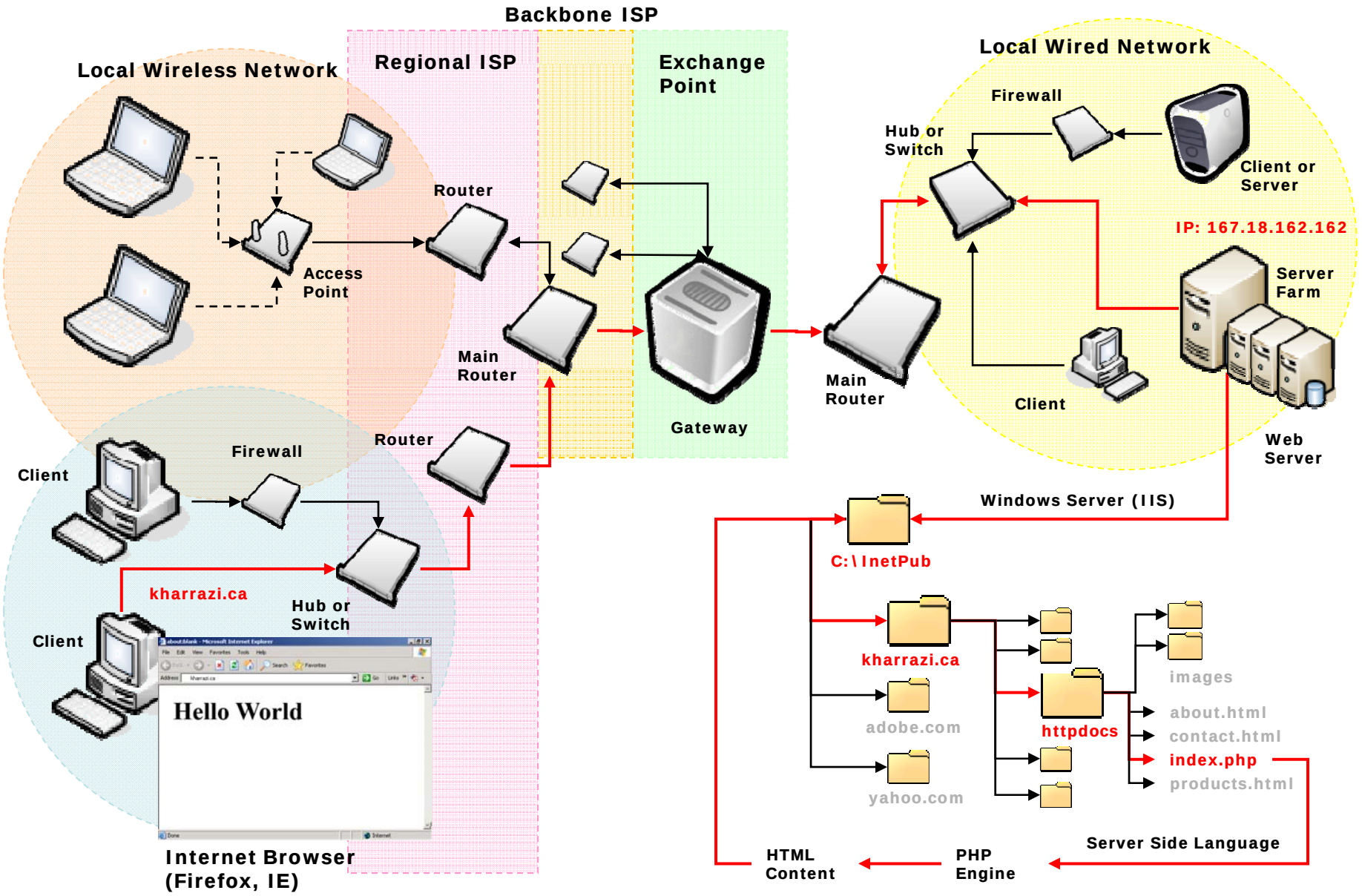
- **What is PHP?**

  - PHP stands for **PHP: Hypertext Preprocessor**
  - PHP is a *server-side scripting language*, like Perl or ASP
  - PHP scripts are *executed on the server*
  - PHP supports many databases (***MySQL***, Informix, Oracle, Sybase, Solid, PostgreSQL, Generic ODBC, etc.)
  - PHP is an open source software (OSS)
  - PHP is *free* to download and use

PHP Introduction (cont.)

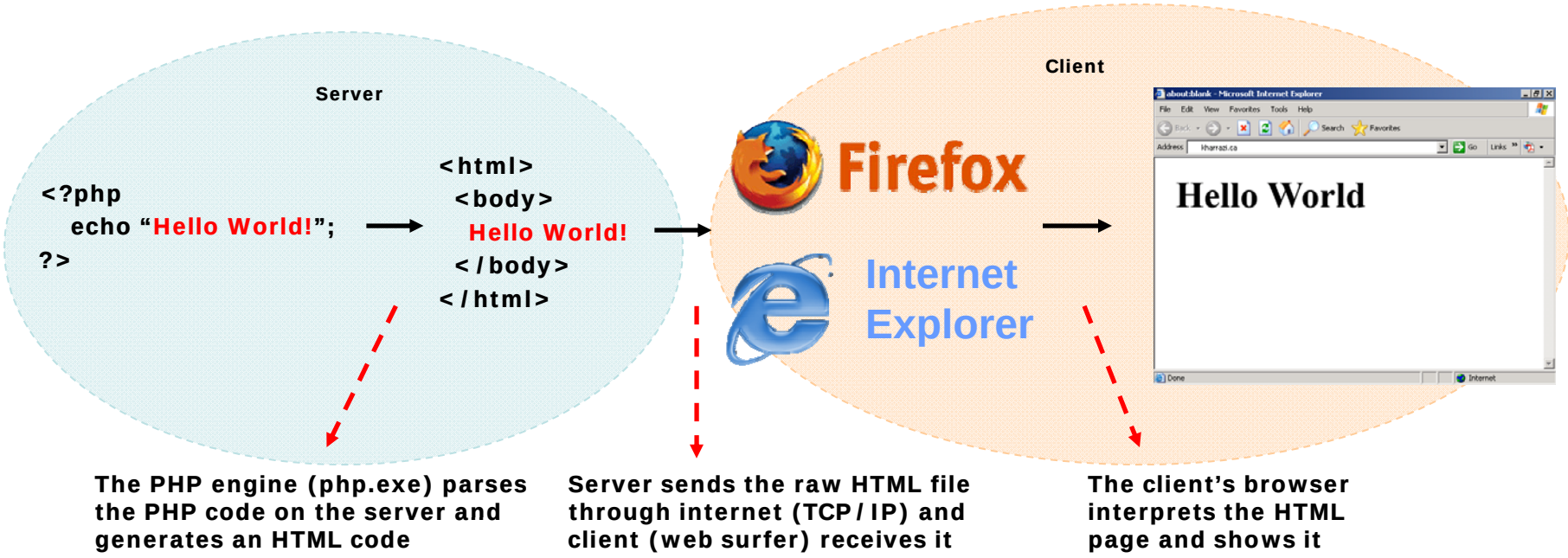
- **What is a PHP File?**
  - PHP files may contain text, HTML tags and scripts
  - PHP files are *returned to the browser as plain HTML*
  - PHP files have a file extension of ".php", ".php3", or ".phtml"

- **Why PHP?**
  - PHP runs on different platforms (Windows, Linux, Unix, etc.)
  - PHP is compatible with almost all servers used today (Apache, IIS, etc.)
  - PHP is *free* to download
  - PHP is easy to learn and runs efficiently on the server side

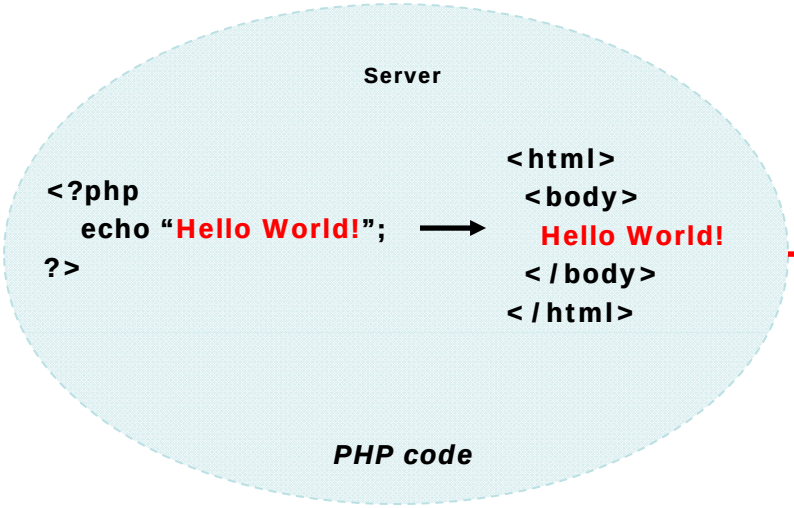

Backbone ISP
Regional ISP
Exchange Point
Local Wireless Network
Local Wired Network
Router
Access Point
Main Router
Gateway
Firewall
Hub or Switch
Client or Server
IP: 167.18.162.162
Server Farm
Main Router
Client
Web Server
Router
Firewall
Client
kharrazi.ca
Hub or Switch
Client
Hello World
Internet Browser (Firefox, IE)
Windows Server (IIS)
C:\InetPub
kharrazi.ca
adobe.com
yahoo.com
httpdocs
images
about.html
contact.html
index.php
products.html
Server Side Language
HTML Content
PHP Engine

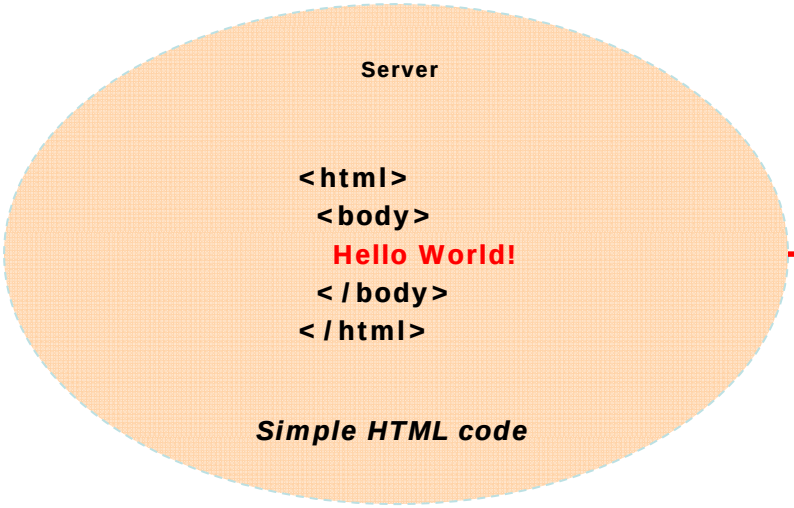
PHP Introduction (cont.)

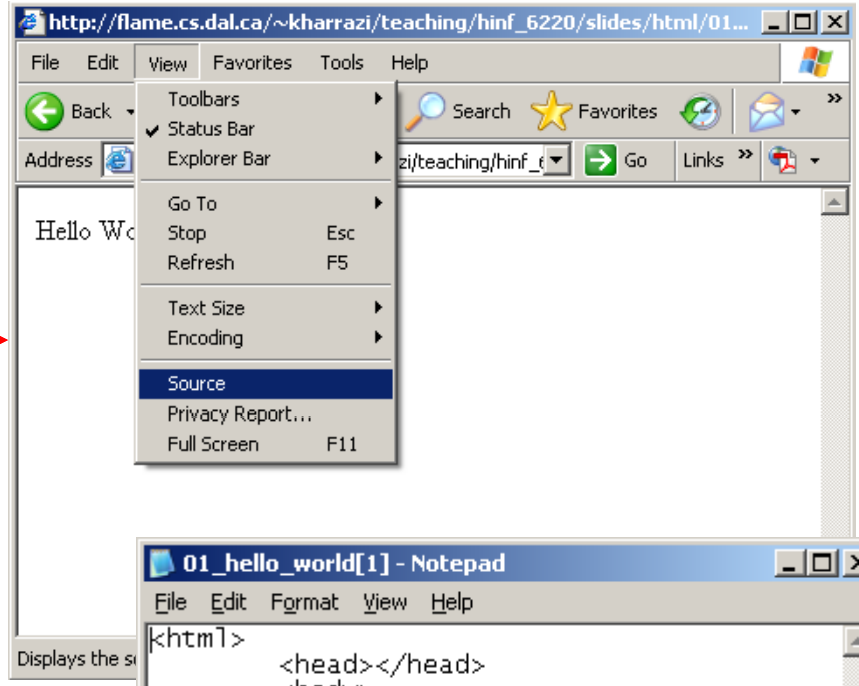
- You cannot view the PHP source code by selecting "View source" in the browser - you will only see the output from the PHP file, which is plain HTML. This is because the scripts are executed on the server before the result is sent back to the browser.

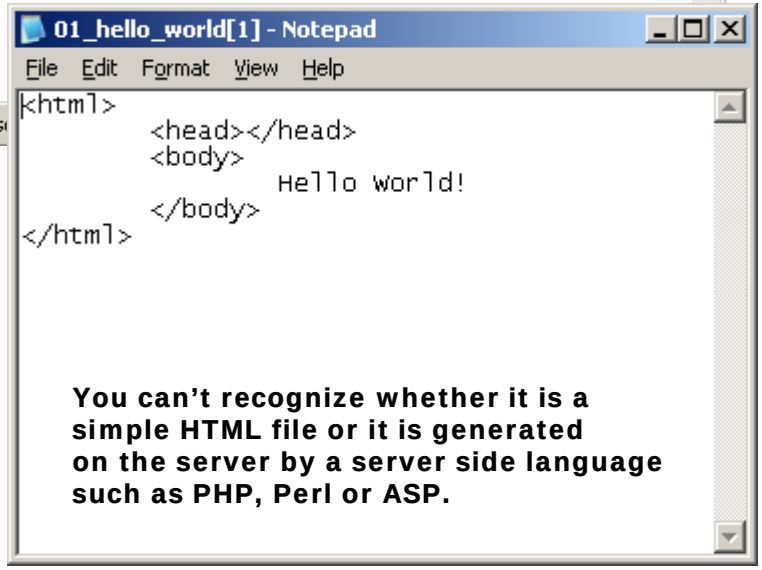


PHP Introduction (cont.)









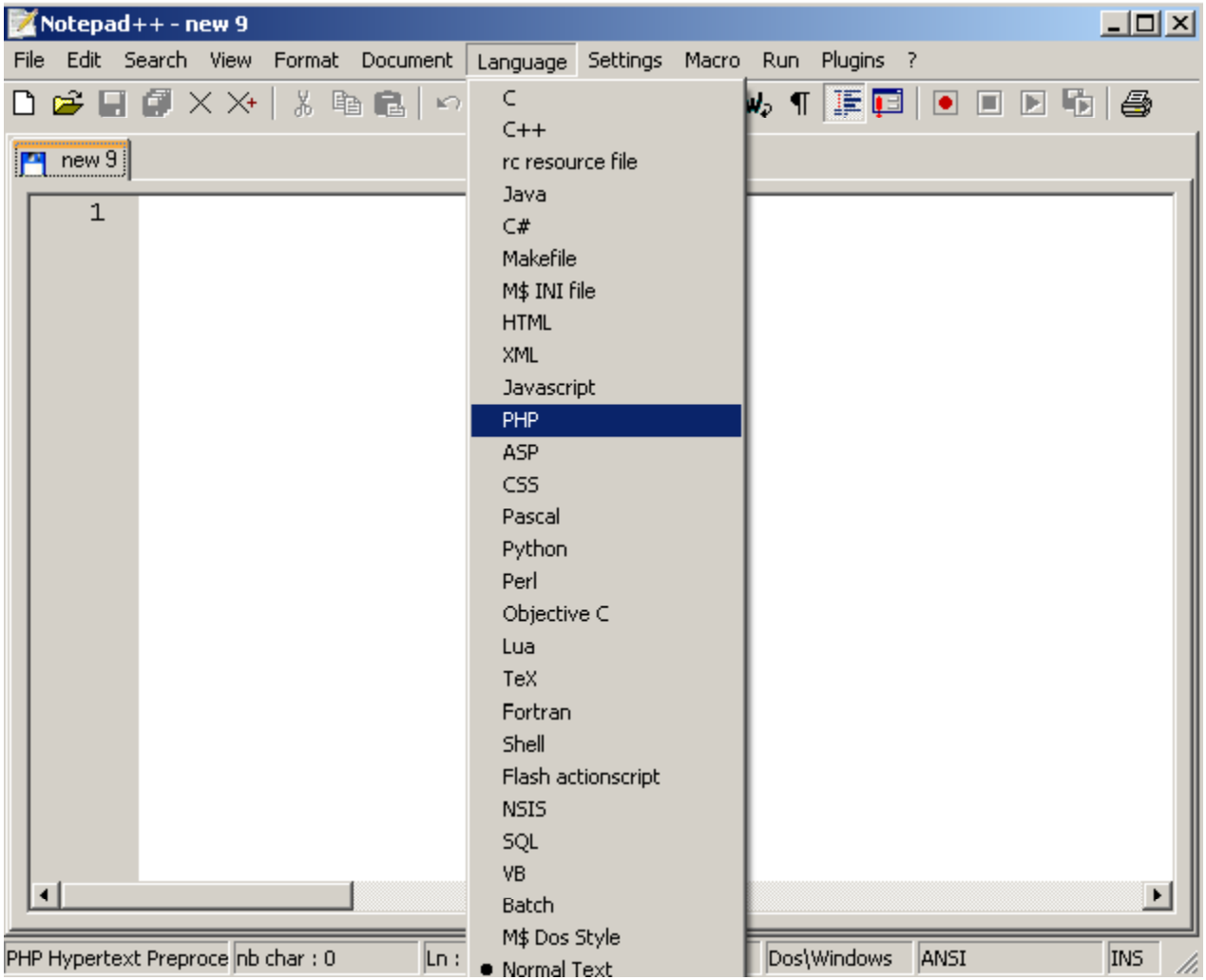
2. PHP Syntax

- A PHP file normally contains HTML tags, just like an HTML file, and some PHP scripting code. A PHP scripting block always starts with **<?php** and ends with **?>**. A PHP scripting block can be placed anywhere in the document.

- Each code line in PHP must end with a semicolon. The *semicolon* is a separator and is used to distinguish one set of instructions from another.

```
<html>
        <body>
                <?php ..commands..; ?>
        </body>
</html>
```

PHP Syntax (cont.)



PHP Syntax (cont.)

PHP Indicator

```php
<?php
    define("_BBCLONE_DIR", "../counter/");
    define("COUNTER", _BBCLONE_DIR."mark_page.php");
    if (is_readable(COUNTER)) include_once(COUNTER);
?>
```

```php
<?php
    $path = "../../tpl/";
    $domain_path = "../../include/";
?>
```

Semicolon

```php
<?php

    include "menu.php";
    include $path."header.php";
?>
```

Embedded HTML

```
<!-- Begining of the index - content -->
<table width="100%" height="100%" border="0" cellpadding="0" cellspacing="0">
<tr>
```

```php
    <!-- Left Column - START -->
    <?php
        include $path."column_left.php";
    ?>
    <!-- Left Column - END -->
```

3. PHP *echo*

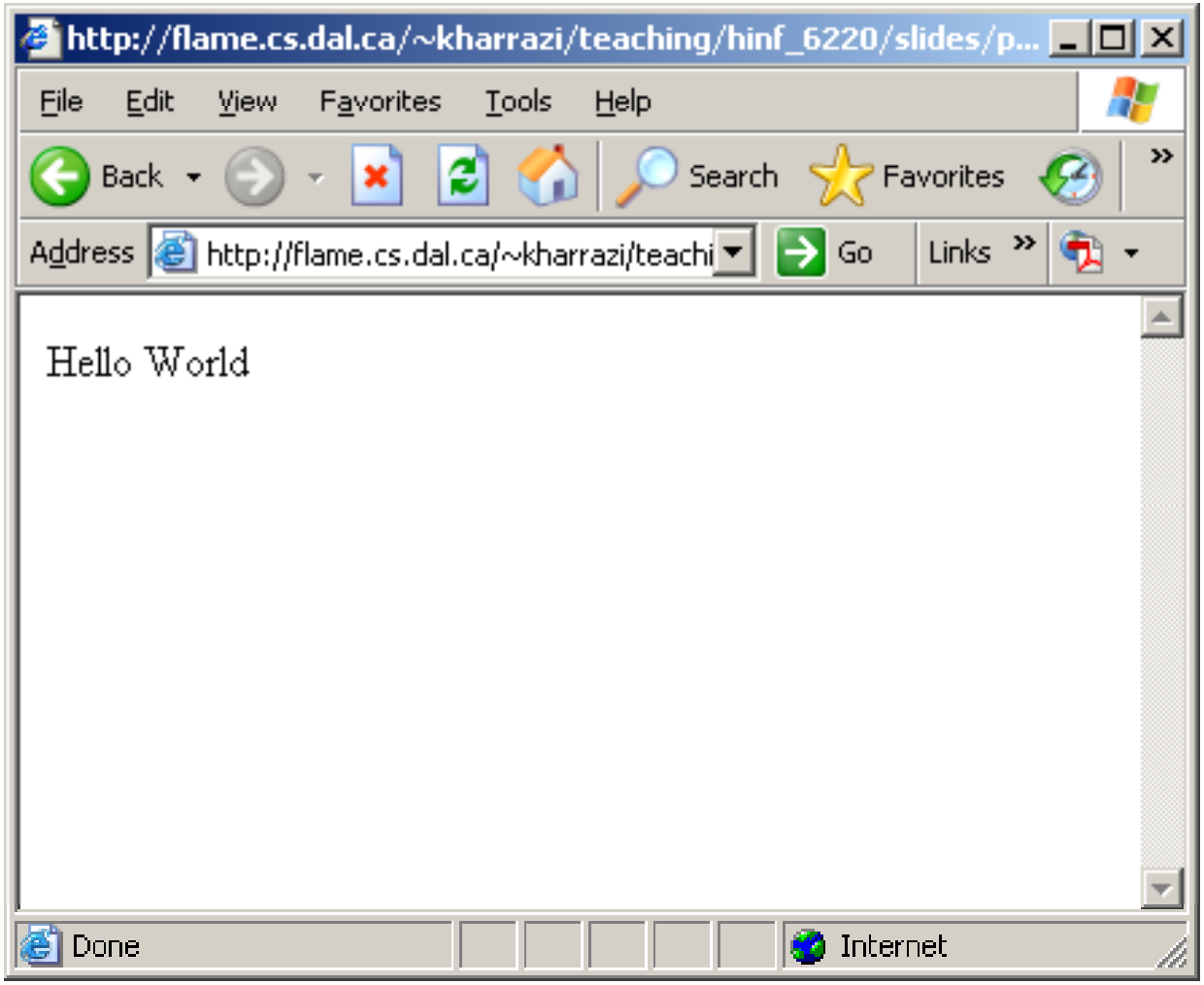
echo

- There are two basic statements to output text with PHP: *echo* and *print*. echo() is not actually a function (it is a language construct), so you are ***not*** required to use parentheses with it.

```
<html>
        <body>
                <?php
                        echo "Hello World";
                ?>
        </body>
</html>
```

PHP echo (cont.)

echo



PHP echo (cont.)

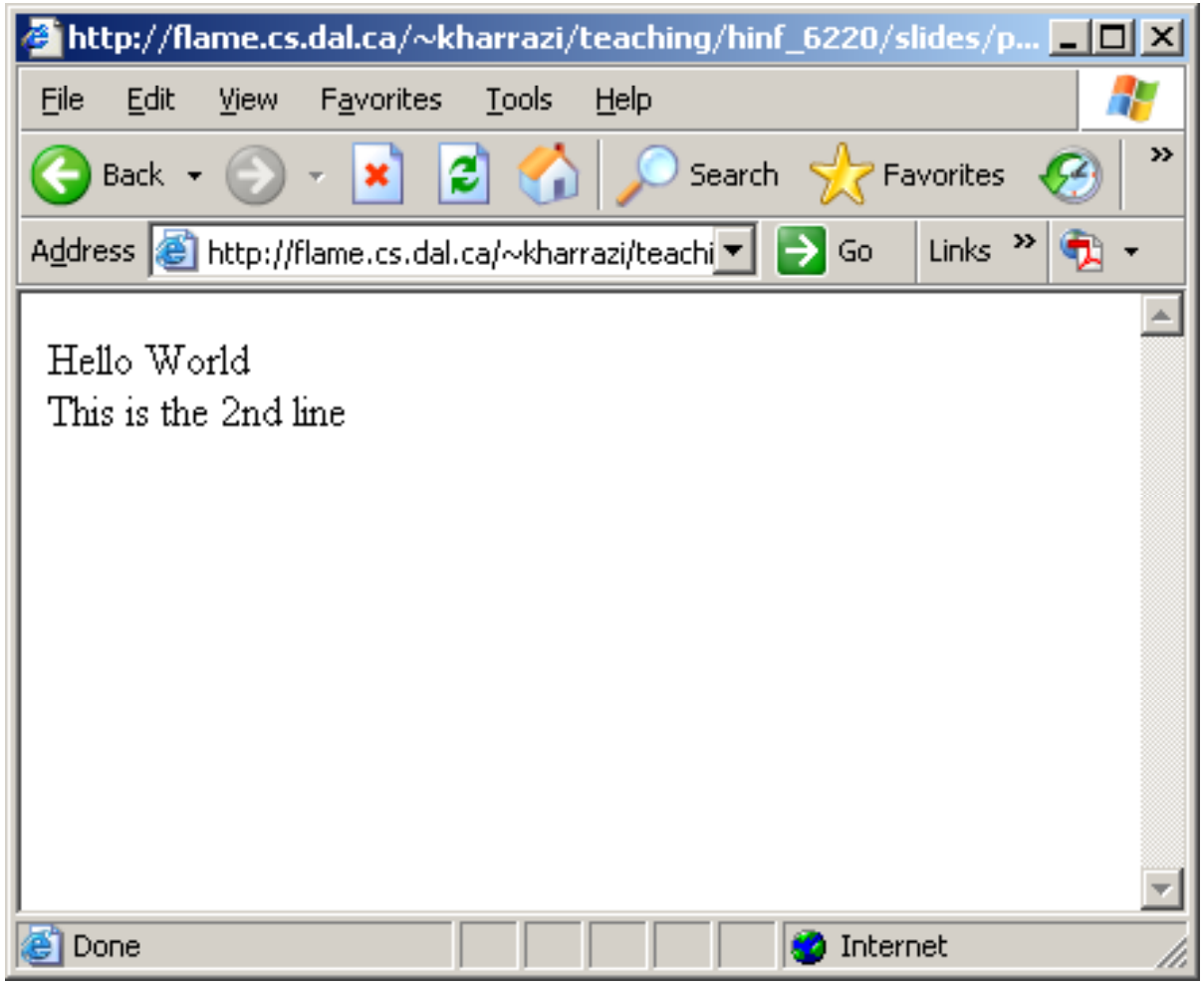
echo

- HTML tags can be constructed in PHP echo statement. In the following case the
 tag is made by the PHP script and will be interpreted by the browser to make a new line in HTML.

```
<html>
        <body>
                <?php
                        echo "Hello World";
                        echo "<br>";
                        echo "This is the 2nd line";
                ?>
        </body>
</html>
```

PHP echo (cont.)

echo



PHP echo (cont.)

echo

- You can create all of the HTML tags by the PHP script but it will be hard to read the code efficiently. This is exactly why Perl's syntax (another server side language) is hard to script and read!

```
<?php
        echo "<html>";
        echo "<body>";
        echo "Hello World";
        echo "<br>";
        echo "This is the 2nd line"
        echo "</body>";
        echo "</html>";
?>
```

PHP echo (cont.)

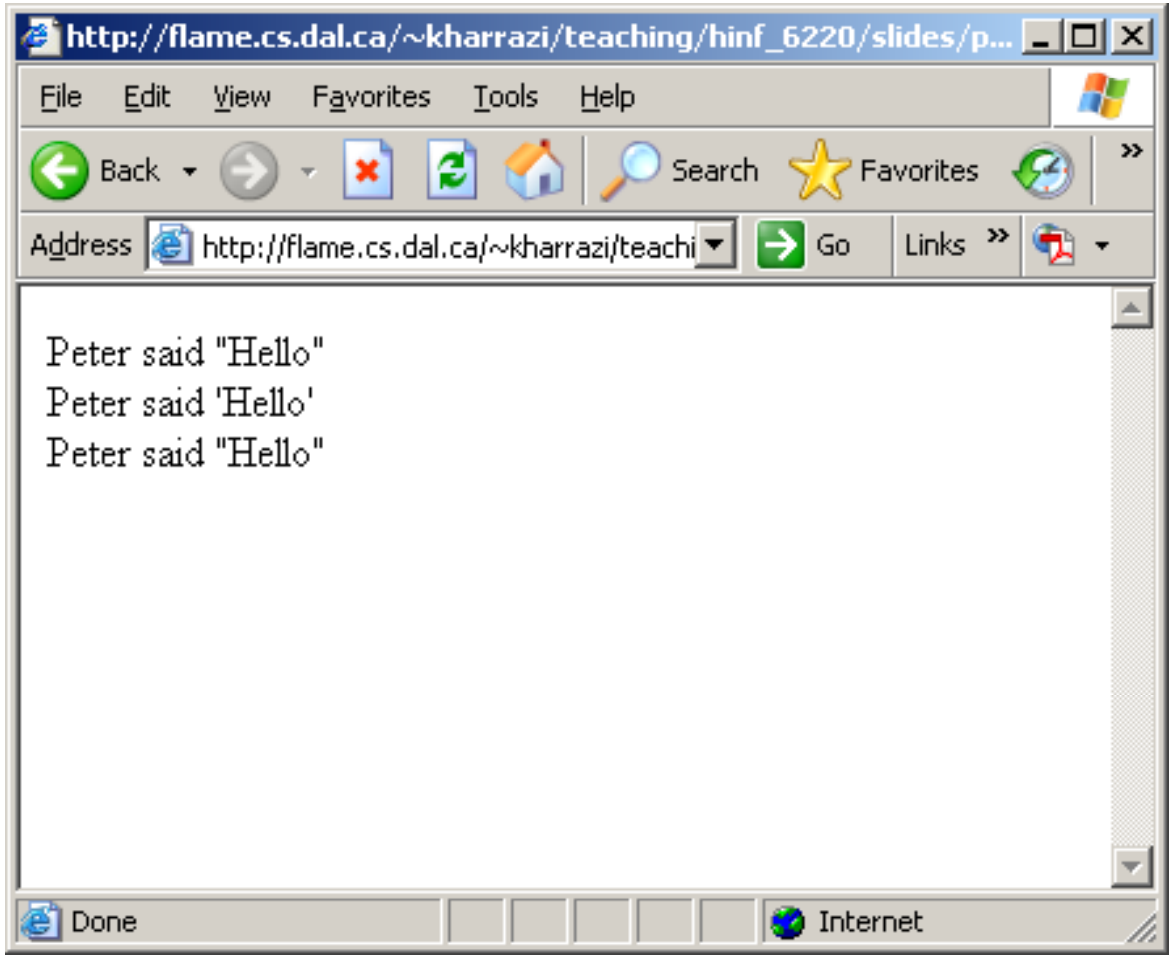
echo

- In order to echo a double quote (") or single quote (') we can either use them following by a back slash (\) or changing the main quote sign from double to single or vice versa.

```
<?php
        echo "Peter said \"Hello\"";
        echo "<br>";
        echo "Peter said 'Hello'";
        echo "<br>";
        echo 'Peter said "Hello"';
?>
```

PHP echo (cont.)

echo



PHP echo (cont.)

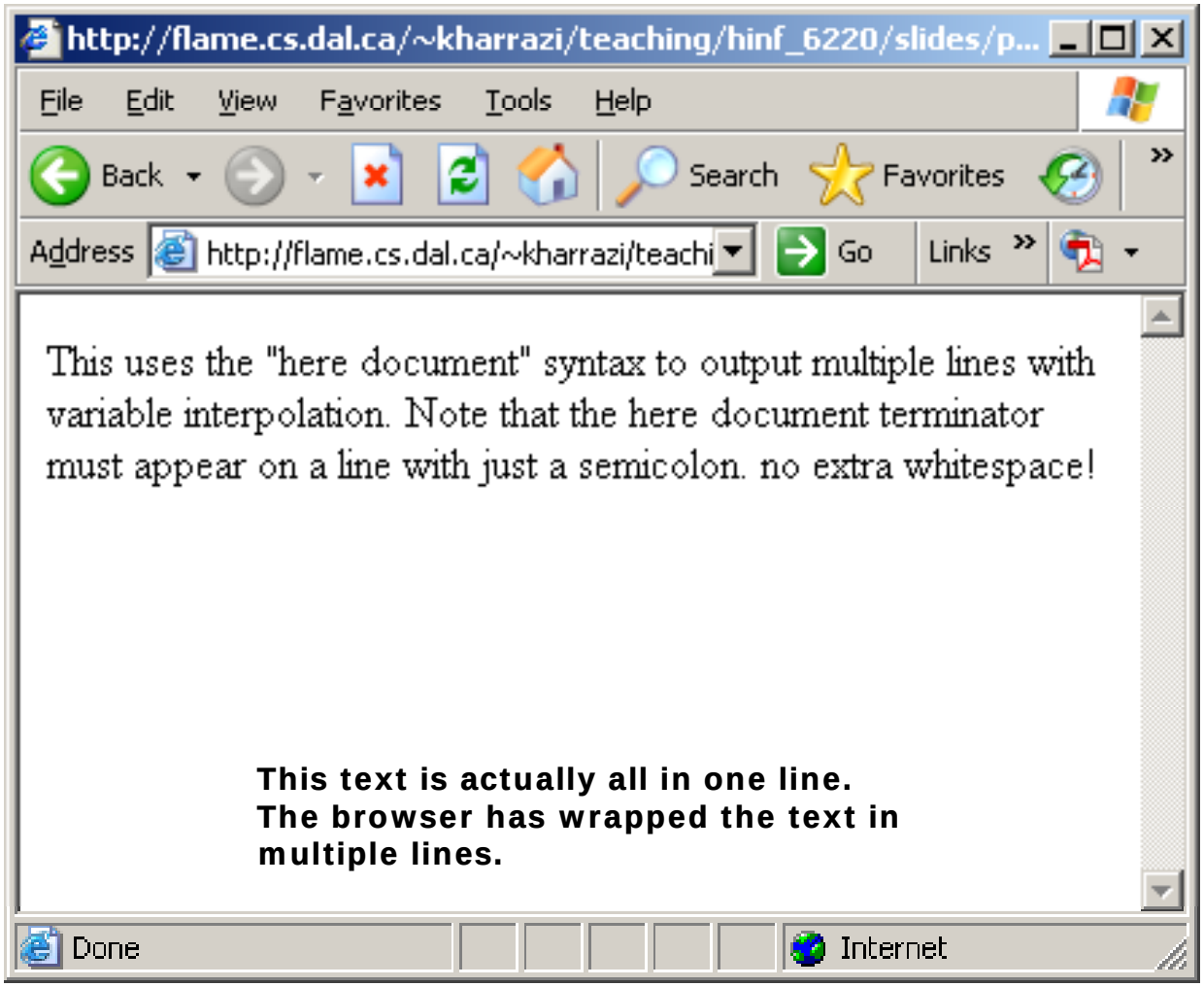
echo

- In order to echo a block of text the following syntax can be used. An indicator (here END) should be used to show the start and end of the text block. Note that the END indicator should be at the beginning of the line in the code. Quotes can be used in this syntax.

```
<?php
echo <<<END
This uses the "here document" syntax to output
multiple lines with variable interpolation. Note
that the here document terminator must appear on a
line with just a semicolon. no extra whitespace!
END;
?>
```

PHP echo (cont.)

echo



PHP echo (cont.)

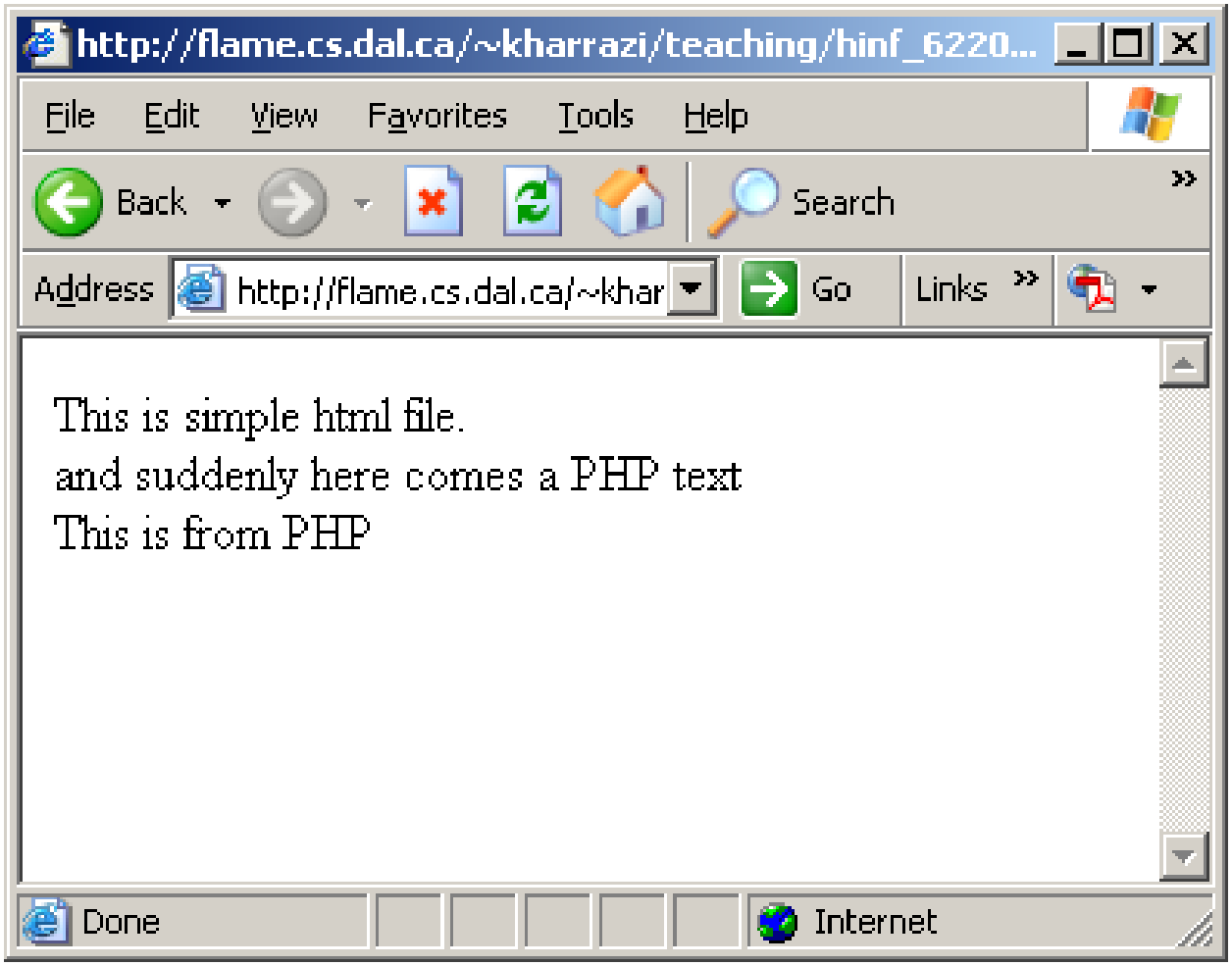
echo

- echo() also has a shortcut syntax, where you can immediately follow the opening tag with an equals sign without the PHP name. This short syntax only works with the *short_open_tag* configuration setting enabled.

```
<html>
        <body>
                This is simple html file.
                <br>
                and suddenly here comes a PHP text
                <br>
                <?="This is from PHP"?>
        </body>
</html>
```

PHP echo (cont.)

echo



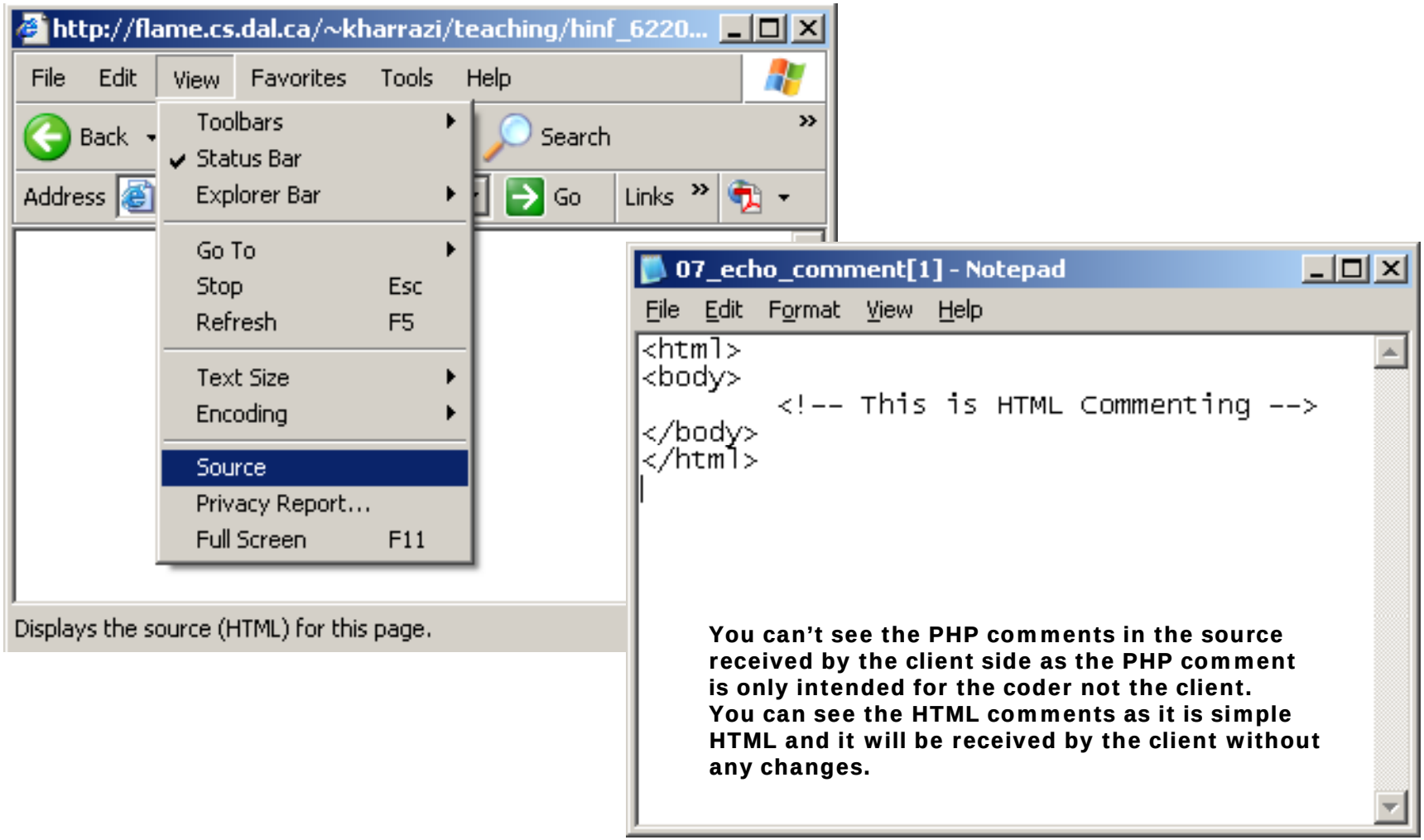
3. PHP Commenting

- In PHP, we use // to make a single-line comment or /* and */ to make a large comment block.

```
<html>
<body>
        <!-- This is HTML Commenting -->
        <?php
                //This is a PHP comment

                /*
                This is a PHP
                comment block
                */
        ?>
</body>
</html>
```

PHP Commenting (cont.)



4. PHP Variables

- All variables in PHP start with a $ sign symbol. Variables may contain different data types. PHP supports the following data types:
  - *Integer*: Used for whole numbers
  - *Double*: Used for real numbers
  - *String*: Used for strings of characters
  - *Boolean*: Used for true or false values
  - *Array*: Used to store multiple data items of the same type
  - *Object*: Used for storing instances of classes

Numbers

```
<?php
        $x = ...
?>
```

PHP Variables (cont.)

- Variables are variable! That means variables can change through a script and accept new values unless they are defined as *constants*.

- In the following example $x will be 5 at the end of the script:

```
<?php
        $x = 2;
        $x = 5;
?>
```

PHP Variables:String (cont.)

string

- To indicate that a variable is a string you can double quote the variables content:

```
<?php
        $x = "This is a string";
        $y = "12345";
?>
```

- Therefore $y="12345" is a string not a number.

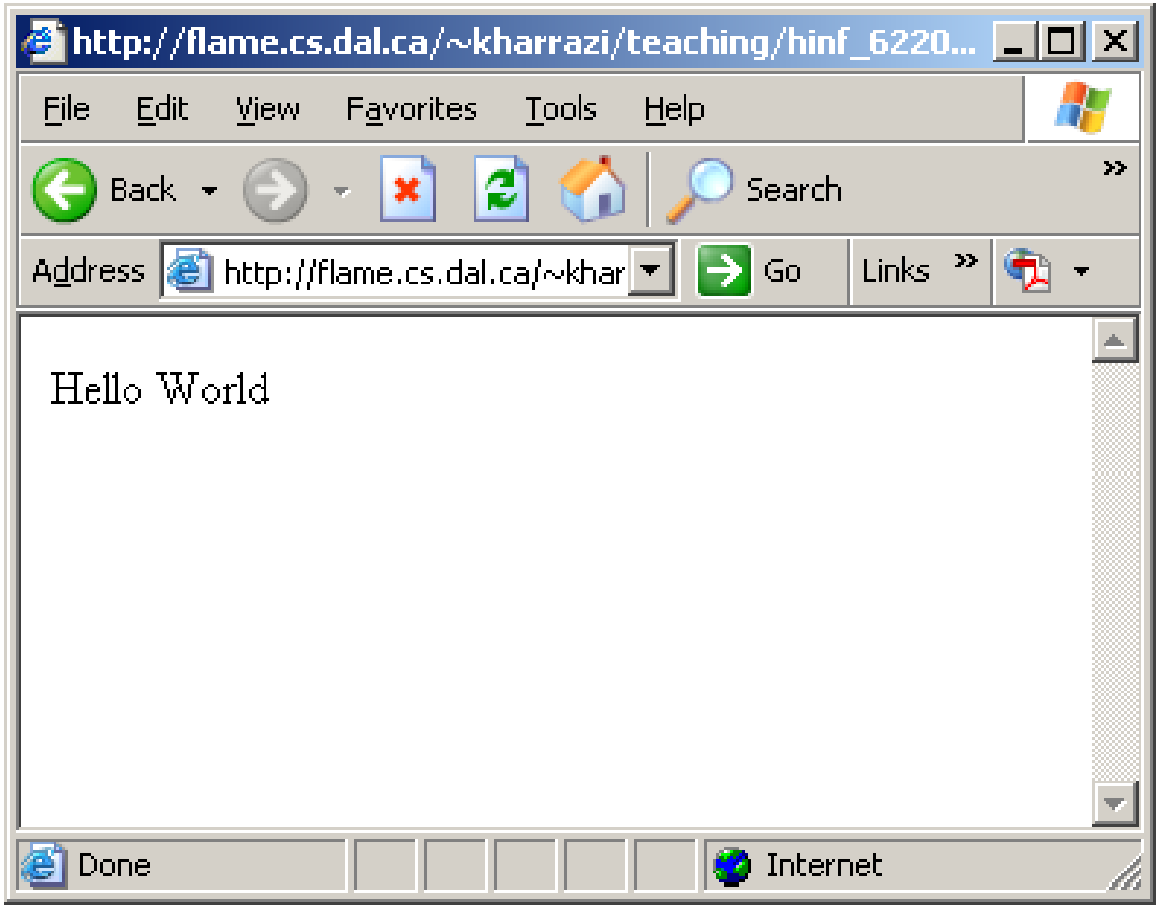
PHP Variables:String (cont.)

string

```
<?php
     $x = "Hello World";
     echo $x;
?>
```

PHP Variables:String (cont.)

string



PHP Variables:String (cont.)

string

- To concatenate two or more string variables together, use the dot (.) operator:

```
<?php
        $x.$y
?>
```

PHP Variables:String (cont.)

string

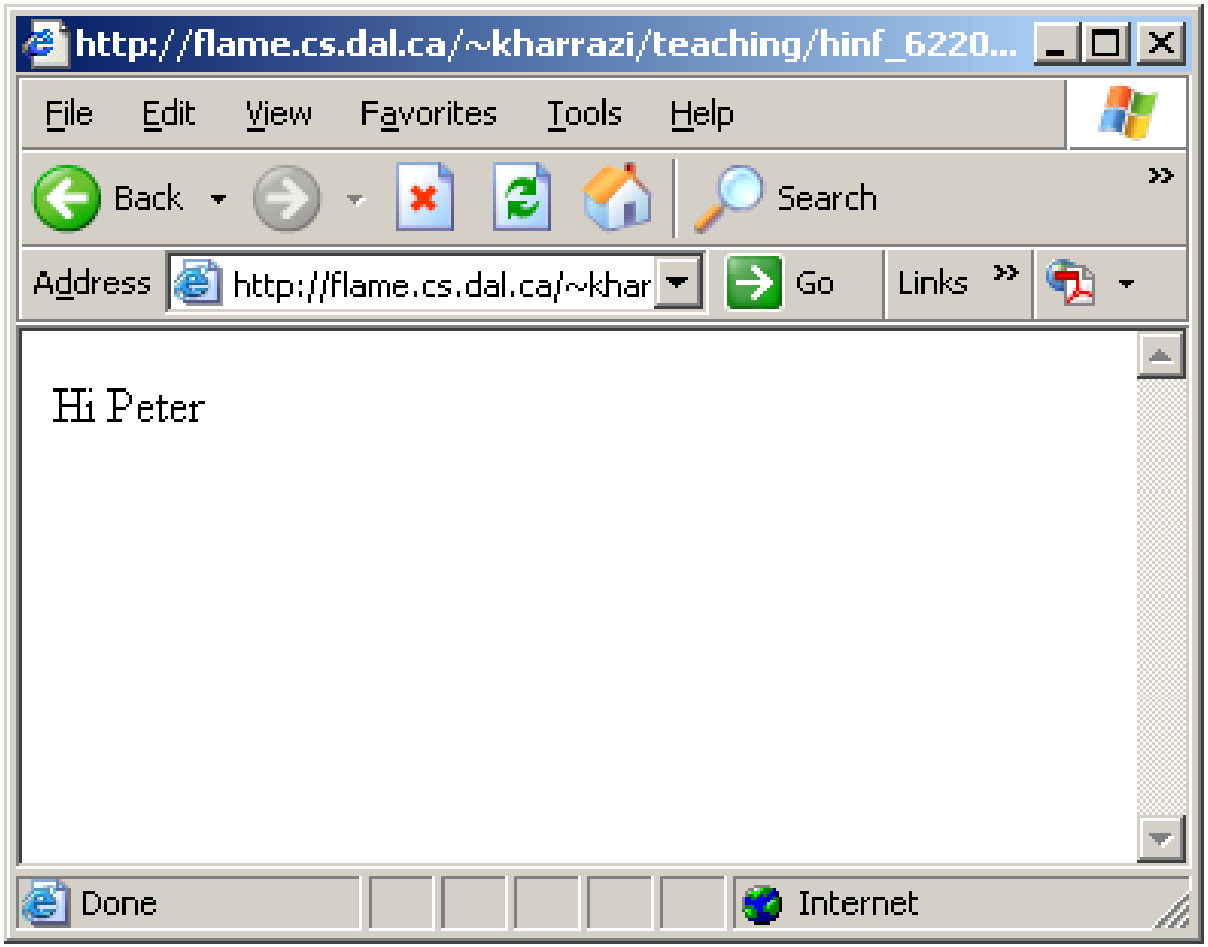
```
<?php
     $x = "Hi ";
     $y = "Peter";
     echo $x.$y;
?>
```

Or we could write:

```
<?php
     $x = "Hi ";
     $y = "Peter";
     $z = $x.$y;
     echo $z;
?>
```

PHP Variables:String (cont.)

string



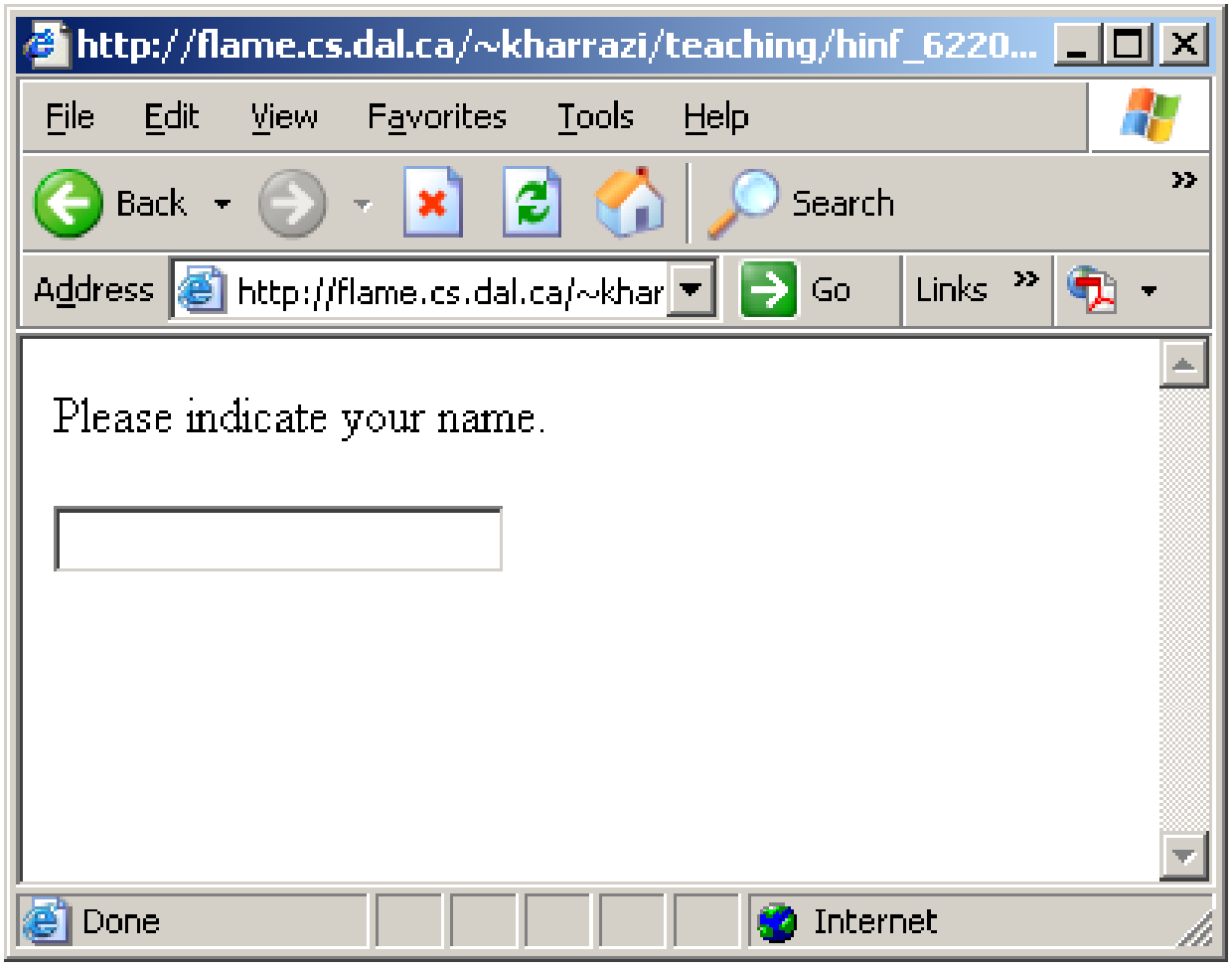
PHP Variables:String (cont.)

string

```
<?php
    $var1 = "Please indicate your name.";
    $var2 = "<br>";
    $var3 = "<form action='../html/print_data.php'
    method='post'>";
    $var4 = "<input type='text' name='firstname'>";
    $var5 = "</form>";
    echo $var1.$var2.$var3.$var4.$var5;
?>
```

PHP Variables:String (cont.)

string



PHP Variables:String (cont.)

string

Or we could write:

```
<?php
    $var = "Please indicate your name.";
    $var = $var . "<br>";
    $var = $var . "<form action='../html/print_data.php'
    method='post'>";
    $var = $var . "<input type='text' name='firstname'>";
    $var = $var . "</form>";
    echo $var;
?>
```

PHP Variables:String (cont.)

string

Or we could write:

```
<?php
    $var = "Please indicate your name.";
    $var .= "<br>";
    $var .= "<form action='../html/print_data.php'
    method='post'>";
    $var .= "<input type='text' name='firstname'>";
    $var .= "</form>";
    echo $var;
?>
```

Shortcut

```
$x = $x . "Some String";
$x.= "Some String";
```

PHP Variables:String (cont.)

string

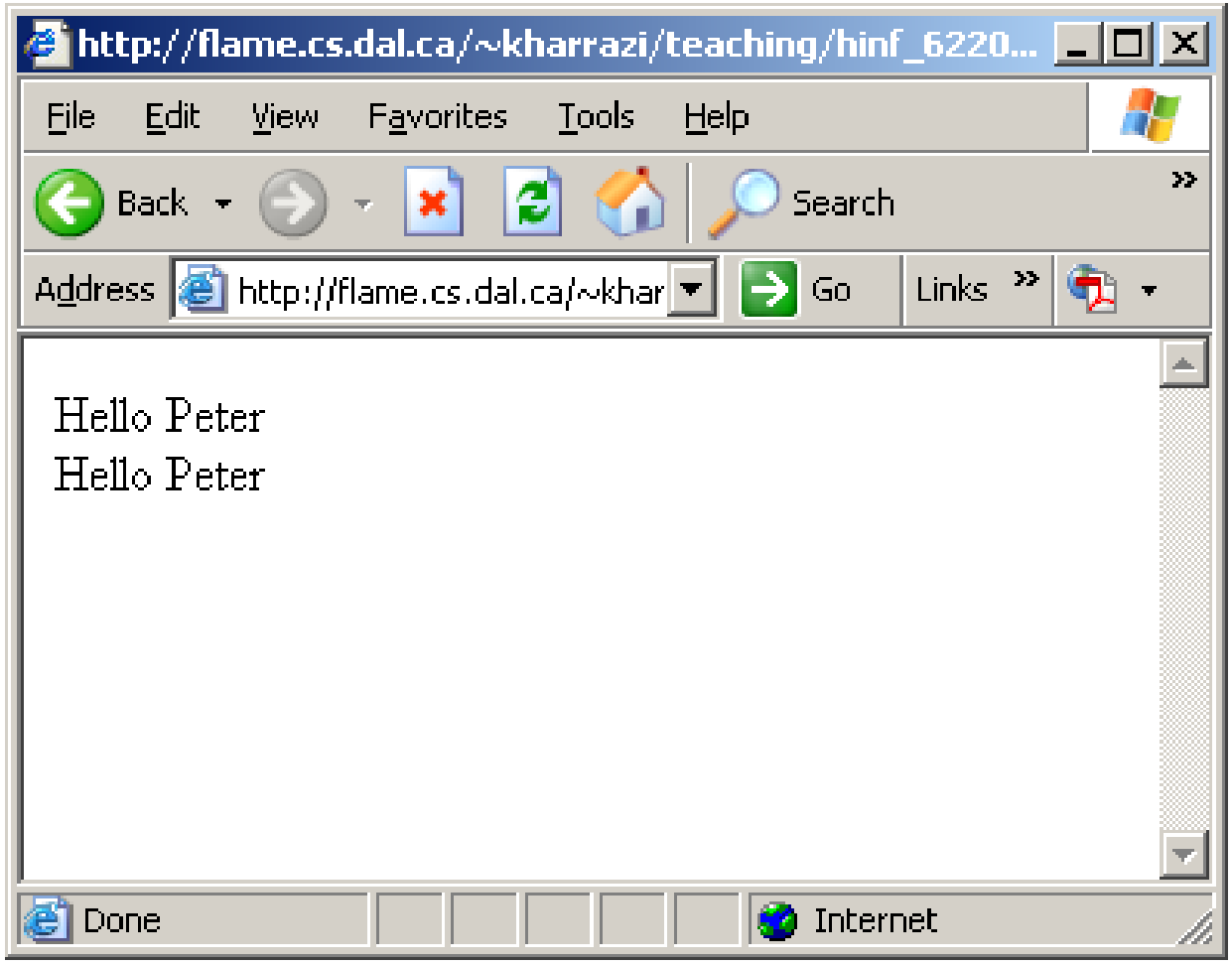
- You can use the variable name inside a quoted string. PHP is smart enough to interpret the variable:

```
$x = "My name";
echo "Some String $x";
```

```
<?php
    $var = "Peter";
    echo "Hello " . $var;
    echo "<br>";
    echo "Hello $var";
?>
```

PHP Variables:String (cont.)

string



PHP Variables:Number (cont.)

number

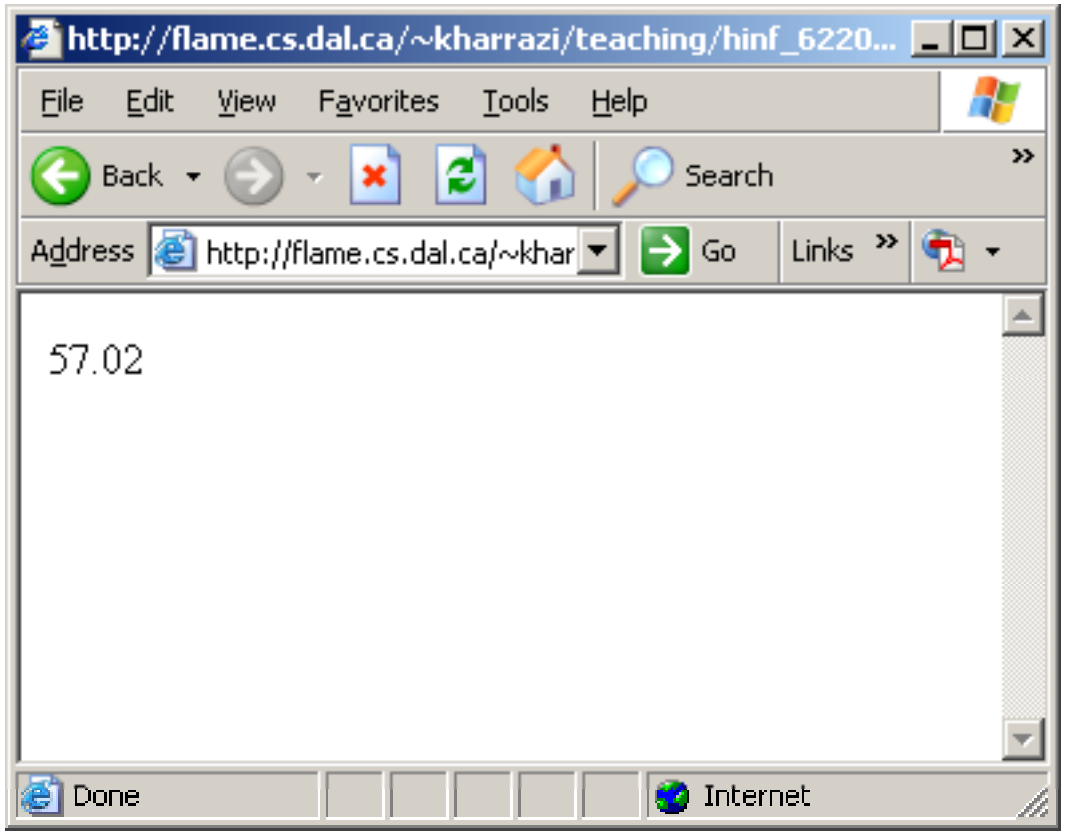
- Numbers could be either integer or double (float with decimals). PHP is smart enough to recognize any situation and can change the number types to each other. In the following example $x is a float number while $y is an integer. Finally $z is the sum of $x and $y (sum of float and integer) and PHP defines $z as an float automatically.

```
<?php
        $x = 12.02;
        $y = 45;
        $z = $x + $y;
?>
```

PHP Variables:Number (cont.)

number

```
<?php
      $x = 12.02;
      $y = 45;
      $z = $x + $y;
      echo $z;
?>
```



PHP Variables:Number (cont.)

number

- Arithmetic Operators:

Operator	Description	Example	Result (echo $y)
+	Addition	$x=12; $y = $x + 5;	17
-	Subtraction	$x=12; $y = $x - 5;	9
*	Multiplication	$x=12; $y = $x * 5;	60
/	Division	$x=12; $y = $x / 5;	2.4
%	Modulus	$x=12; $y = $x % 5;	2
++	Increment	$y=12; $y++; (++$y)	13
--	Decrement	$y=12; $y--; (--$y)	11

PHP Variables:Number (cont.)

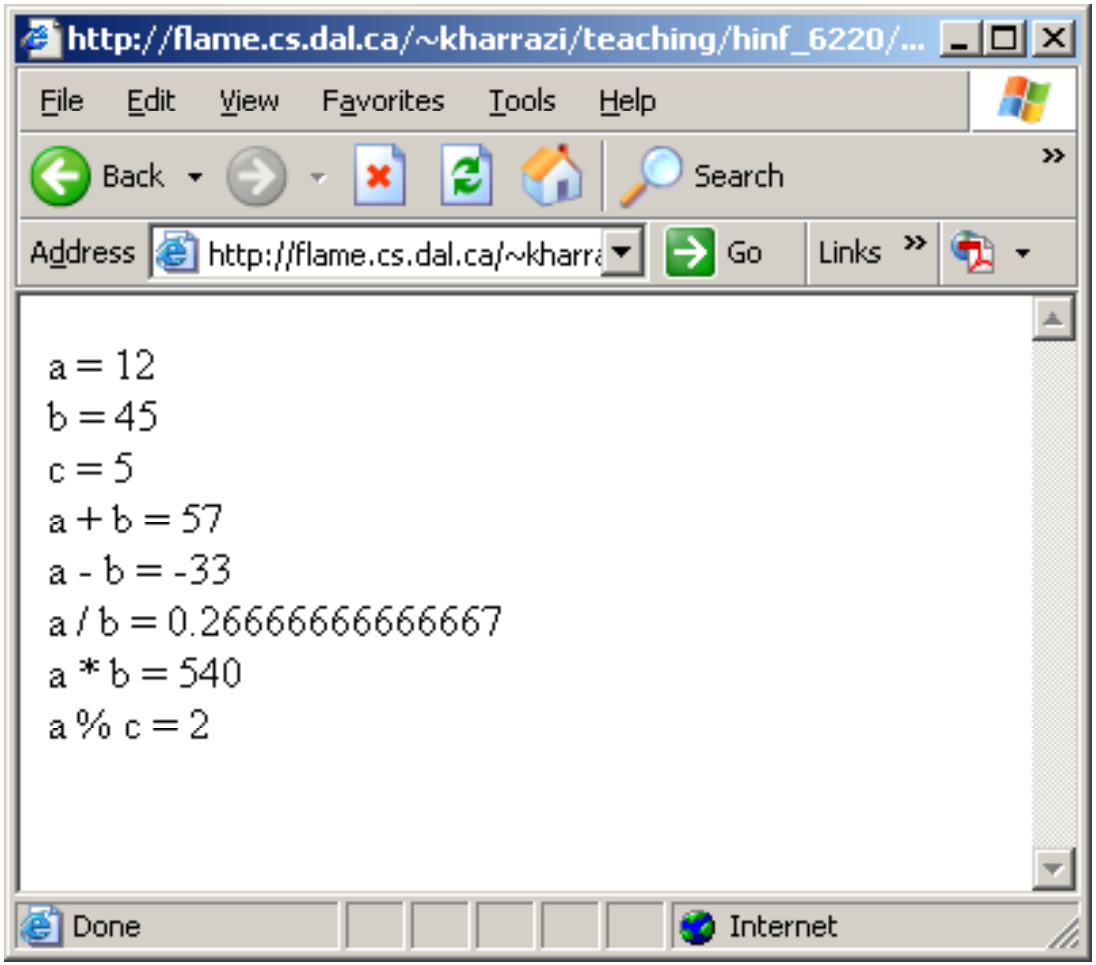
number

```
<?php
    $a = 12; $b = 45; $c = 5;
    echo "a = $a";                          echo "<br>";
    echo "b = $b";                          echo "<br>";
    echo "c = $c";                          echo "<br>";
    echo "a + b = " . ($a + $b);            echo "<br>";
    echo "a - b = " . ($a - $b);            echo "<br>";
    echo "a / b = " . ($a / $b);            echo "<br>";
    echo "a * b = " . ($a * $b);            echo "<br>";
    echo "a % c = " . ($a % $c);            echo "<br>";
?>
```

? Regular Expression

PHP Variables:Number (cont.)

number



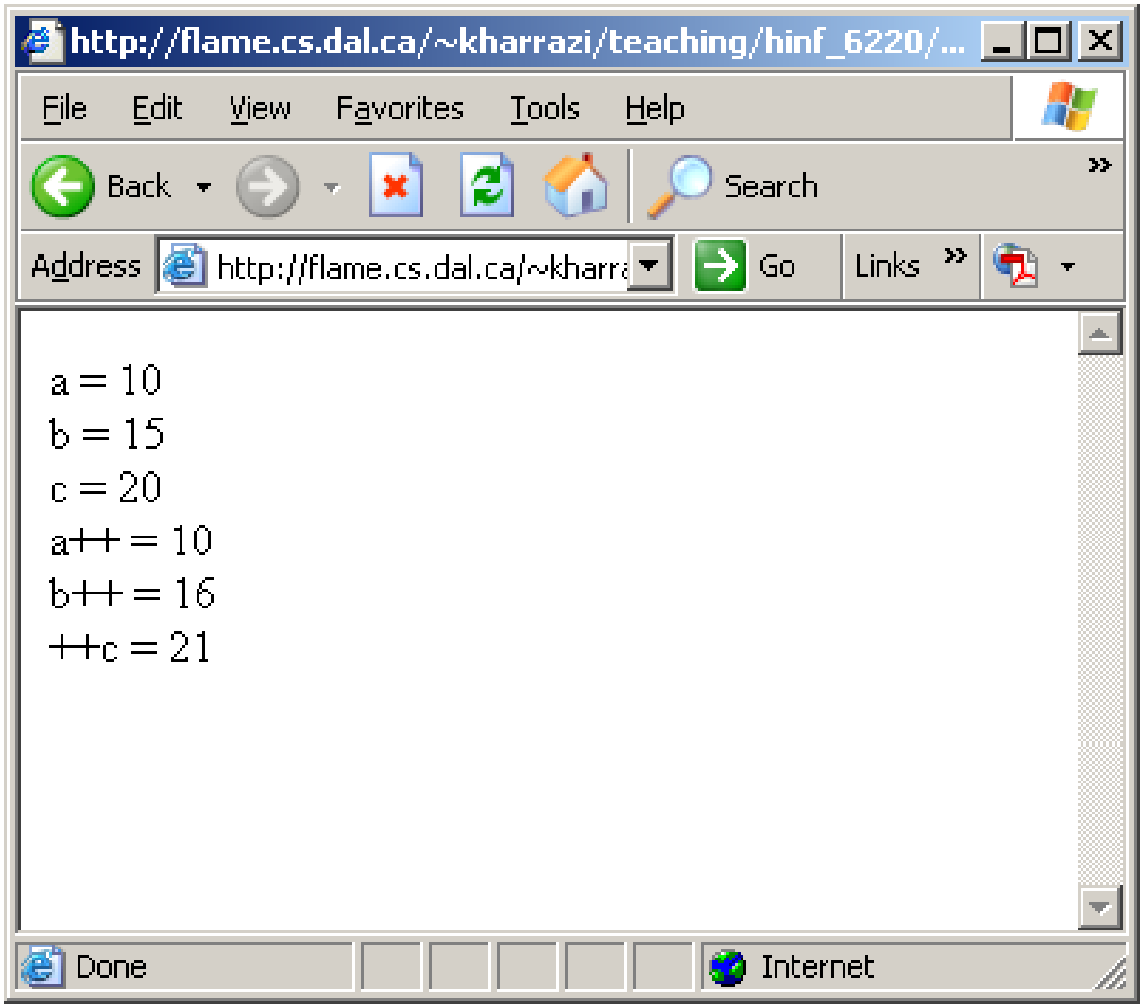
PHP Variables:Number (cont.)

number

```
<?php
    $a = 10; $b = 15; $c = 20;
    echo "a = $a";                    echo "<br>";
    echo "b = $b";                    echo "<br>";
    echo "c = $c";                    echo "<br>";
    // a
    echo "a++ = " . ($a++);           echo "<br>";
    // b
    $b++;
    echo "b++ = " . $b;               echo "<br>";
    // c
    echo "++c = " . (++$c);           echo "<br>";
?>
```

PHP Variables:Number (cont.)

number



PHP Variables:Number (cont.)

number

- Assignment Operators:

Operator	Example	Result (echo $y)
=	$y = 5;	5
+=	$y = 5; $y += 10; ($y = $y + 10)	15
-=	$y = 5; $y -= 10; ($y = $y - 10)	-5
*=	$y = 5; $y *= 10; ($y = $y * 10)	50
/=	$y = 5; $y /= 10; ($y = $y / 10)	.5
%=	$y = 15; $y %= 10; ($y = $y % 10)	5

=	$x=12; $y = $x;	12
+=	$x = 3; $y = 6; $y += $x; ($y = $y + $x)	9

PHP Variables:Number (cont.)

number

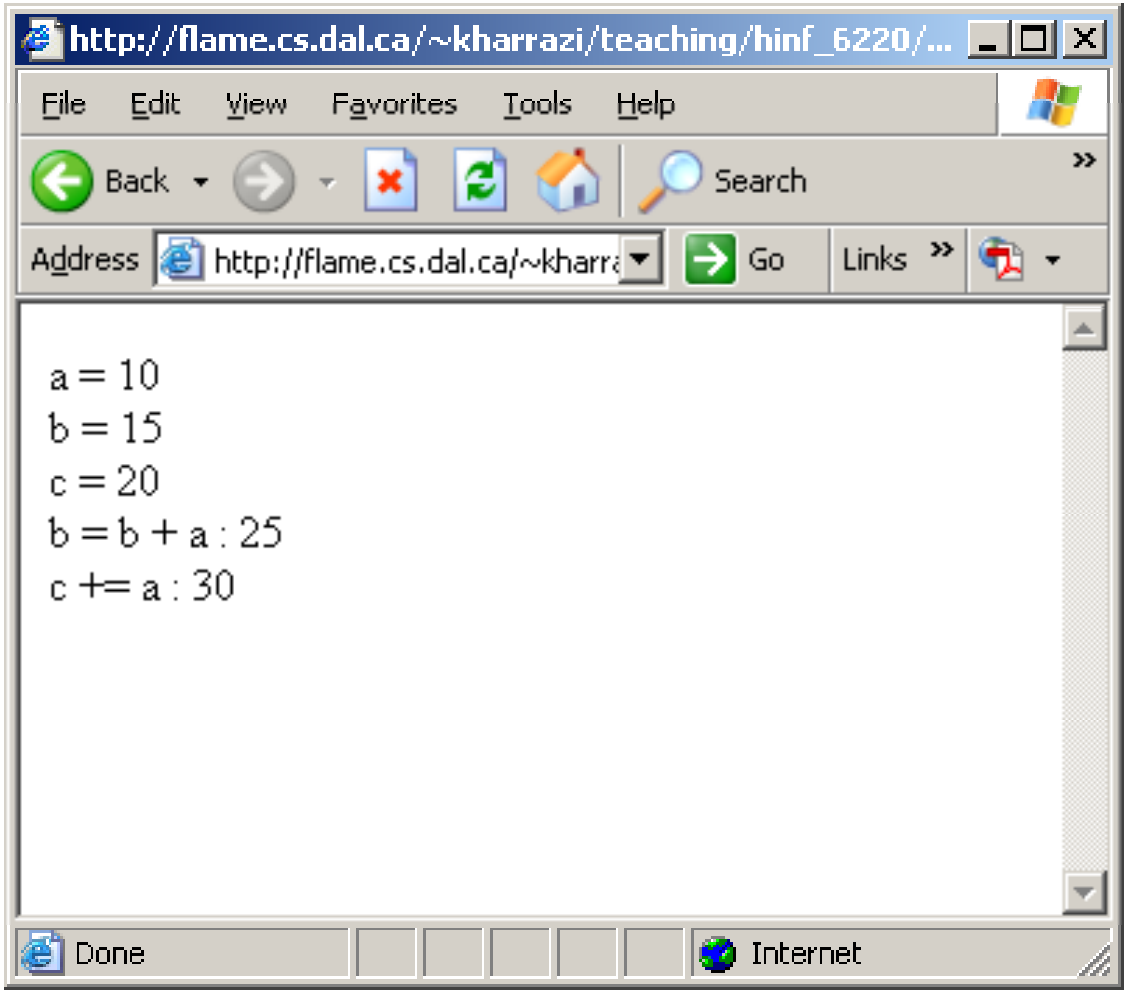
```
<?php
    $a = 10; $b = 15; $c = 20;
    echo "a = $a";                          echo "<br>";
    echo "b = $b";                          echo "<br>";
    echo "c = $c";                          echo "<br>";

    $b = $b + $a;
    echo "b = b + a : " . $b;               echo "<br>";

    $c += $a;
    echo "c += a : " . $c;                  echo "<br>";
?>
```

PHP Variables:Number (cont.)

number



PHP Variables:Array (cont.)

array

- A PHP array is a special data type that can store multiple values in the same variable. The commonly used term for each value is element. Each element can be accessed by the array index, which can be a number or a string. If the index is a string, the array is known as an associative array.

- By default, index values start at zero, and arrays are assumed to be zero-based. This means that the index number of the last element is typically one less than the number of elements. In other words, if an array has four elements, the last value will be referenced as $my_array[3].

PHP Variables:Array (cont.)

array

- These functions allow you to interact with and manipulate arrays in various ways. Arrays are essential for storing, managing, and operating on sets of variables.

- Simple and multi-dimensional arrays are supported, and may be either user created or created by another function. There are specific database handling functions for populating arrays from database queries, and several functions return arrays.

```
<?php
        $x = array(...);
?>
```

PHP Variables:Array (cont.)

array

- PHP arrays have many advantages. First of all, they're flexible. You can store as many items as you need, and work with all of them from one variable.

- In addition, you can work with a single element at a time by using the index value, or you can work with all of the elements. There are many built in PHP functions to help you loop through all the values in the array, count them, shuffle them, sort them, and more.

PHP Variables:Array (cont.)

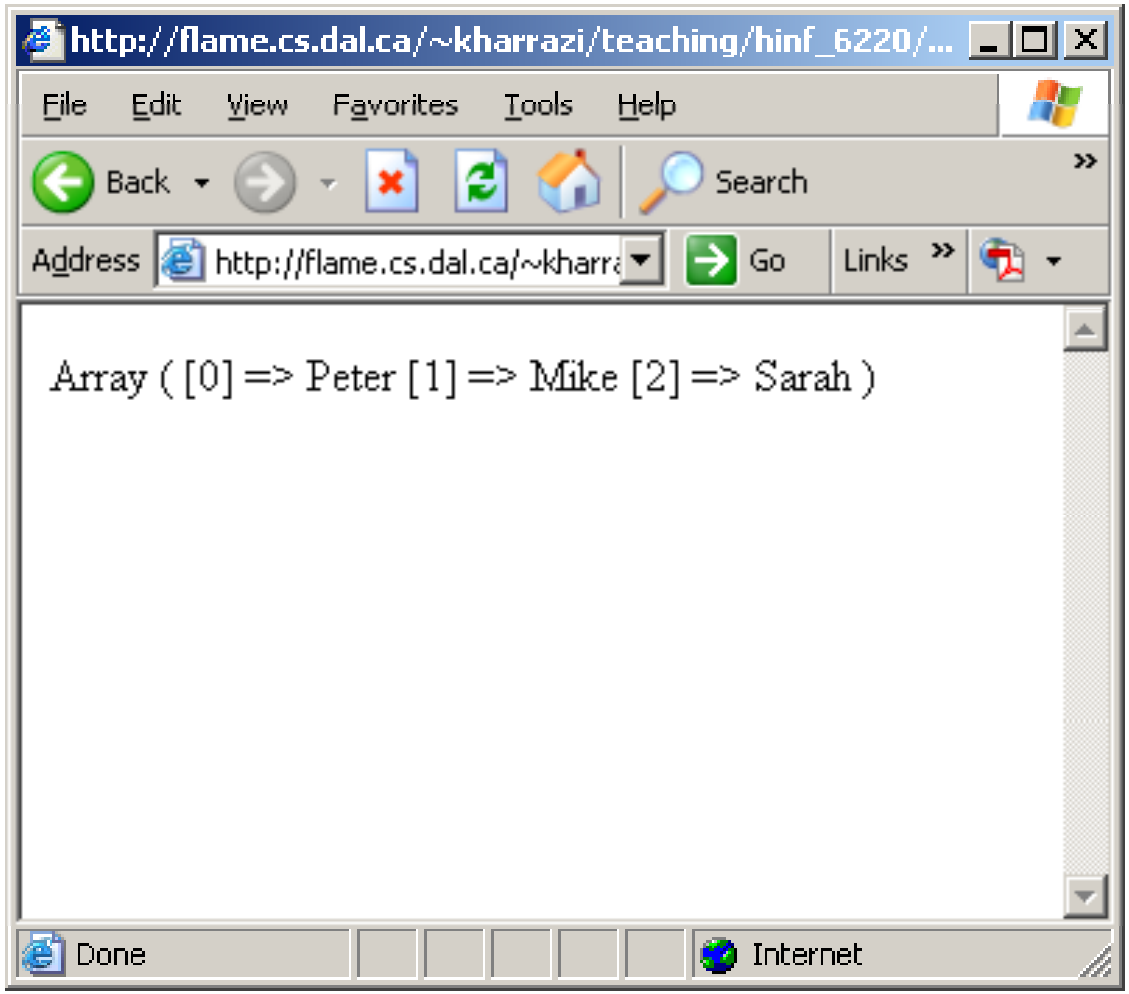
array

```
<?php
     $a = array ('Peter', 'Mike', 'Sarah');

     // print_r will echo array's contents
     print_r($a);
?>
```

PHP Variables:Array (cont.)

array



PHP Variables:Array (cont.)

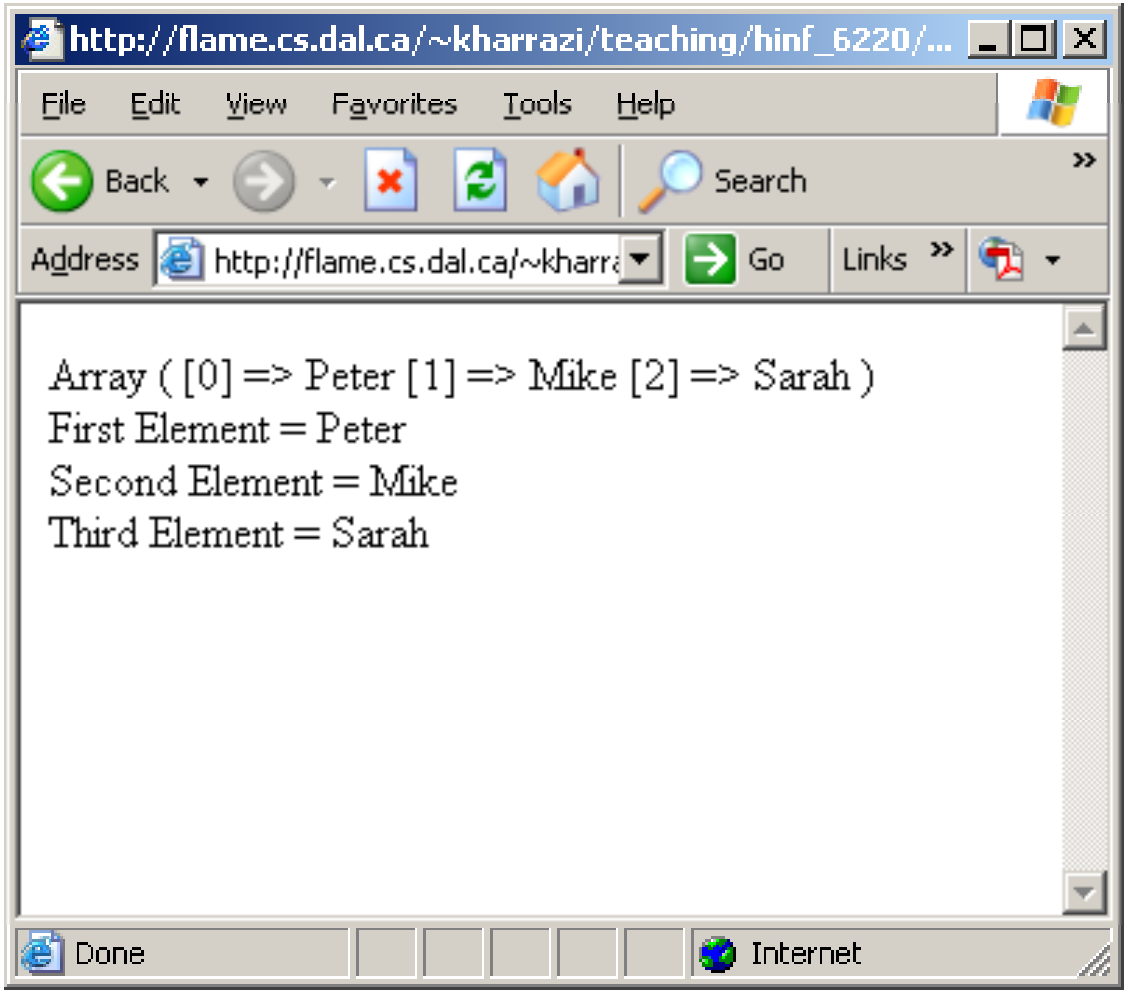
array

```
<?php
     $a[0] = 'Peter';
     $a[1] = 'Mike';
     $a[2] = 'Sarah';

     // print_r will echo array's contents
     print_r($a);
     echo '<br>';
     echo 'First Element = '      . $a[0] . '<br>';
     echo 'Second Element = '     . $a[1] . '<br>';
     echo 'Third Element = '      . $a[2] . '<br>';
?>
```

PHP Variables:Array (cont.)

array



PHP Variables:Array (cont.)

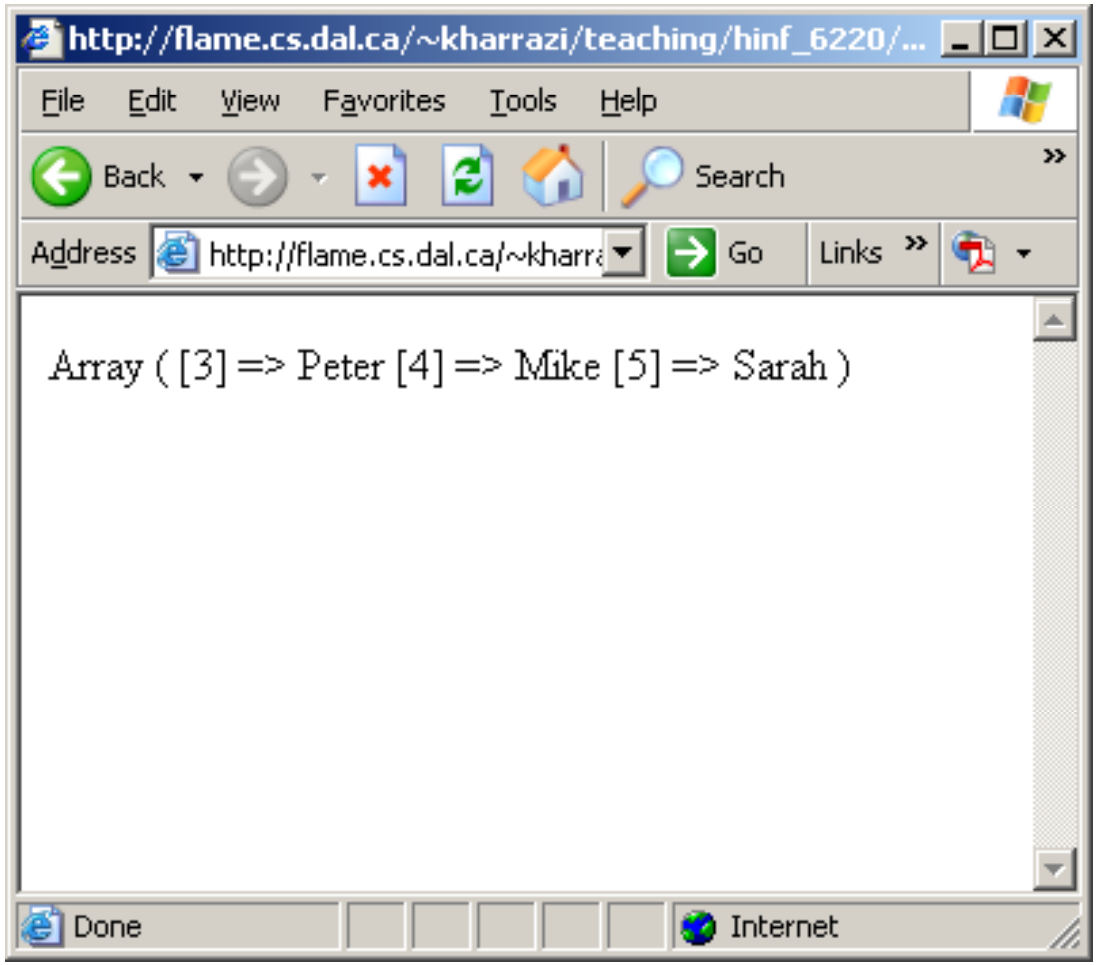
associative array

```
<?php
    $a = array (3 => 'Peter', 4 => 'Mike', 5 => 'Sarah');

    // print_r will echo array's contents
    print_r($a);
?>
```

PHP Variables:Array (cont.)

associative array



PHP Variables:Array (cont.)

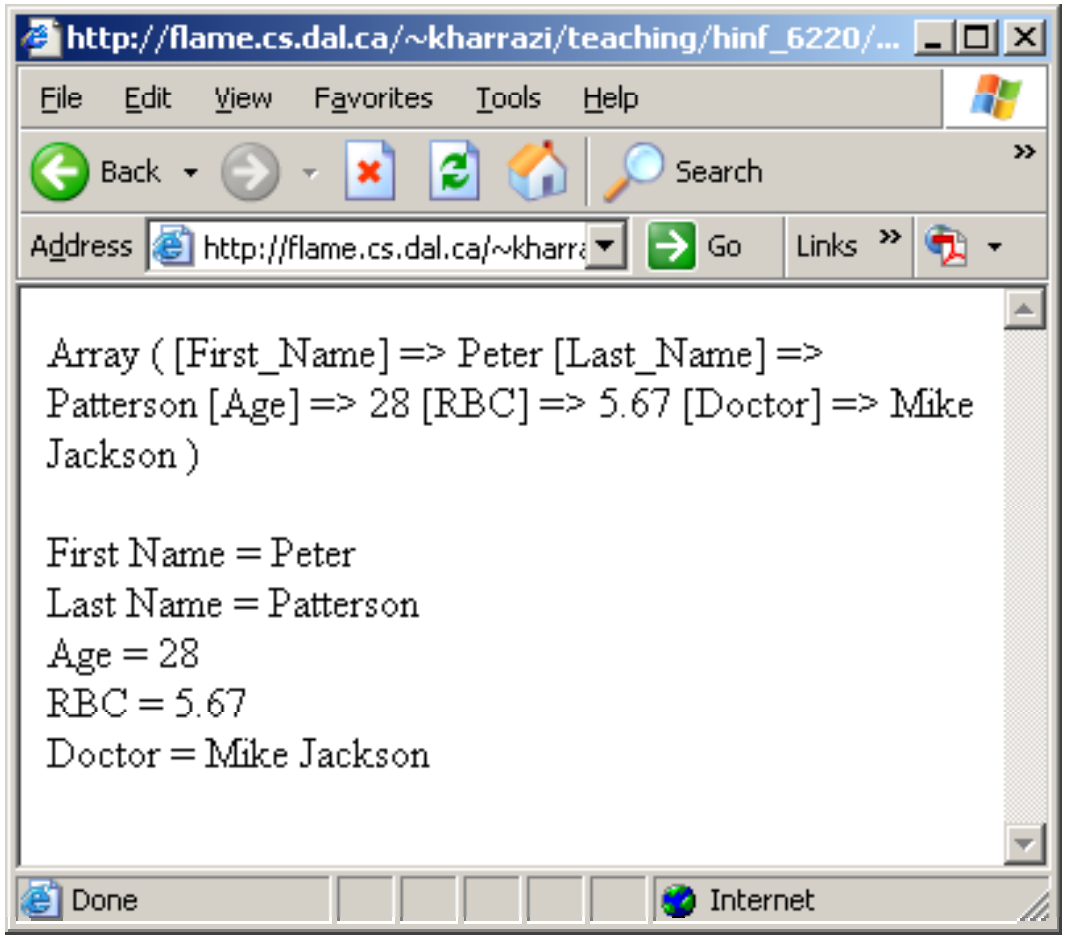
associative array

```
<?php
      $a = array (
                  'First_Name'      => 'Peter',
                  'Last_Name'       => 'Patterson',
                  'Age'             => 28,
                  'RBC'             => 5.67,
                  'Doctor'          => 'Mike Jackson'
                  );

      // print_r will echo array's contents
      print_r($a);
      echo '<br><br>';
      echo 'First Name = '          . $a['First_Name'] .      '<br>';
      echo 'Last Name = '           . $a['Last_Name'] .       '<br>';
      echo 'Age = '                 . $a['Age'] .             '<br>';
      echo 'RBC = '                 . $a['RBC'] .             '<br>';
      echo 'Doctor = '              . $a['Doctor'] .          '<br>';
?>
```

PHP Variables:Array (cont.)

associative array



PHP Variables:Array (cont.)

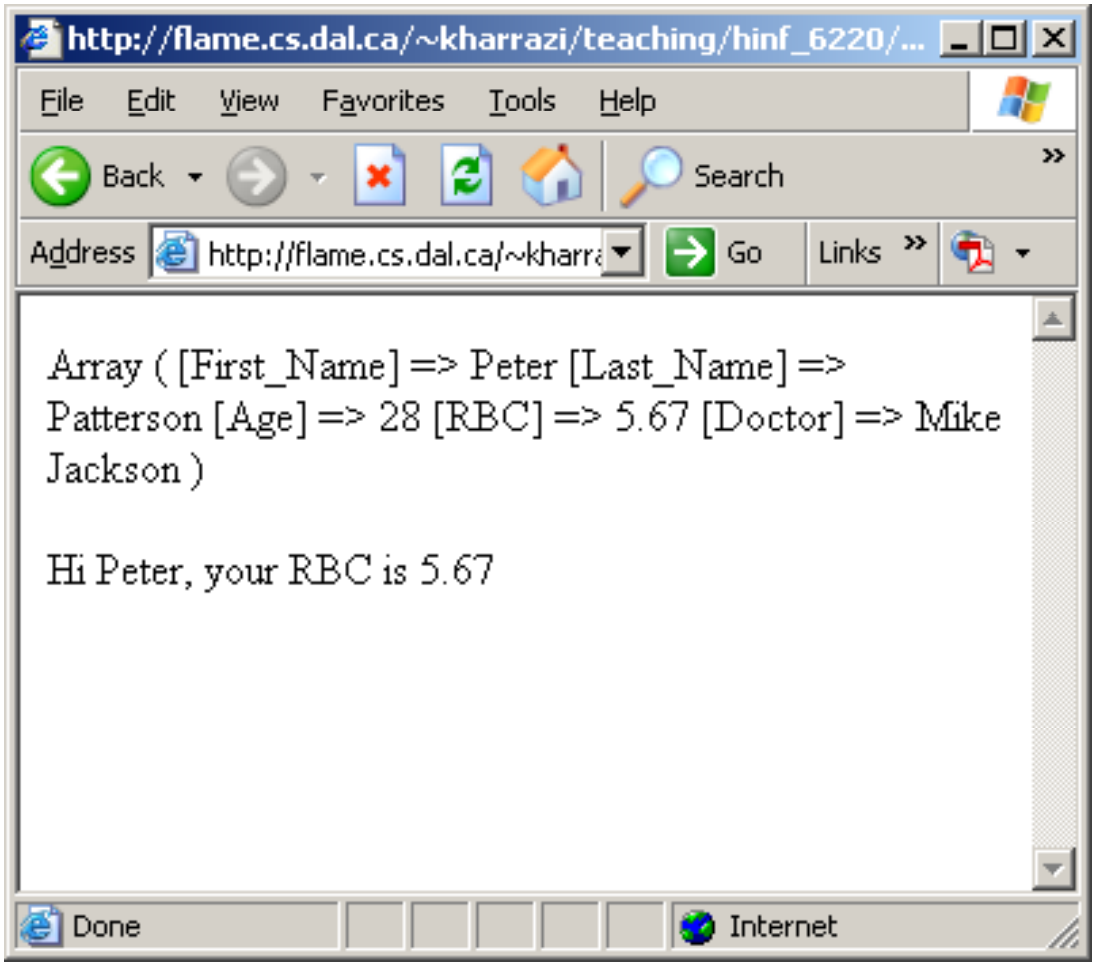
associative array

```
<?php
     $a = array (
               'First_Name'  => 'Peter',
               'Last_Name'   => 'Patterson',
               'Age'          => 28,
               'RBC'          => 5.67,
               'Doctor'       => 'Mike Jackson'
               );

     // print_r will echo array's contents
     print_r($a);
     echo '<br><br>';
     echo "Hi {$a['First_Name']}, your RBC is {$a['RBC']}";
?>
```

PHP Variables:Array (cont.)

associative array



PHP Variables:Array (cont.)

associative array

```
<?php
    $a = array ('Peter', 'Mike', 'Sarah');
    $b = array (23, 45, 19);
    $c = array ('Name' => $a, 'Age' => $b);

    // print_r will echo array's contents
    print_r($c);
    echo '<br><br>';
    echo 'Person 1 = ' . $c['Name'][0] . ' Age = ' .
    $c['Age'][0] .'<br>';
    echo 'Person 2 = ' . $c['Name'][1] . ' Age = ' .
    $c['Age'][1] .'<br>';
    echo 'Person 3 = ' . $c['Name'][2] . ' Age = ' .
    $c['Age'][2] .'<br>';
?>
```

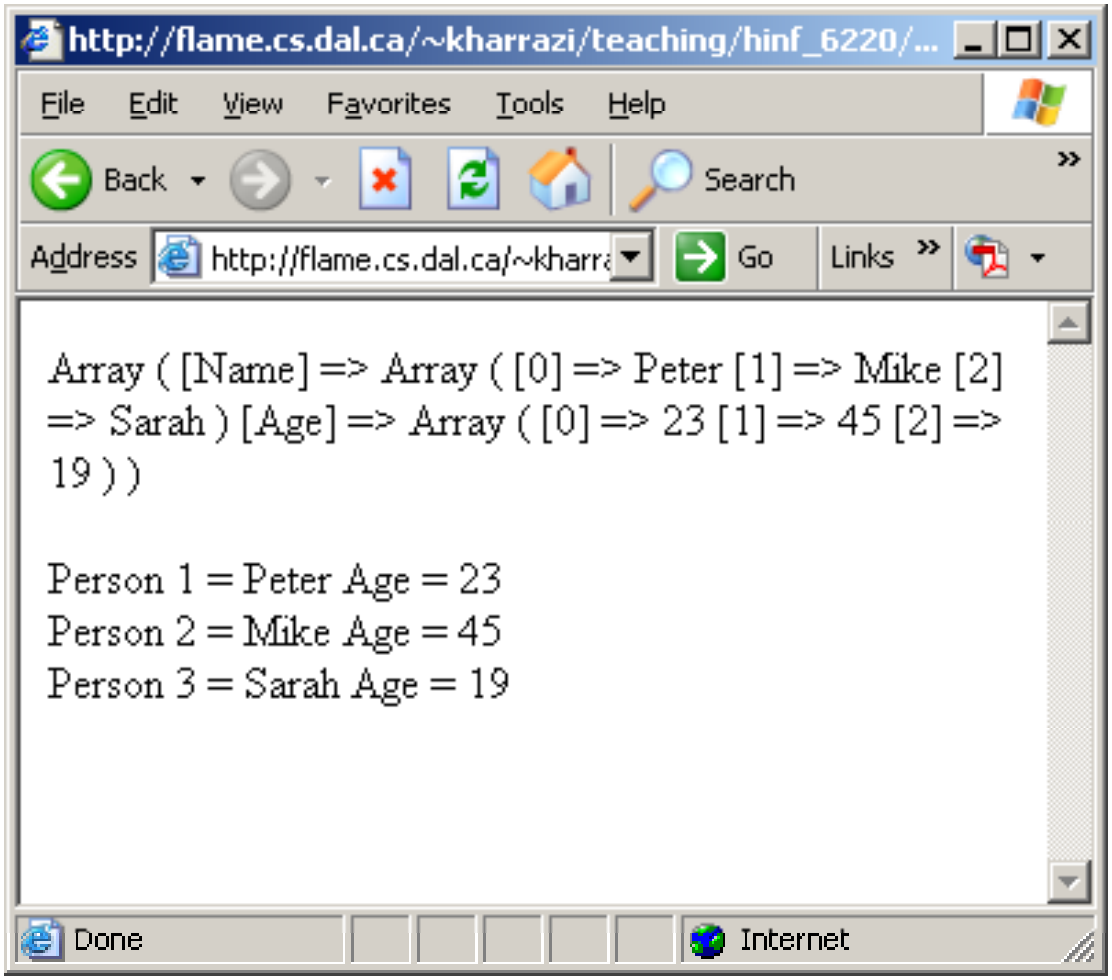
PHP Variables:Array (cont.)

associative array

$c	$a, $b Element 1	$a, $b Element 2	$a, $b Element 3
$c Element1 $c ['Name'] $a	Peter $c ['Name'][0]	Mike $c ['Name'][1]	Sarah $c ['Name'][2]
$c Element 2 $c ['Age'] $b	23 $c ['Age'][0]	45 $c ['Age'][1]	19 $c ['Age'][2]

PHP Variables:Array (cont.)

associative array



PHP Variables:Array (cont.)

array

```
<?php
    $a = array ('Peter', 'Mike', 'Sarah');
    $b = array (23, 45, 19);
    $c = array ($a, $b);

    // print_r will echo array's contents
    print_r($c);
    echo '<br><br>';
    echo 'Person 1 = ' . $c[0][0] . ' Age = ' . $c[1][0]
    .'<br>';
    echo 'Person 2 = ' . $c[0][1] . ' Age = ' . $c[1][1]
    .'<br>';
    echo 'Person 3 = ' . $c[0][2] . ' Age = ' . $c[1][2]
    .'<br>';
?>
```

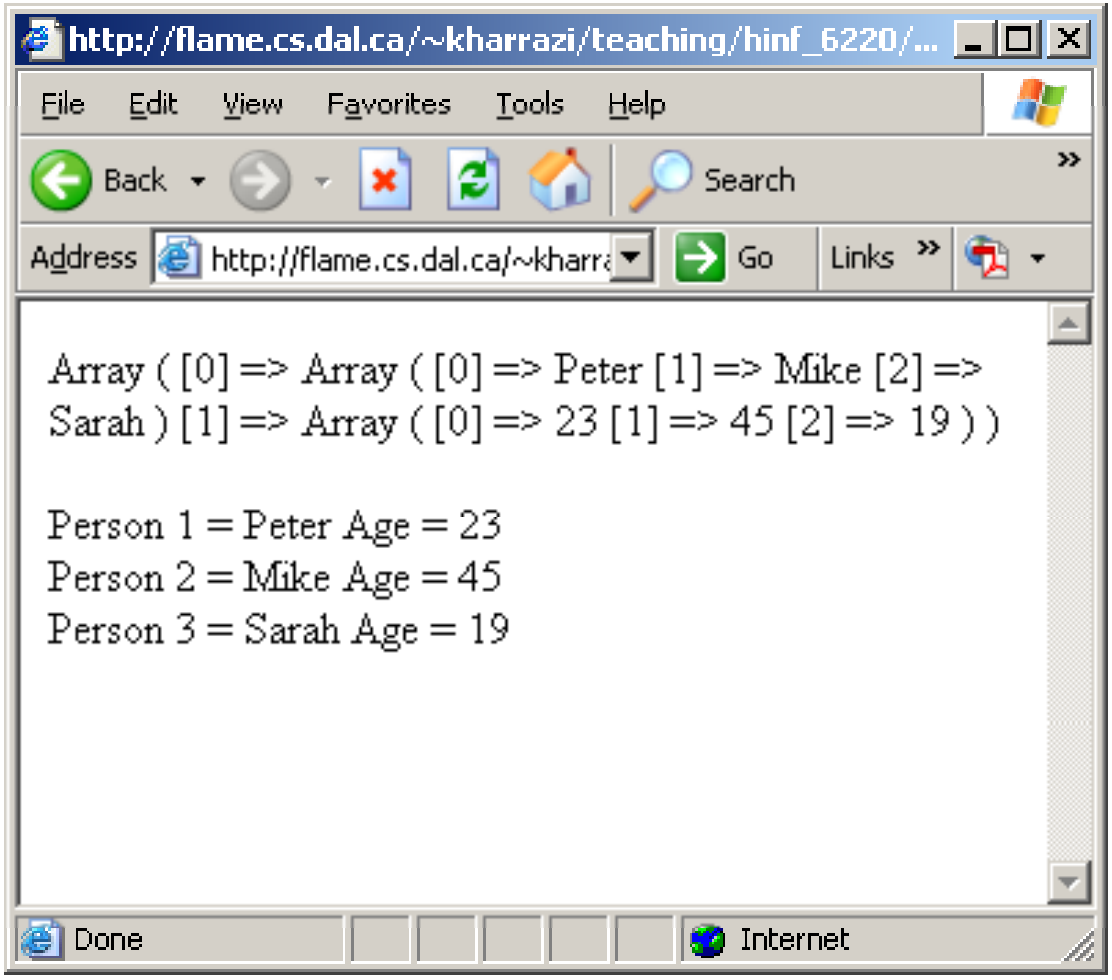
PHP Variables:Array (cont.)

array

$c	$a, $b Element 1	$a, $b Element 2	$a, $b Element 3
$c Element1 $c [0] $a	Peter $c [0][0]	Mike $c [0][1]	Sarah $c [0][2]
$c Element 2 $c [1] $b	23 $c [1][0]	45 $c [1][1]	19 $c [1][2]

PHP Variables:Array (cont.)

array



Summary

1. PHP Intro
2. PHP Syntax `<?php ?>`
3. PHP *echo* `echo 'something'`
4. PHP Commenting `// or /* */`
5. PHP Variables `$x = string, number, array`

Next Sessions

- PHP String/Array Manipulations
- PHP Conditions
- PHP Loops
- PHP Functions
- PHP Cookies/Sessions
- PHP SSI
- PHP Forms
- PHP/MySQL Integration

Exercise

- Please refer to the available text file in the slides section for this session on the course website:
- http://info510.com/core/public_page.php?page_name=slides