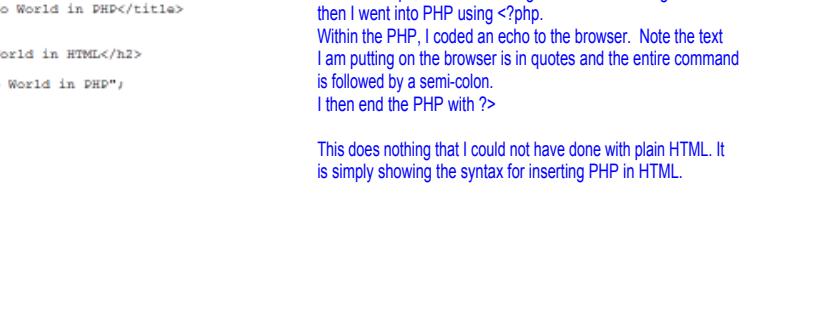


## Hello World in HTML

## Hello World in PHP



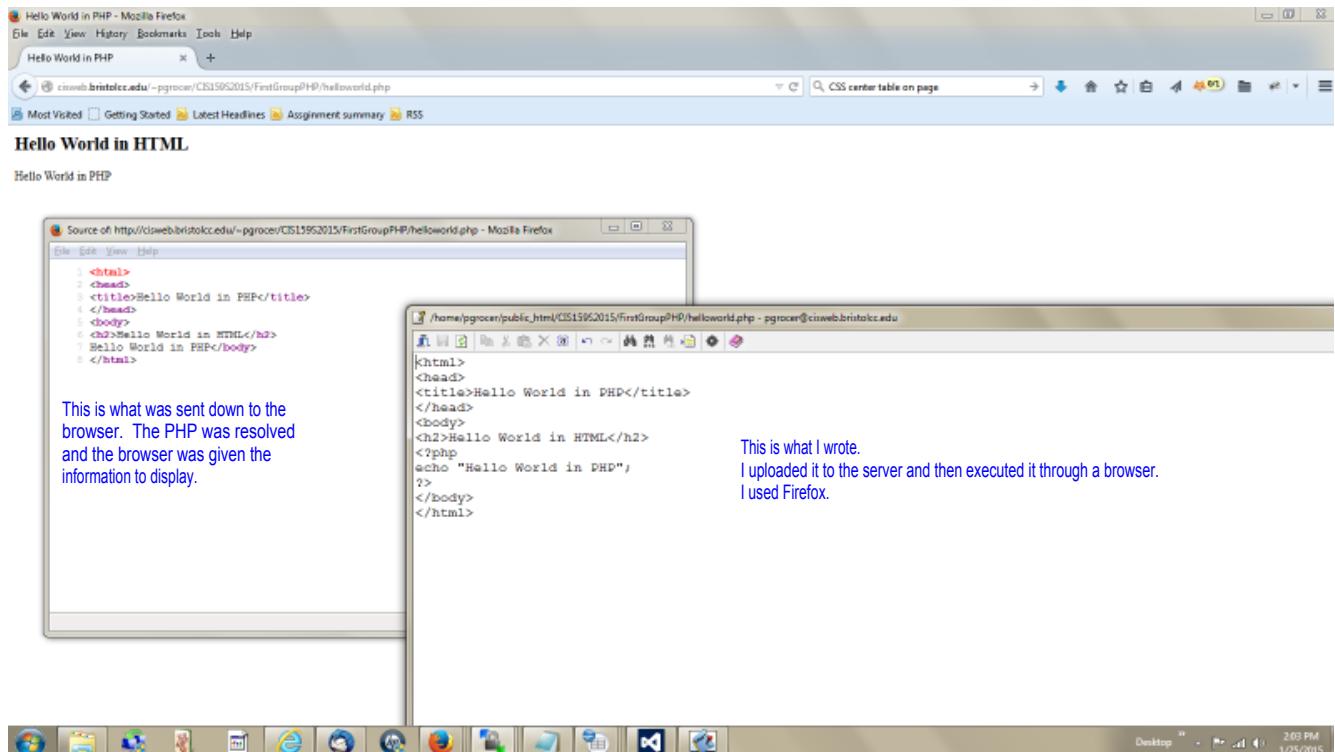
The screenshot shows a web browser window with the URL `/home/pgrocer/public_html/CIS1505.2015/FirstGroup/HP/helloworld.php`. The page content is as follows:

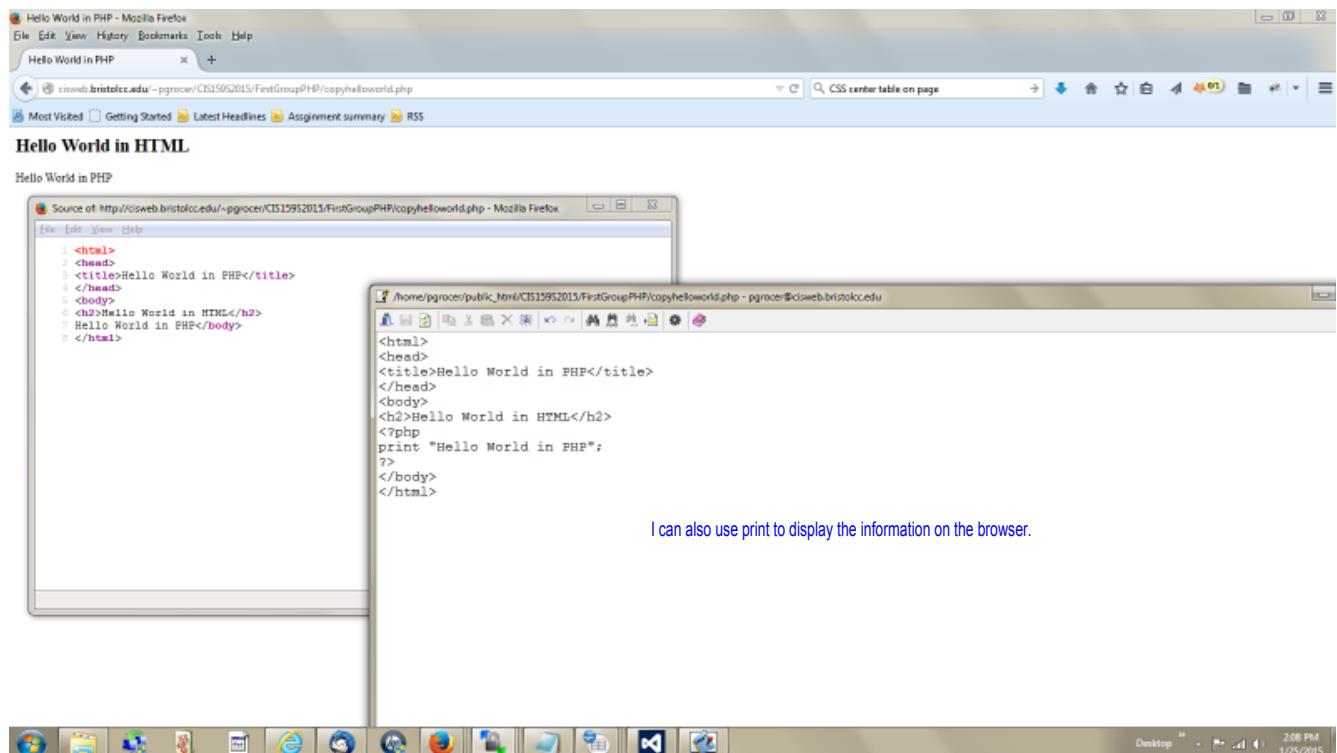
```
<html>
<head>
<title>Hello World in PHP</title>
</head>
<body>
<h2>Hello World in HTML</h2>
<?php
echo "Hello World in PHP";
?>
</body>
</html>
```

In this example I wrote some regular HTML including a <h2> and then I went into PHP using <?php. Within the PHP, I coded an echo to the browser. Note the text I am putting on the browser is in quotes and the entire command is followed by a semi-colon. I then end the PHP with ?>

This does nothing that I could not have done with plain HTML. It is simply showing the syntax for inserting PHP in HTML.







The screenshot shows a Windows desktop environment. In the center, there is a Mozilla Firefox window displaying the output of a PHP script. The page title is "Hello with variable" and the content is "Hello John Doe". Below this, another Firefox window is open, showing the source code of the same script. The source code is as follows:

```
1 <html>
2 <head><title>Hello with variable</title></head>
3 <body>
4 Hello John Doe</body>
5 </html>
```

To the left of the source code window, a text box contains the explanatory text: "The echo generated Hello John Doe which was sent to the browser." To the right of the source code window, another text box contains notes about PHP syntax: "Here I am setting up two variables - note that the first character must be the \$. I am then assigning values to the variables. The // are notes that I can insert on a line that contains PHP or on a separate line." At the bottom of the screen, a taskbar is visible with various icons, and the system tray shows the date and time as "1/25/2015 2:10 PM".

The screenshot shows a Windows desktop environment with three windows open:

- Firefox Browser Window:** Title: "Hello with variable - Mozilla Firefox". Address bar: "cinnweb.bristolcc.edu/~pgrocer/CS15952015/FirstGroupPHP/hellonameotherv.php". Content: Displays the output of the PHP script, showing multiple lines of text where variables are concatenated.
- Firefox Developer Tools Window:** Title: "Source of http://cinnweb.bristolcc.edu/~pgrocer/CS15952015/FirstGroupPHP/hellonameotherv.php - Mozilla Firefox". Content: Shows the raw HTML output of the browser, which includes the PHP code and its execution results.
- Code Editor Window:** Title: "/home/pgrocer/public\_html/CS15952015/FirstGroupPHP/hellonameotherv.php - pgrocer@cinnweb.bristolcc.edu". Content: Displays the original PHP code with annotations in blue text explaining specific parts of the code.

Annotations in the code editor window:

- #1 basically the same as it was on the previous example except it now includes #1. Note I used double quotes .
- #2 used the . to concatenate the data I wanted to display. Note that #4 is the same. I suspect it started out different... Ah well.
- Some things to note:
  - I am using single quotes.
  - I have a <br> within the single quotes to skip to the next line.
- #3 is the same as #2 except that I am using double quotes
- #5 and #6 are on the next page.

At the bottom right of the desktop screen, there is a status bar with the text: "Desktop 2:16 PM 1/25/2015".

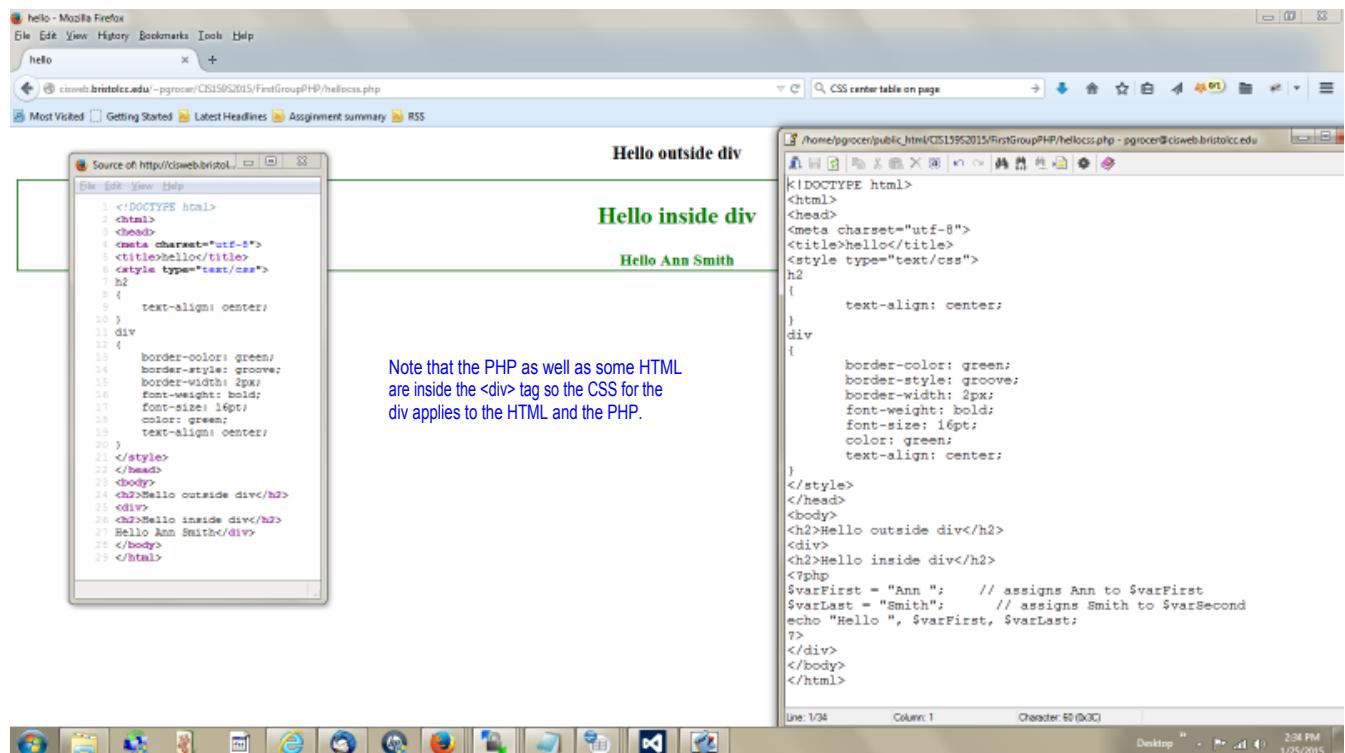
The screenshot shows a Windows desktop environment with two windows open. The top window is a Mozilla Firefox browser displaying the output of a PHP script. The output shows six lines of text: "#1 Hello John Doe", "#2 Hello John Doe", "#3 Hello John Doe", "#4 Hello John Doe", "#5 Hello John Doe", and "#6 Hello \$varFirst \$varLast". Below this, the browser shows the source code of the PHP script, which includes variable assignments and echo statements. The bottom window is a terminal or command-line interface window showing the same PHP source code. A note in the terminal window explains the behavior of double versus single quotes in PHP variable evaluation. The desktop taskbar at the bottom shows various icons for common applications like File Explorer, Internet Explorer, and Microsoft Word.

```
#1 Hello John Doe
#2 Hello John Doe
#3 Hello John Doe
#4 Hello John Doe
#5 Hello John Doe
#6 Hello $varFirst $varLast

<html>
<head><title>Hello with variables</title></head>
<body>
#1 Hello John Doe<br>#2 Hello John Doe<br>#3 Hello John Doe<br>#4 Hello John Doe<br>#5 Hello John Doe<br>#6 Hello SvarFirst SvarLast
</body>
```

```
<html>
<head><title>Hello with variables</title></head>
<body>
<?php
SvarFirst = "John"; // assigns John to SvarFirst
SvarLast = "Doe"; // assigns Doe to SvarSecond
echo "#1 Hello ", SvarFirst, " ", SvarLast;
echo "<br>#2 Hello " . SvarFirst . " " . SvarLast;
echo "<br>#3 Hello " . SvarFirst . " " . SvarLast;
echo "<br>#4 Hello " . SvarFirst . " " . SvarLast;
echo "<br>#5 Hello SvarFirst SvarLast"; //variables are evaluated when enclosed in double quotes like this
echo '<br>#6 Hello SvarFirst SvarLast'; //with single quotes, the variables are not evaluated
?>
</body>
</html>

#5 has the <br>, the literal #5 Hello and the two variables all enclosed in
double quotes and PHP figures out what to send down to the browser.
Do note the mixture of literals and variables within the one set of double
quotes.
Also note that when I change to single quotes it does not work! Then the
names of the variables get displayed.
```



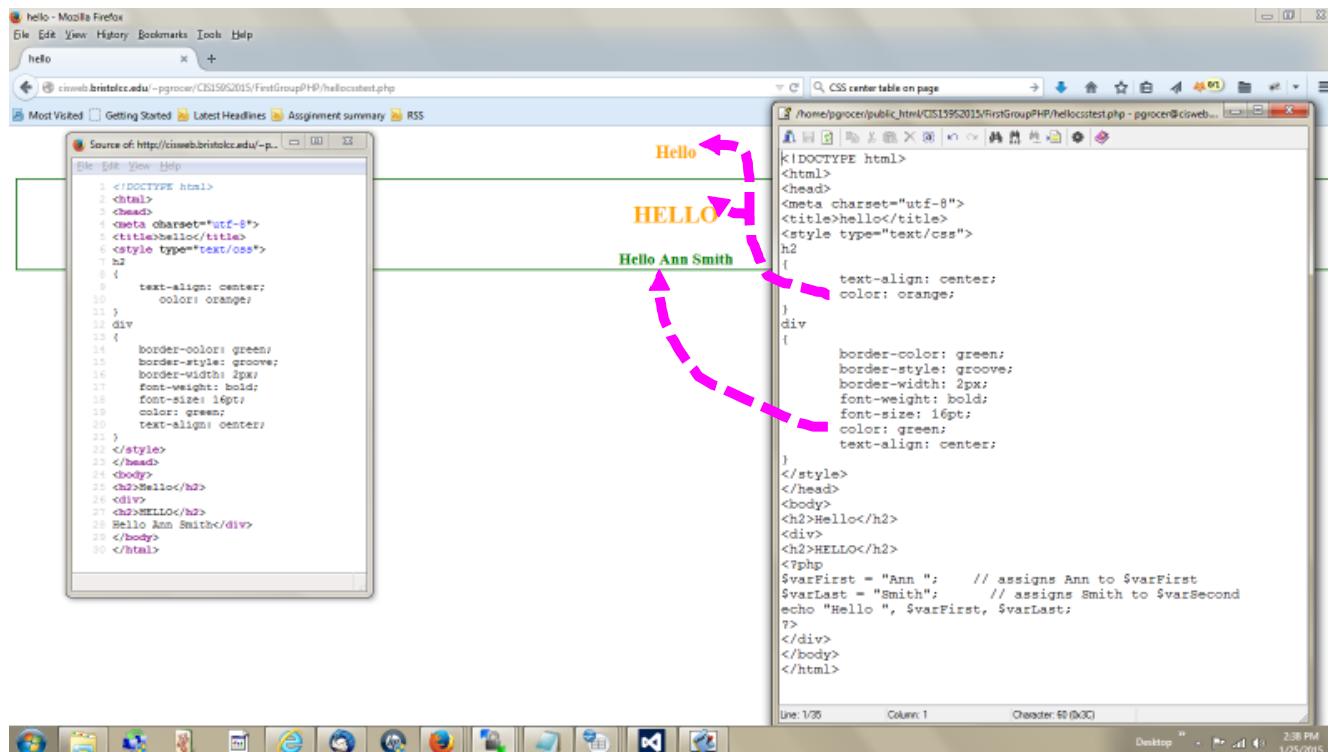
The screenshot shows a Windows desktop environment. In the center, there is a Mozilla Firefox browser window titled "hello - Mozilla Firefox". The address bar shows "http://ciweb.bristolcc.edu/~pgrocer/CS13952015/FirstGroupPHP/hellocss.php". The page content displays two sections: "Hello outside div" and "Hello inside div", both in green font. Below the browser is a terminal window with the title "/home/pgrocer/public\_html/CS13952015/FirstGroupPHP/hellocss.php - pgrocer@ciweb.bristolcc.edu". The terminal shows the PHP source code and its corresponding HTML output. A green box highlights the browser's source code panel, which contains the following code:

```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<title>Hello</title>
<style type="text/css">
h2
{
    text-align: center;
}
div
{
    border-color: green;
    border-style: groove;
    border-width: 2px;
    font-weight: bold;
    font-size: 16pt;
    color: green;
    text-align: center;
}
</style>
</head>
<body>
<h2>Hello outside div</h2>
<div>
<h2>Hello inside div</h2>
Hello Ann Smith</div>
</body>
</html>
```

Note that the PHP as well as some HTML are inside the <div> tag so the CSS for the div applies to the HTML and the PHP.

The terminal window also shows the PHP code being executed:

```
$varFirst = "Ann "; // assigns Ann to $varFirst
$varLast = "Smith"; // assigns Smith to $varSecond
echo "Hello ", $varFirst, $varLast;
?>
</div>
</body>
</html>
```



A screenshot of a Windows desktop environment. In the center, a Mozilla Firefox window displays a web page titled "Hello" with the text "Hello Ann Smith" repeated twice. Below the browser is a code editor window showing the corresponding HTML, CSS, and PHP code. To the left of the code editor is another window showing the source code of the same page. At the bottom of the screen is a taskbar with various icons.

**Source of http://ciseb.bristolcc.edu/~pgrocer/CIS1595/HelloTest2.php**

```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<title>Hello</title>
<style type="text/css">
h2
{
    text-align: center;
}
div
{
    border-color: green;
    border-style: groove;
    border-width: 2px;
    font-weight: bold;
    font-size: 16pt;
    color: green;
    text-align: center;
}
</style>
</head>
<body>
<h2>Hello</h2>
<div>
Hello Ann Smith<br>Hello Ann Smith</div>
</body>
</html>
```

**Hello**

Hello Ann Smith  
Hello Ann Smith

```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<title>Hello</title>
<style type="text/css">
h2
{
    text-align: center;
}
div
{
    border-color: green;
    border-style: groove;
    border-width: 2px;
    font-weight: bold;
    font-size: 16pt;
    color: green;
    text-align: center;
}
</style>
</head>
<body>
<h2>Hello</h2>
<div>
$varFirst = "Ann ";      // assigns Ann to $varFirst
$varLast = "Smith";      // assigns Smith to $varSecond
echo "Hello ", $varFirst, $varLast;
echo "<br>Hello $varFirst $varLast";
?
</div>
</body>
</html>
```

Two echos. One with the comma separator and one with the literals and variables enclosed in double quotes.

Makes information about the installed PHP available.

PHP Version 5.5.7	
<b>System</b>	Linux canewb.britolcc.edu 2.6.9-103.ELamp #1 SMP Fri Nov 11 14:34:02 EST 2011 x86_64
<b>Build Date</b>	Jan 2 2014 12:42:59
<b>Configure Command</b>	'./configure' '--prefix=/opt/php-5.5.7' '--with-config-file-path=/opt/php-5.5.7/lib' '--with-apxs2' '--with-mysqli' '--with-openssl' '--with-mcrypt' '--with-pcre-regex' '--with-png-dir=/usr/lib' '--with-jpeg-dir=/usr/lib' '--with-gd' '--with-freetype-dir=/var' '--enable-gd-native-ttf' '--with-pdo-mysqli=/opt/mysql/5.5.22-linux2.6-686' '--with-mysqli'
<b>Server API</b>	Apache 2.0 Handler
<b>Virtual Directory Support</b>	disabled
<b>Configuration File (php.ini) Path</b>	/opt/php-5.5.7/lib
<b>Loaded Configuration File</b>	/opt/php-5.5.7/lib/php.ini
<b>Scan this dir for additional .ini files</b>	(none)
<b>Additional .ini files parsed</b>	(none)
<b>PHP API</b>	20121113
<b>PHP Extension</b>	20121212
<b>Zend Extension</b>	20121212
<b>Zend Extension Build</b>	API20121212,NTS
<b>PHP Extension Build</b>	API20121212,NTS
<b>Debug Build</b>	no
<b>Thread Safety</b>	disabled
<b>Zend Signal Handling</b>	disabled
<b>Zend Memory Manager</b>	enabled
<b>Zend Multithread</b>	disabled

The screenshot shows a Windows desktop environment. In the center is a Mozilla Firefox browser window titled "Pass data - Mozilla Firefox". The address bar shows "http://ciseweb.bristolcc.edu/~pgrocer/CIS1595/2015/FirstGroupPHP/passtest.html". The page content is a simple HTML document with a title "Pass data" and a paragraph "Passing data through the href". Below the browser is a terminal window titled "Source of http://ciseweb.bristolcc.edu/~pgrocer/CIS1595/2015/FirstGroupPHP/passtest.html - M...". It displays the same HTML code. To the right of the terminal is a file explorer window showing the file "passtest.html" in the directory "/home/pgrocer/public\_html/CIS1595/2015/FirstGroupPHP". A blue note overlay on the desktop states: "Note that this is an html page - it contains only html." At the bottom of the screen is a taskbar with various icons and a system tray indicating the date and time as "1/25/2015 2:51 PM".

Passing data through the href

Passing that the school is BCC

```
<html>
<head><title>Pass data</title></head>
<body>
<h3>Passing data through the href</h3>
<a href="receive.php?school=BCC">Passing that the school is BCC</a>
</body>
</html>
```

Note that this is an html page - it contains only html.

```
<html>
<head><title>Pass data</title></head>
<body>
<h3>Passing data through the href</h3>
<a href="receive.php?school=BCC">Passing that the school is BCC</a>
</body>
</html>
```

In this example, I am linking to the program receive.php and using the ? school=BCC I end up sending BCC to the receive.php to be stored in a variable called school.  
This happens when I click on the link.

Note that when I click on the link, receive.php?school=BCC telling me that the receive.php was linked to and the assignment of BCC to school was passed.

```
<html>
<head><title>Receive data from href</title></head>
<body>
<?php
$school = $_GET['school']; //Note the use of the get because the data was passed without a form using the URL
echo "Welcome to $school!";
?>
</body>
</html>
```

school=BCC was processed using the `$_GET` and BCC was assigned to the variable `$school`.  
Then the echo displayed the literal and the contents of `$school`.

This is one approach, we will look at another technique that is more frequently used next.

This is the form

Name:   
Grade:

I have set up an HTML page with a form to collect data.  
Within the form, I use action to tell what page to load next and specify that  
I will be using the get method to handle the data being passed from the  
form. In this case when I click on submit, the action will be executed and  
the data named stuname and the data named grade will be sent.

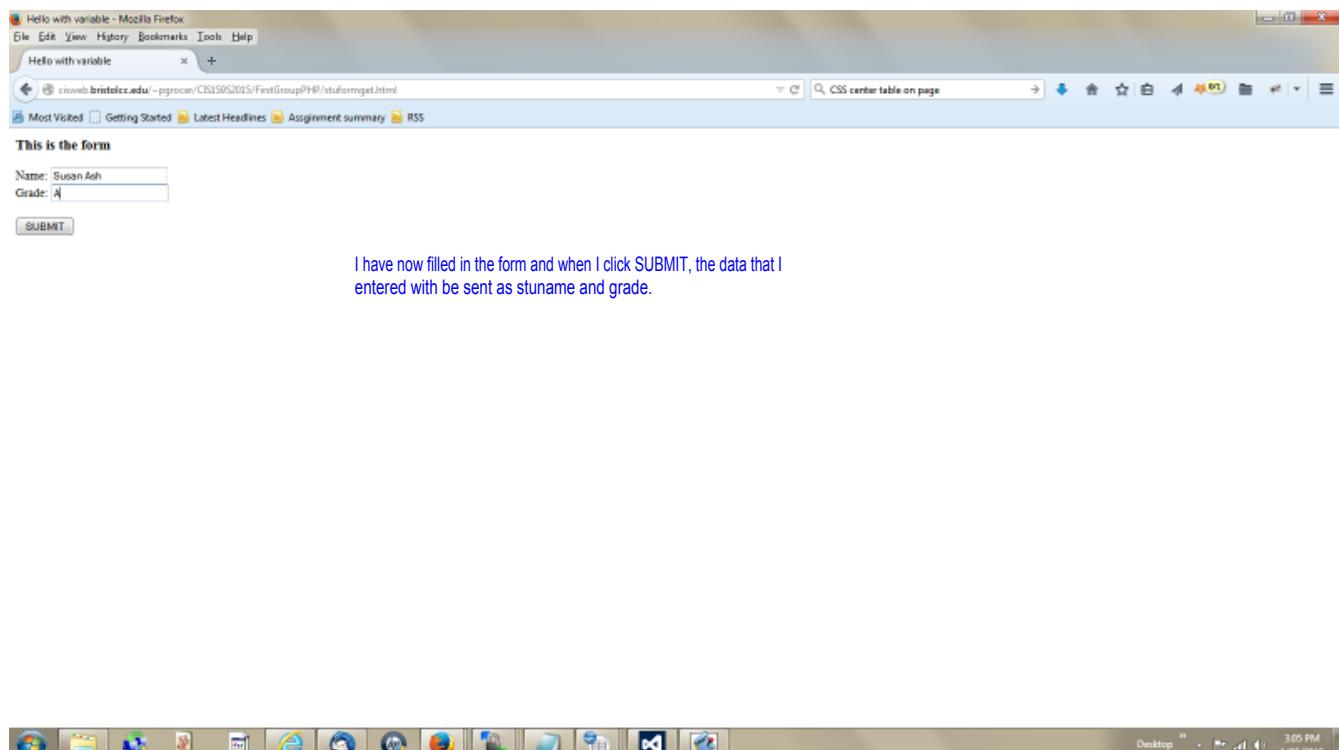
Source of: http://cnweb.bmstlcc.edu/~pgrocer/CIS150S2015/FirstGroupPHP/stuformget.html - Mozilla Firefox

```
1 <html>
2 <head><title>Hello with variables</title></head>
3 <body>
4 <h3>This is the form</h3>
5 <form action="formuseget.php" method="get">
6 Name: <input type="text" name="stuname"><br>
7 Grade: <input type="text" name="grade"><br><br>
8 <input type="submit" value="SUBMIT">
9 </form>
10 </body>
11 </html>
12
```

/home/pgrocer/public\_html/CIS150S2015/FirstGroupPHP/stuformget.html - pgrocer@cnweb.bmstlcc.edu

```
<html>
<head><title>Hello with variables</title></head>
<body>
<h3>This is the form</h3>
<form action="formuseget.php" method="get">
Name: <input type="text" name="stuname"><br>
Grade: <input type="text" name="grade"><br><br>
<input type="submit" value="SUBMIT">
</form>
</body>
</html>
```

Note the page that will be loaded next is formuseget.php



Using the get method I can see the data being sent - great for testing.

Student Name: Susan Ash Note the receiving of the data sent using the get method. The parts of the name have a + between them and there is an & before the grade.

Student Grade: A

The resolved information received by the browser.

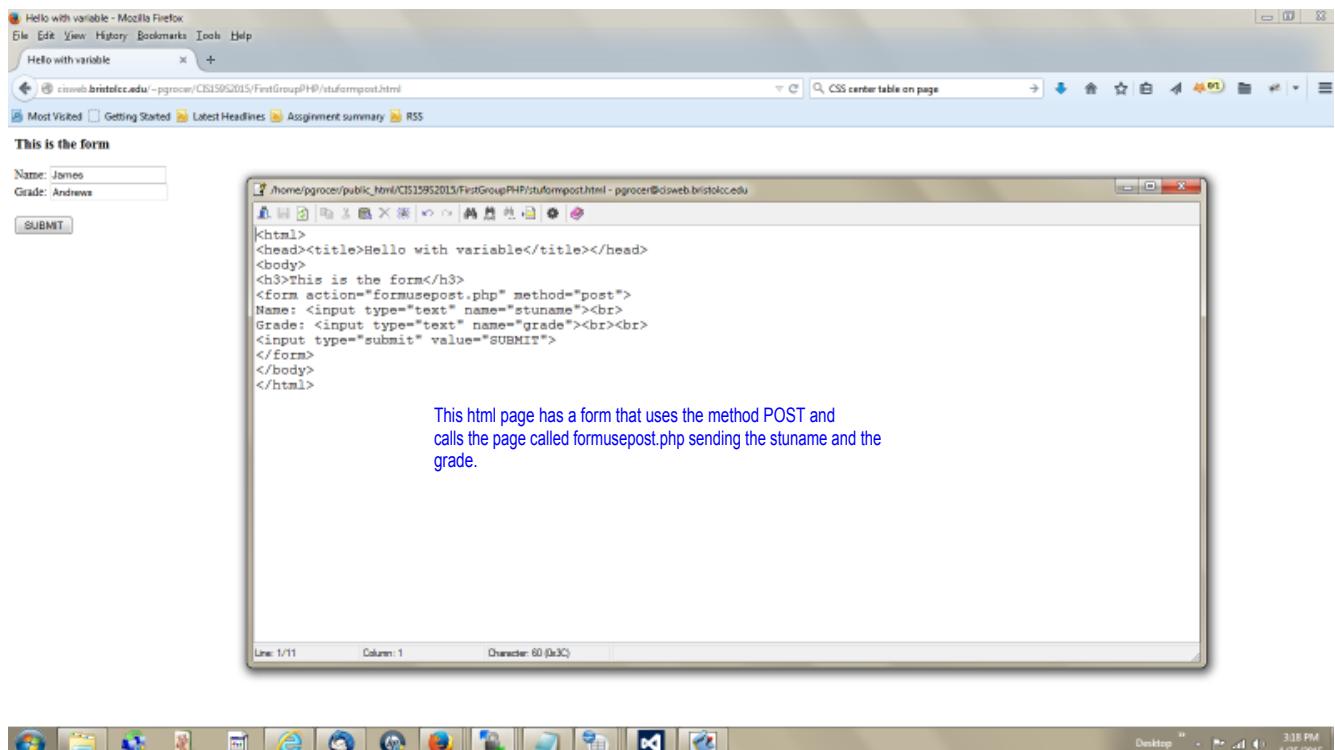
Source of: http://cineab.bristolcc.edu/~pgrocer/CIS15952015/FirstGroupPHP/formusingget.php

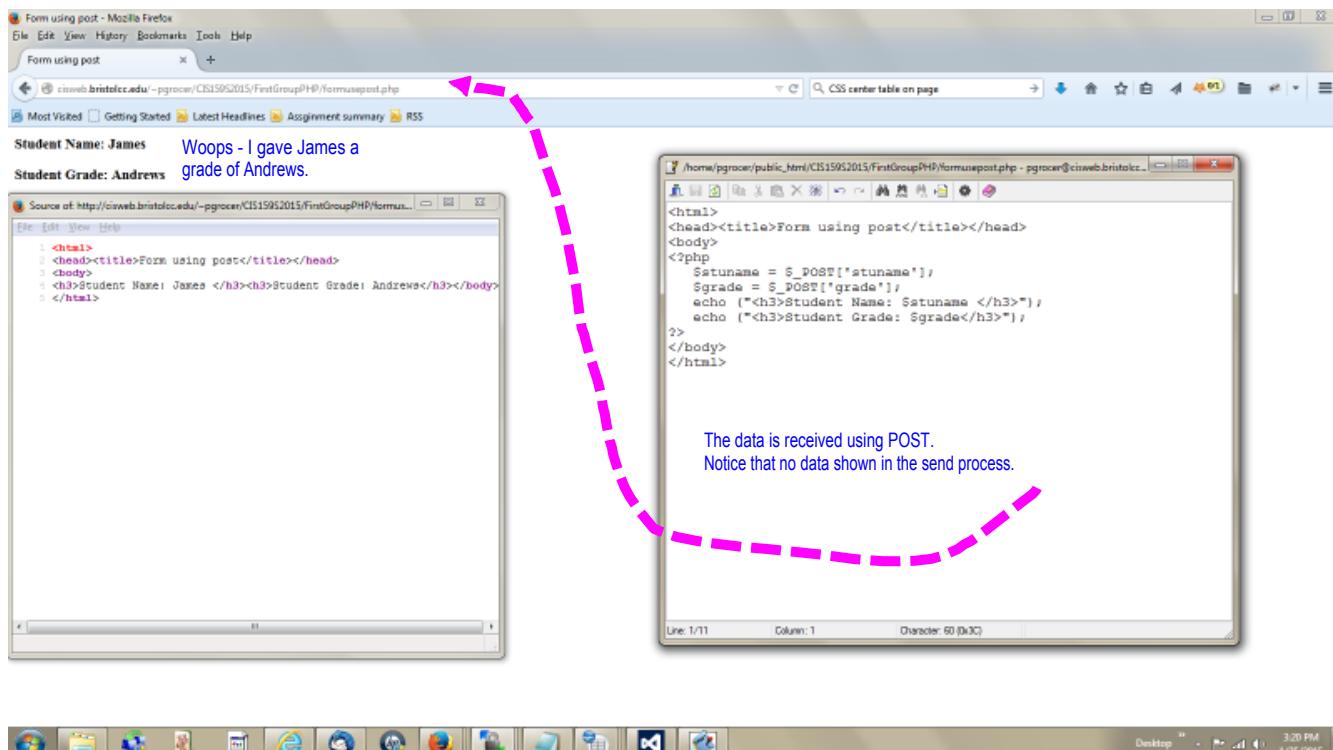
```
<html>
<head><title>Form using get</title></head>
<body>
<h3>Student Name: Susan Ash </h3><h3>Student Grade: A</h3></body>
</html>
```

The stuname that was sent using the GET method is assigned to the variable \$stuname and the grade is similarly assigned to \$grade. Then the echo displays the literal and the variable inside an <h3>.

Note that the parenthesis with the echo are optional.

Desktop 3:08 PM 1/25/2015





This is the form

Name: James Andrews  
Grade: A

SUBMIT

Student Name: James Andrews  
Student Grade: A

```
<html>
<head><title>Form using post</title>
</head>
<body>
<?php
    $stuname = $_POST['stuname'];
    $grade = $_POST['grade'];
    echo ("<h3>Student Name: $stuname
</h3>");
    echo ("<h3>Student Grade: $grade
</h3>");
?>
</body>
</html>
```