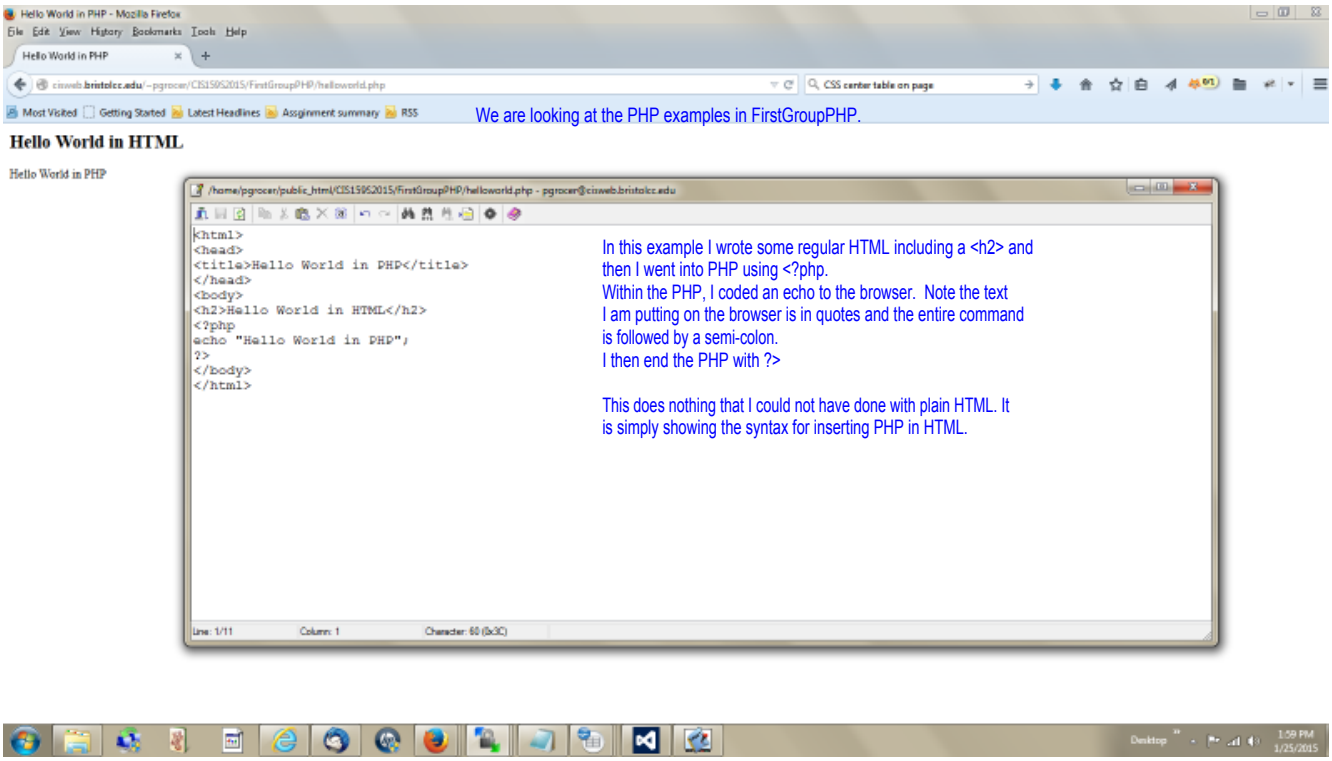
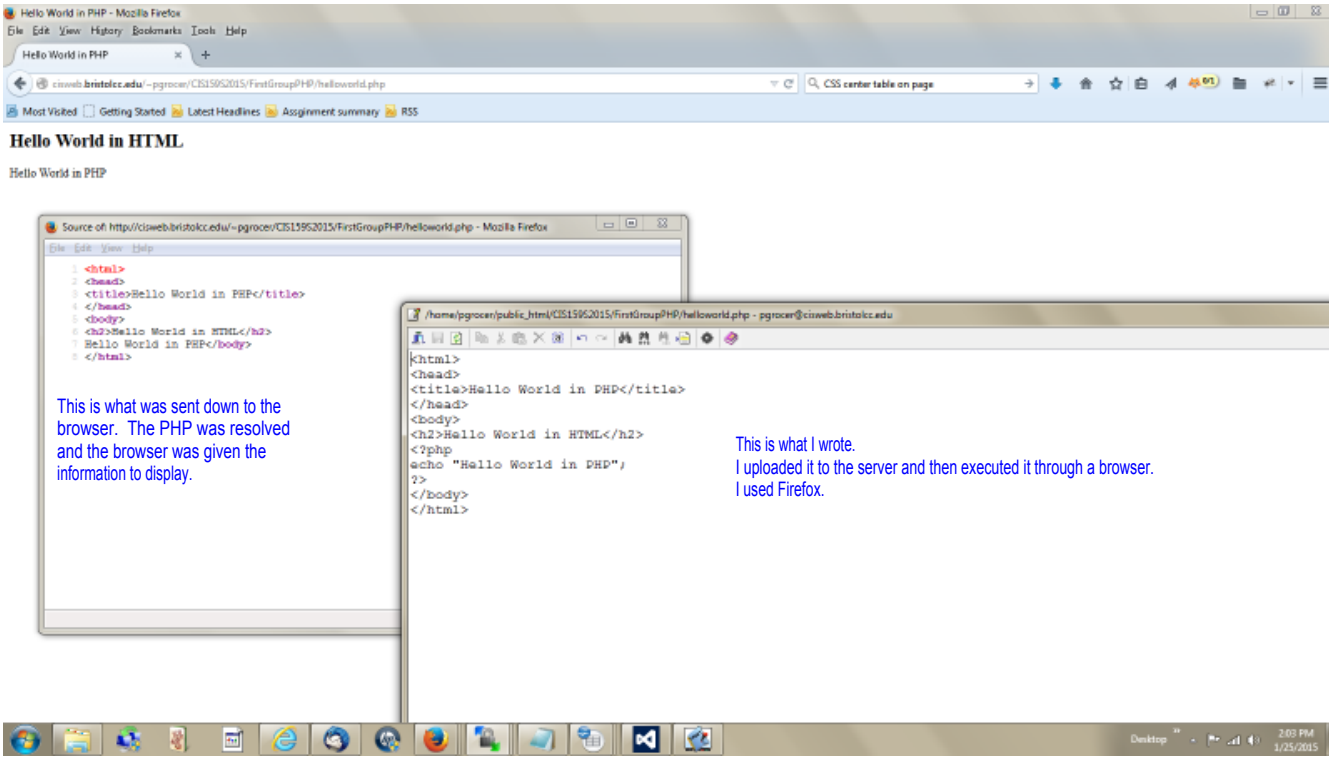
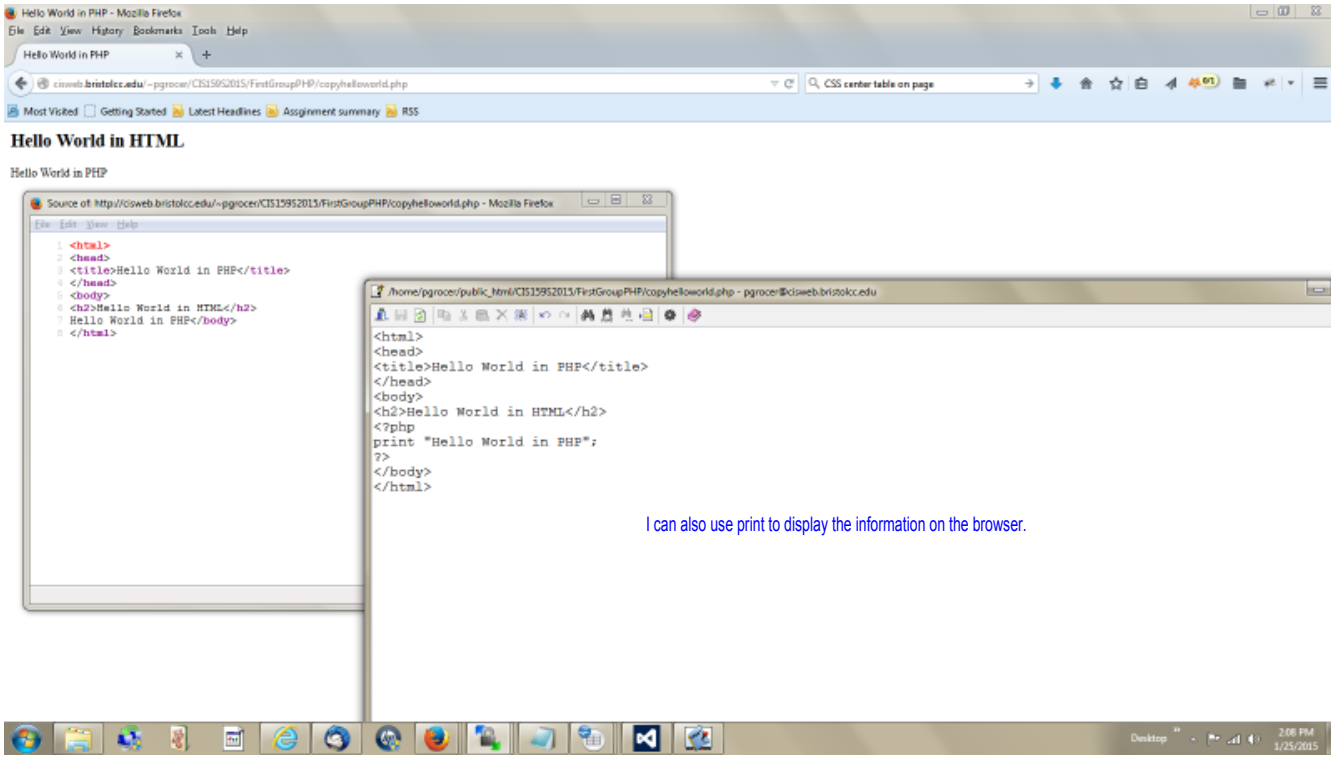


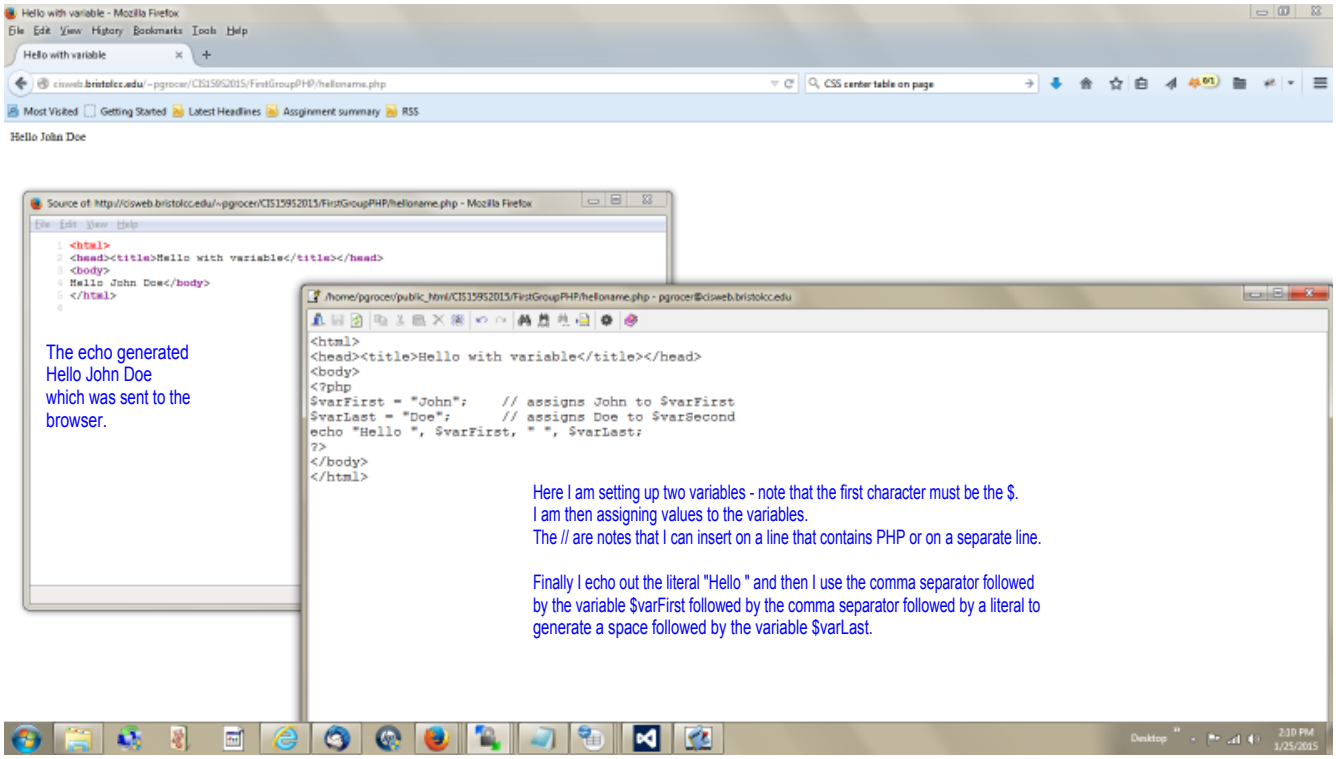
We are looking at the PHP examples in FirstGroupPHP.

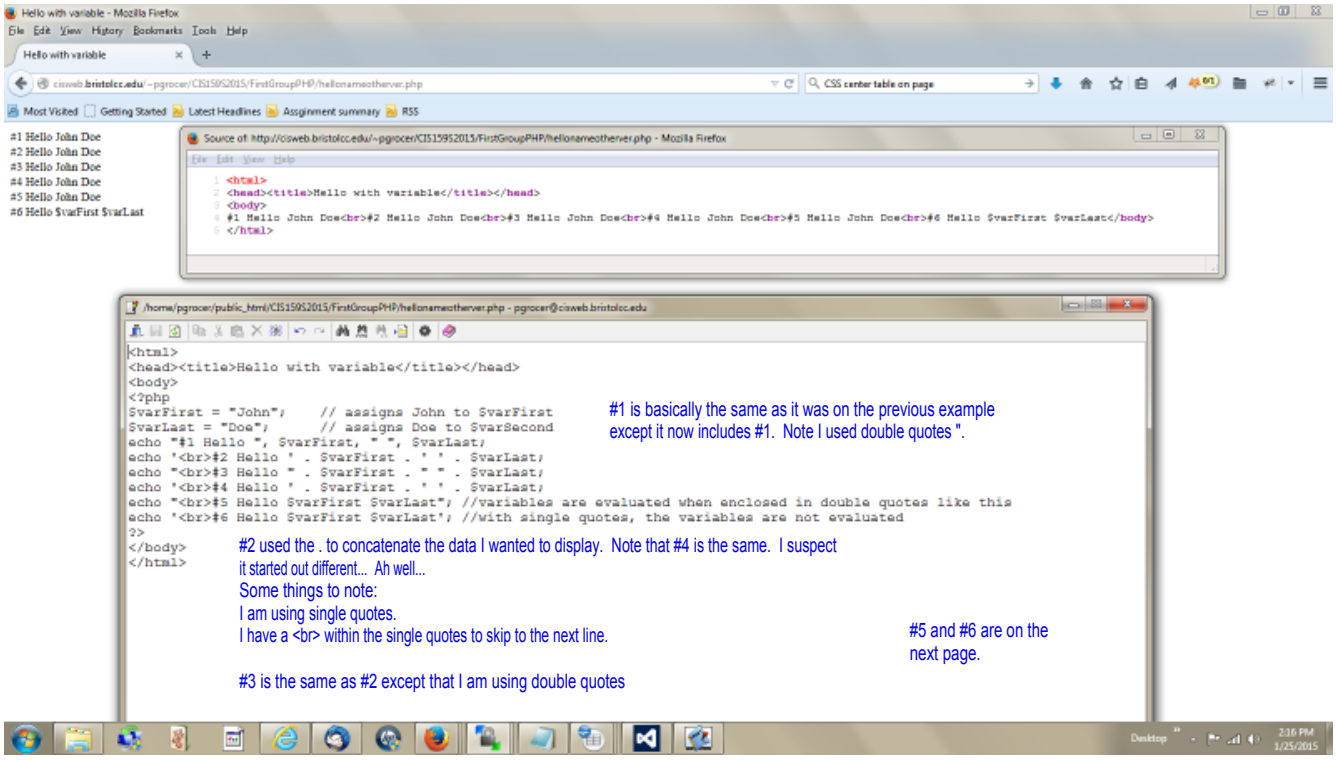
In this example I wrote some regular HTML including a <h2> and then I went into PHP using <?php.
Within the PHP, I coded an echo to the browser. Note the text I am putting on the browser is in quotes and the entire command is followed by a semi-colon.
I then end the PHP with ?>

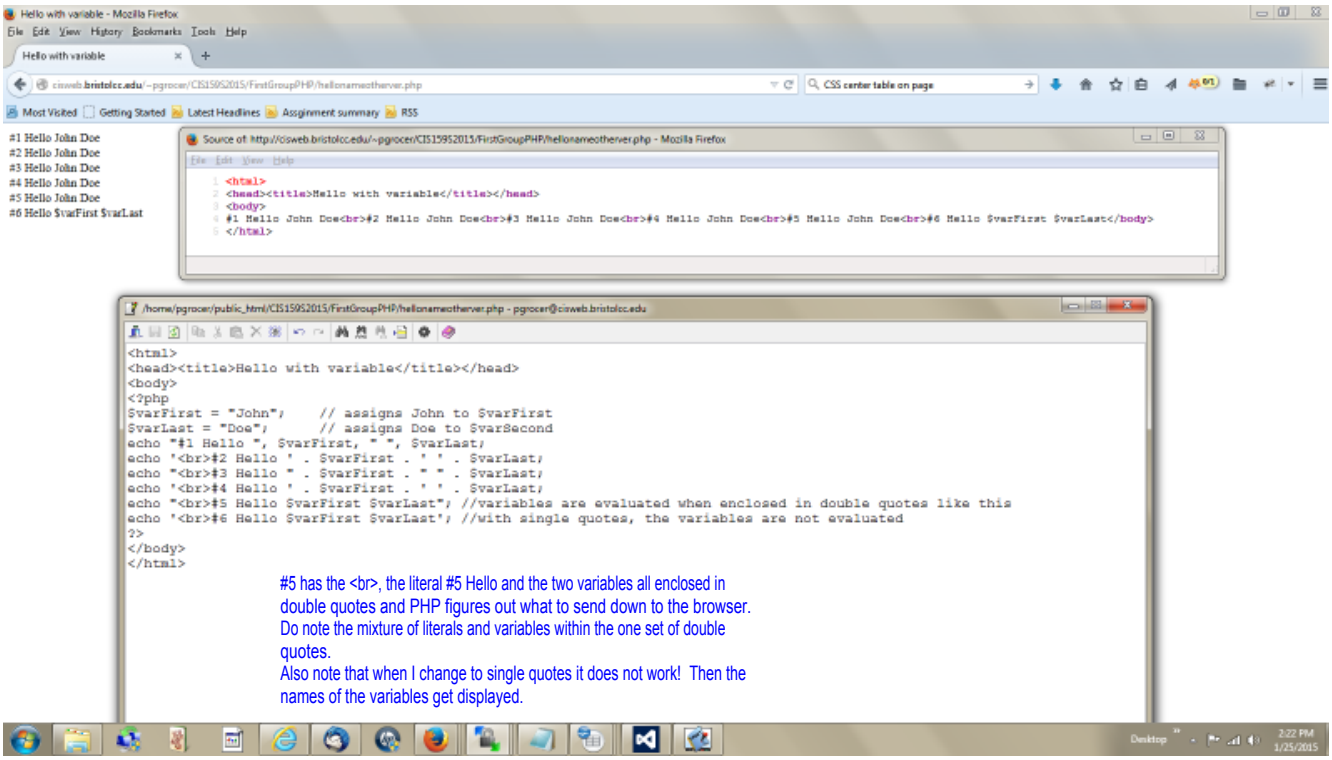
This does nothing that I could not have done with plain HTML. It is simply showing the syntax for inserting PHP in HTML.

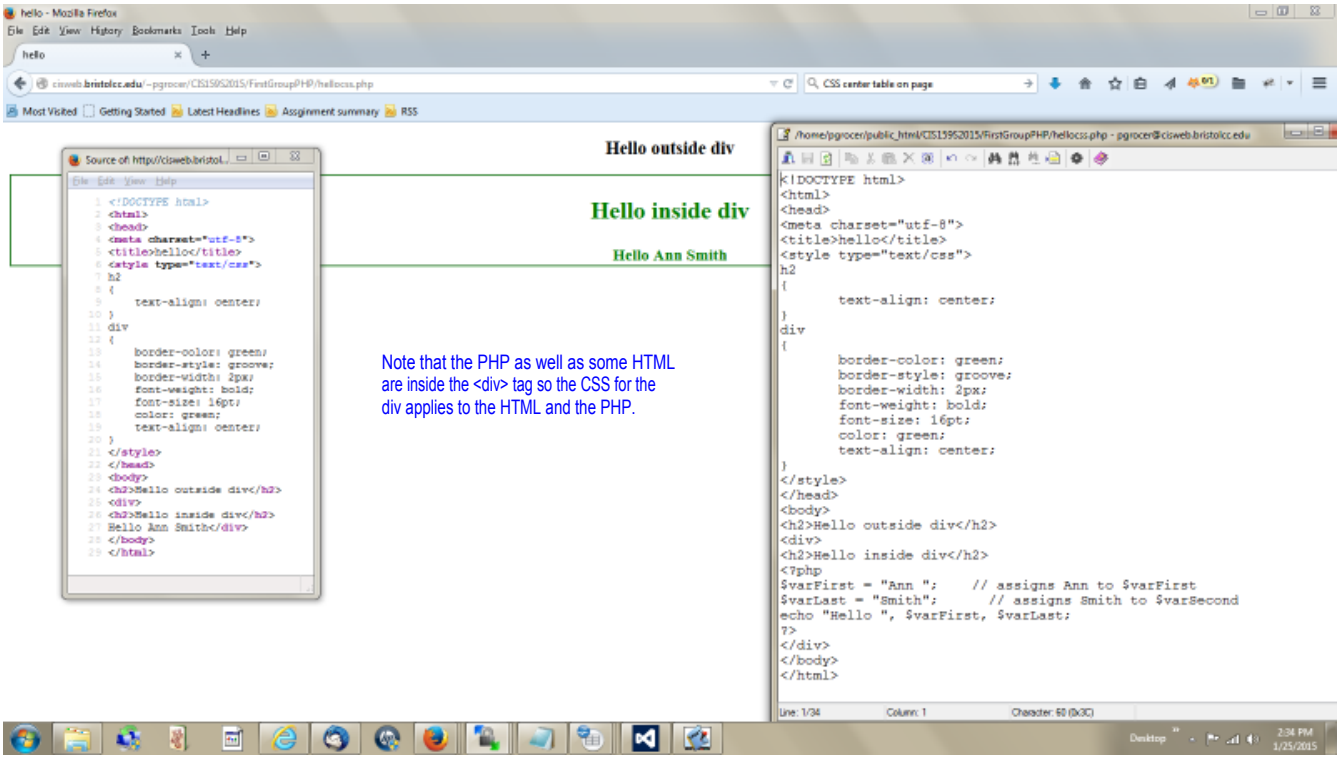

Hello World in PHP - Mozilla Firefox
Hello World in HTML
Hello World in PHP
Source of: http://ciswebbristolcc.edu/~pgrocer/CIS15952015/FirstGroupPHP/helloworld.php - Mozilla Firefox
<html>
<head>
<title>Hello World in PHP</title>
</head>
<body>
<h2>Hello World in HTML</h2>
Hello World in PHP</body>
</html>
This is what was sent down to the browser. The PHP was resolved and the browser was given the information to display.
/home/pgrocer/public_html/CIS15952015/FirstGroupPHP/helloworld.php - pgrocer@ciswebbristolcc.edu
<html>
<head>
<title>Hello World in PHP</title>
</head>
<body>
<h2>Hello World in HTML</h2>
<?php
echo "Hello World in PHP";
?>
</body>
</html>
This is what I wrote.
I uploaded it to the server and then executed it through a browser.
I used Firefox.

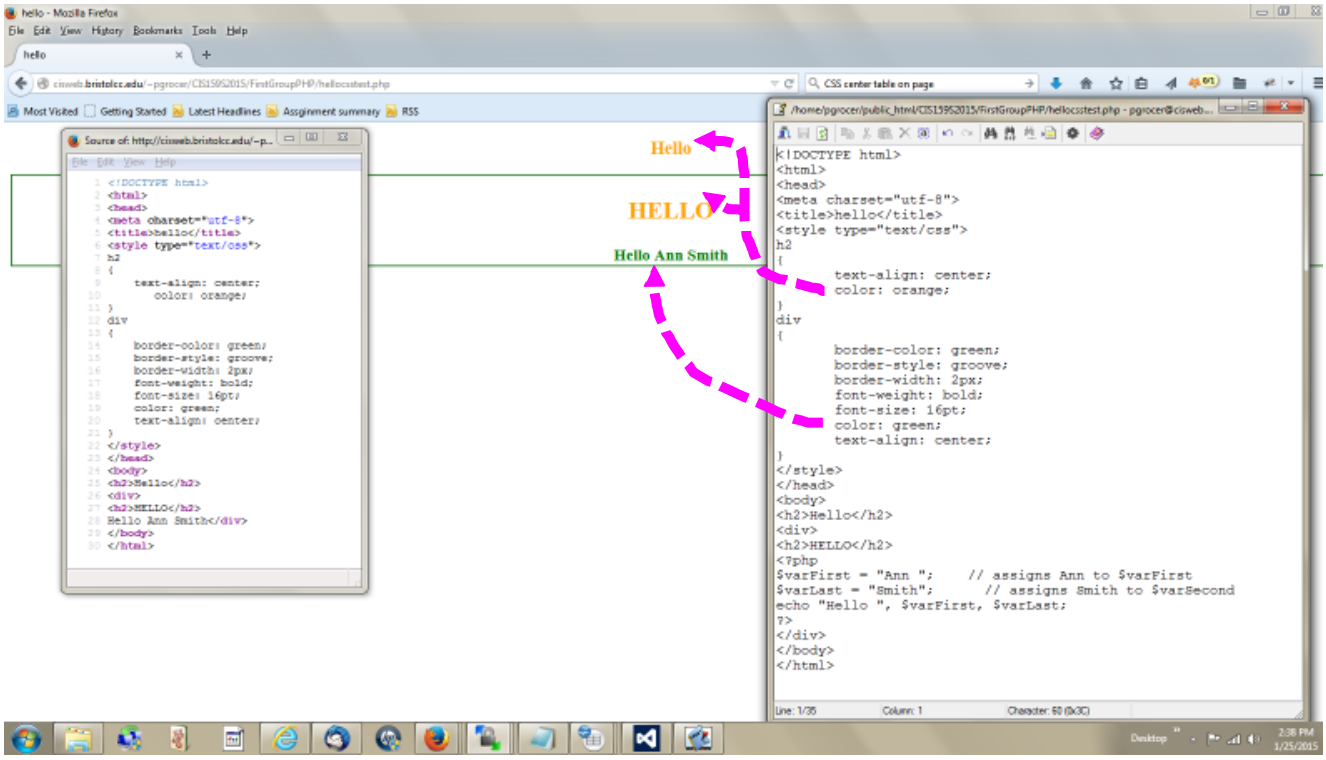

Hello World in PHP - Mozilla Firefox
Hello World in HTML
Hello World in PHP
<html>
<head>
<title>Hello World in PHP</title>
</head>
<body>
<h2>Hello World in HTML</h2>
<?php
print "Hello World in PHP";
?>
</body>
</html>
I can also use print to display the information on the browser.

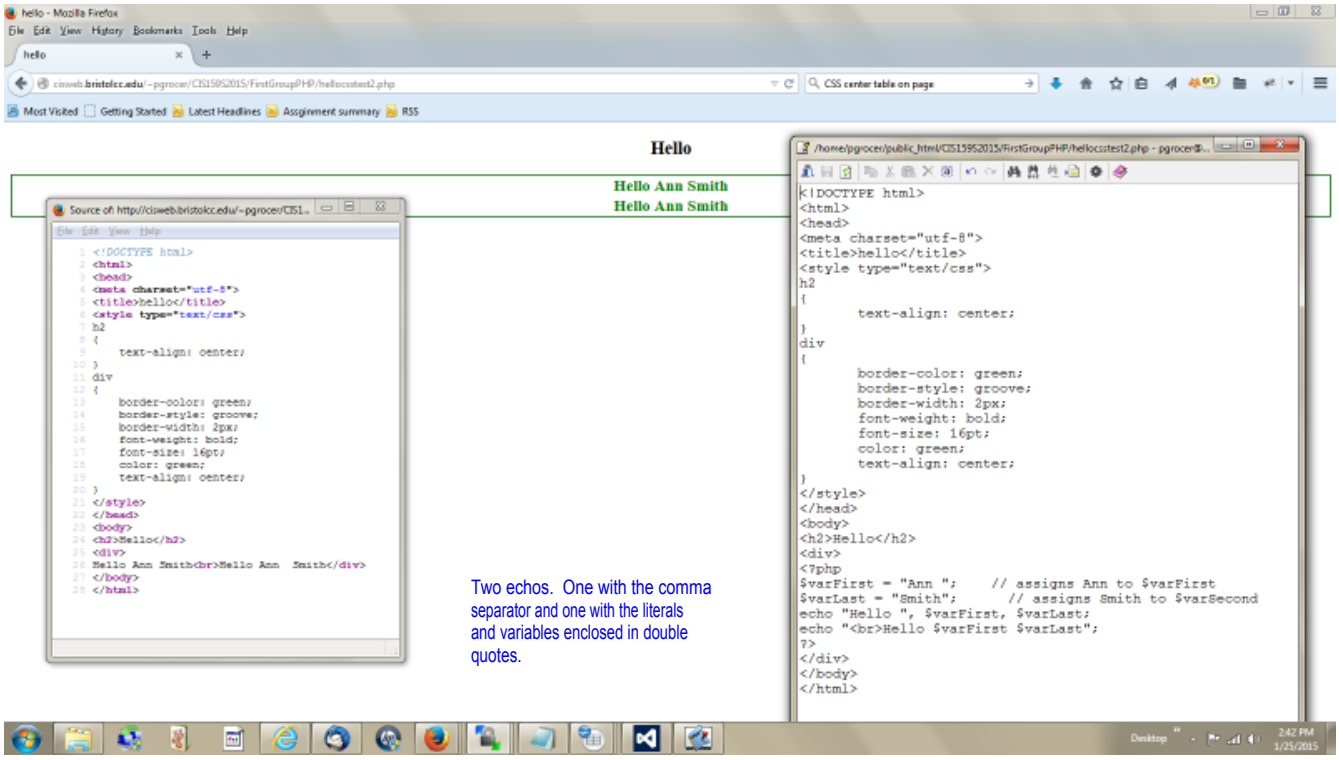

Hello with variable - Mozilla Firefox
Hello John Doe
Source of: http://cisweb.bristolcc.edu/~pgrocer/CIS15952015/FirstGroupPHP/helloname.php - Mozilla Firefox
<html>
<head><title>Hello with variable</title></head>
<body>
Hello John Doe</body>
</html>
The echo generated Hello John Doe which was sent to the browser.
/home/pgrocer/public_html/CIS15952015/FirstGroupPHP/helloname.php - pgrocer@cisweb.bristolcc.edu
<html>
<head><title>Hello with variable</title></head>
<body>
<?php
$varFirst = "John"; // assigns John to $varFirst
$varLast = "Doe"; // assigns Doe to $varSecond
echo "Hello ", $varFirst, " ", $varLast;
?>
</body>
</html>
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.
Finally I echo out the literal "Hello " and then I use the comma separator followed by the variable $varFirst followed by the comma separator followed by a literal to generate a space followed by the variable $varLast.

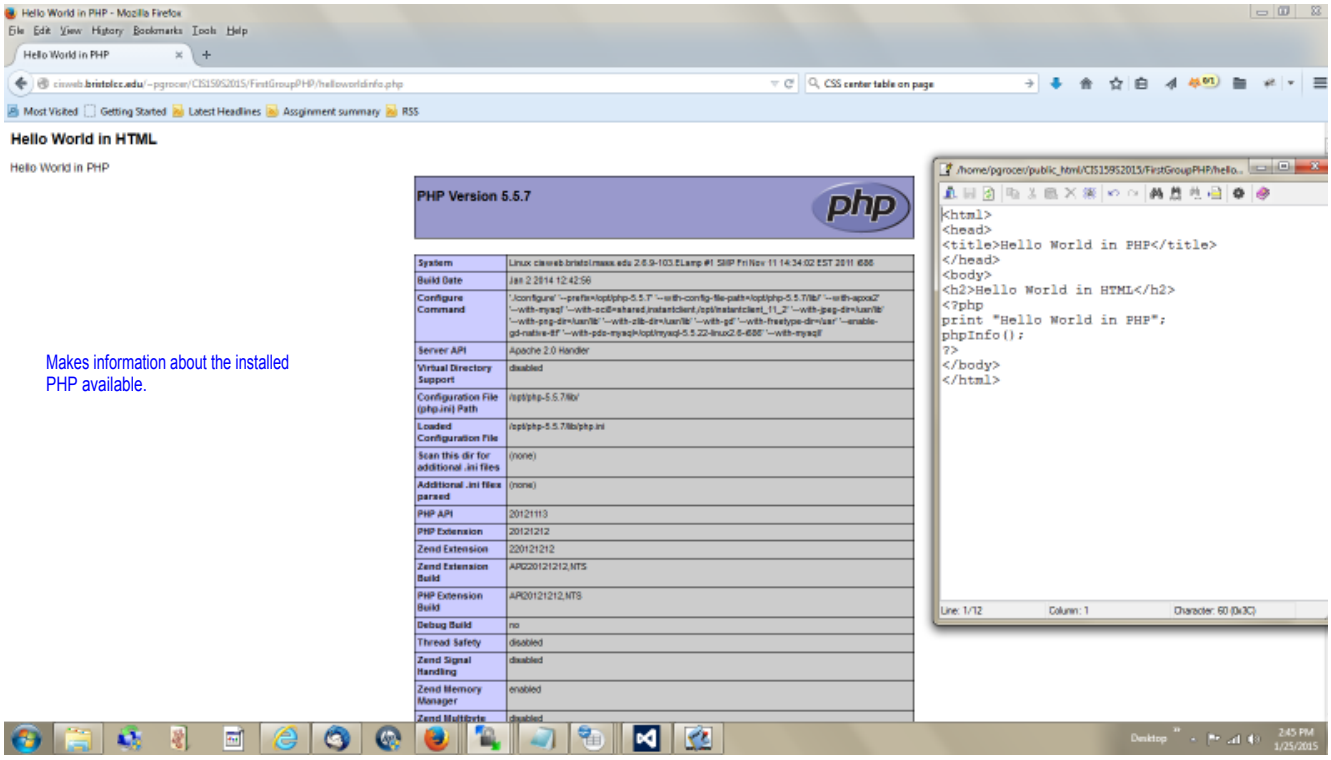

Hello with variable - Mozilla Firefox
#1 Hello John Doe
#2 Hello John Doe
#3 Hello John Doe
#4 Hello John Doe
#5 Hello John Doe
#6 Hello $varFirst $varLast
Source of: http://cisweb.bristolcc.edu/~pgrocer/CIS15952015/FirstGroupPHP/hellonameotherver.php - Mozilla Firefox
<html>
<head><title>Hello with variable</title></head>
<body>
#1 Hello John Doe
#2 Hello John Doe
#3 Hello John Doe
#4 Hello John Doe
#5 Hello John Doe
#6 Hello $varFirst $varLast</body>
</html>
/home/pgrocer/public_html/CIS15952015/FirstGroupPHP/hellonameotherver.php - pgrocer@cisweb.bristolcc.edu
<html>
<head><title>Hello with variable</title></head>
<body>
<?php
$varFirst = "John"; // assigns John to $varFirst
$varLast = "Doe"; // assigns Doe to $varSecond
echo "#1 Hello ", $varFirst, " ", $varLast;
echo '
#2 Hello ' . $varFirst . ' ' . $varLast;
echo "
#3 Hello " . $varFirst . " " . $varLast;
echo '
#4 Hello ' . $varFirst . ' ' . $varLast;
echo "
#5 Hello $varFirst $varLast"; //variables are evaluated when enclosed in double quotes like this
echo '
#6 Hello $varFirst $varLast'; //with single quotes, the variables are not evaluated
?>
</body>
</html>
#1 is 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.


Hello with variable - Mozilla Firefox
#1 Hello John Doe
#2 Hello John Doe
#3 Hello John Doe
#4 Hello John Doe
#5 Hello John Doe
#6 Hello $varFirst $varLast
Source of: http://cisweb.bristolcc.edu/~pgrocer/CIS15952015/FirstGroupPHP/hellonameotherver.php - Mozilla Firefox
<html>
<head><title>Hello with variable</title></head>
<body>
#1 Hello John Doe
#2 Hello John Doe
#3 Hello John Doe
#4 Hello John Doe
#5 Hello John Doe
#6 Hello $varFirst $varLast</body>
</html>
/home/pgrocer/public_html/CIS15952015/FirstGroupPHP/hellonameotherver.php - pgrocer@cisweb.bristolcc.edu
<html>
<head><title>Hello with variable</title></head>
<body>
<?php
$varFirst = "John";   // assigns John to $varFirst
$varLast = "Doe";   // assigns Doe to $varSecond
echo "#1 Hello ", $varFirst, " ", $varLast;
echo "
#2 Hello " . $varFirst . " " . $varLast;
echo "
#3 Hello " . $varFirst . " " . $varLast;
echo "
#4 Hello " . $varFirst . " " . $varLast;
echo "
#5 Hello $varFirst $varLast"; //variables are evaluated when enclosed in double quotes like this
echo '
#6 Hello $varFirst $varLast'; //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.

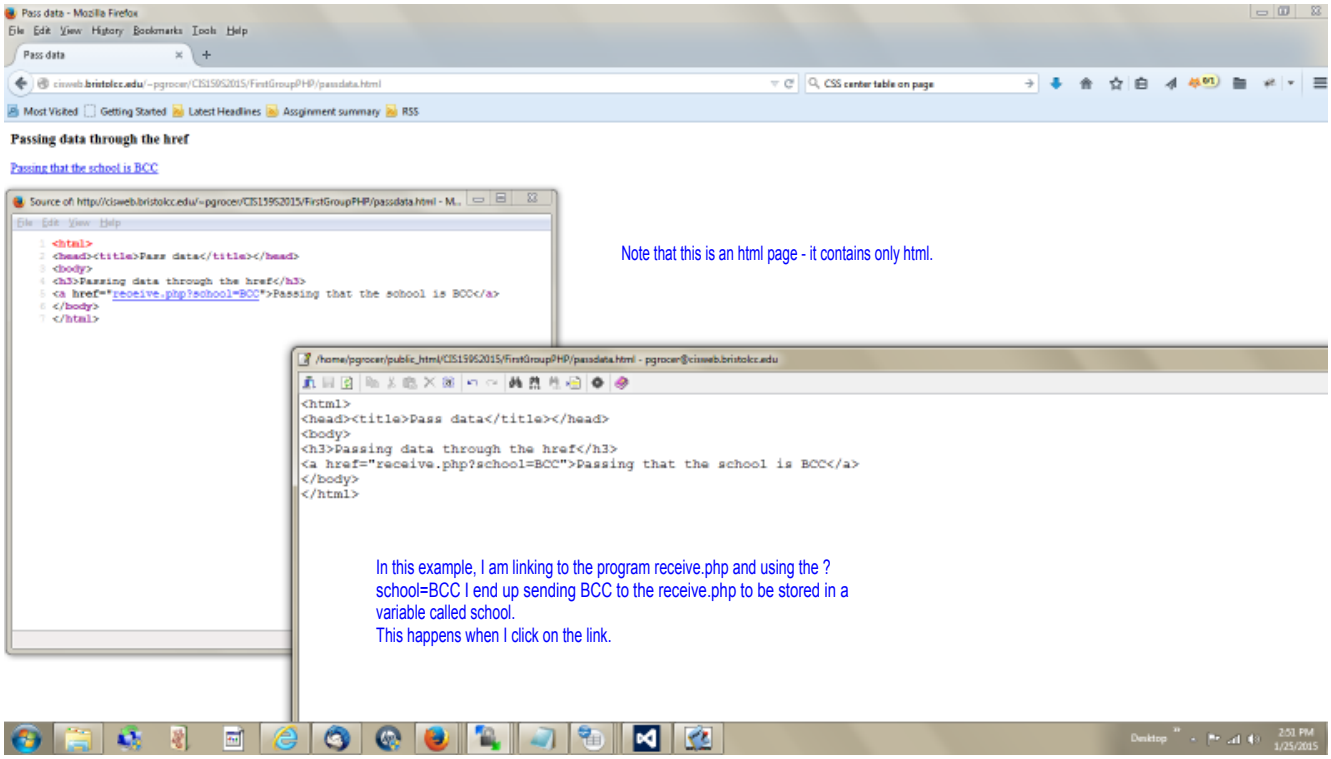

Hello outside div
Hello inside div
Hello Ann Smith
Note that the PHP as well as some HTML are inside the tag so the CSS for the div applies to the HTML and the PHP.


Hello
HELLO
Hello Ann Smith


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

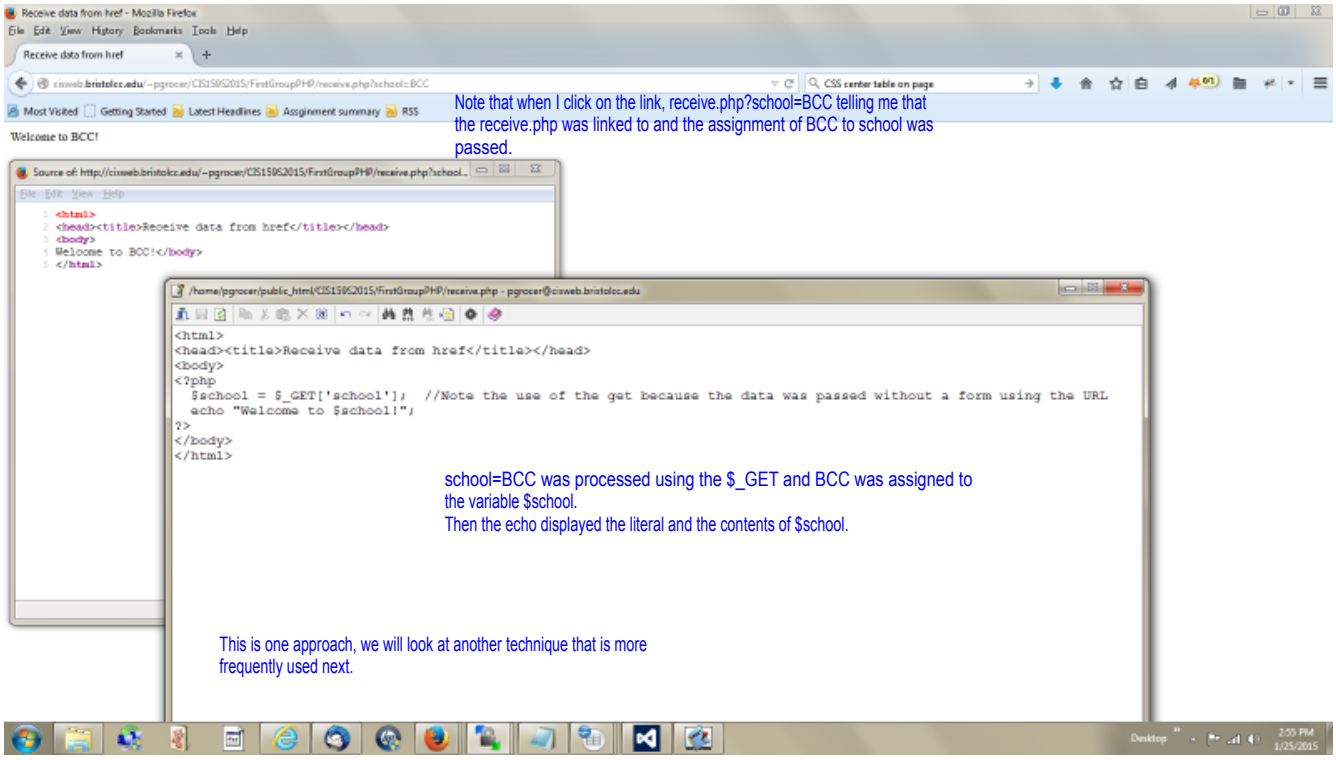


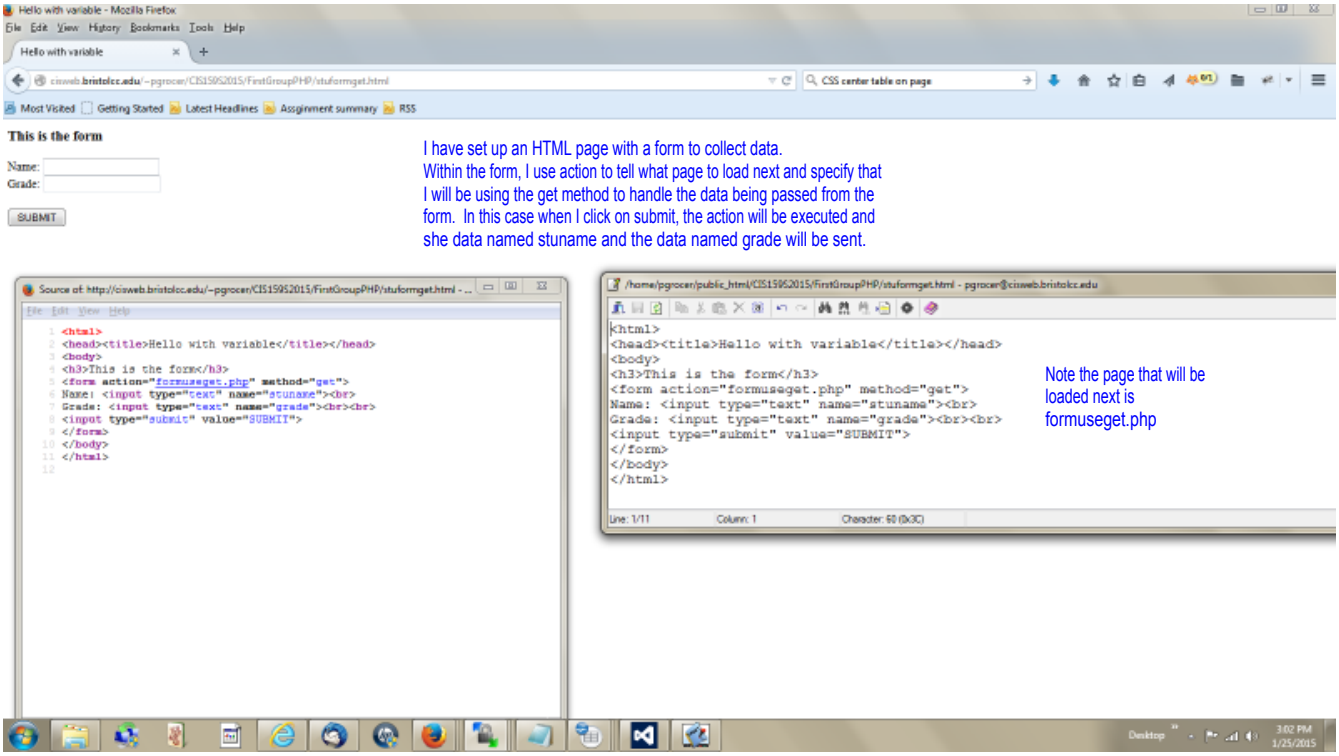
Makes information about the installed PHP available.



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

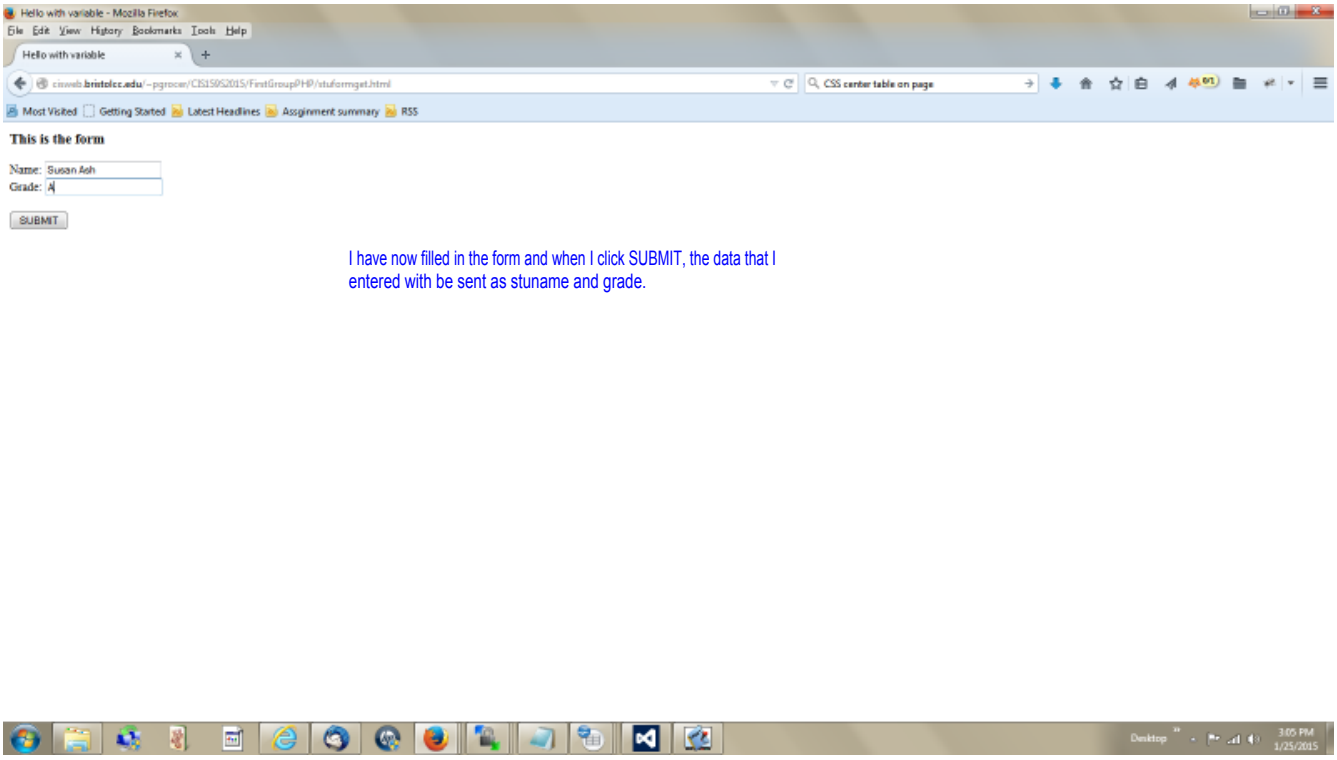

Receive data from href - Mozilla Firefox
Welcome to BCC!
Note that when I click on the link, receive.php?school=BCC telling me that the receive.php was linked to and the assignment of BCC to school was passed.
<html>
<head><title>Receive data from href</title></head>
<body>
Welcome to BCC!</body>
</html>
<?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!";
?>
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.



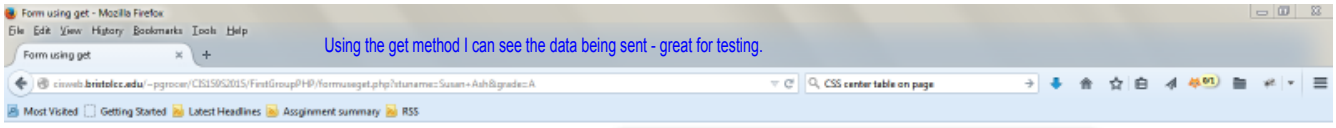
I have set up an HTML page with a form to collect data.
Within the form, I use action to tell what page to load next and specify that I will be using the get method to handle the data being passed from the form. In this case when I click on submit, the action will be executed and she data named stuname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

```
<html>
<head><title>Hello with variable</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



I have now filled in the form and when I click SUBMIT, the data that I entered with be sent as stuname and grade.

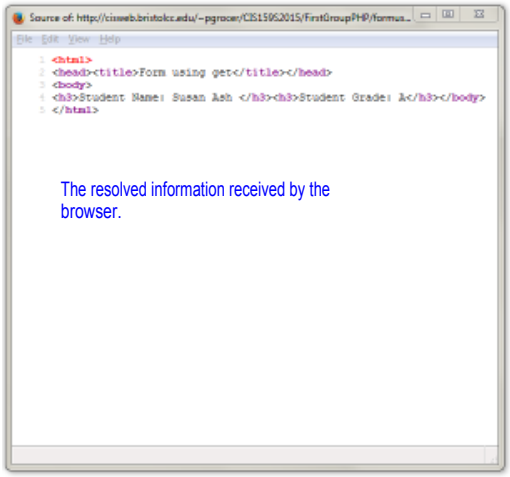


Using the get method I can see the data being sent - great for testing.

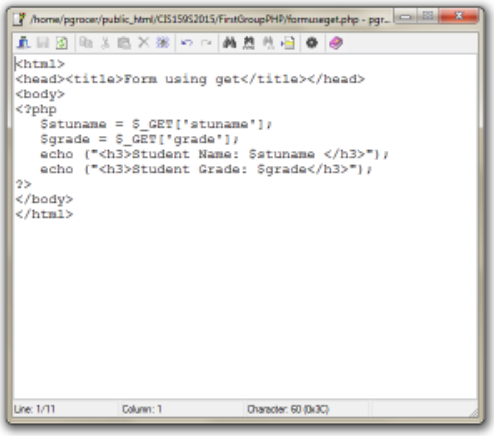
Student Name: Susan Ash

Student Grade: A

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.

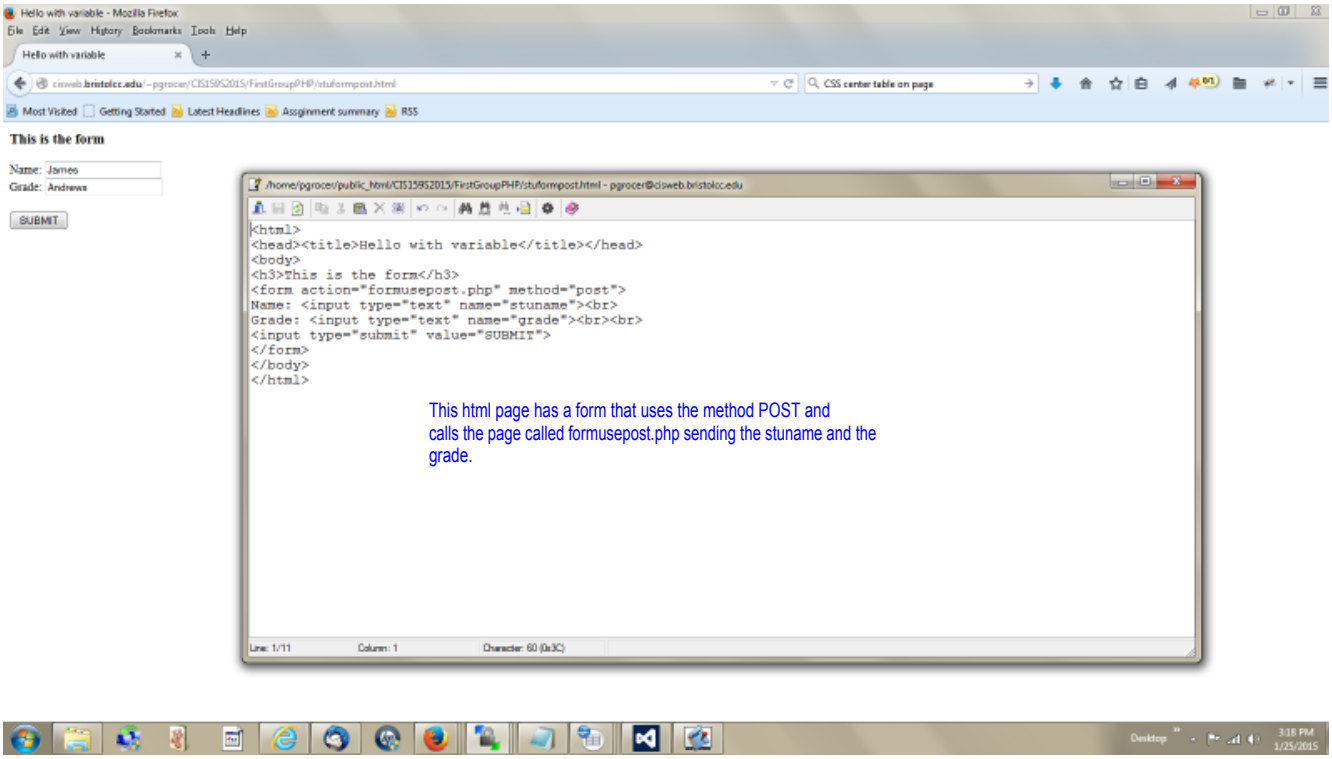


The resolved information received by the browser.

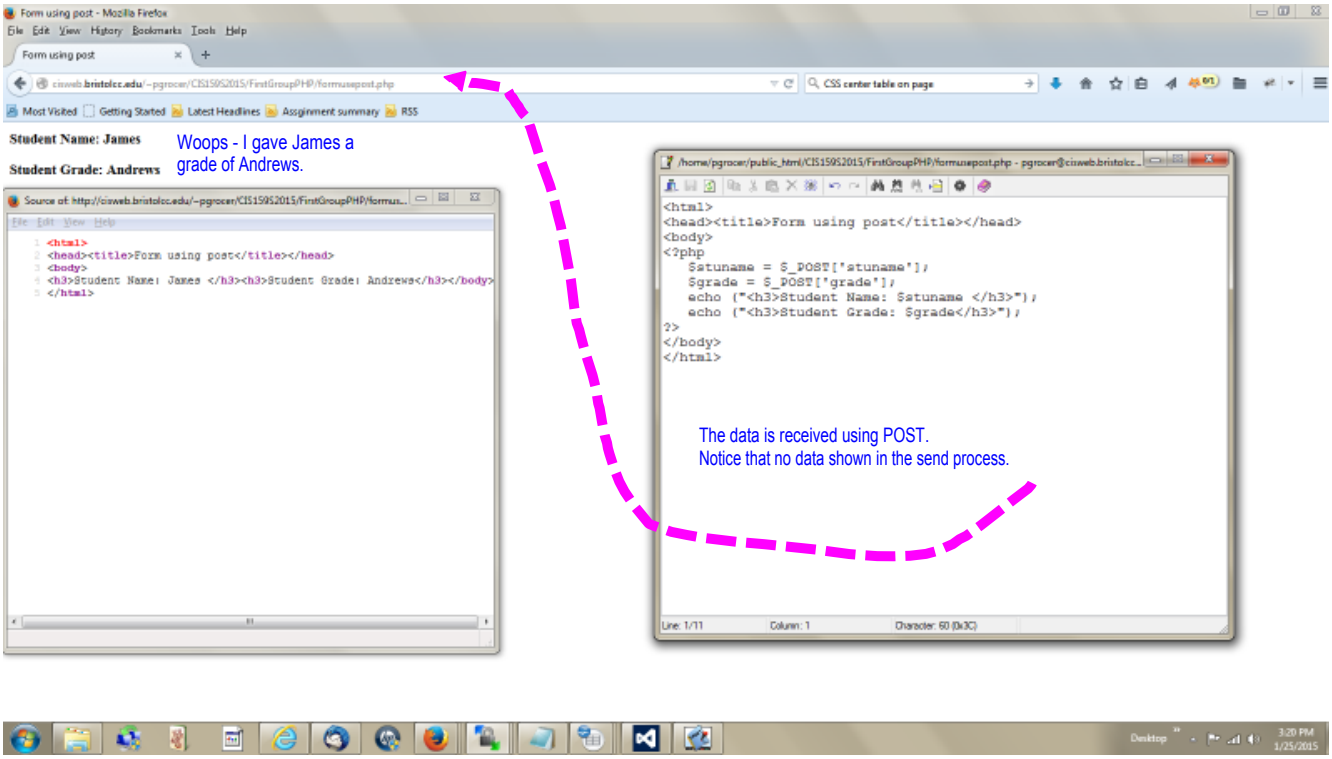


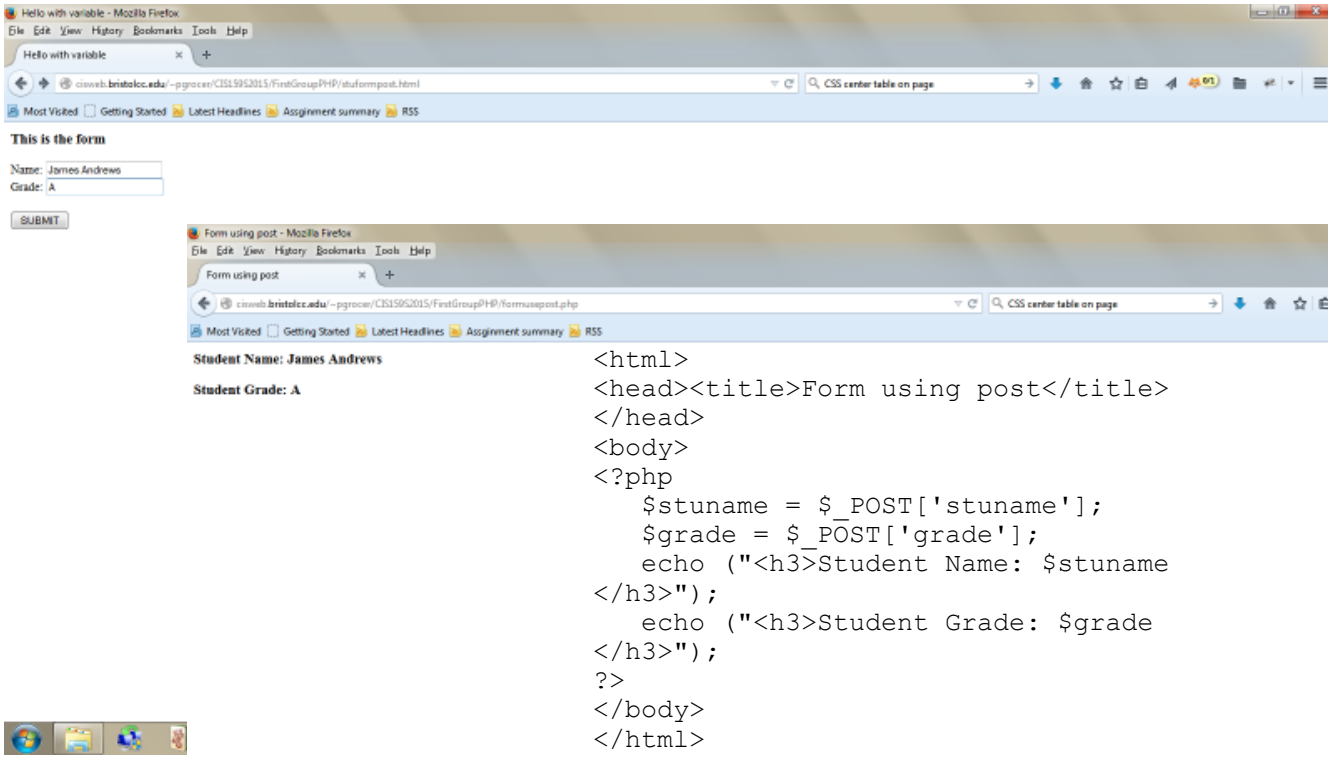
Note that the parenthesis with the echo are optional.

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>.


Hello with variable - Mozilla Firefox
This is the form
Name: James
Grade: Andrews
SUBMIT
/home/pgrocer/public_html/CIS1595S2015/FirstGroupPHP/stuformpost.html - pgrocer@cisweb.bristolcc.edu
<html>
<head><title>Hello with variable</title></head>
<body>
<h3>This is the form</h3>
<form action="formusepost.php" method="post">
Name: <input type="text" name="stuname">
Grade: <input type="text" name="grade">

<input type="submit" value="SUBMIT">
</form>
</body>
</html>
This html page has a form that uses the method POST and calls the page called formusepost.php sending the stuname and the grade.


Form using post - Mozilla Firefox
Student Name: James
Student Grade: Andrews
Woops - I gave James a grade of Andrews.
<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>
The data is received using POST.
Notice that no data shown in the send process.




```
<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>
```