



ASP 101

Course Handbook Supplement

By Richard Rost

Published By

Amicron Computing

PO Box 1308, Amherst NY 14226 USA

www.MYOLP.com

First Printing **12/6/2004**

Copyright 2004 by Amicron Computing

All Rights Reserved



Welcome

Welcome to **ASP 101**.

This handbook is designed to be a **supplement** to the full 599CD video course for **ASP 101**. We recommend you use this handbook to follow along with the class videos. This handbook is not meant as a stand-alone study guide.

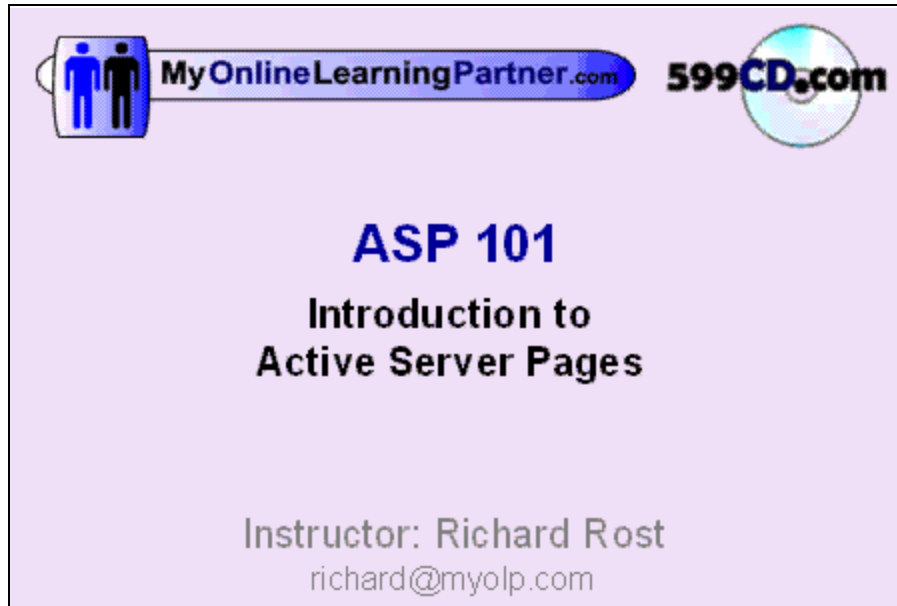
We do recommend that you watch the course videos one time through, paying attention to the lessons covered. Follow along with the course videos using this guide. Take notes on the pages where needed. Then, watch the videos a second time, practicing the examples yourself on your own website.

Table of Contents

Welcome	2
Table of Contents	2
Introduction	3
Definitions	4
Our First Script.....	5
If-Then.....	12
Looping & Variables.....	17
Mathematical Operatoins	20
If-Then-ElseIf-Else.....	22
Processing Form Input	26
Reading the Query String.....	31
Review.....	34

Introduction

Welcome to **ASP 101**, brought to you by **599CD.com**. I am your instructor, Richard Rost.



Objectives for today's class:

- What is ASP (Active Server Pages)
- What you need to use ASP
- Basic ASP commands
- Building simple ASP pages

We will not be discussing database connectivity in today's class. The class follows Windows 101, 102 or 110, and HTML 101. I strongly recommend that you watch all previous classes before you start with this class. We will be **using ASP 3.0 (Windows 2000 Server, IIS 5.0), FrontPage XP, and Windows XP** in this class, but the lessons in this class are really common to all versions of Windows.

We also recommend that you take our Visual Basic 101 and FrontPage 101 classes.

If you need help locating an Internet provider that can give you a website to host your ASP pages, please contact us. We have a list of Internet providers that offer ASP hosting.

Definitions

In the old days, web pages were static. That means the content on them didn't change. Active Server pages allow pages to change. ASP makes pages come alive with dynamic content.

ASP pages are simply HTML pages with a scripting language added to them to include dynamic content.

A script is a series of programming codes embedded in a web page to execute commands. For example, to show the current date and time, or include some data value from a database.

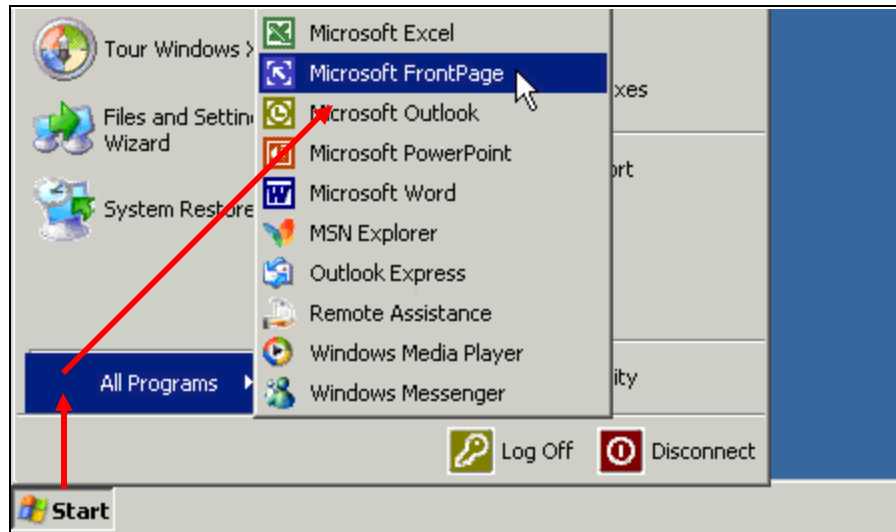
Some scripts, like JavaScript run in the browser. These are called **client-side scripts**. Client-side scripts run in the client, or the browser. The bad thing about client-side scripts is all the scripting code goes to the user's browser. If the user knows how to view the source of a page, he can see how the program is running.

ASP is a **server-side** script. The server interprets the scripts first, and then sends the page to the browser. The browser receives only HTML. If the user were to look at the source code of the page, all he would see is the HTML.

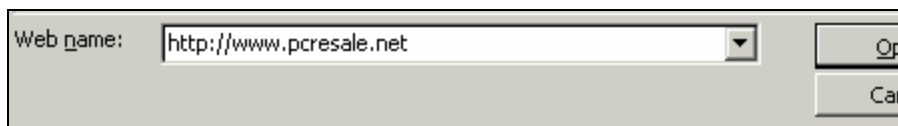
Our First Script

Let's get started today by creating our first ASP script. This will be a test script just to make sure your web server runs ASP pages.

I'm going to start by opening Microsoft FrontPage.



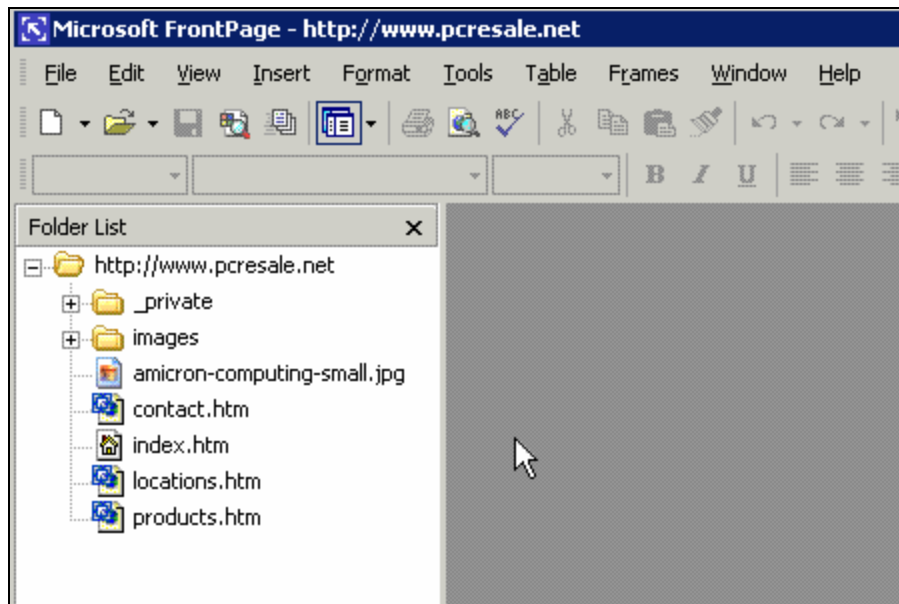
I'm going to click on **File** – and then **Open Web**. Then I'm going to enter my website and then click **Open**:



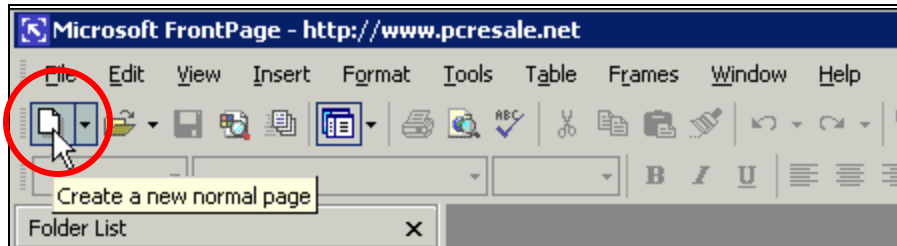
I'll enter in my username and password and then click **OK**:



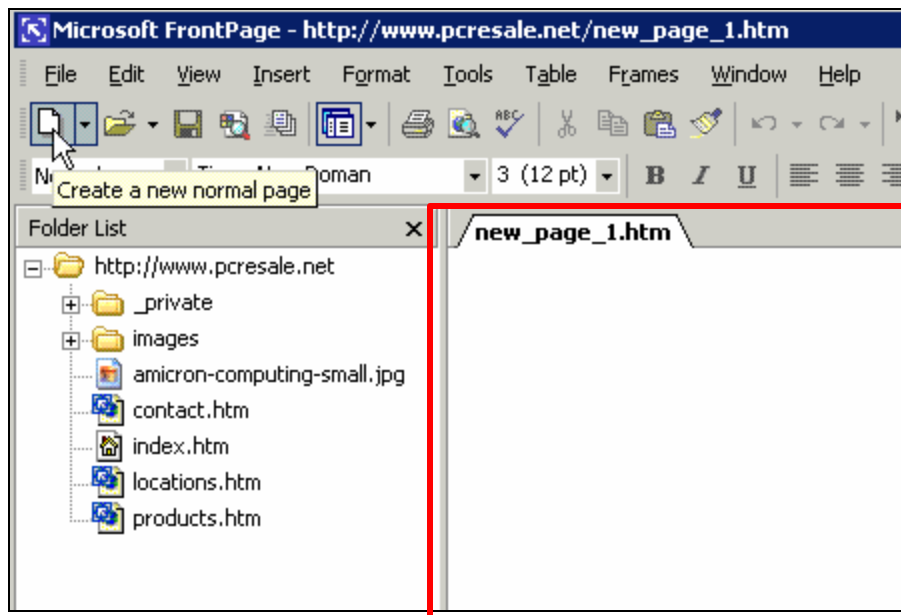
Now here I am in my web.



I'm going to click on the toolbar's blank page to create a new page.



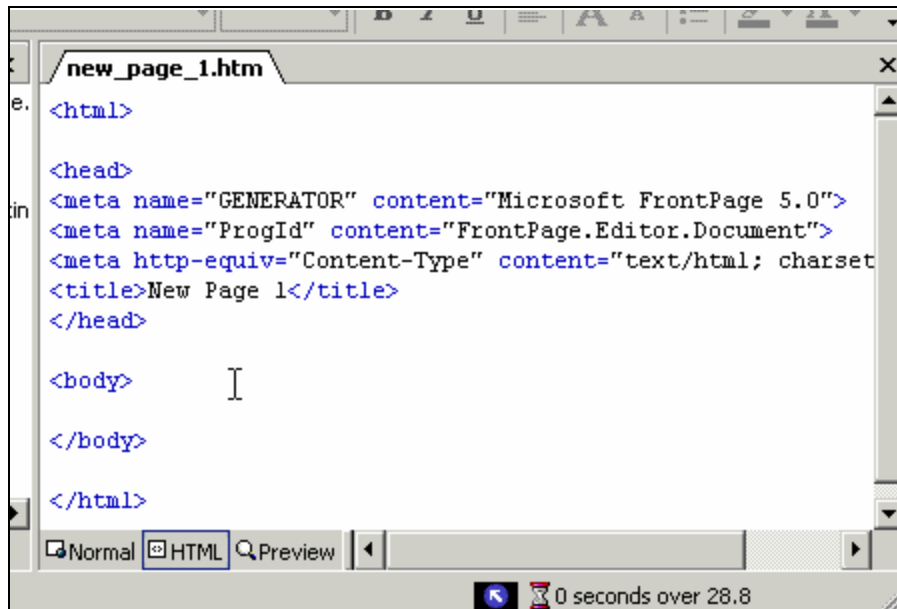
New Page 1 appears.



Let's come down to the very bottom and click on HTML.



This will take us to the HTML view for this particular page.



```
<html>
<head>
<meta name="GENERATOR" content="Microsoft FrontPage 5.0">
<meta name="ProgId" content="FrontPage.Editor.Document">
<meta http-equiv="Content-Type" content="text/html; charset=
<title>New Page 1</title>
</head>
<body>
</body>
</html>
```

For the purpose of this class, I'm going to delete the entire header section of this page (everything from the <head> tag to the </head> tag). All we need to concentrate on will be inside the body tags (<body> and </body>), where I'll type "Hello There."

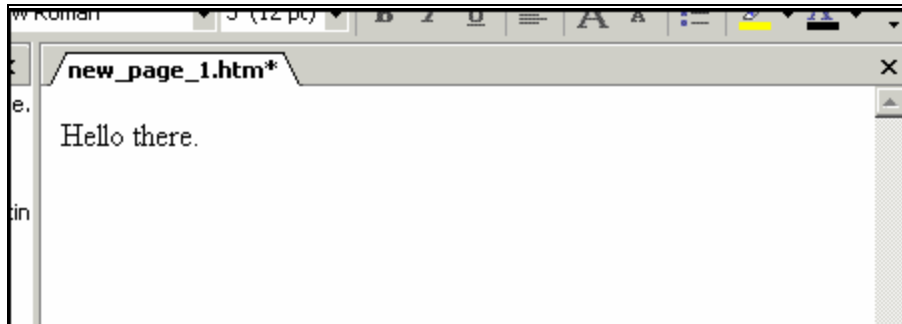


```
<html>
<body>
Hello there.
</body>
</html>
```

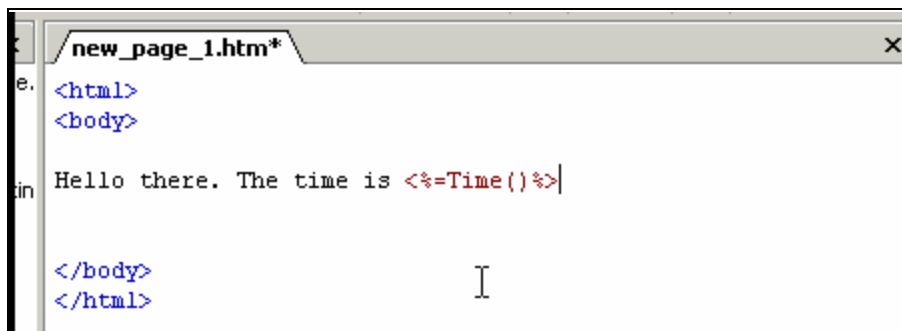
Now I'll go back to the bottom and click on the **Normal** tag.



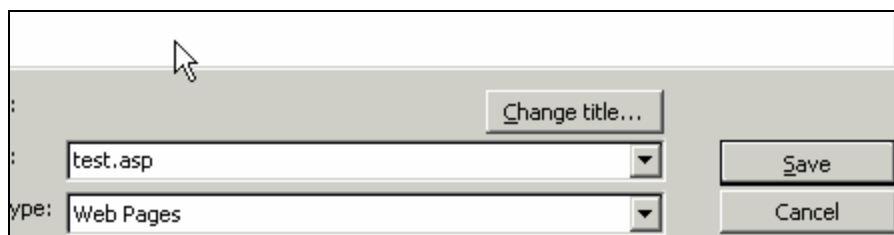
You'll see **Hello There.**



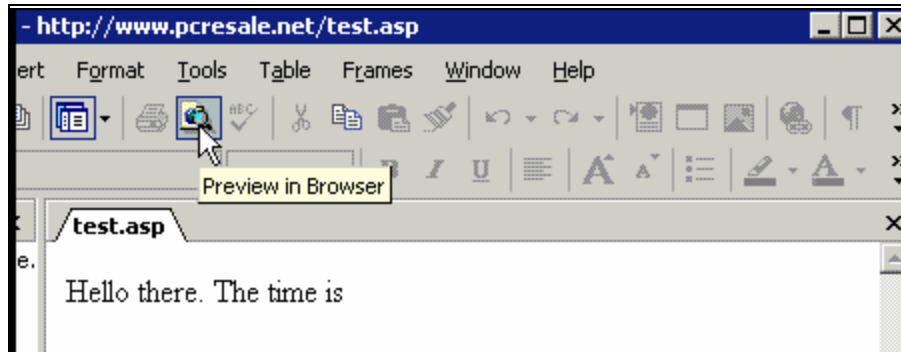
Back on the HTML tab, I want to add “The time is” and then the time after “Hello There.” Here’s how to code that with ASP tags. Everything between `<%` and `%>` is our ASP script. This tells the server to put the correct time there.



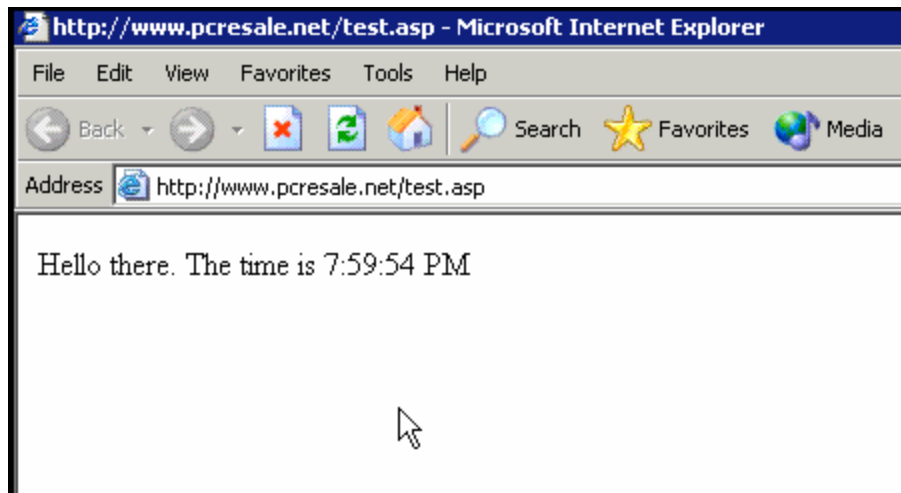
Let's save this page as **test.asp**. ASP pages have to end in **.ASP**.



Now let's click on the **Preview in Browser** button to see what we get.

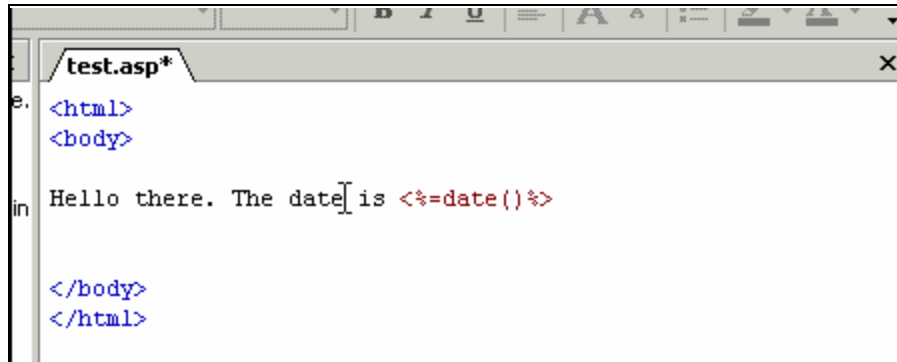


Internet Explorer opens up and it shows the following:



That's because the server is handing out that page with the time in it. If you tried the same thing as we did above, but did not see the time here, then your server or your site does not support ASP. You'll have to talk to your Internet provider.

Let's close Internet Explorer and go back to HTML view. And instead of time, let's change the code to date.

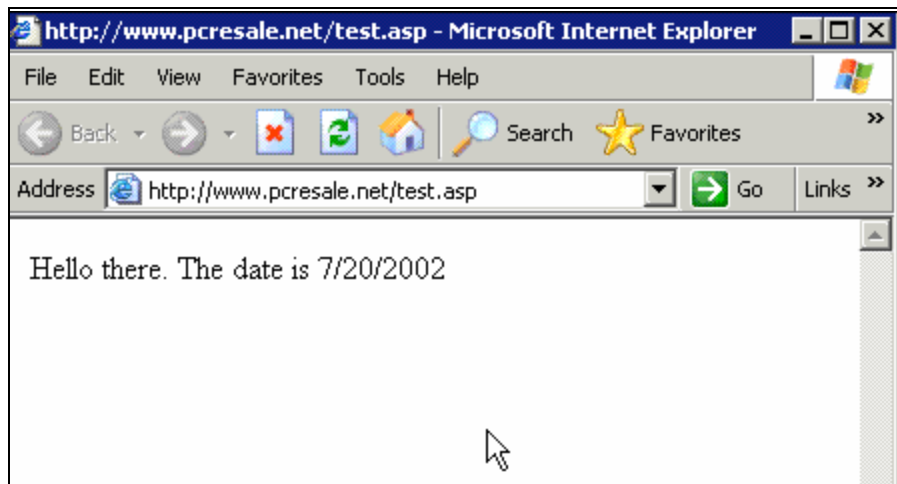


```
test.asp*
<html>
<body>

Hello there. The date is <%=date()%>

</body>
</html>
```

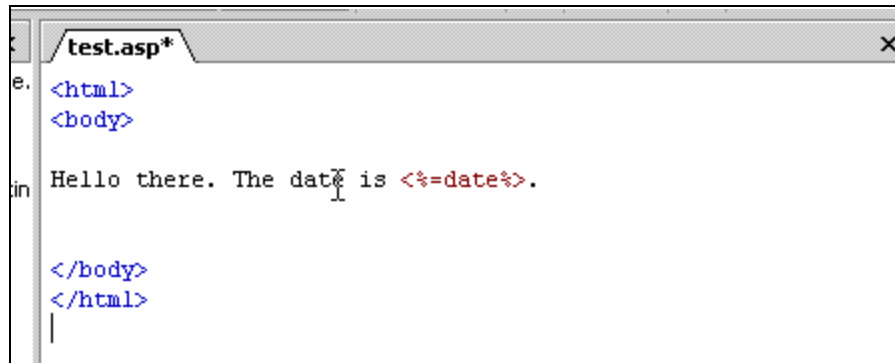
Save the page and preview it in the browser. And there we go!



If-Then

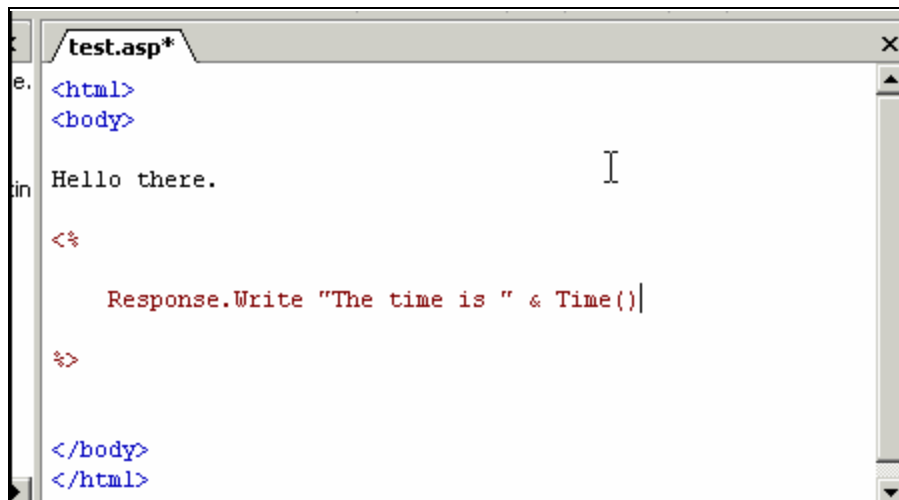
In this lesson, we're going to learn about three new ASP commands: **Response.Write**, **If / Then**, and **DataPart**.

In the previous example, we had a mixture on this line of regular HTML text and ASP code.



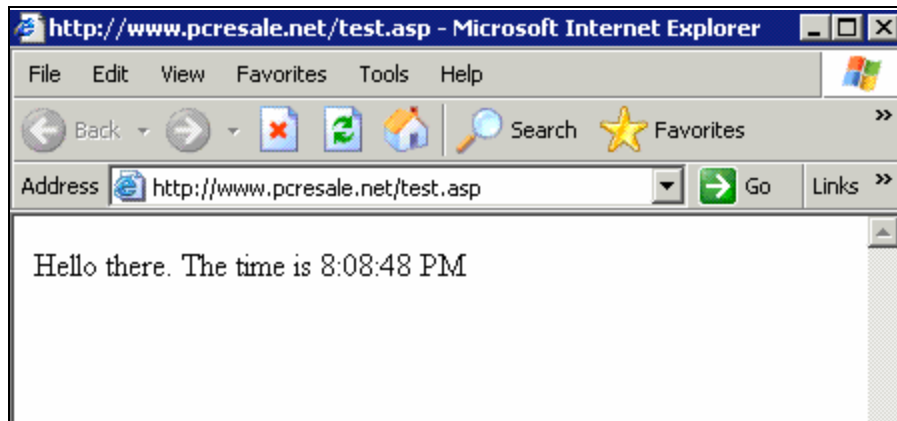
```
test.asp*
<html>
<body>
Hello there. The date is <%=date%>.
</body>
</html>
```

You can have ASP generate standard text using the **Response.Write** command. Try this. Copy the following onto your page.



```
test.asp*
<html>
<body>
Hello there.
<%=
    Response.Write "The time is " & Time()
%>
</body>
</html>
```

Save that and preview it in your browser.



The Response.Write command is like the old “Print” command in BASIC. The Response.Write command makes it much easier to create dynamic lines of text. Anything inside of quotes is displayed in the browser. It’s an actual string of text . The ampersand (&) is used to put together two strings (bits of information).

Now I’d like the computer to tell me “Good Morning” or “Good Afternoon.” Add the following to your code below:

```

<%
    Response.Write "The time is " & Time()

    If DatePart("h",Time()) >= 12 then
        'It is after noon
        Response.Write ". Good Afternoon"
    Else
        'It is before noon
        Response.Write ". Good Morning"
    End If

```

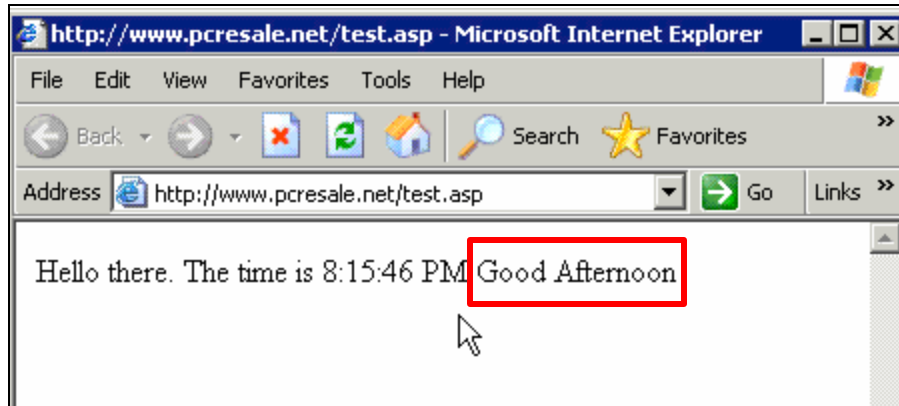
Let’s talk about this for a minute. **DatePart** is a function that will get part of the date or time. In this case we’re saying, “Give me the H” or the hour of the current time.

The IF part is going to check and see if the date part that we get back is greater than or equal to 12. If so, THEN print “Good Afternoon” onto the webpage.

The single quote (‘) precedes a comment. Everything after a single quote is ignored. It’s just for you or another programmer to see and understand what you’re doing.

IF the date part is not greater than 12, the ASP script will do something ELSE. It will run the command under the ELSE part.

Let's save this page and preview it.

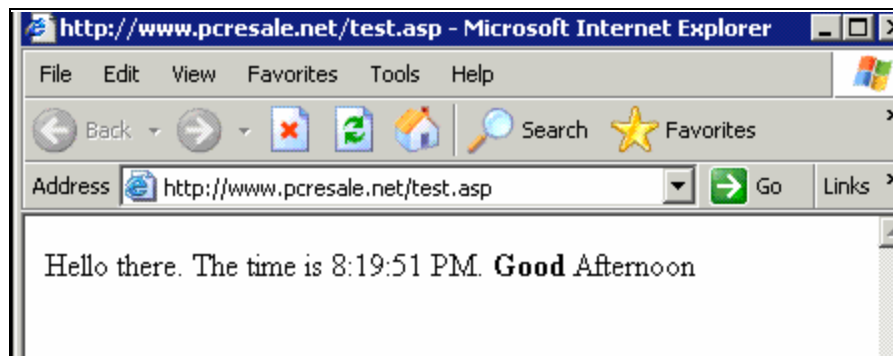


HTML Tags in ASP Scripts

In this lesson, we'll see how we can include HTML tags in ASP scripts. Let's say in our previous example, I want the word, "Good" to be bolded. I can do that by surrounding "Good" with HTML bold tags.

```
If DatePart("h",Time()) >= 12 then
    'It is after noon
    Response.Write "<B>Good</B> Afternoon"
Else
    'It is before noon
    Response.Write ". Good Morning"
End If
```

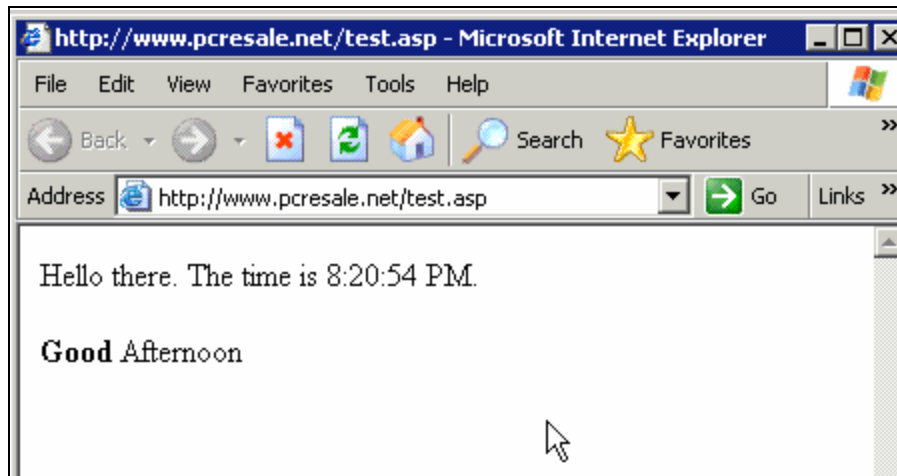
I'll save it and preview it to see if that worked. (It did!)



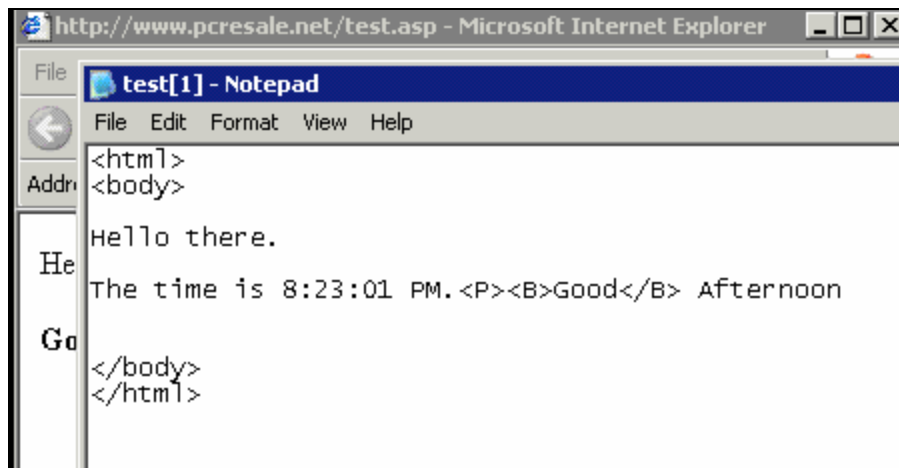
How about putting a paragraph break in here as well?

```
If DatePart("h",Time()) >= 12 then
    'It is after noon
    Response.Write "<P><B>Good</B> Afternoon"
Else
    'It is before noon
    Response.Write "<P><B>Good</B> Morning"
End If
```

I'll save it and preview it.



As you can see, you can put almost any HTML tag inside your ASP scripts. Let's take a look at what exactly the server sends to the browser. On the browser menu bar, click on **View** – and then **Source**. This is how you can see the source of any HTML page. The server replaced our ASP script with the actual values. That's all that's sent to the browser.

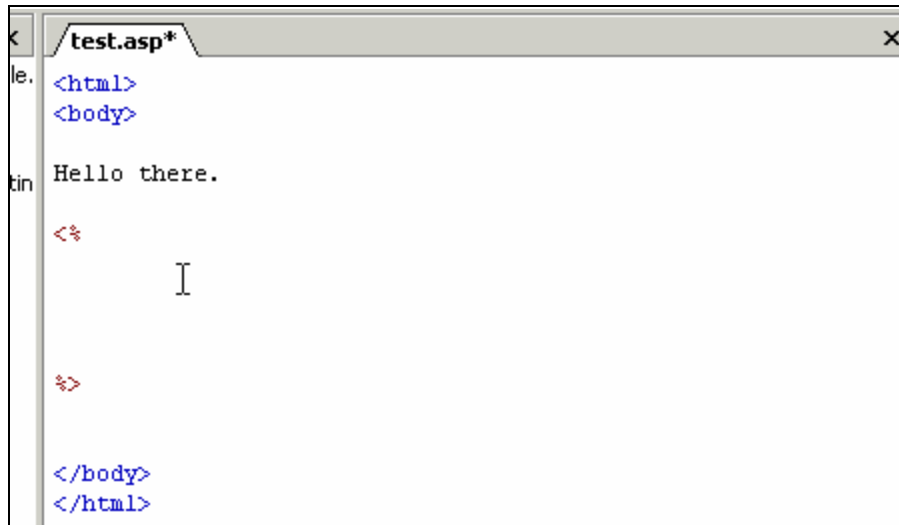


Let's go ahead and close Notepad and the Internet Explorer.

Looping & Variables

In this lesson, we're going to talk about looping and variables. If you've done any work in Visual Basic, you may already know this material. Here, you'll get to see how it works in ASP.

Let's start off with a blank ASP script.



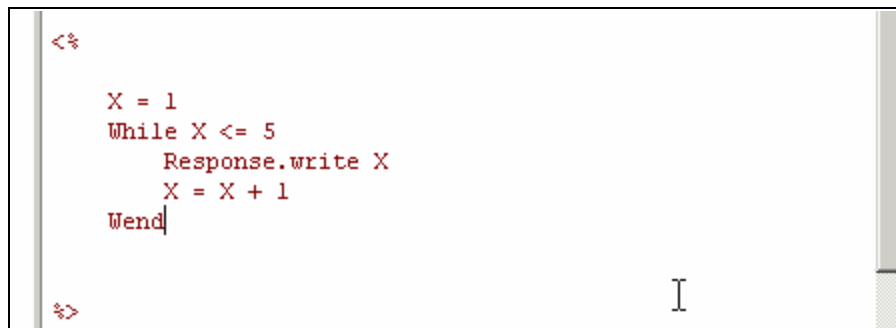
```
<html>
<body>

Hello there.

<?
    I
%>

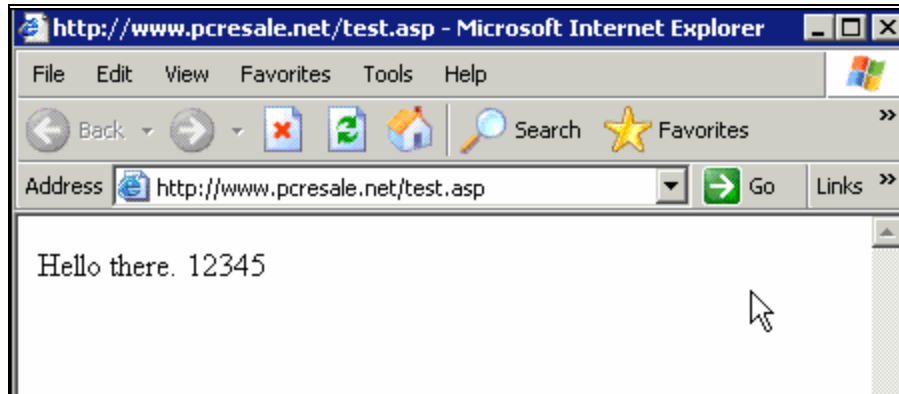
</body>
</html>
```

I'd like to loop through one through five. Copy what you see below onto your own ASP page (ASP is not case sensitive, so don't worry about case).



```
<?
    X = 1
    While X <= 5
        Response.write X
        X = X + 1
    Wend
%>
    I
```

Save the page and preview it in the browser. We get exactly what we told it to do.



Let's close this and analyze the code.

```
<%  
  
    X = 1  
    While X <= 5  
        Response.write X  
        X = X + 1  
    Wend  
  
%>
```

X = 1 creates a variable. **While X <= 5** means that as long as X is less than or = to 5, write the value of X (**Response.write X**). **X = X + 1** will increment the value of X. **Wend** is the end of the **While** loop. When it finally gets to six, it drops out.

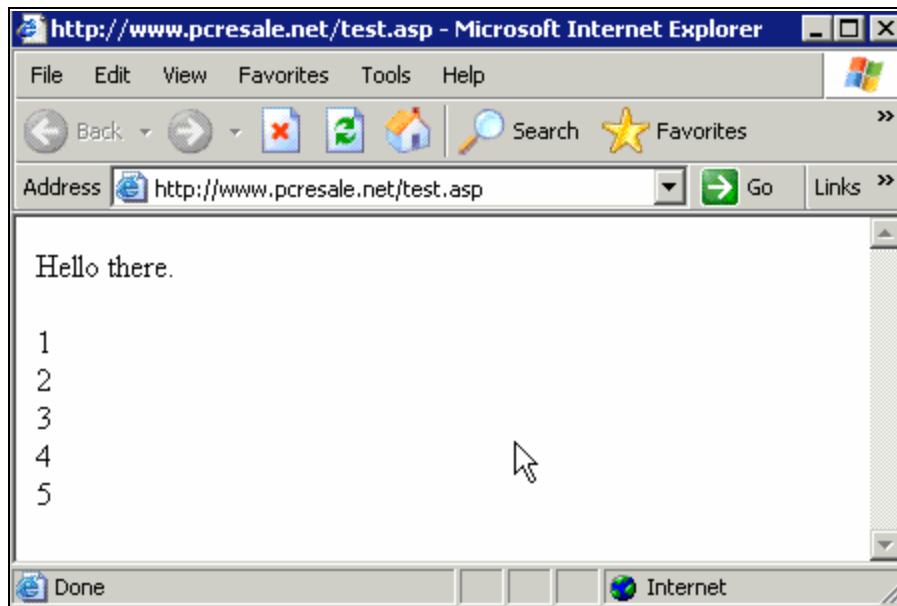
That's your basic While loop. Here's another way to write the same stuff using a different loop – **the For-Next** loop. Type this as your ASP code.

```
<%  
  
    For X = 1 to 5  
        Response.write X  
    Next  
  
%>
```

If we save it and preview it, we get the same results. But if you want to have each number on a line of its own, change your code to include a line break (
 tag):

```
<%  
  
    For X = 1 to 5  
        Response.write X & "<BR>"  
    Next  
  
%>
```

When we save that, we get the following:

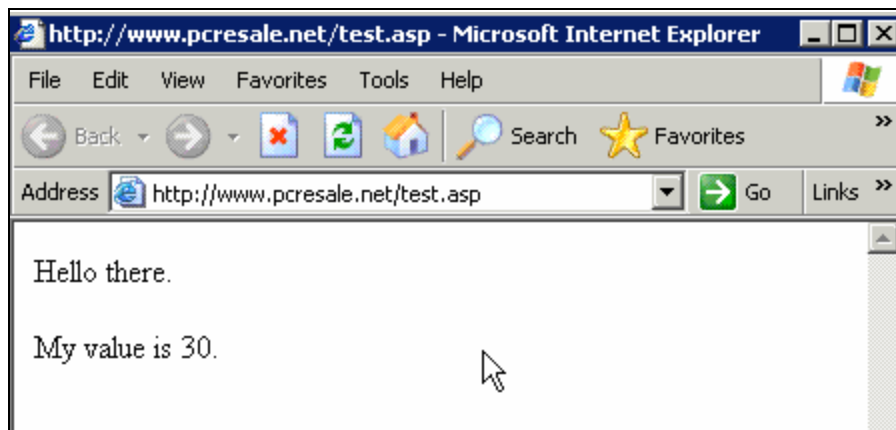


Mathematical Operations

Let's see how we can do some mathematical operations in ASP. Let's create new code and type in the following:

```
<%  
  
    MyValue = 10  
    MyValue = MyValue + 20  
  
    Response.Write "My value is " & MyValue & "."  
  
%>
```

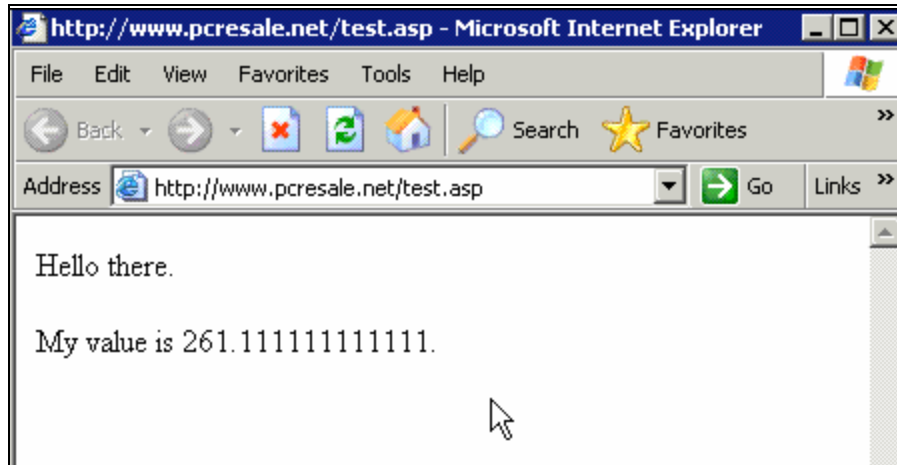
Save it and preview it.



Let's go back and add some more to it.

```
<%  
  
    MyValue = 10  
    MyValue = MyValue + 20  
    MyValue = MyValue - 5  
    MyValue = MyValue * 2  
    MyValue = MyValue / 3  
    MyValue = MyValue + (MyValue - 1)|  
  
    Response.Write "My value is " & MyValue & ". "  
  
%>
```

Save it and preview it.



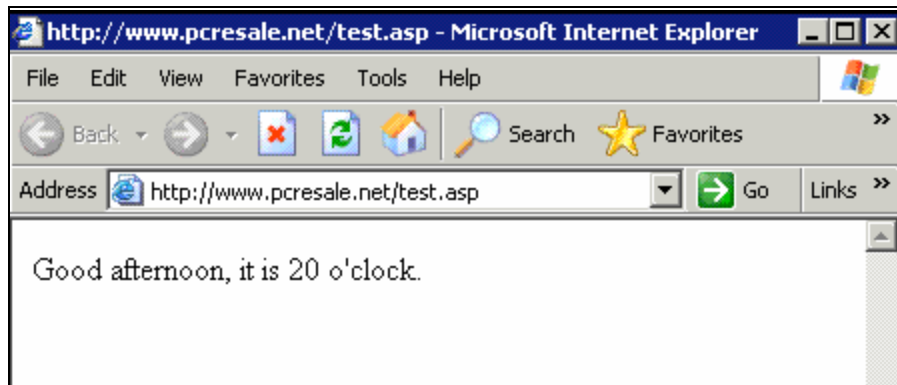
If we add `MyValue = Round(MyValue,2)` to the code, **261.11** is displayed in the web browser.

If-Then-Else-Else

Now that we know how to use variables, let's get the hour variable and store it in a value called, "MyHour". Type the following as new ASP code.

```
in <*>  
  
    MyHour = DatePart("h",Time())  
  
    If MyHour < 12 then  
        response.write "Good morning, it is " & MyHour _  
            & " o'clock."  
    Else  
        response.write "Good afternoon, it is " & MyHour _  
            & " o'clock."  
    End If
```

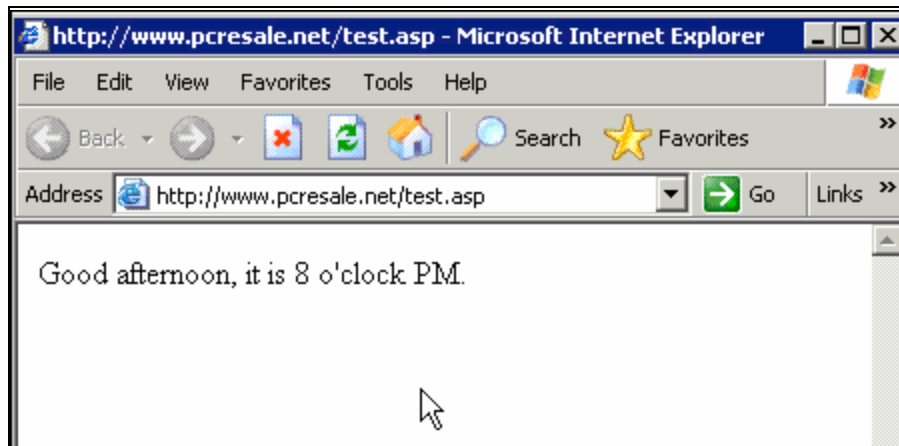
Let's save that and see what we get so far.



Maybe, we should have it where if it's after 12, we can subtract 12 from the value, add some AM and PM text, and then display that.

```
*  
  
    MyHour = DatePart("h",Time())  
  
    If MyHour < 12 then  
        response.write "Good morning, it is " & MyHour _  
            & " o'clock AM."  
    Else  
        response.write "Good afternoon, it is " & MyHour-12 _  
            & " o'clock PM."  
    End If
```

Save it and preview it.



Now, what happens if it's noon?! We need something to take into consideration that it could be exactly 12. Let's insert the following code:

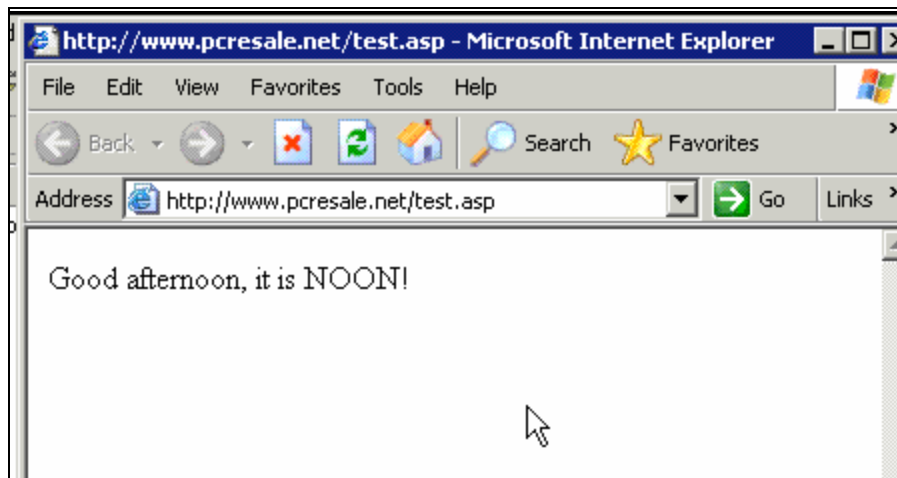
```
<%  
  
MyHour = DatePart("h",Time())  
  
If MyHour < 12 then  
    response.write "Good morning, it is " & MyHour _  
        & " o'clock AM."  
Elseif MyHour = 12 then  
    response.write "Good afternoon, it is NOON!"  
Else  
    response.write "Good afternoon, it is " & MyHour-12 _  
        & " o'clock PM."  
End If
```

And let's cheat a little bit. Let's make our own time to test this script, instead of using the server time. We can do that by commenting out the code that grabs the server time, and make our own time equal 12.

```
<?
  'MyHour = DatePart("h",Time())
  MyHour = 12

  If MyHour < 12 then
    response.write "Good morning, it is " & MyHour _
      & " o'clock AM."
  ElseIf MyHour = 12 then
    response.write "Good afternoon, it is NOON!"
  Else
    response.write "Good afternoon, it is " & MyHour-12 _
      & " o'clock PM."
  End If
```

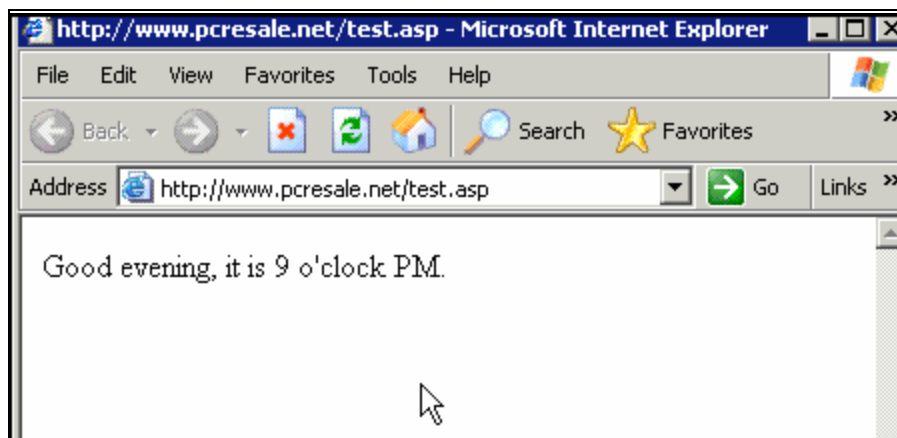
Let's save that and see what happens.



Now how are we going to handle midnight? We need to change our code to the following:

```
If MyHour = 0 then
    response.write "Good morning, it is MIDNIGHT!"
Elseif MyHour < 12 then
    response.write "Good morning, it is " & MyHour _
        & " o'clock AM."
Elseif MyHour = 12 then
    response.write "Good afternoon, it is NOON!"
Elseif MyHour < 18 then
    response.write "Good afternoon, it is " & MyHour-12
        & " o'clock PM."
Else
    response.write "Good evening, it is " & MyHour-12 _
        & " o'clock PM."
End If
```

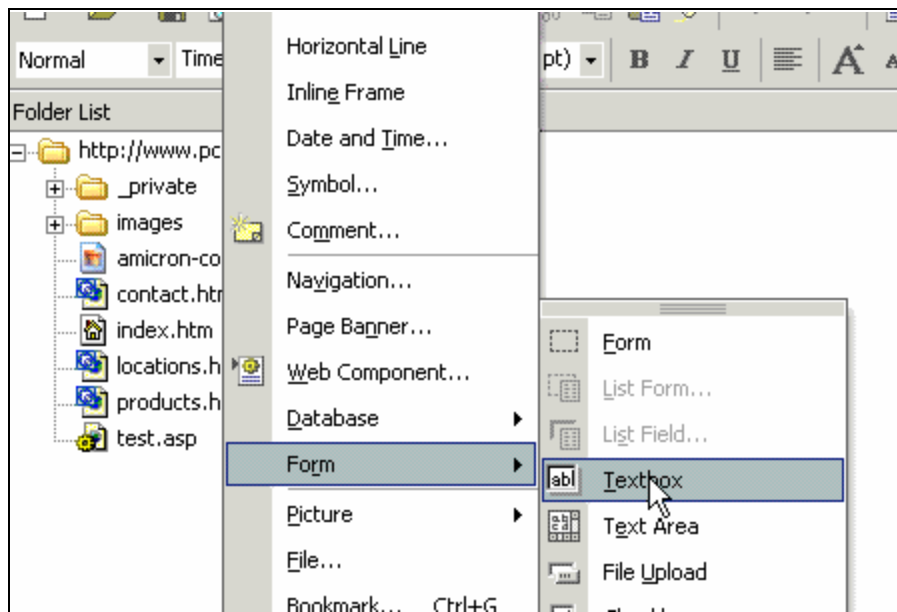
That ought to do it. Let's get rid of our fake time and un-comment the real (server) time. Save this and see what we get.



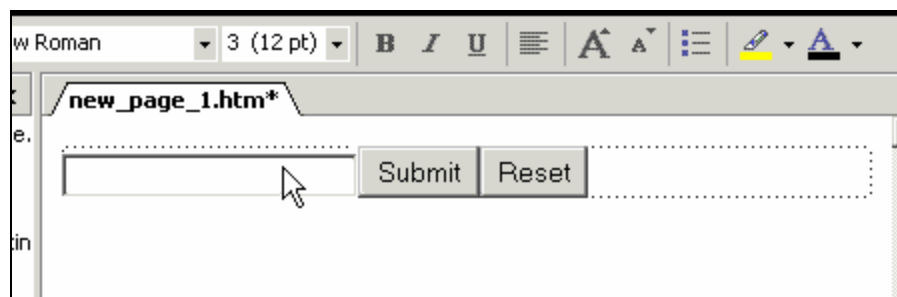
Processing Form Input

In this lesson, we're going to see how to process form input. The way you can make your website dynamic, is to have a user type some information into a form – hit a submit button – and submit that information to a server so the server can process it. If you have any pages opened from previous lessons, go ahead and close them.

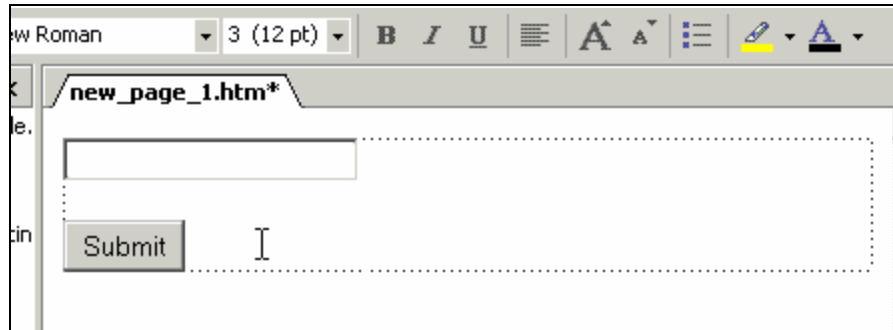
First, let's create a basic form. Make a new page and drop a form on it. On the menu bar, click **Insert** – then **Form**– then **Textbox**.



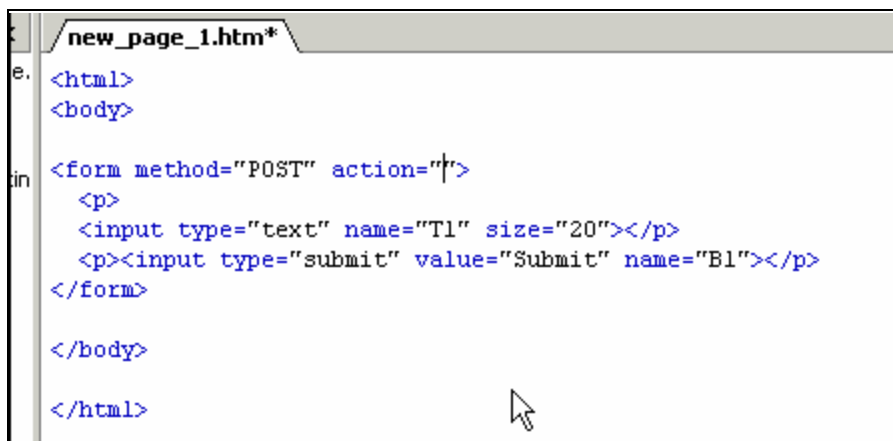
This will give us a form with one textbox, a submit button, and a reset button.



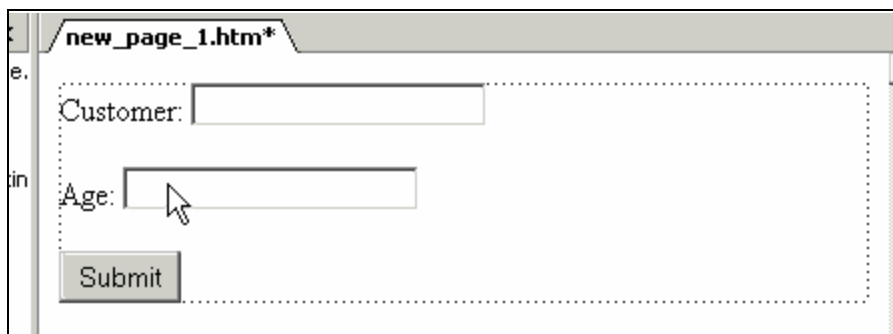
I'm going to place the buttons on a new line and get rid of the reset button.



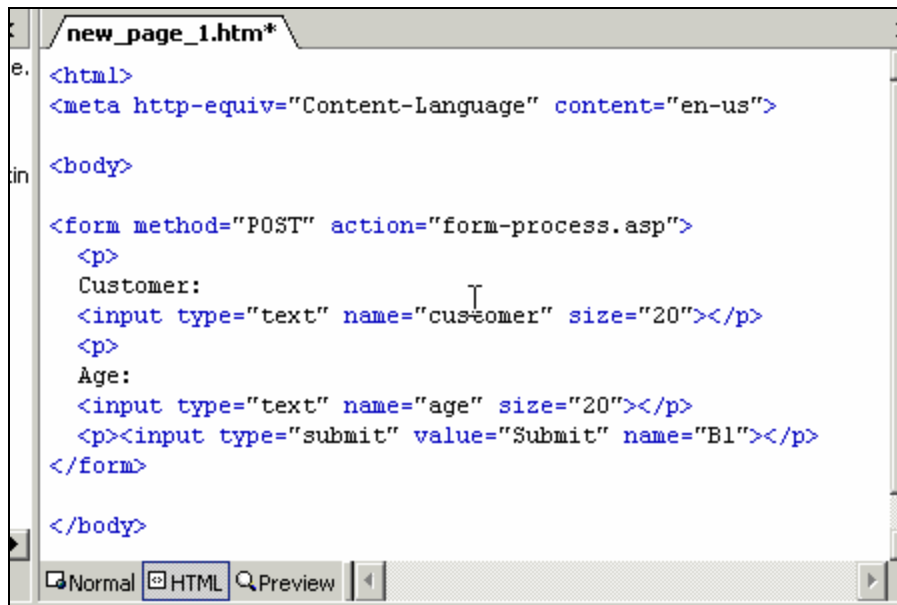
Let's take a look at the HTML code of this page real quick. Click the HTML tab and get rid of the header stuff, plus the extra FrontPage gunk that we don't need until your HTML looks like this.



Let's give this thing a better name like, "customer" and put another one in here from **Normal** view. Just copy the box and paste it after "Age".



Back in **HTML** view, we can change the name of this text box to “age”. Now when the user clicks the Submit button, I want this page to send its information to another page. Inside the action value of our HTML code, we can type in the name of that page. We’ll call it “**form-process.asp**”.



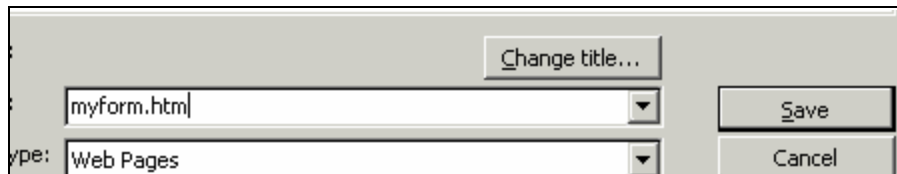
```
<html>
<meta http-equiv="Content-Language" content="en-us">

<body>

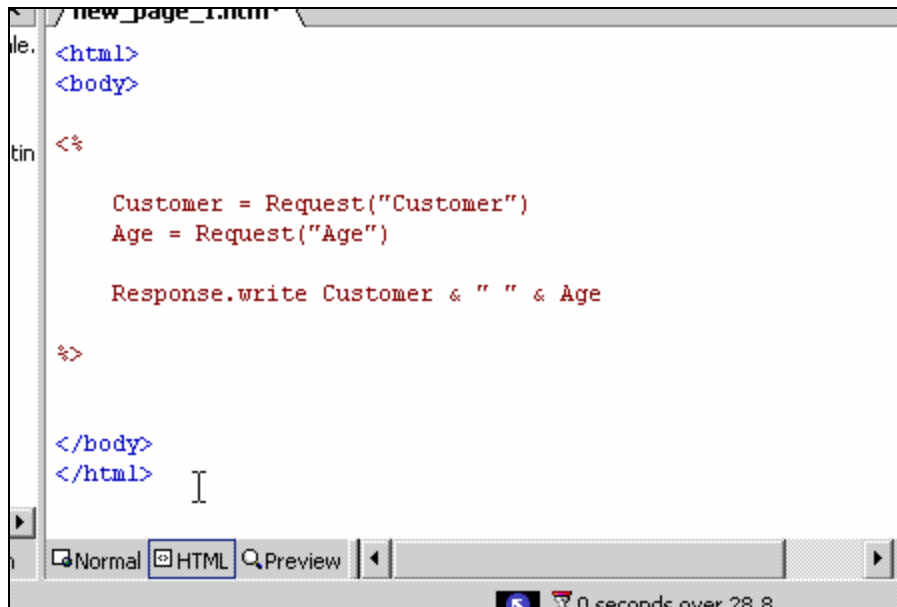
<form method="POST" action="form-process.asp">
  <p>
  Customer:
  <input type="text" name="customer" size="20"></p>
  <p>
  Age:
  <input type="text" name="age" size="20"></p>
  <p><input type="submit" value="Submit" name="B1"></p>
</form>

</body>
```

Let’s save this page as **myform.htm**.



Now let's make that other page (**form-process.asp**) real quick. Create a new blank page and go to **HTML** view. Delete all the header stuff and type in some ASP code that will process the information submitted through our form.



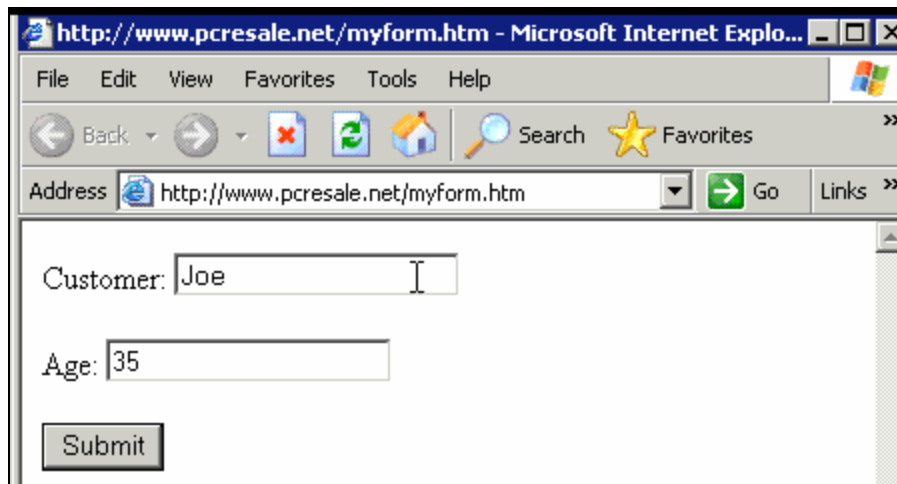
```
<html>
<body>

<%
    Customer = Request("Customer")
    Age = Request("Age")

    Response.write Customer & " " & Age
%>

</body>
</html>
```

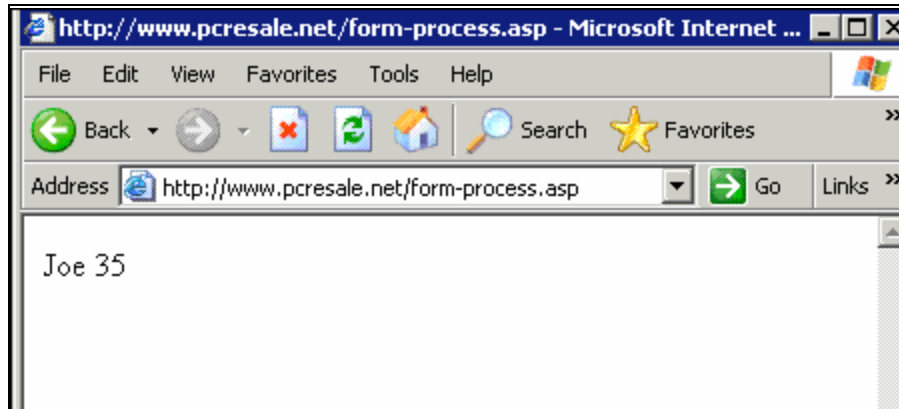
Save this as **form-process.asp**. Now go back to **myform.htm** and then preview it in the browser. I'll fill in the form with the following:



Customer:

Age:

I'll hit the Submit button and voila!



Let's analyze this code and see what's going on. We made a variable called **Customer** and we set it equal to **Request("Customer")**. ASP will request the value entered in the textbox named "Customer" from our form. Same thing for the age. The **Response.write** just writes it all out.

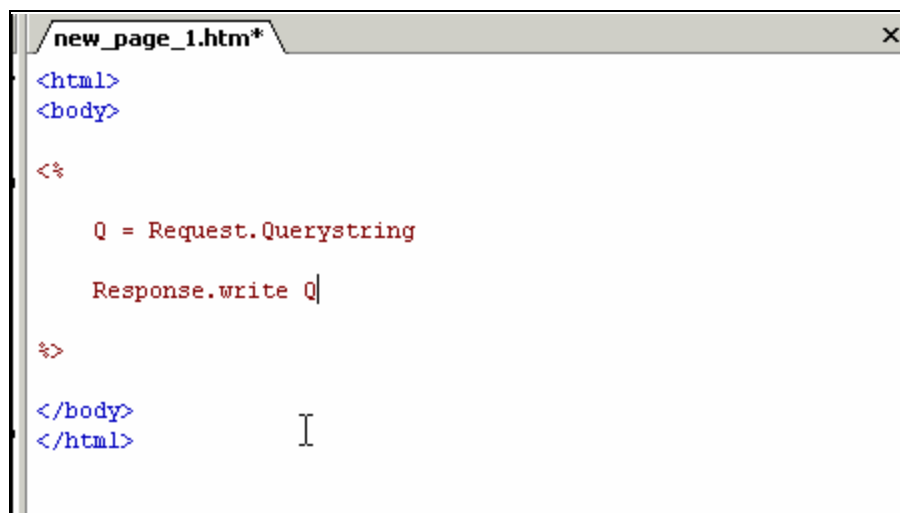
```
tin <%  
  
    Customer = Request("Customer")  
    Age = Request("Age")  
  
    Response.write Customer & " " & Age  
  
%>
```

Reading the Query String

In this lesson, we're going to learn how to read data from a URL or hyperlink. Here's a regular url:
<http://www.pcesale.net>

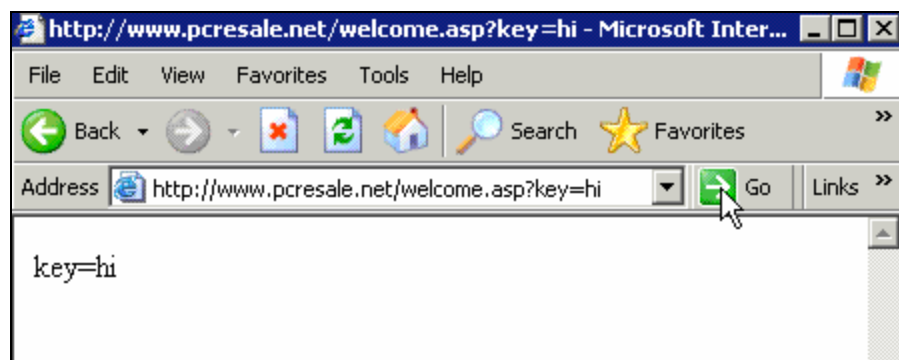
You can specify values after a URL and you do that with a question mark and then some value, like "key". We'll say, <http://www.pcesale.net?key=Google>. (The stuff that comes after the question mark is called the query string.) Now I can use this URL in any hyperlinks that I advertise on Google. That way when people come to my site, I'll know that they came from Google.

Let's see how I can put this in action. I'm going to create a new page in HTML view without all the header stuff, and then type in the following ASP code. This code should get the query string and show me what it is.



```
new_page_1.htm*
<html>
<body>
<%=
    Q = Request.QueryString
    Response.write Q
%>
</body>
</html>
```

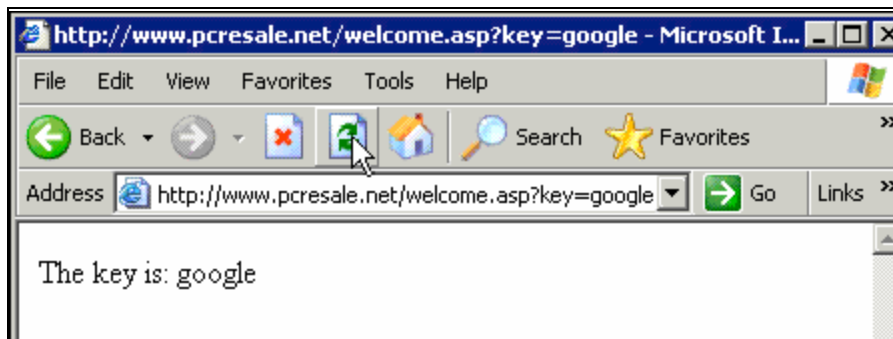
Let's save this page as **welcome.asp** Now in my web browser, I'm going to go to <http://www.pcesale.net/welcome.asp?key=hi>. Our ASP page will display anything after the question mark in our URL.



Back in our script, I don't want to see the whole query string. I want just one value out of the query string. So I'm going to change the ASP code to this now:

```
inputin <%  
n  
    Key = Request("Key")  
m  
    Response.write "The key is: " & Key  
m  
%>
```

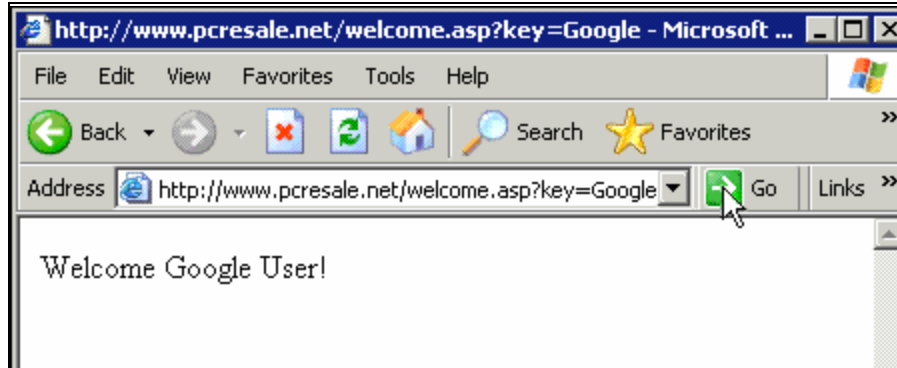
Let's save this and see what this will do in the browser. I'll go to <http://www.pcsale.net/welcome.asp?key=google>.



We can even do something funny. We can use:

```
inputin <%  
n  
    Key = Request("Key")  
m  
    If Key = "Google" then  
        response.write "Welcome Google User!"  
    end if  
n  
ss.asp %>
```


Then when we save it and load up <http://www.pcesale.net/welcome.asp?key=google>, we get the following in our browser:



Review

Let's take a moment now to review what we covered in class.

- We learned about ASP, what ASP is, and a few definitions
- We built a test script to see if our server can run ASP scripts
- We learned about If, Then, Else, EndIf statements
- We learned how to use HTML tags in ASP
- We learned about While and For loops
- We learned how to do a little math in ASP
- We learned how to submit and process form data
- We learned how to get data from a query string

RICK'S NOTE: I really do enjoy getting surveys from you! Make sure you visit the web page above and fill out the survey for this class. Let me know if I've moved too fast, and whether or not I covered material that was helpful to you!

Take your **skills check** quiz at www.599CD.COM/Test. If you pass, you can print out a Certificate of Completion.

What's next? ASP 102.

Contact Us. If you have any questions, visit www.599cd.com/contact



This course, handbook, videos, and other materials are **copyright** 2002, 2003, 2004 by Amicron Computing. All rights reserved. No portion of this course, handbook, videos, or other course materials may be reproduced, copied, edited, or otherwise distributed without the express written permission of Amicron Computing. Amicron Computing shall not be held liable for any errors or omissions in this document.

This document may **not** be used as part of a training course without express, written permission from Amicron Computing and the purchase of an **Instructional License**. For details, contact:

Amicron Computing
PO Box 1308
Amherst NY 14226 USA
www.MYOLP.COM