

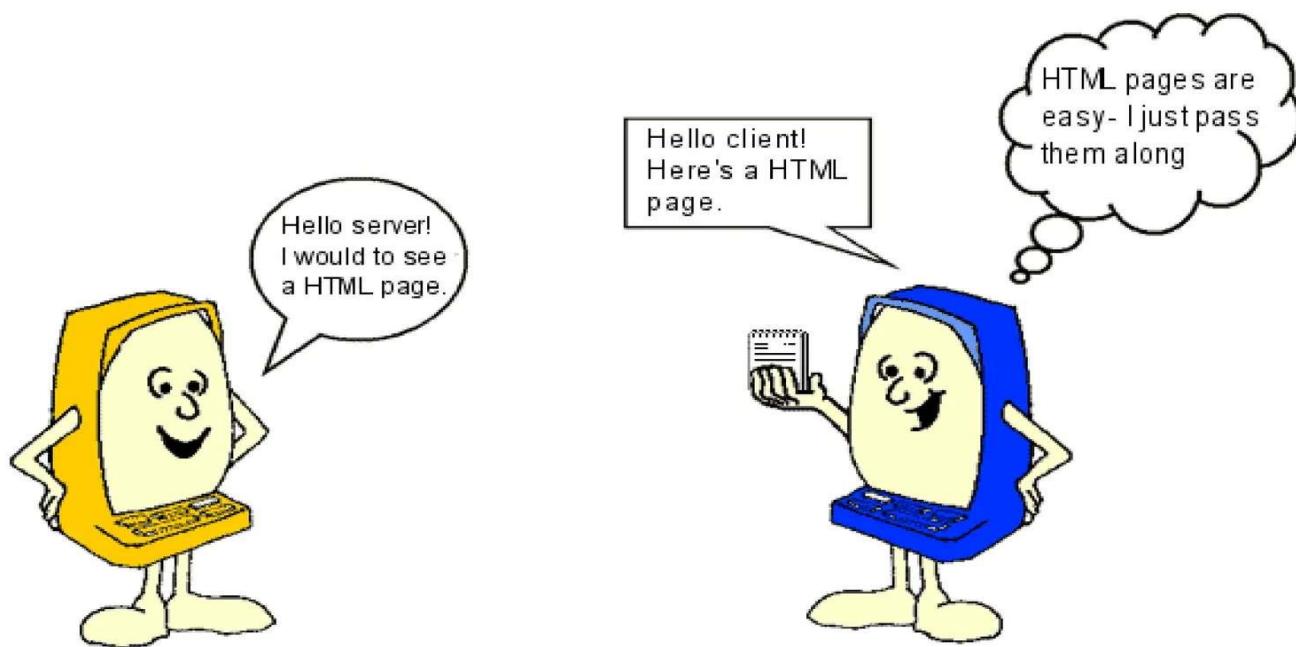
ASP is a feature of the Microsoft Internet Information Server (IIS), but, since the server-side script is just building a regular HTML page, it can be delivered to almost any browser. You can create an ASP file by including a script written in VBScript or JScript in an HTML file or by using ActiveX Data Objects (ADOs) program statements in the HTML file. You name the HTML file with the ".asp" file suffix. Microsoft recommends the use of the server-side ASP rather than a client-side script, where there is actually a choice, because the server-side script will result in an easily displayable HTML page. Client-side scripts (for example, with JavaScript) may not work as intended on older browsers.

Important!

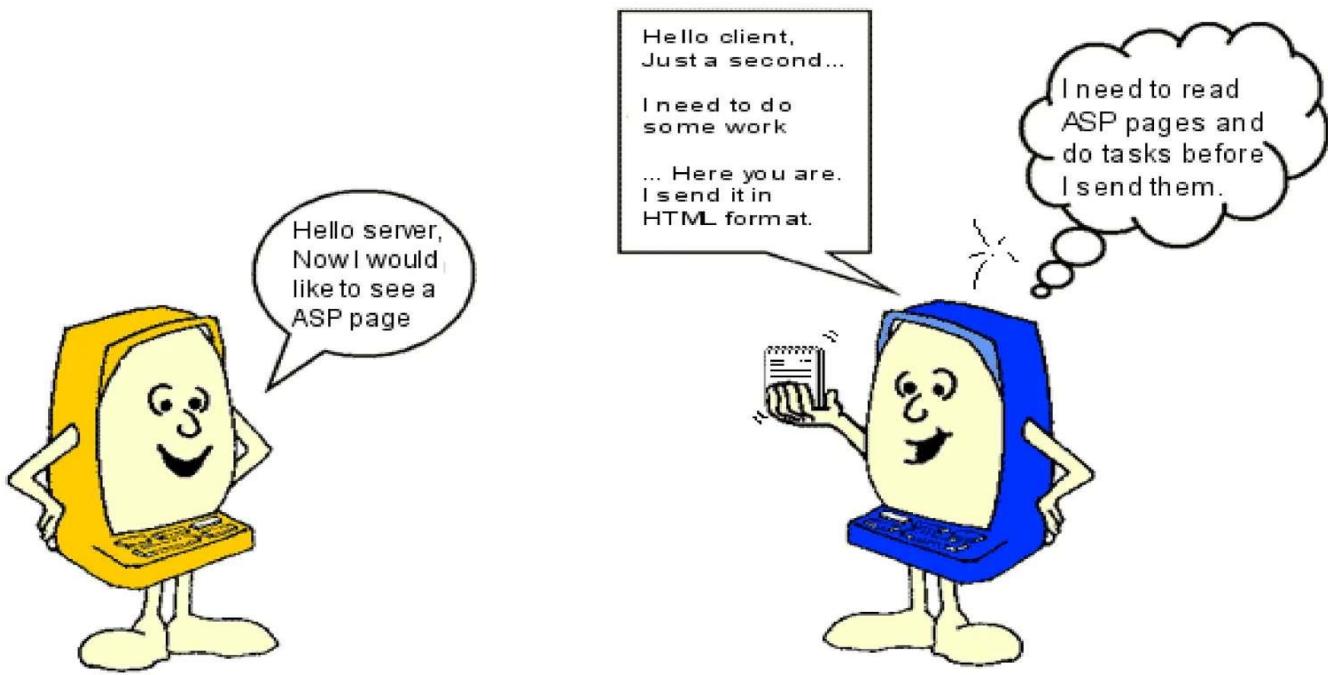
ASP is no longer being further developed by Microsoft - it has been replaced by ASP.NET. But ASP is still a fully functional scripting language and you may choose to continue to learn and use ASP (i.e., if your server only supports ASP, you need to maintain an ASP website, etc.). If it's not necessary for you to use ASP, we recommend you learn PHP instead:

How does ASP work?

The best way to explain how ASP works is by comparing it with standard HTML. Imagine you type the address of an HTML document (eg. <http://www.mysite.com/page.htm>) in the address line of the browser. This way you **request** an HTML page. It could be illustrated like this:



As you can see, the server simply sends an HTML file to the client. But if you instead type <http://www.mysite.com/page.asp> - and thus request an **ASP page** - the server is put to work:



The server first reads the ASP file carefully to see if there are any tasks that need to be executed. Only when the server has done what it is supposed to do, the result is then sent to the client. It is important to understand that the client only sees the **result** of the server's work - not the actual instructions.

This means that if you click "view source" on an ASP page, you do not see the ASP codes - only basic HTML tags. Therefore, you can not see how an ASP page is made by using "view source". You have to learn ASP in other ways, for example, by reading this tutorial.

What you learn in this tutorial is to write commands to a server!

So, the first thing you need to get hold of is... a server! But don't worry - you don't need to buy a new computer. You just need to install some software on your computer that makes it function as a server. Another option is to have a website on a hosted server that supports ASP. Then you just need to be online while coding.

The next lesson is about how to get your computer to act as a server.

Servers

ASP is a *server-side* technology. Therefore, you need to have a server to run ASP. But it doesn't need to cost you anything to make this upgrade and there are several options for doing so.

Since you ultimately only need to choose one option this lesson is divided into different parts. First comes a little introduction on the different options (just choose the one that suits you best). When your server is up and running, we'll pick up with Lesson 3 to make your first ASP page.

If you use Windows 7: Windows 7 includes Internet Information Server 7.5 (IIS) and can therefore run ASP.

If you use Windows XP Professional: Windows XP Professional includes Internet Information Server 5.1 (IIS) and can therefore run ASP.

If you use Windows XP Home Edition: Internet Information Server (IIS) is not included in this edition of Windows XP. However, according to this Usenet post it is possible to install IIS on Windows XP Home Edition. You may want to try this approach. But we recommend you use Windows 7 or Windows XP Professional instead.

If you use Linux: It is possible to run ASP on Linux, but we recommend that you use PHP on Linux instead. With PHP you can do pretty much the same as ASP.

If you use Macintosh: We know of no option for running ASP on a Macintosh.

If you have a website on a hosted server: You can choose to have a website on hosted server which run ASP (like AspHost4Free (free host)), then you are ready to go and can just ftp your ASP pages to the server.

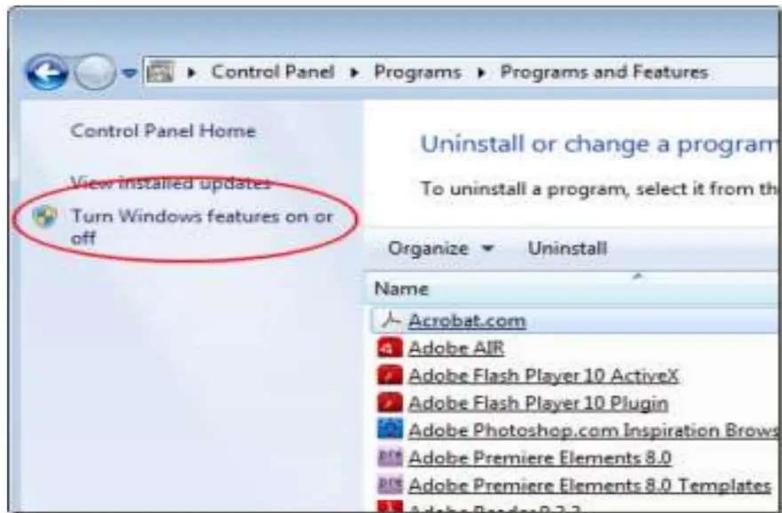
a) Internet Information Server (IIS) on Windows 7

By default, IIS is not installed on Windows 7. But you can easily install and configure IIS by following the instructions below:

1. Click Start and then click Control Panel:



2. In Control Panel, click Programs and then click Turn Windows features on or off:



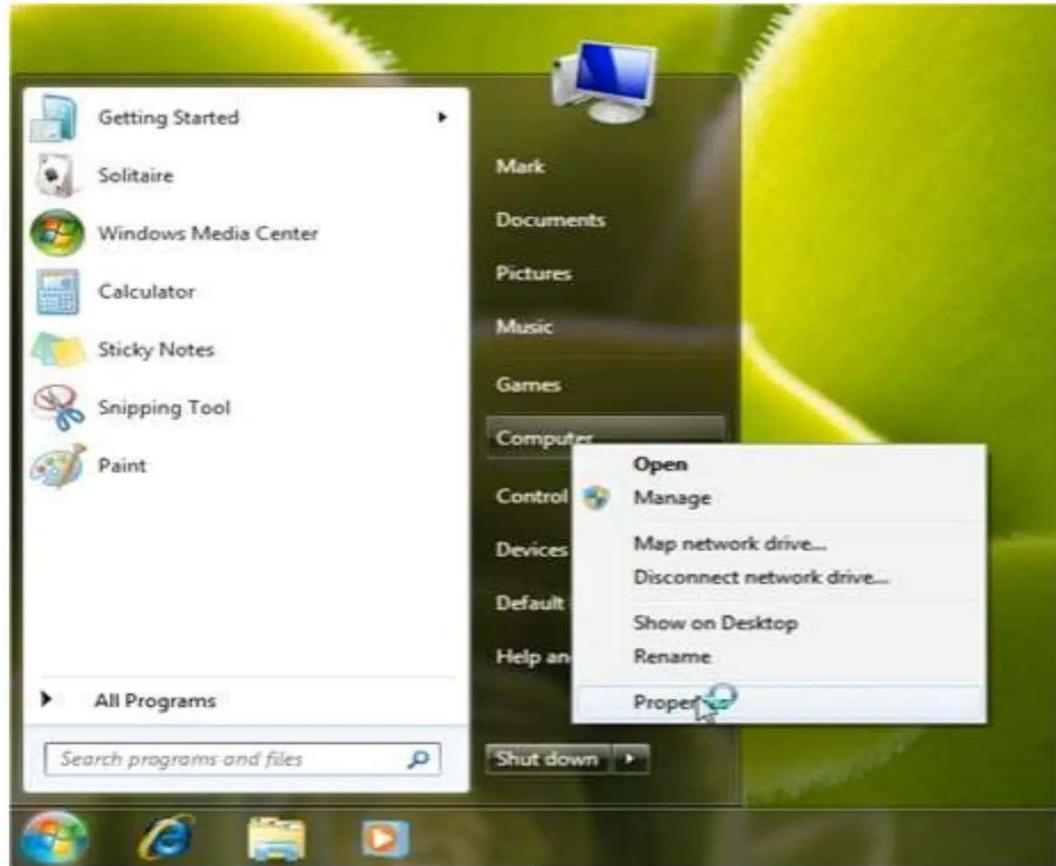
3. In the Windows Features dialog box, click Internet Information Services and then click OK:



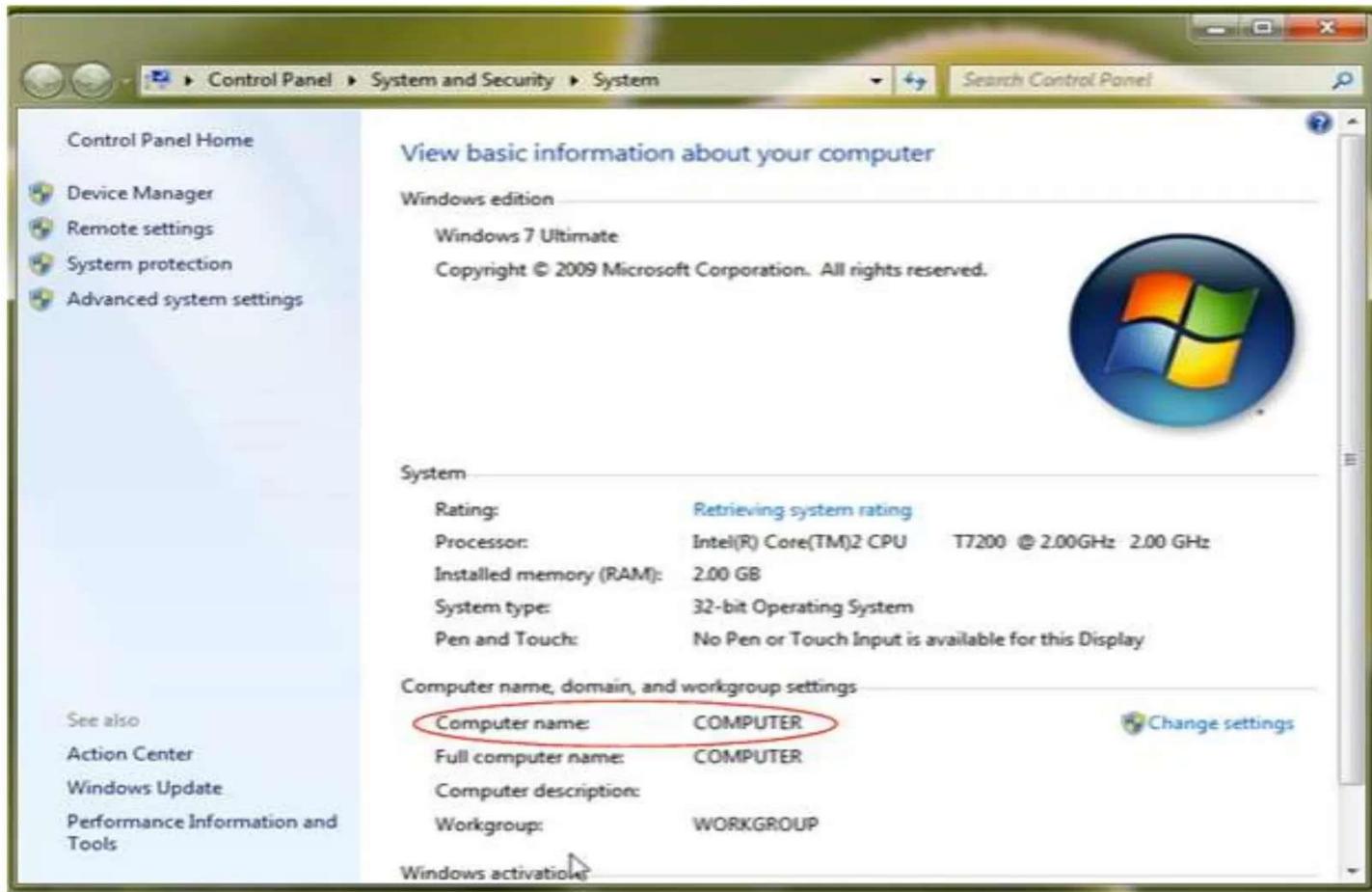
Congratulations! Your computer is now a web server!

Browsing your server

Next, you need to find your computer name. You'll find it by clicking the Start button, right-clicking Computer, and then clicking Properties.



Under Computer name, domain, and workgroup settings, you can find your computer name. I have here stated my computer's name in red - it's called COMPUTER (but yours are probably called something else).



Now you simply type the computer name in the address bar of your browser...



... and when you click return you will see a page like the one pictured below. It means that you now have contact to your server. Now we just need to look at where you need to save your ASP pages.



How to work with ASP and HTML files on the server

The last thing you need to know before you are ready to create your first ASP pages is where to save them.

The root of your website is c:\inetpub\wwwroot. This means that the file c:\inetpub\wwwroot\default.asp can be accessed in your browser at the address <http://COMPUTER/default.asp> (replace "COMPUTER" with the name of your computer (look above)).



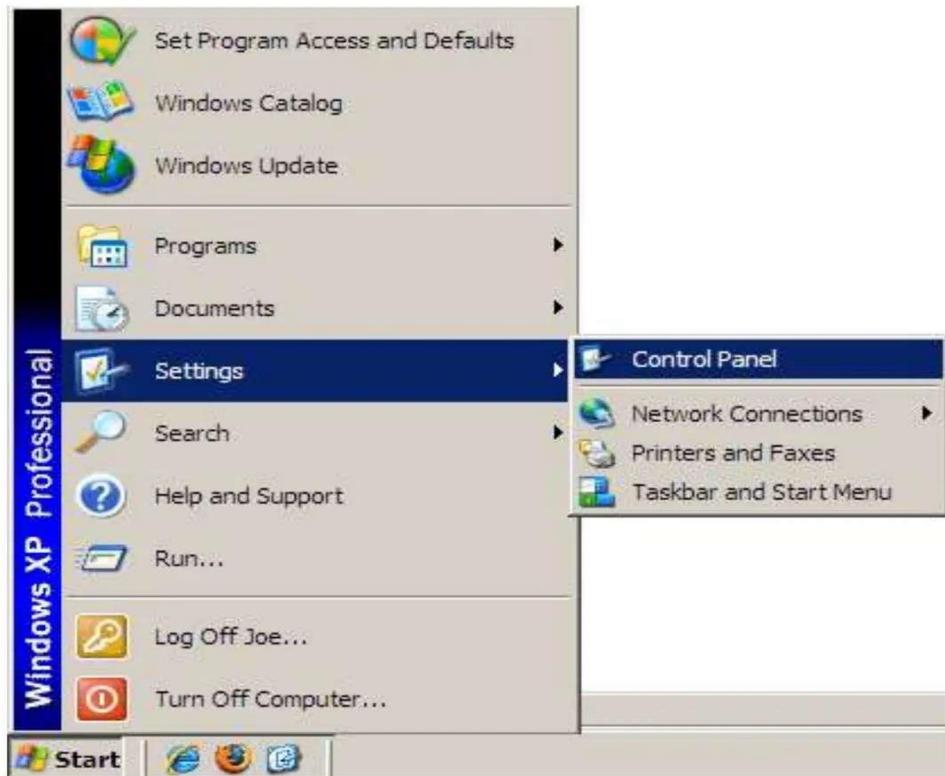
And in the same way, the folder c:\inetpub\wwwroot\images will be available in your browser at <http://COMPUTER/images>.

That's it! Now are you ready to code your first ASP page. Hurry on to Lesson 3!

b) Internet Information Server (IIS) on Windows XP Professional

By default, IIS is not installed on Windows XP Professional. But you can easily install and configure IIS by following the instructions below:

1. Click Start, click Control Panel, and click Add or Remove Programs:



2. Click Add/Remove Windows Components. The Windows Components Wizard appears:



3. Follow the on-screen instructions to install, remove, or add components to IIS:



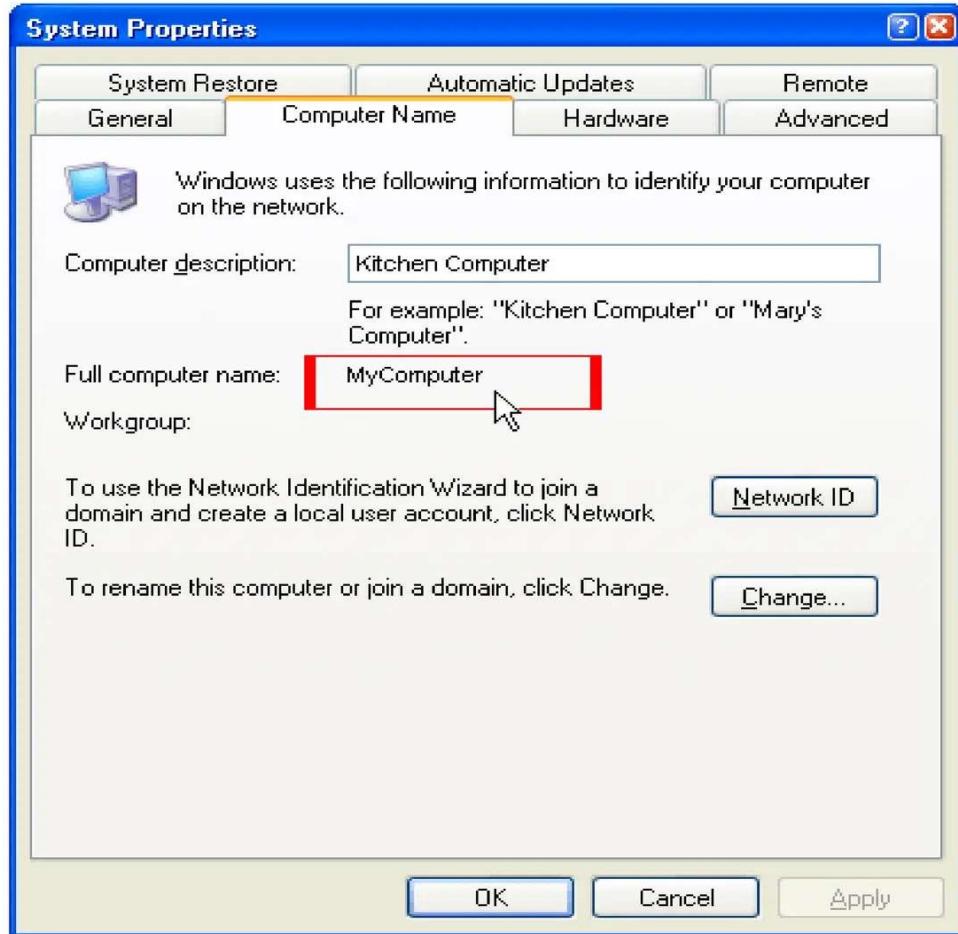
Congratulations! Your computer is now a web server!

Browsing your server

Next, you need to find your computer name. You'll find it by right clicking "My Computer" on your desktop and select "properties":



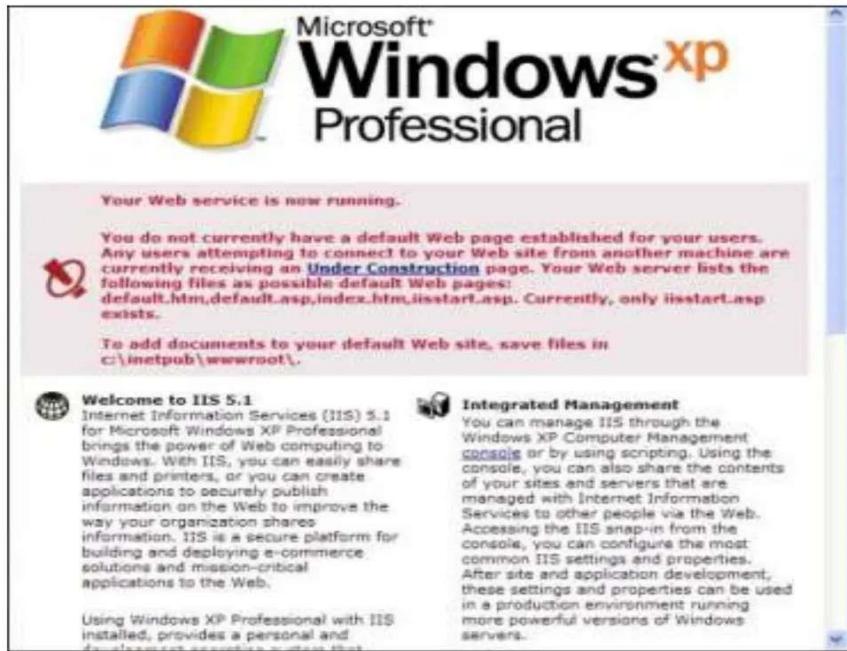
In the dialog box, you find your computer name under the tab "Network Identification". I have here stated my computer's name in red - it's called MyComputer (but yours are probably called something else).



Now you simply type the computer name in the address bar of your browser...



... and when you click return you will see a page like the one pictured below. It means that you now have contact to your server. Now we just need to look at where you need to save your ASP pages.



How to work with ASP and HTML files on the server

The last thing you need to know before you are ready to create your first ASP pages is where to save them.

The root of your website is `c:\inetpub\wwwroot`. This means that the file `c:\inetpub\wwwroot\default.asp` can be accessed in your browser at the address `http://MyComputer/default.asp` (replace "MyComputer" with the name of your computer (look above)).



And in the same way, the folder `c:\inetpub\wwwroot\images` will be available in your browser at `http://MyComputer/images`.

That's it! Now are you ready to code your first ASP page. Hurry on to Lesson 3!

Your first ASP page

You now know a little about what ASP is, and you've installed (or have access to) a server. Now we are ready to begin making our first ASP page. We keep it simple and easy, but after you have gone through this lesson, you will understand much more about what ASP is and what you can do with it.

Basically, an ASP file is a text file with the extension **.asp** which consists of:

- Text
- HTML tags
- ASP Scripts

You already know what text and HTML tags are. So let's look a little more at ASP scripts.

ASP Scripts

ASP scripts can be written in different languages. The examples in this tutorial are written in Microsoft Visual Basic Scripting Edition (VBScript), but could also be written in another language - eg. JScript.

Let's get started with your first ASP page.

Example: Hello World!

Start by making an ordinary HTML document. But name the file *default.asp* and place it in the root of the site. On your computer (which is now a server), the path is *c:\inetpub\wwwroot\default.asp*.

The HTML code should look like this:

```
<html>
<head>
<title>My first ASP page</title>

</head>
<body>

</body>
</html>
```

As you probably remember from lesson 1, ASP is all about **writing commands to a server**. So let's write a command to the server.

First, we need to tell the server when the ASP will **start** and **end**. In ASP you use the tags **<%** and **%>** to mark the start and end for the ASP codes that the server must execute.

Now try to add the following simple code snippet to your HTML code:

```
<html>
<head>
<title>My first ASP page</title>
</head>
<body>
<%
Response.Write "<h1>Hello World!</h1>"
%>
</body>
</html>
```

When we look at the ASP document in a browser, it should look like this:



But it gets interesting when you look at the HTML code in the browser (by selecting "view source"):

A screenshot of a "Source of..." dialog box from a browser. The title bar says "Source of: http://...". The menu bar includes "File", "Edit", "View", and "Help". The main content area displays the following HTML code:

```
<html>
<head>
<title>My first ASP page</title>
</head>
<body>
<h1>Hello World!</h1>
</body>
</html>
```

The ASP codes are gone! As you may remember from lesson 1, it is only the server that can see the ASP codes - **the client (the browser) only sees the result!**

Let's look at what happened. We asked the server to write `<h1> Hello World!</h1>`. In a more technical language, one would say that we used the object `Response` and the method `Write` to write a specified string to the client. But do not worry! In this tutorial we try to keep the technical language at a minimum.

Our first example is obviously not particularly exciting. But just wait! From now on, it's only going to be more and more interesting. Let's look at another example.

Example: Now!

Let's make the server write something else. We could, for example, ask it to write the current date and time:

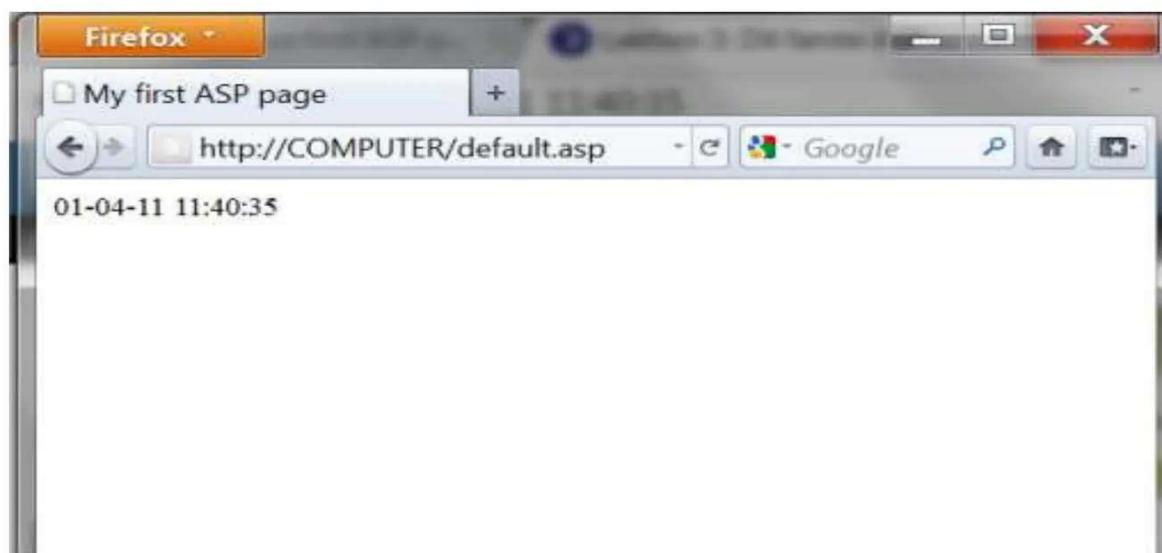
```
<html>
<head>
<title>My first ASP page</title>

</head>
<body>

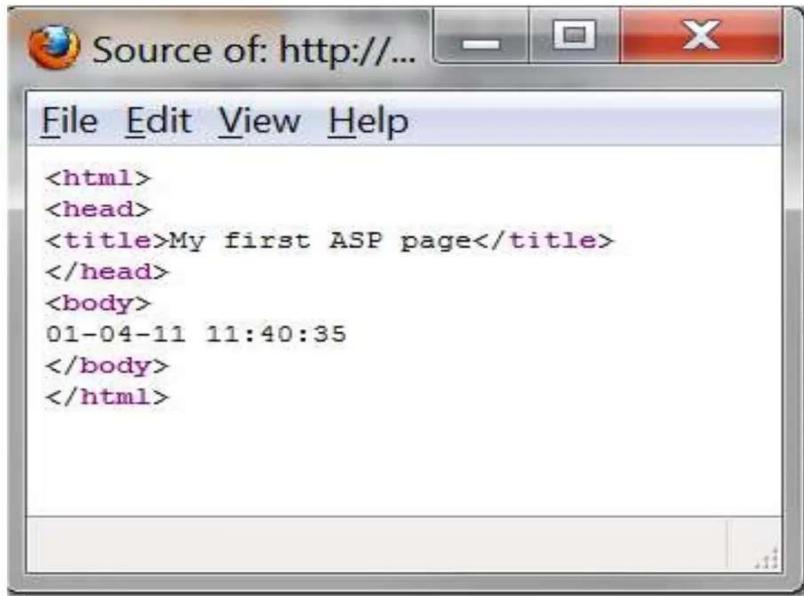
<%
Response.Write Now
%>

</body>
</html>
```

Will look like this in the browser:



And the corresponding HTML code:



The screenshot shows a Firefox browser window with the title bar "Source of: http://...". The menu bar includes "File", "Edit", "View", and "Help". The main content area displays the following ASP code:

```
<html>
<head>
<title>My first ASP page</title>
</head>
<body>
01-04-11 11:40:35
</body>
</html>
```

Now things are getting interesting, right?

We make the server write the date and time when the ASP page is displayed. Note that if you refresh the page in the browser, a new time is written. The server writes the current date and time each time the page is sent to a client.

It is also important to note that the HTML code contains only the date - not the ASP codes. Therefore, the example is not affected by which browser is used. Actually, all functionalities that are made with *server-side* technologies always **work in all browsers!**

In the example, we used `Now` - which is a function that returns the current date and time on the server.

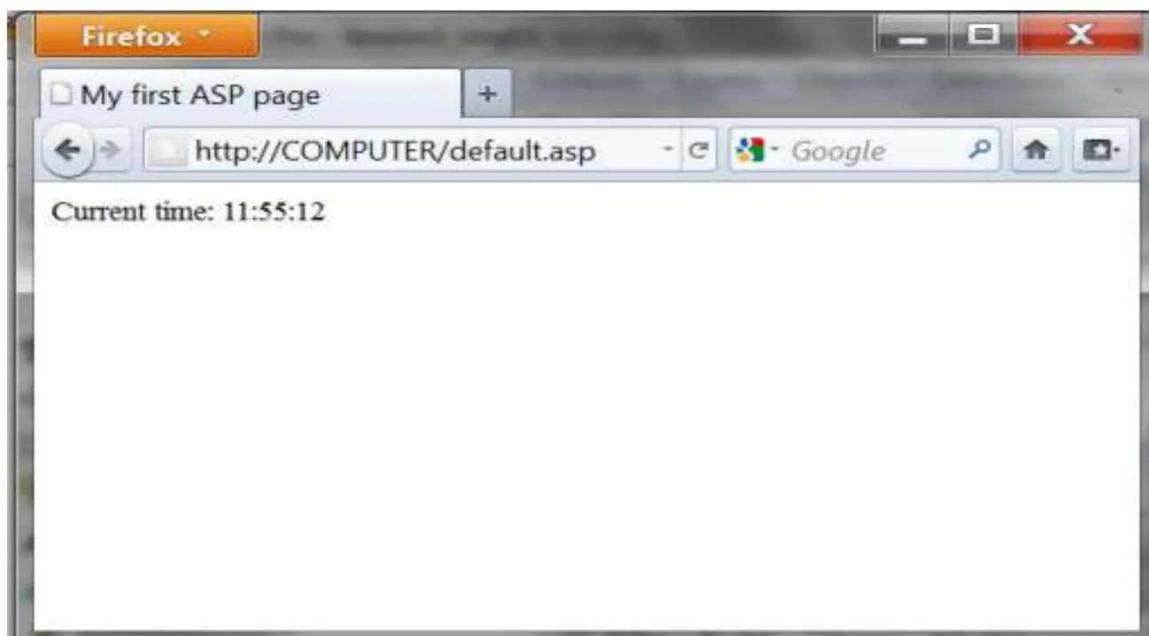
Let's try to extend the example by writing both a *string* and a *function* - separated by `&` - it's done like this:

```
<html>
<head>
<title>My first ASP document</title>
</head>
<body>

<%
Response.Write "<p>Current time: " & Time & "</p>"

</body>
</html>
```

Will look like this in the browser:



And the corresponding HTML code:

A screenshot of a browser developer tools window titled "Source of: http://...". It shows the raw HTML code generated by the ASP page:

```
<html>
<head>
<title>My first ASP page</title>
</head>
<body>
<p>Current time: 11:55:12</p>
</body>
</html>
```

As you can see, the function `Time` returns the current time. There are several functions that relate to time and date. And that's exactly what we'll look at in the next lesson.