

note

default.aspx .designer.cs

You've probably noticed that when you expand an .aspx file, there's a file called `[filename].aspx.designer.cs` as well as `[filename].aspx.cs`.

This file is used by the *Design* view of the Visual Studio interface. All of the code inside it is automatically generated and you will never have to edit it.

important

C# and semicolons (;)

You'll notice that the line of code you add in this example ends in a semi-colon.

Semi-colons are required by C# to indicate the end of a line of code. You might have noticed in *Lesson 2-11: Work with JavaScript* that JavaScript uses the same convention.

You'll notice, however, that the code that defines the method doesn't have a semi-colon at the end of it.

Lines that 'contain' other code using `{` and `}` don't need semi-colons because the `{` and `}` define the area they affect.

Because C# doesn't consider a line of code to be finished until it sees a semi-colon, you can split a long line of code onto multiple lines without causing any problems.

If you are not completing the course incrementally use the sample file: **Lesson 3-1** to begin this lesson.

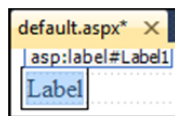
Sample files with the starting point for each lesson are also provided for all of the other lessons in this session.

Lesson 3-1: Change properties with C#

You've already worked with HTML controls and briefly viewed ASP.NET's code-behind files.

In this lesson, you'll use some very basic C# code to change the properties of controls on a web page.

- 1 Open *CSharpTest* from your sample files folder.
- 2 Open *default.aspx* in *Design* view.
- 3 Add a *Label* control from the *Standard* category of the *Toolbox*.



- 4 Set the *ID* of the new *label* control to: **LabelOutput**
- 5 Open the code-behind file of *default.aspx*

You learned how to do this in: *Lesson 1-7: Manage a project with the Solution Explorer*.

- 6 Add some code to set the *Text* property of the label.

You can see the C# code that goes with *default.aspx* on this page. At the moment, the important part is:

```
protected void Page_Load(object sender, EventArgs e)
{
}
```

```
protected void Page_Load(object sender, EventArgs e)
{
}
```

This is a *method*. Methods are the way C# code is organized into pieces that can be run individually.

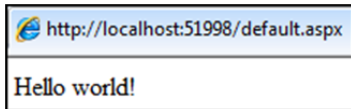
The *Page_Load* method is an *event handler*. An event handler is a special type of method that runs in response to something happening. In this case the event that triggers the event handler is the page loading. You'll learn more about event handlers in: *Lesson 3-2: Add event handlers to Controls*.

The `{` and `}` symbols show where the method begins and ends. Everything between the `{` and `}` is part of the method. As you can see, at the moment the method is empty.

1. Add the following line of C# code in the gap between the `{` and `}`.

```
LabelOutput.Text = "Hello world!";
```

```
protected void Page_Load(object sender, EventArgs e)
{
    LabelOutput.Text = "Hello world!";
}
```



2. View the page in your browser.

You'll see that the *Text* property of the label was changed when the page loaded.

7 Examine what ASP.NET has done.

1. View *default.aspx* in your browser, if it isn't visible already.
2. View the source of the page.

To do this in *Internet Explorer*, right-click on the page and then click *View Source* from the shortcut menu.

You should see a line of code that says:

```
<span id="LabelOutput">Hello world!</span>
```

```
<div>
  <span id="LabelOutput">Hello world!</span>
</div>
```

3. Close your browser and return to *default.aspx* in *Source* view.

Compare the label code here. It should say:

```
<asp:Label ID="LabelOutput" runat="server"
Text="Label"></asp:Label>
```

```
<asp:Label ID="LabelOutput" runat="server" Text="Label"></asp:Label>
```

When a user requests your page, the web server converts the *asp:* controls into HTML that their web browser can understand.

The real power of ASP.NET is in the use of event handlers and C# code to change the properties of elements on the page in response to user interaction, making pages interactive.

8 Add some code to make the label's text bold.

1. Switch back to *default.aspx.cs* (the code-behind file).
2. Add the following line of code on the next line of the *Page_Load* event handler:

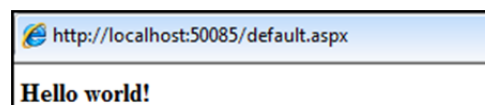
```
LabelOutput.Font.Bold = true;
```

```
protected void Page_Load(object sender, EventArgs e)
{
    LabelOutput.Text = "Hello world!";
    LabelOutput.Font.Bold = true;
}
```

This piece of code is a little different to the last one (see sidebar).

3. View the page in your browser.

You'll see that the text has been made bold by the C# code that you added.



note

String and Boolean properties

The first property you set in this lesson is *Text*, which is a *string*. A string is literally a piece of text.

When you set the property in C#, you put the value in inverted commas (""). C# automatically assumes that anything in inverted commas is a string.

If you'd left out the inverted commas, the project would have failed to build and displayed an error.

The second property is a *boolean*. A boolean can only be either *true* or *false*.

You'll learn about the different types of value in C# in more detail in *Session Five: C# Variables*.