Lesson 2-3: Understand the aspx page structure

Although Visual Studio automatically creates the basic elements of an HTML page for you, it's important to understand the purpose of each element.

In this lesson you'll manually create the different parts of a structured HTML web page.

1 Open *HTMLTest* from your sample files folder.

2 Open *emptypage.aspx* in *Source* view.

This page contains only the bare bones of an *.aspx* web page.

3 Understand the *Page* tag.

The first tag is enclosed in <% tags (see sidebar). This line is never sent to web browsers and is used instead to inform ASP.NET which programming language and code-behind file to use for this page.

```
<%@ Page Language="C#" AutoEventWireup="true" CodeBehi
```

You should never have to change this line in the course of this book.

4 Understand the *DOCTYPE* tag.

The *DOCTYPE* tag is the first thing that a web browser reads. It tells the web browser which web standard was used to create the page so that it can interpret the HTML code correctly.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitic
```

This line is needed because there are several different versions of HTML defined by the Web Consortium or W3C (see sidebar).

Again, you should never have to change this line.

5 Add a *<head>* tag.

The next tags you'll see are opening and closing *<html>* tags. These tell the browser that the code inside the tags is HTML.

Between the *<html>* and *</html>* tags, add the following:

<head>

</head>

```
<html xmlns="http://www.w3.org/1999/xhtml">
    <head>

    </head>
</html>
```

The *head* tag doesn't contain any of the page's content, but instead is used to contain other information about the page. You'll learn more about the *head* tag in: *Lesson 2-4: Use the title, meta, link and script tags*.

6 Add a *<body>* tag.

note

<% %> Tags

In ASP.NET it is actually possible to put C# code directly onto an HTML web page by putting it between <% and %> tags.

ASP.NET's precursor was simply known as ASP and had no code-behind files, so all of the C# code was put on the aspx page inside <% %> tags.

Since ASP.NET introduced code-behind files, it is very rare to see this done today.

trivia

HTML and the W3C

You've probably noticed the links to *w3.org* in the *DOCTYPE* and *html* tags. This is the web site of the World Wide Web Consortium or W3C.

The W3C are responsible for defining international standards for the Internet.

Before the W3C began to define recognized standards, web browsers often disagreed hugely on the correct way to interpret HTML.

Since the introduction of the W3C's standards there are a lot fewer incompatibilities between browsers.

Newer versions of HTML may appear at any time as the W3C refines and improves the HTML and CSS languages. You can see that, in this case, the web page complies with the *XHTML 1.0 Transitional* standard. This is a very mature standard (finalized in 2002) that is well supported by all modern browsers.

Unfortunately, some browsers still don't fully comply with the W3C's standards, so it is still necessary to test your sites thoroughly in every type of browser you think your users may have.

Everything that you actually see in a web browser is contained between two *body* tags. In the last lesson, all of the text you added was nested inside the *body* tags.

After the *</head>* line, add the following:

<body>

</body>

```
<html xmlns="http://www.w3.org/1999/xhtml">
    <head>

    </head>
    <body>

    </body>
</html>
```

7 Add a form.

If you look back at *default.aspx*, you'll notice that the only thing missing from this page is the *form* tag. You'll learn more about forms in: *Lesson 2-12: Work with HTML Forms*.

For ASP.NET controls to work correctly, they need to be nested inside *form* tags. The code between the form tags runs on the server (see sidebar).

Since ASP.NET controls will be displayed on the web page, the *form* tag needs to be inside the *body* tags.

Add the following text inside the *body* tag:

<form id="form1" runat="server">

</form>

```
<html xmlns="http://www.w3.org/1999/xhtml">
    <head>

    </head>
    <body>
        <form id="form1" runat="server">

        </form>
    </body>
</html>
```

You've now created the full framework of an ASP.NET web page, capable of using ASP.NET controls and C# code.

Of course, this is normally created automatically when you add a new page to the project.

8 Save your changes and close Visual Studio.

note

runat="server"

When an HTML tag is marked with *runat="server"* it becomes available to ASP.NET instead of being a normal 'static' tag.

This means that you can modify its contents using C# code, which you will start doing in: *Lesson 3-1: Change properties with C#*.

For this reason, any ASP.NET controls have to be inside a form with *runat="server"* to work correctly.

Although you could place 'static' HTML content outside the *form* tag, it is best to keep all of your page's content between the *<form>* and *</form>* tags.

ASP.NET will only allow you to have one form with *runat="server"* per page.

You'll learn more about forms in: *Lesson 2-12: Work with HTML Forms.*