

HTML/XHTML, CSS, and Dreamweaver Manual

A beginner's guide to HTML/XHTML, CSS, and using Dreamweaver to create an XHTML page.

Table of Contents

Introduction	ii
1. HTML/XHTML	1
1.1 HTML/XHTML tags	2
1.2 XHTML rules	8
1.3 Defining file paths	8
2. CSS	9
2.1 Class selectors	9
2.2 ID selectors	10
2.3 Specifying styles in XHTML	10
3. Dreamweaver	11
3.1 The interface	11
3.2 Creating a new XHTML page	13
3.3 Basic XHTML example	14
3.4 Defining a site	17
3.5 Creating a template	18
3.6 Creating an XHTML page from your template that includes a style sheet	19
4. List of Resources	24

Introduction

This guide will provide you with a basic understanding of HTML, XHTML, and CSS. It will also introduce you to Dreamweaver, an editing tool used to create Web pages. You do not need to have prior knowledge of any of the markup languages or Dreamweaver in order to use this guide. However, if you have had some exposure to HTML, you already have a tremendous head start because the language of XHTML is the same as HTML, with the exception of a few rules that must be followed to create a valid XHTML document.

While this guide describes some of the HTML/XHTML tags that you will use most often, it is not meant to be a reference guide. At the end of the guide is a list of resources, some of which are meant to serve as references for the markup languages we will be exploring. The goal here is to get you started on the basics of how use these tools so you can extend your knowledge of them as far as you would like.

Section 1 - HTML/XHTML

XHTML (Extensible Hypertext Markup Language) is basically an extension of HTML (Hypertext Markup Language). It is a bit stricter than HTML because XHTML was developed to be a transition from HTML to XML (Extensible Markup Language), which has much more rigid standards than HTML. Everything you will learn about HTML applies to XHTML except for the few rules that are required for a valid XHTML document. First we will discuss the “language” of HTML, and then we will examine how HTML and XHTML differ.

HTML is a markup language comprised of a set of tags that tell a browser how to display content. HTML can be written in any text editor including a multitude of HTML editors such as Dreamweaver. Dreamweaver can be used to follow along in the examples used in this guide, and it is advisable to do so since it will help you to learn the syntax of HTML and XHTML. To learn how you can use Dreamweaver to enter code and see how it will display in a browser, refer to section 3.

HTML does not recognize traditional carriage returns in text. It will run text across the entire width of the screen unless a line break is specified using the line break tag or the width of a text tag is specified so that the text only extends to a certain width of the screen. Figure 1 shows the words one, two, three, four listed vertically in an HTML document. Figure 2 shows how those words would appear in a browser. You can see that HTML does not retain the line breaks we placed in the code.

```
<body>
one
two
three
four
</body>
</html>
```

Figure 1 - code view

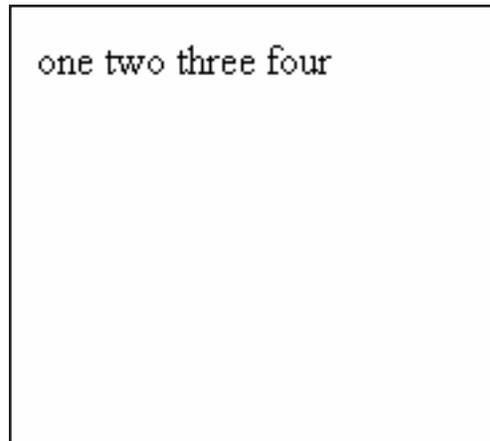


Figure 2 - browser view

1.1 HTML/XHTML Tags

An HTML tag consists of an opening and closing angle bracket containing the tag name, which is a word or abbreviation representing the function of the tag. For example, the HTML tag which begins and ends an HTML document is `<html>`. Tags can contain one or more attributes that affect how the tag is applied to the content between the tags. For example, the size of an image can be increased or decreased using the **height** and **width** attributes. An equal sign connects the attribute value to the attribute. Values are generally enclosed in quotes, but those that consist of one word or number value do not require quotes.

There are three types of tags: starting, ending, and empty tags. Starting tags include only the word or abbreviation of the tag and ending tags have a forward slash before the word or abbreviation e.g. `<html></html>`. The action or function of a tag will be applied to any content between the starting and ending tags.

Empty tags are tags that do not need an end tag. An example is the line break tag `
`, which inserts a line break in the flow of the text.

Not all HTML tags require an ending tag. In addition to empty tags, HTML does allow some tags to be used without an ending tag. For example, the paragraph tag (`<p>`) is often used without a closing tag to insert a blank line in a Web page. Even though the browser will display these types of instances correctly, it is good practice to close all tags, as you will see when we discuss XHTML.

Below is a list of some of the HTML/XHTML tags you will likely use most often.

Structure and Meta tags

Body tag - `<body>`

The body tag is required in an HTML document. It is the main section of an HTML document and contains all the content that will be displayed in the browser.

Division tag - `<div>`

The division tag is used to set apart chunks of content, and is typically used to assign specific formatting to text or objects, especially using style sheets.

Header tag - `<head>`

The header tag is required in an HTML document. It contains information about the HTML document. This tag appears before the body tag and its content is not displayed in the browser.

HTML tag - <html>

The html tag is the first and last tag in an HTML document. It lets the browser know that what is contained between the tags is written in HTML.

Title tag - <title>

The title tag defines the title of a Web page and usually appears in the browser's title bar. This tag is required for valid HTML and must be in the header of the HTML code.

Text tags

Emphasis tag -

The emphasis tag applies the italic style to any text between the starting and ending tags.

Heading tags - <h1>, <h2>, <h3>, <h4>, <h5>, <h6>

Text contained between two heading tags is displayed with a varying font size depending on the header level - h1 being the highest and h6 being the lowest. Thus, text enclosed between h1 tags will be displayed much larger than text enclosed between h3 tags.

Line break tag -

The line break tag inserts a line break at the point in the text where the tag is placed. This tag is referred to as an "empty" tag - it does not require an end tag.

Paragraph tag - <p>

Content enclosed in paragraph tags is automatically separated from other content with a blank line.

Strong tag -

The strong tag applies the bold style to the text between its starting and ending tags.

Image tag -

The image tag allows you to insert images into an HTML document. This is possible with the `src` attribute that requires the path to the image file. Other attributes that are widely used with image tags are **alt** or **title**, **height**, and **width**.

The **alt** and **title** attributes let you add text to an image that users see when they mouse over the image. Using these attributes will make your Web pages more accessible for users because the text will display even if the image does not. This will help users to better understand how your Web page should look and what it is about.

The **width** and **height** attributes specify the dimensions of the image. These values tell the browser the size of the image so it can display the contents of the Web page accordingly. This will help to increase the speed with which an image is downloaded onto the Web page. Figure 3 shows an example of code used to insert an image into a Web page.

```

```

Figure 3 - Example of code used to insert an image into a Web page

Note: Two other popular attributes for the image tag, the **border** and **align** attributes, have been deprecated. This means that the tag may still be supported by a browser, but it is recommended to discontinue using it because it will probably be phased out in the near future. It is preferable to use these attributes within a style sheet. The **border** attribute is often used to set the border around an image to zero so that it doesn't appear (browsers automatically put a border around images). The **align** attribute is used to position an image on a Web page.

Link tags

Anchor tag - <a>

The anchor tag is most often used with the **href** attribute to specify a hyperlink to another document or Web page. The value of the **href** attribute is the URL to the hyperlink's destination. The URL must be enclosed in double quotes. The text between the opening and closing anchor tags is the text that appears in the browser. Figure 4 displays the HTML code for a hyperlink to the Museum of Modern Art, and Figure 5 shows how the hyperlink would be displayed in a browser.

```
<a href="http://www.moma.org">My favorite museum</a>
```

Figure 4 - Example of code for a hyperlink

[My favorite museum](http://www.moma.org)

Figure 5 - Example code as viewed in a browser

Note: Notice that the browser will underline the link and set the link color as the familiar dark blue commonly used for hyperlinks. You can change this format with various attributes, preferably using CSS.

Table tags

HTML tables have been, and are, used as a way to arrange content on HTML pages. However, there is a growing movement to use tables in the traditional sense, which is to display data, and not to use them as a mechanism to lay out content on an HTML page. HTML tables consist of headers, rows, and cells. Figure 6 shows how table elements are defined using HTML vocabulary. The table has three rows (four rows if you include the heading row) and three columns. Each column has a heading, and each row has three cells.

heading, cell 1	heading, cell 2	heading, cell 3
row 1, cell 1	row 1, cell 2	row 1, cell 3
row 2, cell 1	row 2, cell 2	row 2, cell 3
row 3, cell 1	row 3, cell 2	row 3, cell 3

Figure 6 - Table elements

A table in HTML is defined with a <table> tag. A table row is designated with the <tr> tag, and a table cell is designated with a <td> tag. Table headings use the <th> tag. The table in Figure 6 would be constructed in HTML using the code shown in Figure 7.

```
<table>
<th>heading, cell 1</th><th>heading, cell 2</th><th>heading, cell 3</th>
<tr><td>row 1, cell 1</td><td>row 1, cell 2</td><td>row 1, cell 3</td></tr>
<tr><td>row 2, cell 1</td><td>row 2, cell 2</td><td>row 2, cell 3</td></tr>
<tr><td>row 3, cell 1</td><td>row 3, cell 2</td><td>row 3, cell 3</td></tr>
</table>
```

Figure 7 - HTML code used to create the table in Figure 6

Figure 8 displays how the HTML code for our table would appear in a browser. This is actually the design view in Dreamweaver, which we will look at a little later when we discuss Dreamweaver's interface.

heading, cell 1	heading, cell 2	heading, cell 3
row 1, cell 1	row 1, cell 2	row 1, cell 3
row 2, cell 1	row 2, cell 2	row 2, cell 3
row 3, cell 1	row 3, cell 2	row 3, cell 3

Figure 8 - A browser's rendering of the code in Figure 7

Now we can add some formatting to our table using attributes. Some of the more widely-used attributes for tables include **align**, **border**, **width**, **cellpadding**, and **cellspacing**.

The **align** attribute can be used with the row and cell tags to control where the text appears in the cell.

The **border** attribute is used with the table tag and places a border around the outside of the table and between the rows and cells within the table. If you do not wish to have a border in your table, you can set the border attribute equal to zero.

The **width** attribute can be used with the table and row tags. Used with the table tag, the **width** attribute can be specified as a percentage of the screen width or a number of pixels. Used with the row tag, the width can be specified as a percentage of the table width or a number of pixels.

```
<table border="1" width="50%" cellpadding="10" cellspacing="5">
<th>heading, cell 1</th><th>heading, cell 2</th><th>heading, cell 3</th>
<tr align="left">
<td>row 1, cell 1</td><td>row 1, cell 2</td><td>row 1, cell 3</td>
</tr>
<tr align="center">
<td>row 2, cell 1</td><td>row 2, cell 2</td><td>row 2, cell 3</td>
</tr>
<tr align="right">
<td>row 3, cell 1</td><td>row 3, cell 2</td><td>row 3, cell 3</td>
</tr>
</table>
```

Figure 9 - The code for the table in Figure 7 with some formatting added

Now, let's look at what we've added to our table in Figure 9. If you look at the attributes of the table tag, you will see that the table has a border equal to one pixel, and the width of the table is set to fifty percent of the width of the screen. The **cellpadding** attribute is equal to 10 pixels and **cellspacing** is equal to 5 pixels.

Cellpadding affects the space between the contents of the cell and its border. If cellpadding is not specified, the amount of space is set to one pixel. If the cellpadding is set to zero, then the cell contents will touch the cell borders.

Cellspacing affects the space around the cells themselves. This includes the space between cells and the space between the cells and the table border.

Notice the attributes in the row tags. The cell contents of row one are aligned to the left, the cell contents of row two are centered, and the cell contents of row three are aligned to the right.

Figure 10 below displays how the code from Figure 9 would look in a browser. Notice the space between the cells, and the space between the text in the cells and the cell borders.

heading, cell 1	heading, cell 2	heading, cell 3
row 1, cell 1	row 1, cell 2	row 1, cell 3
row 2, cell 1	row 2, cell 2	row 2, cell 3
row 3, cell 1	row 3, cell 2	row 3, cell 3

Figure 10 - The table from Figure 9 as it would be displayed in a browser

1.2 XHTML Rules

Now we get to the differences between HTML and XHTML. Everything you have learned so far about HTML applies to XHTML. There are three main rules XHTML requires for a valid document:

- 1) All tag and attribute names must be in lowercase.
- 2) Attribute values must be enclosed in double quotes.
- 3) Every starting tag must have an end tag.

Since all tags must have an end tag, what do we do about the empty tags such as the tags we learned about above (``, `
`)? Empty tags will use a forward slash before the ending angle bracket with a space before the slash. So, the line break tag will become `
`, and the image tag will become `` or ``.

1.3 Defining File Paths

There are two ways to specify paths to image and document files: relative paths and absolute paths.

An absolute path includes the entire URL path for a linked object.

```

```

A relative path is a URL path that is relative to the home directory containing your pages.

```

```

If the image is located in the same directory as the Web page in which it linked, the path would just be the filename of the image.

```

```

If the image is located in a directory above the Web page, the path URL contains two dots and a forward slash (`../`) to tell the browser that the image is up one directory level.

```

```

This image is located two directory levels up from the Web page directory.

Section 2 - CSS

CSS (Cascading Style Sheets) are a way to create a style for one or more XHTML documents with one style sheet. Using style sheets, you can apply a style or format to text, hyperlinks, images, basically any of the content in your Web page. They are also a way to separate content from style and layout, which is an increasingly important goal in creating Web pages. This is important not only for accessibility for users, but also for ease of creating and updating Web pages, since a change in a particular style will only require editing one page instead of all of the pages that use that style.

The syntax for a CSS style is

```
Selector {Property: Value}
```

For example, a style that applies the sans serif font to all content within paragraph tags is

```
p {font-family: "sans serif"}
```

Values with more than one word require double quotes. If there is more than one property specified, they must be separated with a semicolon. Multiple selectors that are separated by commas can be specified for the same properties:

```
p, div {text-align:center; color:black}
```

There are two types of selectors, class selectors and ID selectors.

2.1 Class Selectors

The class selector utilizes the period to specify styles.

By using the class selector, you can specify multiple styles for one XHTML element.

Two types of paragraphs can be designated with the following code:

```
p.right {text-align:right}  
p.center {text-align:center}
```

The first paragraph style will align the text in that paragraph to the right of the screen, and the second paragraph style will center the text in that paragraph.

The class attribute is required in the paragraph tag in the XHTML code:

```
<p class="right"></p>  
<p class="center"></p>
```

The selector can be omitted in the CSS code, leaving

```
.center {text-align:center}
```

This style can then be applied to any XHTML element.

2.2 ID Selector

The ID selector uses the pound sign. An example of a CSS style titled “text” with the ID selector would look like this:

```
#text {max-width: 600px}
```

This style will set the width of the text in a paragraph to 600 pixels. Here is how the style is inserted into the XHTML code:

```
<p id="text"></p>
```

ID selectors can be applied only once in the XHTML code. So, the style “text” can be used in one distinct XHTML tag.

2.3 Specifying styles in XHTML

There are three ways to specify a style:

1. Inside an XHTML tag as an attribute - Inline
2. Inside a specific XHTML document - Internal
3. In an external file that is linked to one or more XHTML documents - External

An inline style should only be used when a specific XHTML element style differs from another specified style. Inline styles are defined with the style attribute.

```
<p style="color: red"></p>
```

An internal style sheet should be used when an XHTML document has a unique style that differs from an external style sheet. The CSS styles are defined within the header of the HTML document.

```
<head>  
<style type="text/css">  
p {text-align: center; color: red}  
h1 {color: blue}  
</style>  
</head>
```

The type attribute in the style tag tells the browser that what follows is a style sheet that is to be applied to the XHTML.

An external style sheet is defined in the head of an XHTML document:

```
<head>  
<link rel="stylesheet" type="text/css" href="mystyle.css" />  
</head>
```

This definition uses the link tag to link the XHTML document to the style sheet. The filename of the style sheet is mystyle.css.

Section 3 - Dreamweaver

Dreamweaver is one of the more popular HTML/CSS editors. It is an application designed for people with various levels of Web design skills. You can be a beginning user of HTML or you can be an advanced Web designer. Dreamweaver is convenient for experienced Web designers because it can insert much of the HTML code for you.

Note: The information in this section pertains to Dreamweaver 8. The current version of Dreamweaver is CS3. The previous version and the current version will have the same functions, but CS3 may have a different look to some of its windows and features.

3.1 The Interface

The Menu and Insert Bars at the top of the Dreamweaver window (Figure 11) let you select HTML tags and objects such as links and images. Dreamweaver will insert the HTML and will prompt you to insert the content.

The menus allow you to perform various operations in Dreamweaver. Some of these will be familiar to you from your use of other programs. For example, the File menu lets you create new files, open existing files, and save files, to name a few of the operations. The Edit menu allows you to choose from various operations for editing HTML code. The Insert menu contains options to insert tags, images, links, and other objects into your code. The Text menu lets you select operations that affect text, such as paragraph formats, fonts, font sizes, font styles, and even colors.

Different object (icon) toolbars can be selected with the drop-down selector (where the word “Common” is displayed on the Insert bar). For example, the Common toolbar allows you to insert many of the same objects found in the Insert menu. The Forms toolbar lets you insert HTML form objects, such as text fields, checkboxes, and radio buttons. The Data and Text toolbars contain objects and styles for data and text.



Figure 11 - Menu bar and Insert bar

The Document toolbar (Figure 12) appears above the document window. The filename of the document appears in the tab on the left side. You can have multiple files open and switch between them easily by clicking on that file’s tab. The three buttons below that tab will be the ones you will probably use most often.

The code button displays the HTML code of the document, and the design button displays the visual editor in the window which shows how the page would look in a browser. The split button displays both the code and design view at the same time. The icons on the right allow you to perform various operations such as change the view options or validate your code against current standards, but the icon that resembles the earth is especially useful because it will let you preview your code in the browser of your choice.

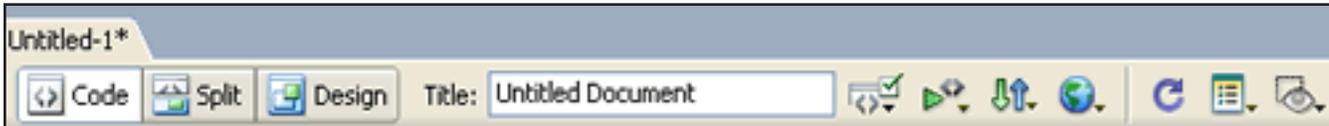


Figure 12 - Document toolbar

At the bottom of the Dreamweaver window is the Properties inspector that lets you edit properties of objects without having to edit the HTML code. Figure 13 below shows the Properties inspector containing information for a smiley face image. You can use the inspector to edit many of the properties of the image including the width and height of the image, the source of the image, alternate text, and any border we would like to apply to the image.

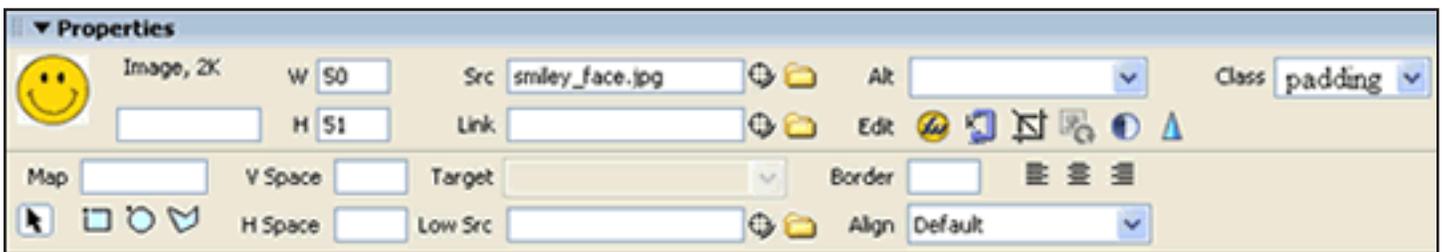


Figure 13 - Properties inspector

On the right side of the Dreamweaver window is the Files panel (Figure 14) that will display the file structure of your computer and let you access the HTML, CSS, and image files you are using to create your Web pages.

You can also use this panel to upload your files to a server from your local computer, or you can edit pages directly on the Web server.

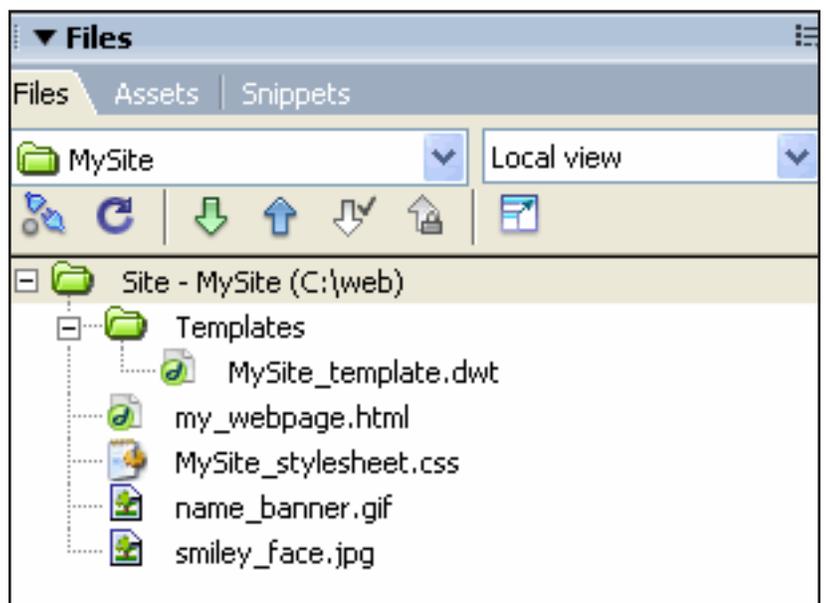


Figure 14 - Files panel

3.2 Creating a New XHTML Page

To create a blank XHTML page, go to the File menu and select **New**. The New document window will appear (Figure 15). Select the **General** tab if it is not already selected. Under the Category heading, select **Basic page**. Under the Basic page category, select **HTML**. Or you can select other types of pages depending on what type of document you want to create. For now, we just want to create some basic XHTML code.

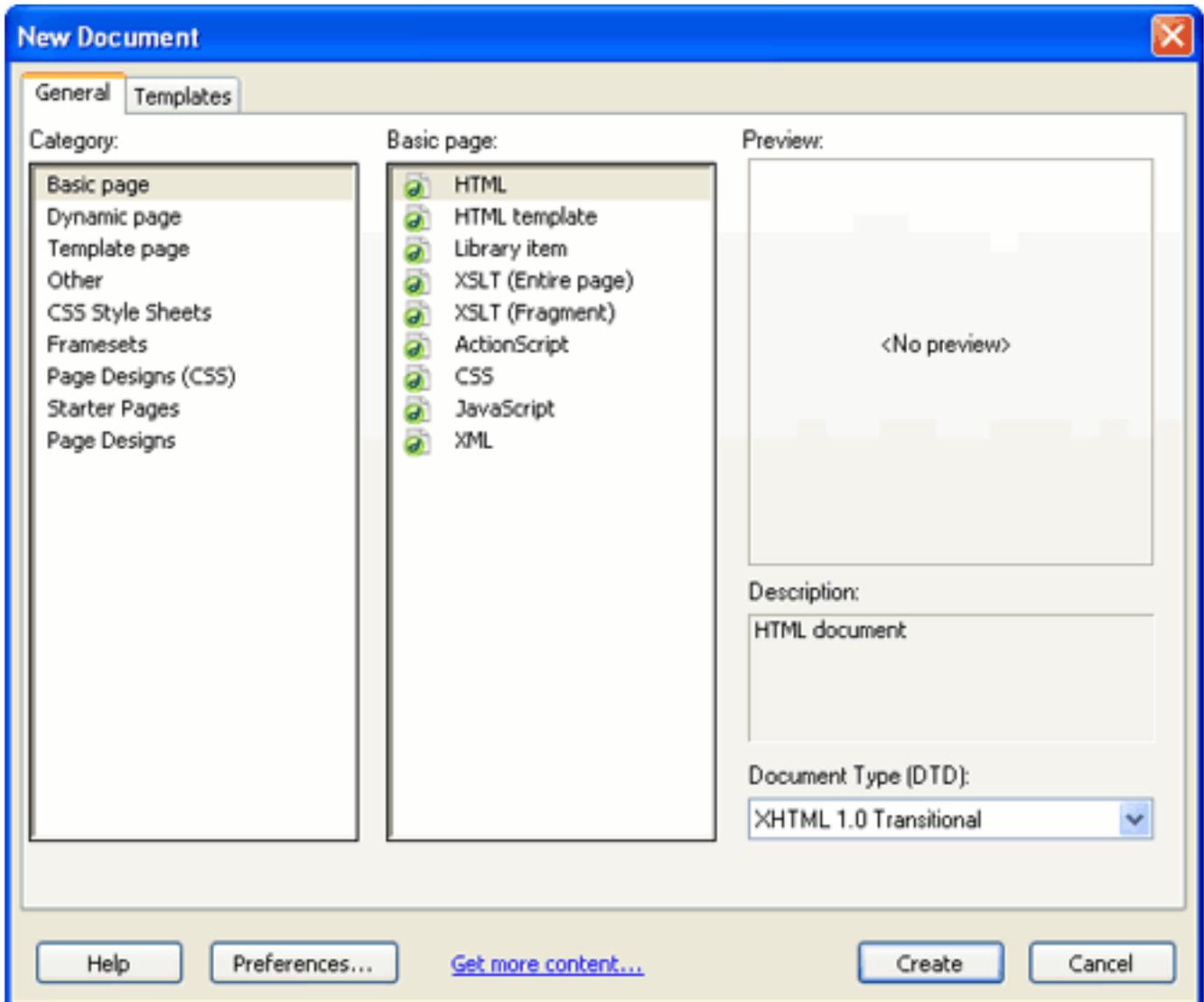


Figure 15 - New Document window

3.3 Basic XHTML Example

Figure 16 shows code for an example of a basic XHTML page which contains some text, an image, hyperlinks, and a small table. Notice the various tags used to surround the text. We have used a level two heading, a paragraph tag, and a div tag. My Website uses the level two heading because it serves as the heading for the Web page and should stand out. The use of paragraph and div tags was arbitrary. We could have exchanged the paragraph and div tags.

You can insert this code into a new page using Dreamweaver and then preview it to see how it would look in a browser.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<title>My webpage</title>
</head>

<body>
<h2>My Website</h2>
<p>Hello!<br />
Welcome to my website.</p>
<div>Insert personal summary here.</div>

<br />
<a href="http://www.moma.org">My favorite museum</a><br />
<a href="http://www.italianfoodforever.com/">My favorite food</a><br />

<table>
<tr><td>address</td><td>123 Main St. Anytown USA</td></tr>
<tr><td>phone</td><td>(123) 456-7890</td></tr>
<tr><td>email</td><td>my_email@yahoo.com</td></tr>
</table>

</body>
</html>
```

Figure 16 - Basic XHTML example

Rather than typing all of the code into a page, you can utilize some of Dreamweaver's shortcuts by going to the Menu bar or the Insert bar mentioned in the interface section to select the object you would like to insert into your page.

For example, if you click on the Insert image icon in the Insert bar (Figure 17) and select **Image**, the Select Image Source window appears (Figure 18). You can now browse to the image you want to insert. When you click **OK**, another window will appear asking you to insert alternate text and a long description. Long description will link to another document if you have an especially lengthy description for your image.



Figure 17 - Insert image icon

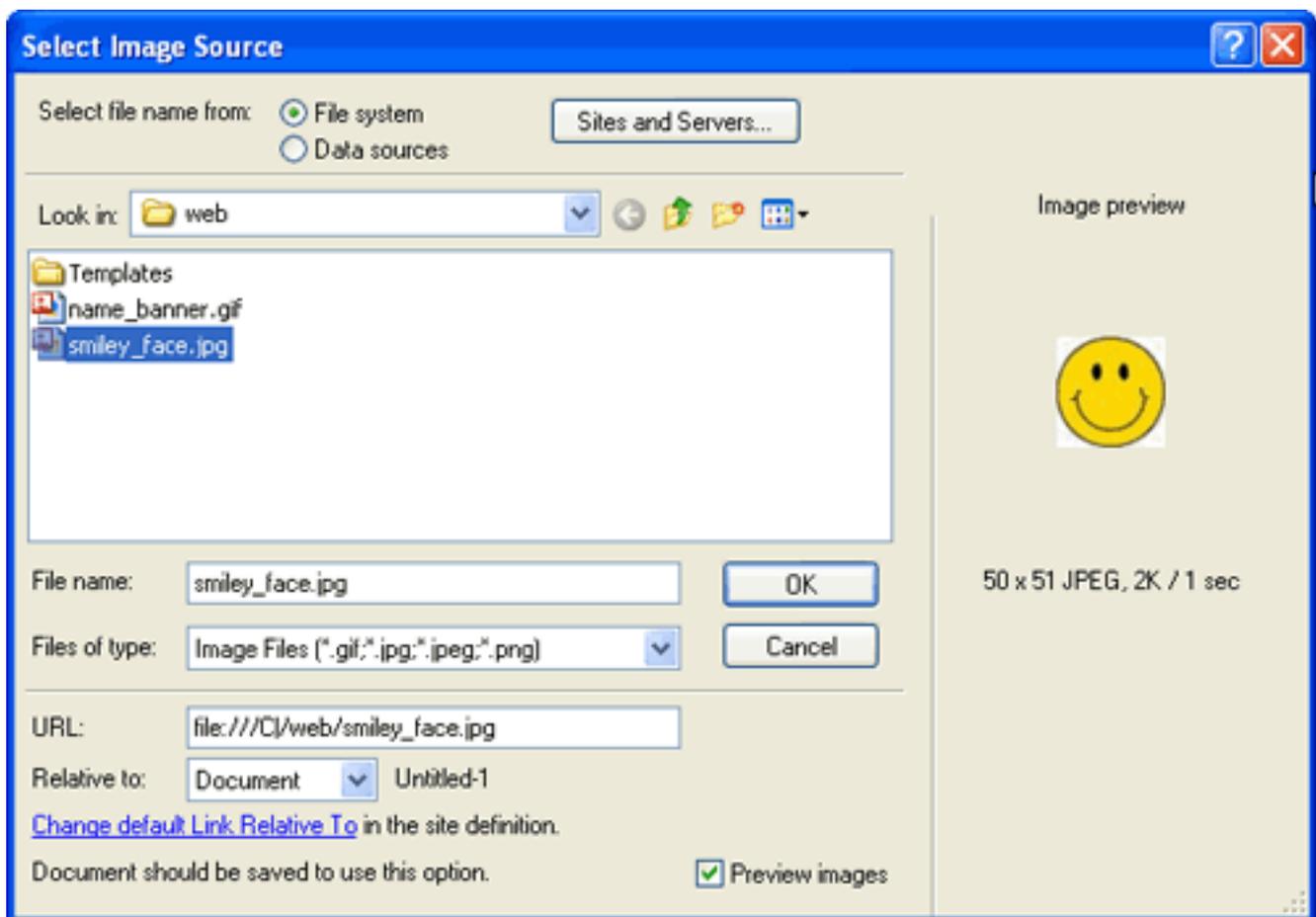


Figure 18 - Select Image Source window

Figure 19 shows how the design view of Dreamweaver will display your code. Visible dotted lines represent the outline of the div section and the rows and cells of the table.

You can also click on the earth icon in the Document toolbar (Figure 20) to preview the code in a browser window. This way you will not see the section outlines that Dreamweaver displays.



Figure 20 - Preview/Debug in browser icon

My Website

Hello!
Welcome to my website.

Insert personal summary here.



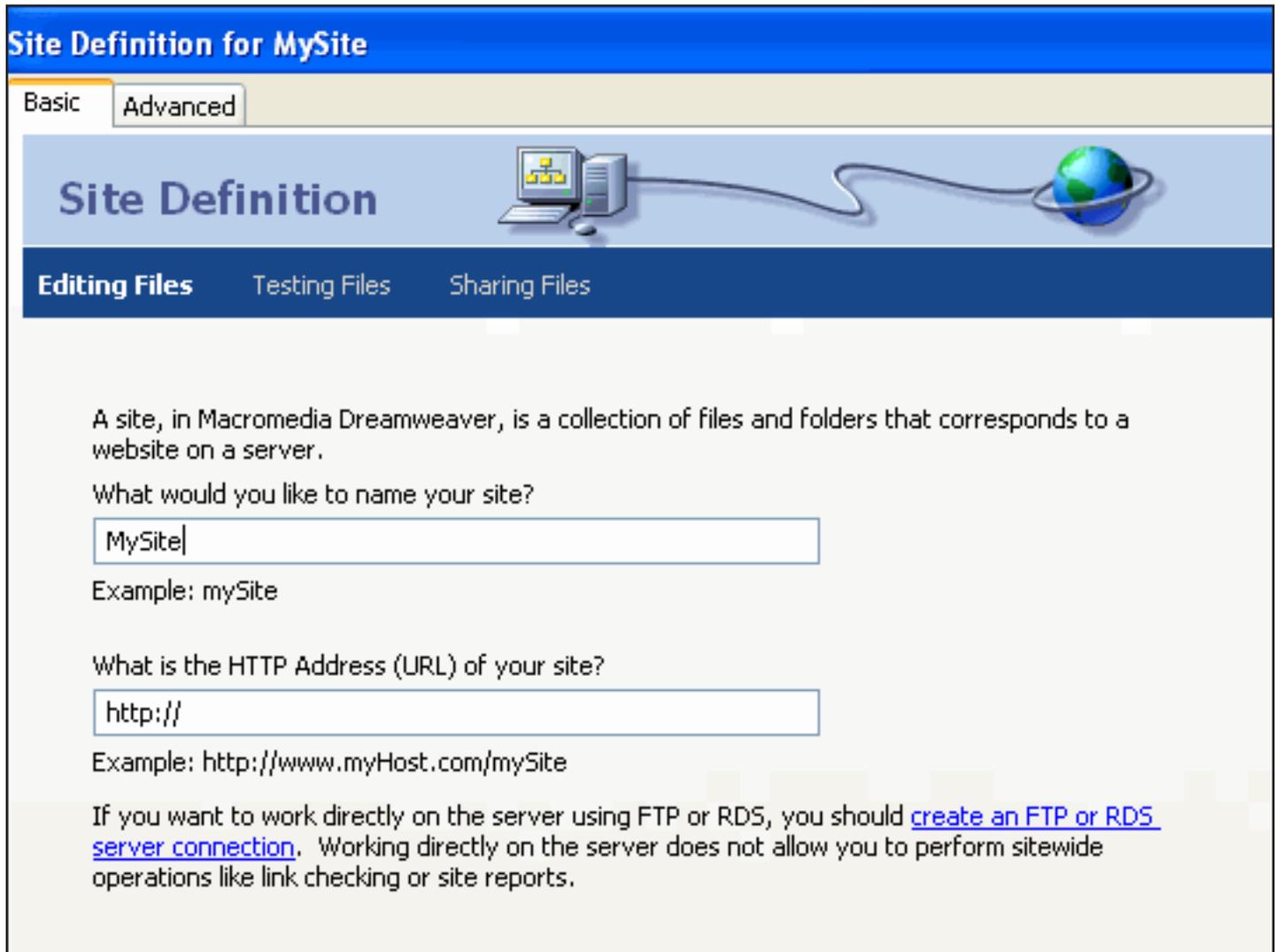
[My favorite museum](#)
[My favorite food](#)

address:	123 Main St. Anytown USA
phone	(123)456-7890
email	my_email@yahoo.com

Figure 19 - The XHTML example as it would appear in the design view of Dreamweaver

3.4 Defining a Site

Defining a site will allow you see your files in the file panel to the right of the screen. To define a new Web site in Dreamweaver to which you can add Web pages, click on the **Site** menu and select **New site**. A window will appear like the one in Figure 21 that will allow you to define a new site. Enter a name for the site and a URL for the site if your site is located on a server. If you will be creating files on your local machine, then leave the URL section blank.



Site Definition for MySite

Basic Advanced

Site Definition

Editing Files Testing Files Sharing Files

A site, in Macromedia Dreamweaver, is a collection of files and folders that corresponds to a website on a server.

What would you like to name your site?

Example: mySite

What is the HTTP Address (URL) of your site?

Example: http://www.myHost.com/mySite

If you want to work directly on the server using FTP or RDS, you should [create an FTP or RDS server connection](#). Working directly on the server does not allow you to perform sitewide operations like link checking or site reports.

Figure 21 - Site Definition window

The next screen will ask you if you would like to use a server technology. Select **No** and advance to the next screen. Select whether you will be working on your files on the server or on your local network. Browse to the directory where your files will be stored. On the next screen, select how you will connect to the server. If you are editing on your local machine, select **None**. The final screen will summarize your selections. Select **Done**. Your site name should appear in the file structure to the right of the screen.

You can now create pages and add them to your site.

3.5 Creating a Template

Now that you have defined a site, you can create a template that you can use to create all of the Web pages in your Web site. The template can include any text or objects you would like to appear on the pages of your site.

To create a template, go the File menu and select **New**. The New Document window will open. Under Category, select **template page** and then **HTML template**.

You can now add whatever you would like to be part of the template. In the example we are using for this exercise, we have added an image of a blue banner that would display your name (Figure 22). The banner will appear at the top of our page.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<!-- TemplateBeginEditable name="doctitle" -->
<title>This is my website</title>
<!-- TemplateEndEditable -->
<!-- TemplateBeginEditable name="head" -->
<!-- TemplateEndEditable -->
</head>
<body>

<!-- TemplateBeginEditable name="Edit_content" -->
Edit_content<!-- TemplateEndEditable -->
</body>
</html>
```

Figure 22 - Example template page

Once you have finished creating the content for the template, you will have to decide where you want to insert an editable region. An editable region is a section of the template that you will be able to add and edit content in a Web page that uses your template. Once you use the template to create another Web page, you will not be able to edit the template content on the new Web page. Template content can only be edited in the template page. To insert an editable region, go to the Insert menu, select **Template Objects** and then **Editable Region**.

When you save the template, Dreamweaver creates a template folder and saves the template file in that folder. The template file will be easy to locate because Dreamweaver assigns it a file extension of “dwt.” Thus, your template filename will look something like this: my_template.dwt.

3.6 Creating an XHTML Page From Your Template That Includes a Style Sheet

Go to the File menu and select **New**. The New Document window will appear. Select the **Templates** tab. Under the “Templates for” heading, select the site whose template you want to use to create a Web page. The site you have chosen will appear above the second panel, and any templates you have created for that site will appear under the Site heading. In our case, we only have one template for MySite. Select the template you wish to use to create your page. A preview of the template will appear in the Preview panel. Be sure to check the box next to “Update page when template changes.” Then any changes you make to the template will be applied to any pages that use that template. Click **Create**. You can now add code to the editable regions of your page.

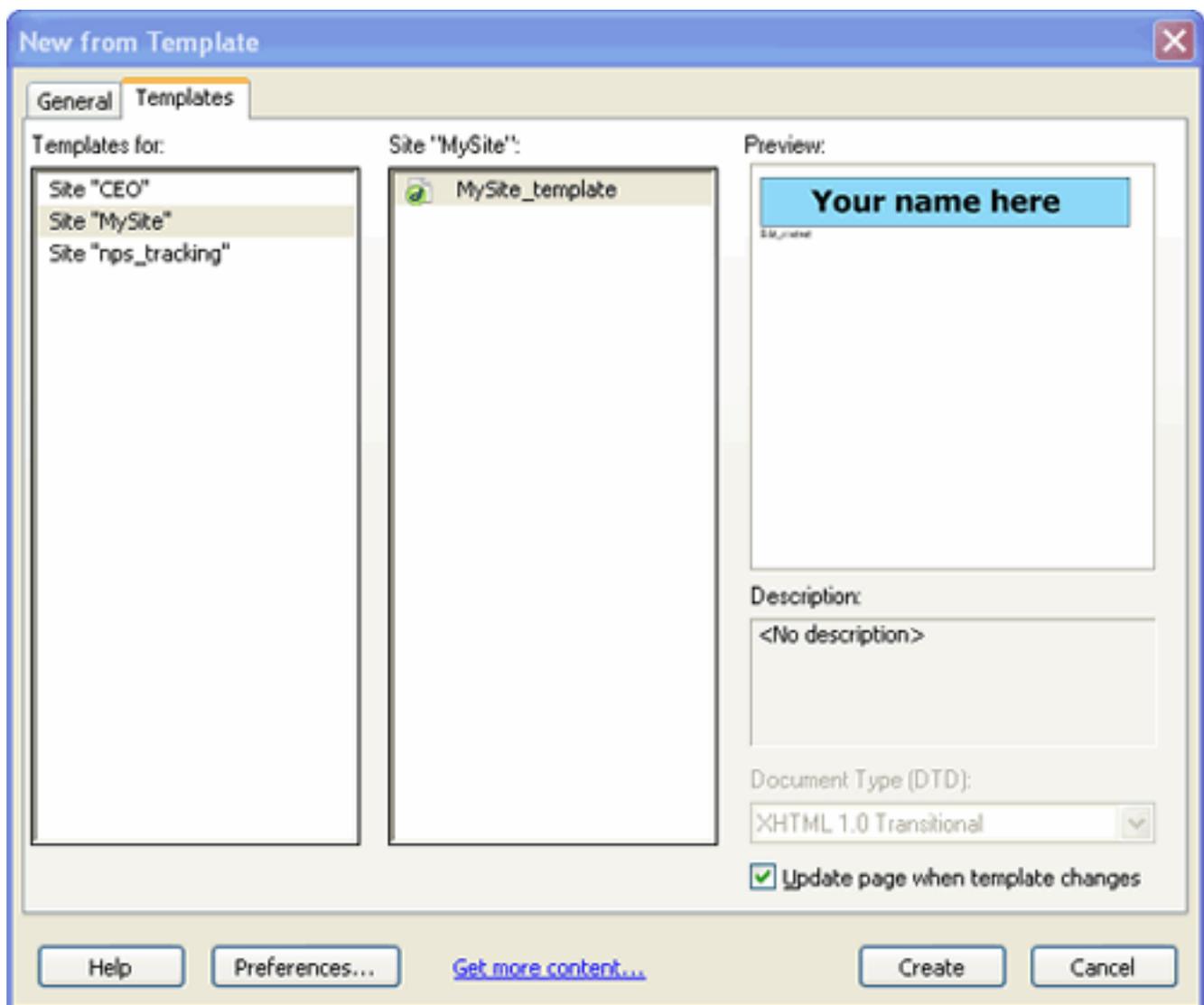


Figure 23 - New from Template window

Below is the code for our example page. The code is the same as in the previous example, except the template we created earlier is now part of the Web page and a style sheet has been applied to the page. Notice the code from the template is grayed out in the Web page since it is not editable. Also, notice the `link` tag that links the style sheet to the XHTML document. The `rel` attribute describes the relationship of the linked document to the XHTML document. The `type` attribute specifies the content type of the linked file. In this case, the CSS document is a text file, but has an extension of “css.” The value of the `href` attribute is the URL of the style sheet. In this case, the URL is just the filename since the file is in the same directory as the XHTML document.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml"><!-- InstanceBegin
template="/Templates/MySite_template.dwt" codeOutsideHTMLOIsLocked="false" -->
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<!-- InstanceBeginEditable name="doctitle" -->
<title>This is my website</title>
<link rel="stylesheet" type="text/css" href="MySite_stylesheet.css" />
<!-- InstanceEndEditable -->
<!-- InstanceBeginEditable name="head" -->
<!-- InstanceEndEditable -->
</head>
<body>

<!-- InstanceBeginEditable name="Edit_content" -->
<h2>My Website</h2>
<p>Hello!<br />
Welcome to my website.</p>
<div id="center">Insert personal summary here.</div>

<br />
<a href="http://www.moma.org">My favorite museum</a><br />
<a href="http://www.italianfoodforever.com/">My favorite food</a><br />

<table class="padding">
<tr><td>address</td><td>123 Main St. Anytown USA</td></tr>
<tr><td>phone</td><td>(123) 456-7890</td></tr>
<tr><td>email</td><td>my_email@yahoo.com</td></tr>
</table>

<!-- InstanceEndEditable -->
</body>
<!-- InstanceEnd --></html>
```

Figure 24 - Code for template page that includes CSS

Figure 25 displays the style sheet for our example. There are two styles applied to the body tag, which will affect all the content between the body tags. The **font-family** property applies the fonts in the order in which they are listed. The Verdana font will be applied to the text, if Verdana is not available on a user's computer, then Arial is applied, etc. The **width** property limits the width of the content of the body tag to 600 pixels.

The table on our example page is set to half the width of the screen width. It also has a border that is 5 pixels wide.

Next on the style sheet is an ID selector named "center," which will center the text within any text container in which the selector is applied, but remember that it can be applied only once in the code.

The class selector "padding" inserts 15 pixels of space along the top and bottom of whatever object the class applies, in this case the smiley face image and the table on our page.

The last two selectors in the style sheet affect the appearance of the hyperlinks on our page. The "a:link" selector affects the style of links before a user has clicked on it. The "a:visited" selector affects the style of links that have been clicked. The **text-decoration** property with a value of "none" reverses the underlining that browsers often apply to links. The **color** property sets the color of the links. The color values are the hexadecimal notation used for Web colors. The List of Resources at the end of the manual provides a website with a chart of colors for use on the Web.

```
body {
font-family: Verdana, Arial, Helvetica, sans-serif;
width: 600px;}

table {
width: 50%;
border: 5px;}

#center {text-align: center}

.padding{
padding-top: 15px;
padding-bottom: 15px;}

a:link {
text-decoration: none;
color: #3366ff;}

a:visited {
text-decoration: none;
color: #663399;}
```

Figure 25 - Style sheet for the template page

Figure 26 shows how our Web page looks in the design view of Dreamweaver. Notice the space inserted at the top and bottom of the image, the border around the table, and the style of the hyperlinks that we specified in our style sheet.



Figure 26 - Our Web page in the design view of Dreamweaver

Congratulations! You have successfully created a Web page using a template and a style sheet! Now that you have some basic knowledge of the tools you need to create Web pages, you can explore these tools further to build more complex Web pages. The List of Resources on the next page will point you to other online and prints materials that will contain more in-depth information about these tools. Good Luck!

Section 4 - List of Resources

XHTML

HTML, XHTML, and CSS (Visual Quickstart Guide) 6th edition. Elizabeth Castro.

HTML & XHTML: The Definitive Guide - 6th edition. Chuck Musciano and Bill Kennedy. O'Reilly Inc.

A complete of XHTML tags can be found on the HTML Dog website:
<http://www.htmldog.com/reference/htmltags/>

The W3Schools offer online tutorials for HTML and XHTML at:
<http://www.w3schools.com/xhtml/default.asp>

CSS

CSS: The Definitive Guide. Eric Meyer. O'Reilly Inc.

Beginning CSS Web Development: From Novice to Professional. Simon Collison

A complete list of CSS properties and their possible values can be found on the HTML Dog website: <http://www.htmldog.com/reference/cssproperties/>

W3Schools tutorial: <http://www.w3schools.com/css/default.asp>

Dreamweaver

Dreamweaver 8: The Missing Manual. David Sawyer McFarland. (There is also an edition for CS3 - the current version of Dreamweaver)

Dreamweaver CS3 Bible. Joseph W. Lowery.

Dreamweaver 8 for Dummies. Janine C. Warner. (Also for Dreamweaver CS3).

Web Safe Color Chart

Web Source Color Chart: http://www.web-source.net/216_color_chart.htm