

HTML Chapter 1

By Dwight Watt

Copyright 2003-2014

HTML is an acronym for Hypertext Markup Language. It is hypertext because we can control the text or images to use it as a link to something else. It is a markup language as it is merely a way to make our text appear not plain. Web Pages are written in HTML as a base language and sometimes use other languages also to work with the HTML such as Flash, JavaScript, Java, XML, PHP, etc. Although HTML is considered as many as a language it is more just a document markup method. You will be describing the document you want to display in a browser such as Internet Explorer or Chrome, supplying the codes to make that document appear the way you want them to appear.

In this chapter we will be looking at how to do a very basic document in HTML that will display a paragraph and a heading line before the paragraph.

Figure 1 shows the document as we want it to appear in a browser. Figure 2 shows the document as we will write it in HTML. We will look at this code line by line shortly

Figure 1

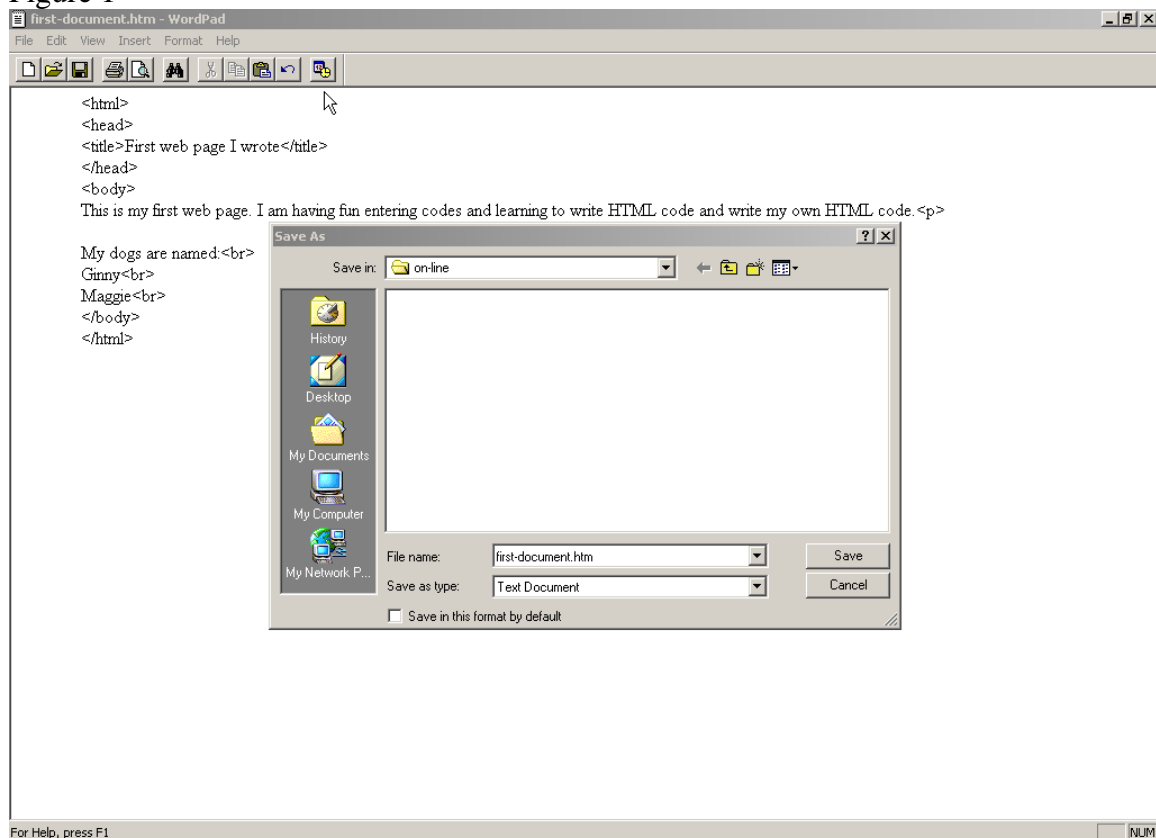


Figure 2

<html>

```

<head>
<title>First web page I wrote</title>
</head>
<body>
<p>This is my first web page. I am having fun entering codes and learning to
write HTML code and write my own HTML code.</p>

My dogs are named:<br>
Ginny<br>
Maggie<br>
</body>
</html>

```

You may be asking why write this HTML code and learn HTML when there are lots of HTML code generators out there such as Word, Dreamweaver, etc. The first reason to actually learn HTML is that even if you use one of the generators you will be able to correct errors it may generate and secondly fix the code to do something you want, but it does not understand. The second reason is that generators are very verbose. You can write the same code in a much smaller code than the generators. Typically you want your HTML code, web page, to go across the Internet or a network. The smaller and more efficient the HTML code is, the quicker the receiver will receive the code, web page, and the quicker the user will see the web page. In figure 3 is the HTML code generated by Microsoft Word to do the same web page as shown in figure 1. Notice how much shorter the code in figure 2 is than in figure 3. Also figure 2 should be more readable, or easier to read, than figure 3 although you have not learned about the codes yet.

Figure 3

```

<html xmlns:o="urn:schemas-microsoft-com:office:office"
xmlns:w="urn:schemas-microsoft-com:office:word"
xmlns="http://www.w3.org/TR/REC-html40">

<head>
<meta http-equiv=Content-Type content="text/html; charset=windows-
1252">
<meta name=ProgId content=Word.Document>
<meta name=Generator content="Microsoft Word 10">
<meta name=Originator content="Microsoft Word 10">
<link rel=File-List
href="This%20is%20my%20first%20web%20page_files/filelist.xml">
<title>This is my first web page</title>
<!--[if gte mso 9]><xml>
<o:DocumentProperties>
<o:Author>Dwight</o:Author>
<o:LastAuthor>Dwight</o:LastAuthor>
<o:Revision>1</o:Revision>
<o:TotalTime>0</o:TotalTime>
<o:Created>2003-12-01T03:52:00Z</o:Created>
<o:LastSaved>2003-12-01T03:52:00Z</o:LastSaved>
<o:Pages>1</o:Pages>
<o:Words>29</o:Words>
<o:Characters>123</o:Characters>

```

```

<o:Lines>2</o:Lines>
<o:Paragraphs>1</o:Paragraphs>
<o:CharactersWithSpaces>151</o:CharactersWithSpaces>
<o:Version>10.4219</o:Version>
</o:DocumentProperties>
</xml><![endif]--><!--[if gte mso 9]><xml>
<w:WordDocument>
  <w:SpellingState>Clean</w:SpellingState>
  <w:GrammarState>Clean</w:GrammarState>

<w:DisplayHorizontalDrawingGridEvery>0</w:DisplayHorizontalDrawingGridE
very>

<w:DisplayVerticalDrawingGridEvery>0</w:DisplayVerticalDrawingGridEvery
>
  <w:UseMarginsForDrawingGridOrigin/>
  <w:Compatibility>
    <w:FootnoteLayoutLikeWW8/>
    <w:ShapeLayoutLikeWW8/>
    <w:AlignTablesRowByRow/>
    <w:ForgetLastTabAlignment/>
    <w:LayoutRawTableWidth/>
    <w:LayoutTableRowsApart/>
    <w:UseWord97LineBreakingRules/>
  </w:Compatibility>
  <w:BrowserLevel>MicrosoftInternetExplorer4</w:BrowserLevel>
</w:WordDocument>
</xml><![endif]-->
<style>
<!--
/* Style Definitions */
p.MsoNormal, li.MsoNormal, div.MsoNormal
  {mso-style-parent:"";
  margin:0in;
  margin-bottom:.0001pt;
  mso-pagination:widow-orphan;
  font-size:10.0pt;
  font-family:"Times New Roman";
  mso-fareast-font-family:"Times New Roman";}

p
  {mso-margin-top-alt:auto;
  margin-right:0in;
  mso-margin-bottom-alt:auto;
  margin-left:0in;
  mso-pagination:widow-orphan;
  font-size:12.0pt;
  font-family:"Times New Roman";
  mso-fareast-font-family:"Times New Roman";}

span.GramE
  {mso-style-name:"";
  mso-gram-e:yes;}

@page Section1
  {size:8.5in 11.0in;
  margin:1.0in 1.25in 1.0in 1.25in;
  mso-header-margin:.5in;
  mso-footer-margin:.5in;
  mso-paper-source:0;}

```

```

div.Section1
    {page:Section1;}
-->
</style>
<!--[if gte mso 10]>
<style>
    /* Style Definitions */
    table.MsoNormalTable
        {mso-style-name:"Table Normal";
        mso-tstyle-rowband-size:0;
        mso-tstyle-colband-size:0;
        mso-style-noshow:yes;
        mso-style-parent:"";
        mso-padding-alt:0in 5.4pt 0in 5.4pt;
        mso-para-margin:0in;
        mso-para-margin-bottom:.0001pt;
        mso-pagination:widow-orphan;
        font-size:10.0pt;
        font-family:"Times New Roman";}
</style>
<![endif]-->
</head>

<body lang=EN-US style='tab-interval:.5in'>

<div class=Section1>

<p class=MsoNormal><span style='font-size:12.0pt'>This is my first web
page. I
am having fun entering codes and learning to write HTML code and write
my own
HTML code. <o:p></o:p></span></p>

<p class=MsoNormal style='mso-margin-top-alt:auto;mso-margin-bottom-
alt:auto'><span
style='font-size:12.0pt'>My dogs are named<span
class=GramE></span><br>
Ginny<br>
Maggie<o:p></o:p></span></p>

<p class=MsoNormal><o:p>&nbsp;</o:p></p>

</div>

</body>

</html>

```

You will use the following codes to write a very simple web (HTML) page: <html>, <head>, <title>, <body>, <p>, and
. Each of these codes is also called a tag. We will also look at a variation of several of these tag codes.

The case of the letters inside these tag codes was not important. They are not case sensitive, however code validators usually require them to be lower case and with

HTML5 it is generally considered good practice to use lowercase.. Later there will be some information inside “” (quotes in some tag codes that will be case sensitive but not these in this chapter. So <HTML> is the same as <html> although you should use <html>.

Some tag codes have similar tag codes that mark the end of the area marked by that tag code. These ending codes will merely be different from the beginning code by adding a / in the code. So the ending code for <html> is </html>.

You can write all your code as a single line with no returns or put lots of returns (enter key) through out the document. HTML will not pay attention to your returns so either way will work for the computer. However to make it easier to read the code, and make changes later, you should put each of the different type codes on separate lines. You can also indent lines if you want. HTML views a space or a bunch of spaces as a space. So both of the following mean same to computer, but you should use second example to do your code:

Example 1

```
<html><head><title>bad code method</title></head><body><p>This is a bad coding method.</p></body></html>
```

Example 1

```
<html>
<head>
<title>good code method</title>
</head>
<body>
<p>This is a good coding method.</p>
</body>
</html>
```

All documents written in HTML (which is almost all documents you will find on the web) must start with the <html> tag code. This code appears in some documents that are generated as the second line, but first is a comment. This merely tells the computer this document is in HTML. You should also always include a </html> code at the end of your document to mark the end of the document. If the page is viewed in Internet Explorer this is optional at the end, but other browsers may not display your page correctly if you leave out the closing tags.

So far our new page looks like:

```
<html>
```

```
</html>
```

Each html document can be split in two major sections. There is a heading section and a body section. All HTML documents must have a body section, but the heading section is optional. The body section will actually contain the information you want displayed in the browser. The heading section may contain a title and other directions to the browser and search engines.

The heading section will come first. It uses the <head> tag code. The <head> tag code also has a closing </head> tag code. This goes immediately after the <html> code. So our code is now:

```
<html>
<head>

</head>

</html>
```

Within the heading section we will put a title. The text you put in the title tags goes in the bar at the top of the page when displayed in the browser. Depending on the browser you can enter a varying number of characters. No HTML tag codes can go in the title text and it will only appear on one line. A good idea is always to include a title as the title reminds people at top of page what they are viewing. Also this is what most search engines will display as the title of your page, so make it suggest the content. We are going to put a title of “First web page I wrote”. You start with the <title> tag code and then put the text and then use a closing </title> tag code. So the line we will add is:

```
<title>First web page I wrote</title>
```

The case of the text is important as it will appear as you type it. Our example program is now:

```
<html>
<head>
<title>First web page I wrote</title>
</head>

</html>
```

We have now finished the heading section so we will start on the body section. The body section will have a <body> tag code to start it and a </body> tag code to end the section. Between the two will come the text or body of the document. Our example code now includes:

```
<html>
<head>
<title>First web page I wrote</title>
</head>
<body>

</body>
</html>
```

We will now be able to type in our text in the body section. Remember that HTML pays no attention to any returns we type in our code. So we will have to use tag codes to tell HTML to actually do a return or enter. There are two different codes that we can use to do this.

One is the <p> tag code or the paragraph code. When you enter </p> in the text a hard return will be done, a blank line added and new text starts on next line. So basically when you are typing in a paragraph do as you have learned in word processing and just type to end of paragraph with no returns. The browser will enter soft returns where needed. You will put a </p> code to mark the end of the paragraph. You use a <p> to mark the beginning of the paragraph. There is no HTML requirement that you put a blank line after a <p> in the code but generally it will help code readability for humans.

The second code merely does a hard return and starts on next line but leaves no blank lines. The code is
. There is no </br> code as you are not marking an area but telling browser what to do then. It is a self closing tag so the / is put after br code and a space.

So we could use the two codes to enter the following:

“This is my first web page. I am having fun entering codes and learning to write HTML code and write my own HTML code.

My dogs are named:
Ginny
Maggie”

The code to do this is:

```
<p>This is my first web page. I am having fun entering codes and learning to write HTML code and write my own HTML code.</p>
```

```
My dogs are named:<br />  
Ginny<br />  
Maggie<br />
```

The example code we now have is:

```
<html>  
<head>  
<title>First web page I wrote</title>  
</head>  
<body>  
<p>This is my first web page. I am having fun entering codes and learning to write HTML code and write my own HTML code.</p>
```

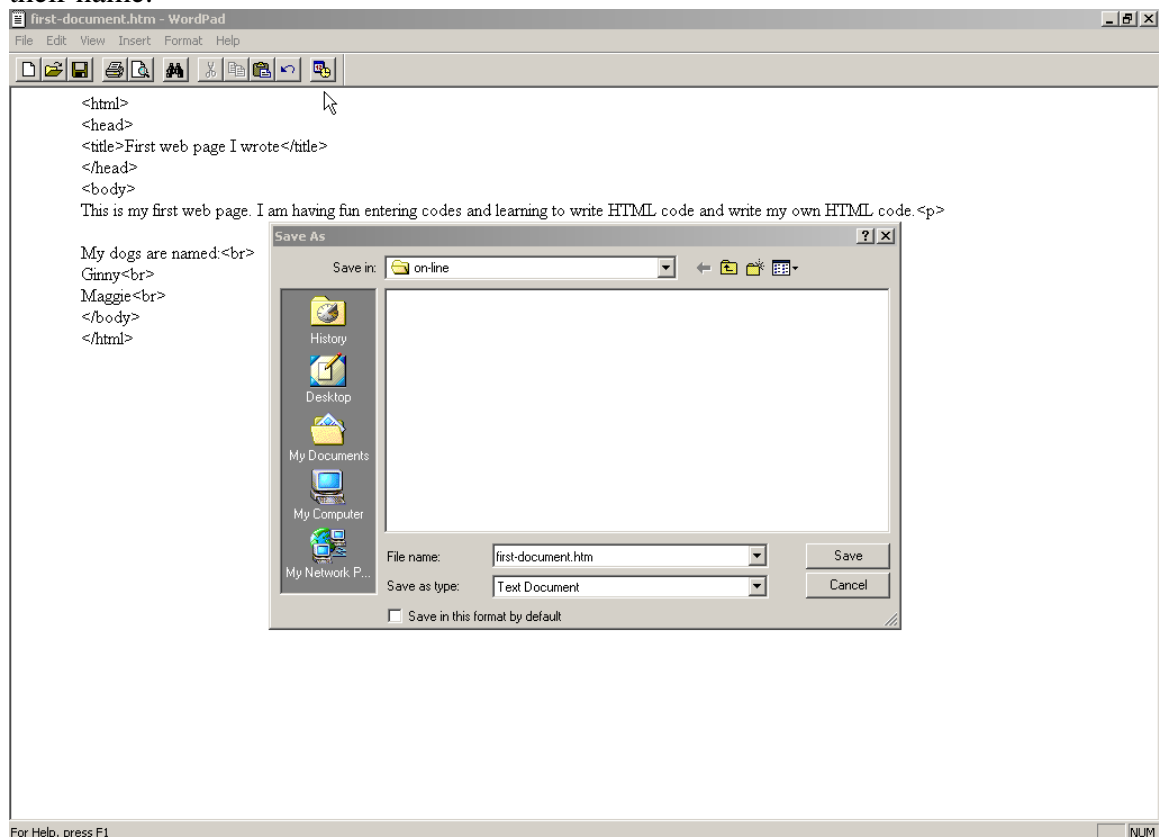
```
My dogs are named:<br />  
Ginny<br />  
Maggie<br />  
</body>  
</html>
```

We have now written the code for our web page. Now how do I put it in the computer?

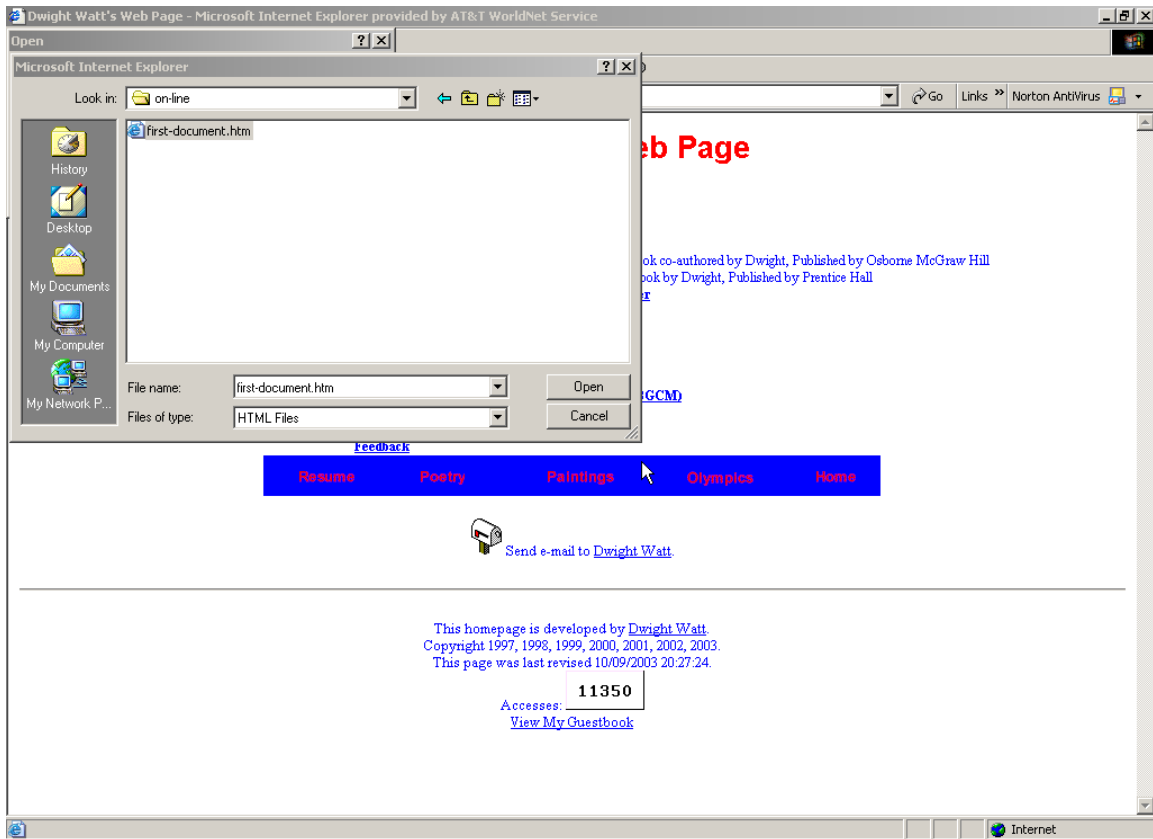
You could use a word processor to enter it, but word processors by default will add their own codes. The best way is to use either NotePad or WordPad to enter the code. Both come with Windows so you will generally have them available unless you are using a different operating system. In which case find out what the text entry program comes with it. These directions will use WordPad.

Start WordPad which is found in Accessories in the Programs list in your Start Menu or by doing Search for WordPad. Then merely type in your program. Go back and double check your typing as there is no spell or grammar checker in WordPad.

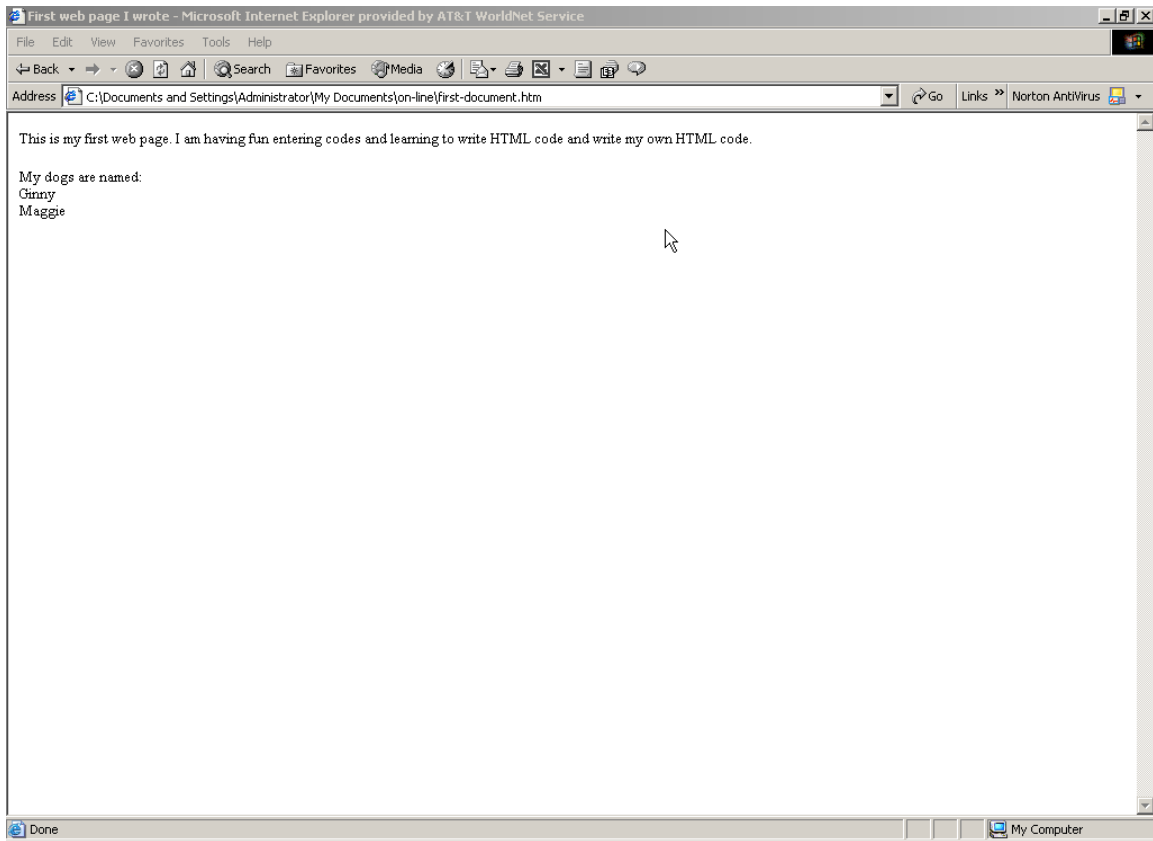
Then choose the File Menu, Save on the list and you will get a Save As pop-up box. You will change the Save as type to Text Document which means only what you typed in is saved and no special codes. In file name enter the name you want to call this HTML document followed by .htm or .html to show it is a HTML document. (extensions of .htm and .html are both acceptable for HTML files) When you name the document do not include spaces as some browsers have a hard time displaying documents with spaces in their name.



Now to run the code you just wrote, open your web browser. Then go to File menu and choose Open. You can then choose to browse to find where your document is on your computer and choose it. So in our example using Internet Explorer here is the step:



Then choose Open and the document will open in your browser window. If there are errors go back to the document in WordPad and fix and save and then press Refresh or Reload on your browser and the browser will load the latest version.



You can print the actual page using File Print in your browser and you can print the code by using File Print in WordPad.