# Anatomy of a Web Page

The source for a page on a website a file with the extension *.htm* or *.html* whose content is made up of plain text.

## Your First Web Page

Let’s get started by creating our first web page. Then we’ll go back and I’ll explain what all that markup stuff means. Are you ready?

Open your favorite text editor and enter the following text:

<!DOCTYPE html>

<html lang=”en”>

<head>

<meta charset=”utf-8” />

<title>My First Web Page</title>

</head>

<body>

<h1>Welcome to the World Wide Web!</h1>

<p>This is my first ever web page.</p>

<address>

<!-- Replace the following -->

Your Name<br>

you@example.com

</address>

</body>

</html>

Save the file into a folder on your hard drive. A good choice for the location of the file on Microsoft Windows is a new folder in the “My Documents” folder. Perhaps you could call it “MyFirst.html” or something similar. Give yourself a round of applause. You have created your first web page.

**NOTE:** Mac users can create a directory under the \Users\UserName\Documents and Linux users can create a new directory under /home/UserName/Documents.

Now open the file in whatever web browser you use. In most cases, Ctrl+O (Command+O on a Mac) allows you to browse to a file on your local computer and open it. Your web page should look something like Figure 2. (NOTE: I only included the actual page contents here to save space.)



Figure - Your first HTML file in the browser.

## What does it all mean?

Let’s go through the contents of your first HTML page bit by bit. The first line tells the browser that this document will be using HTML 5. Earlier versions of HTML required a more complicated DOCTYPE tag but unless you have a compelling reason to use an earlier version of HTML, use this DOCTYPE and HTML 5. By including this line, we tell the browser that it doesn’t have to try to work around the quirky behaviors that were part of earlier versions of HTML.

The second and last line of the document work together to define a container for the rest of the page. <html lang=”en”> is the opening tag. The part that follows “html” is an *attribute* that tells the browser that the language for this page is English. Other possibilities include: fr (French), de (German), it (Italian), and es (Spanish). If you want to know more about the lang attribute, you can check out the definition on the w3c website, Specifying the language of content: the lang attribute (https://www.w3.org/TR/html5/dom.html#attr-lang).

Next, in lines 3 to 6, we define the head section of the document. The head section contains metadata for the current page. (Metadata is data about the page, as opposed to data that will be displayed in the browser.) In our example, I have included two bits of information. The tag on line 4 tells the web browser how to interpret the text on the page. The character set used in this case is “utf-8”. If you include the character set information, it should be within the first 1,024 characters of the page definition. The best way to ensure that is to place it first in the head section of the page, right after the opening <head> tag.

The title tag, which is the only required child tag for the head section of a web page. The text between <title> and </title> will usually be displayed either in the title bar of the web browser or the tab for the page.

In case you’re curious, UTF stands for Unicode Transformation Format and the 8 means that each character in a string is represented by 8 bits or one byte.

The next section, the body, is where all the visible content of the page is placed. In the case of our sample document, we have a few bits of information. We have a heading, the information between <h1> and </h1>, a paragraph, and finally a section that contains my name and email address.

The <address> tag is usually formatted as italic text, but that depends, of course, on what formatting the web designer may have applied using CSS, but also on how the browser you are using chooses to interpret the tag.

Notice that each line of the address ends with <br />? That tag indicates that the browser should begin a new line.

Browsers and white space -- When the web browser interprets an HTML page, it ignores whitespace. Characters such as spaces and carriage return/line feed pairs are removed. You cannot format text using extra spaces to produce indentations, or extra paragraph marks to create new lines.  
  
If you want a new line, you have to use the <br /> tag. We’ll talk about spaces later when we get to character entities.

This would be a good time to play around a little bit. See what you can do with the few tags I’ve already shown you. It won’t be fancy, but the goal is to be able to type any one of the tags you learn from this book without having to look up what it means or how to use it.

## Working with Tags

Now that we have created a page, let’s back up a bit and talk about some of the most commonly used tags. Tags define HTML elements.

### Creating Paragraphs

The first tag you need to know how to use is the paragraph element, defined by the <p> tag.

With the tags I have shown you so far, you could actually create a web page. It would be boring to look at, but it would be a page that people could read. Let’s add a little bit of interest. I’m going to give you a few more tags to play with. And that is exactly what you should do, by the way, PLAY!

### Headings

HTML defines headings with levels from one to six. You use the heading tags the same way as the paragraph tag. The heading tags look like this:

<h1>This is heading level 1</h1>

<h2>This is heading level 2</h2>

<h3>This is heading level 3</h3>

<h4>This is heading level 4</h4>

<h5>This is heading level 5</h5>

<h6>This is heading level 6</h6>

If you were to look at that code as it is rendered in the browser, it would look something like this:

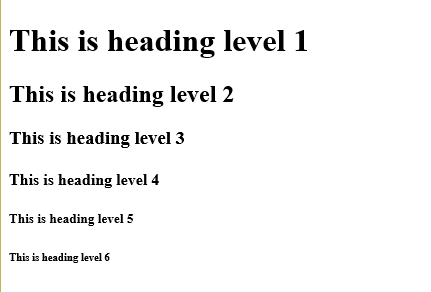


Figure : HTML Heading Tags in the browser.

To tell the truth, I’ve seldom gone any further than heading level three, but all six of them are there for your use. Remember that the look of each of these headings can be modified through the use of CSS. What we are looking at here is the default formatting supplied by my browser.

### Text formats

Here are some tags you can use to control the look of your text. These tags are used on sections of text within a paragraph or other container. So far you’ve seen the heading tags and paragraph tags; we’ll talk about other containers later.

Here are the text formatting tags:

**<em>** - The em tag defines text that should be emphasized in some way. You might use it to set off a technical term, for example. Most browsers will render text between the <em> and </em> tag using italics, but once we start working with CSS, you’ll see that you can change that to suit the content of the page.

**<strong>** - The strong tag indicates text that should stand out. In most cases the text marked as strong will be display in boldface type but this isn’t necessarily so because, as with <em>, you can define other formatting for this tag using CSS. All of the text between the opening <strong> and the closing </strong> will be formatted so that it stands out from the surrounding text.

In case you are wondering why I didn’t tell you about the <i> or <b> tags, that is because HTML 5 is all about meaning rather than formatting. In other words, <i> used to indicate that a run of text should be formatted in italics, while <b> indicated that the text should be bold.  
  
The <em>, emphasis, and <strong>, strong, tags indicate that text should be made to stand out from the rest of the surrounding text without indicated what formatting should be applied to accomplish that task.

**<small>** - Text formatted with the small tag is rendered in a smaller font than the surrounding text. It can be useful for a subtitle, or for information that is less important than the surrounding text.

**<cite>** - Used to mark the name of a work, such as a song, movie, book, or play. This text is usually rendered as italic.

That’s about it for chapter one.

In the next chapter, we’ll talk about using pictures and links in your web page.

Let’s finish by creating a slightly more interesting web page. Let’s create a page about you! Once you’ve entered the following text, save it. The name “aboutme.html” is a good choice for this one. Don’t forget to replace my placeholder text with your own information! I’ve set off the text you need to change with { and }, the curly braces.

<!DOCTYPE html>

<html lang=”en”>

<head>

<title>About {Your Name}</title>

</head>

<body>

<h1>All About {Your Name} <small>{your email}</small></h1>

<p>Hi, my name is <strong>{Your first name}</strong> and I have created

this website to show my skills with HTML. I am a {your job title} and

am learning HTML in order to expand my skills.</p>

<h2>My Family</h2>

<p>{Tell us about your family here.}</p>

<h2>My Job</h2>

<p>{Put stuff about your job here!}</p>

</body>

</html>

For example, take a look at Figure 3 to see my “About Irene” page. Notice that the text between <small> and </small> is slightly smaller than the rest of the line. Also notice the extra space between the headings and the paragraphs. That can be changed. When you get to volume two, CSS for Fun and Profit, you will learn exactly how to do that.

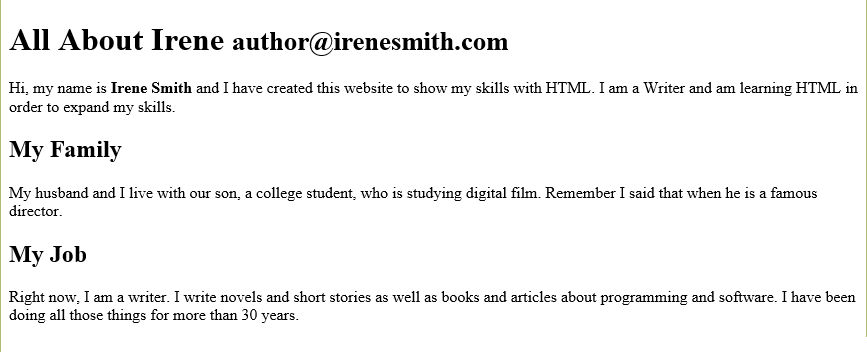


Figure : My "About Irene" page shown in the Edge browser.

## Conclusion

So, you’ve seen how to create a web page and how to use a handful of HTML tags. In coming chapters, well expand your knowledge of tags, and show you a little bit about how to make your web pages more attractive.

## Exercises

1. Create a basic web page that you could use as an ‘About” page on your website. Include your name and a few paragraphs describing who you are.
2. Take the page that you created in Exercise 1 and add some headings. Include multiple heading levels to divide the page into logical sections.

# Adding Pictures

A web page that contained only text would be boring. Remember the saying, a picture is worth a thousand words? A website that includes pictures is far more interesting than one that doesn’t. So let’s roll up our sleeves and add some interest to our pages.

You already know how to create a web page and how to add paragraphs of text. Now let’s talk about pictures. After all, they say that a picture is worth a thousand words, right? To add an image to your web page, you use the img tag. It looks like this:

<img src=”images/myimage.jpg” alt=”A head and shoulders picture of me.”>

The src attribute tells the web browser where to find the image. Your file doesn’t have to be in a folder called images, but it helps keep the website organized if you put all of your images in a single folder and, by convention, that folder is usually called “images” or “imgs” but you can call it “pictures” or “stuff” and nobody will care. Use whatever naming convention feels most comfortable to you.

The alt attribute should contain a description of the image. If someone who is visually impaired visits your page, they will most likely have a screen reading program that will read the contents of the page. By including an alt attribute, you give the screen reader something to read when it gets to the picture.

So let’s make a sample web page that includes a picture. If you still have the sample page we made last time, you can start with that and add an image. The only thing I’ve added so far is an image following the heading that says “All About {Your Name}”

<!DOCTYPE html>

<html lang=”en”>

<head>

<meta charset=”utf=8” />

<title>About {Your Name}</title>

</head>

<body>

<h1>All About {Your Name} <small>{your email}</small></h1>

*<img src=”images/myimage.jpg” alt=”A picture of me.”>*

<p>Hi, my name is <strong>{Your first name}</strong> and I have created this website to show my skills with HTML. I am a {your job title} and am learning HTML in order to expand my skills.</p>

<h2>My Family</h2>

<p>{Tell us about your family here.}</p>

<h2>My Job</h2>

<p>{Put stuff about your job here!}</p>

<h2>More stuff about me!</h2>

<p>{Feel free to add anything you like. Try out all of the tags I showed you and, above all, have fun!}</p>

</body>

</html>

If you followed along with the last tutorial, your web page is in a folder. Add another folder inside the one where your web page is located and call it images. Then place a small photo of yourself in that new folder. Make sure that you edit the src value to match the name and location of your file!

For example, if you file is called irenepic.jpg, and it is in a folder called pics, change the src value to “pics/irenepic.jpg” so the browser will know where to find your picture.

Now when you load the web page in a browser, it will include a picture of you. Try it and see.

## Working with Graphics

Detailed tutorials for creating your own web graphics is beyond the scope of this book, but I will give you a few hints and tips here for working with photographs and other images you might want to include in your website.

# Making Connections (Links)

The whole point of HTML is that it allows parts of a web page to connect to other parts of the same page, to other pages within the site, or to other sites on the web. To do that you use the link element, defined by the <a> tag.

## Connecting within the Same Page

You can connect to locations within the same web page. However, before you can connect to a spot within a page, you must define the spot. There are two ways to do that. The first is to use the <a> tag to define the location like this:

<p><a name=”MyFirstExample”>This is my first example.</a></p>

The second is to simply add an ID to a tag. For example:

<p id=”MySecondExample”>This is the second example.</p>

Either way works the same.

## Connecting to a Page in the Same Website

## Connecting to a Page in Another Website

# Working with Lists

Writing for the web is a little bit different than writing a book or magazine article. You need to remember that the average web surfer doesn’t spend a lot of time on a website that doesn’t provide the information he or she wants in a compact, easy to scan and digest manner.

Lists are a good way to present information to your visitors and HTML provides multiple types of list. The tags that define a list are either <ol> or <ul> where <ol> stands for ordered list and <ul> stands for unordered list, called a “bulleted list” in word processing.

Both types of list element contain items defined by the <li> “list item” tag. Each of the items in the list needs to start with the <li> tag. But, if you are using HTML 5, as we are, you do not need to add a closing </li> tag.

**NOTE:** According to the W3c recommendations for HTML5, a list item does not have to have a closing tag, “An li element's end tag may be omitted if the li element is immediately followed by another li element or if there is no more content in the parent element.” Therefore, I have omitted the closing tags in the examples.

## Ordered List

The <ol> tag defines an ordered list element. The code looks something like this:

<ol>

<li>This is the first item.

<li>This is the second item.

<li>This is the third and last item.

</ol>

## Unordered List

You could also call this a bulleted list, but calling it unordered will help you remember the tag because an unordered list begins with <ul> instead of <ol> and looks like this:

<ul>

<li>This is the first bulleted item

<li>This is the second

<li>And this is the third and final item

</ul>

## Definition List

The definition list is defined using the <dl> tag. This type of list is used to associate a key with its value. For example, you might have a list of terms for which you provide definitions or a list of categories which includes values.

The items in a definition list are presented differently from the lists we have already looked at. Each “item” in the list is defined by associating <dt> elements, which represent the key or term being defined and <dd> elements which contain the definition.

Let’s look at an example:

<dl>

<dt>HTML</dt>

<dd>HyperText Markup Language, the markup language used to create simple web pages.</dd>

<dt>XML</dt>

<dd>Stands for eXtended Markup Language. It is the language on which HTML is based.</dd>

</dl>

Now let’s look at a slightly different example. In this case the key is a category and the values associated with the key are all movie titles.

<h2>Some of our Movies</h2>

<dl>

<dt>Romance</dt>

<dd>When Harry Met Sally</dd>

<dd>Love, Actually</dd>

<dd>You’ve Got Mail</dd>

<dd>Quet Man, The</dd>

<dt>Comedy</dt>

<dd>Kiss, Kiss, Bang, Bang</dd>

<dd>Sting, The</dd>

<dd>We’re the Millers</dd>

<dt>Western</dt>

<dd>Once Upon a Time in the West</dd>

<dd>Unforgiven</dd>

<dd>Man Who Shot Liberty Valence, The</dd>

</dl>

# Appendix A – Web Design Tools

This appendix lists some of the tools you can use for your web design work. Some of the tools are free to use, others are not. In each case, I will include the cost of the tool when this book was written and a link to where you can get it.

## Text Editors

## Graphic Editors

## Code Editors

## FTP Tools

## Browser Developer Tools

## Other Tools

One of the things you should do when you are done with your website is to validate the HTML. You can do that

# Appendix B – Placeholder Images for Mockups

It is useful when you are trying to create a page layout to have placeholder images to take the place of the pictures you will use when you create the actual site. Whether you want a plain box with an X across it or something fun like kittens, puppies, or Bill Murray, you can find free services to meet your needs.

## Unsplash.It

This place is one of my current favorites. To get usage details, you can go to their website (at https://unsplash.it), but basically, all you have to do is set your img src attribute to this: <https://unsplash.it/200>

The number following the slash tells the website to create a square image that is 200x200 pixels. If you want a rectangular image, add an additional parameter. For example, <https://unsplash.it/200/400>