

Creating an HTML Web Page

In today's world of technology, many people want to share information with others through a universal medium which is accessible to those with a computer and Internet access. Most people want to share photographs, personal stories (blogs), link to their friend's sites or simply venture into the creative world of producing electronic art. The most common way to share this information is by creating a web page with HTML.

In this article, you will learn:

- Create a local folder to store your files, such as your web page and graphics
- How web pages work
- Browsers
- DOCTYPEs
- Hypertext Markup Language (HTML)
 - Heading tags
 - Paragraph tags
 - Images
 - Lists
 - Unordered
 - Ordered
 - Links
 - Tables
 - Break tag
 - Comment tag
 - Formatting of text
 - Different font families
 - Different font colors
 - Use of span tag

If you would like to follow creating a web page's step-by-step development or try your hand at creating one yourself, you will find the project zip link below helpful.

<http://midwestwebdesign.net/tutorials/basics/basics.zip>

http://midwestwebdesign.net/tutorials/basics/index_finished.html

Create your folder

When you develop web pages or websites, you create a folder on your computer which stores all web pages, images, style sheets, etc for your web page. Once you complete your web page on your computer, you upload the files from your computer to a remote server which allows your web page to be viewable to the world. Follow these steps to create your folder:

- From the Desktop, double-click **My Computer**
- From this window, double click Local Disk(C :)
 - You may choose another partition of your choice
- From this window, from the main menu select:

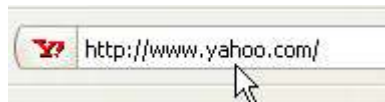
- **File>New>Folder**
- Type your first name
- Press enter
- Keep this window open but minimize it, by left clicking the left most button:



How web pages work

Web pages are delivered via a network protocol, Hypertext Transfer Protocol (HTTP). When a computer connects to a network, it has access to a variety of protocols. Subsequently, when computers connect to the Internet, they have access a few other protocols, which include HTTP and File Transfer Protocol (FTP), which are the two most commonly used on the web. The process of viewing a web page or website happens like this:

1. A visitor opens their web browser (discussed next) and in the address bar types a complete website address, sometimes referred to as Uniform Resource Locator (URL), such as <http://www.yahoo.com>



1. The browser uses the HTTP protocol to identify where the web page or website exists, on a remote server
2. Once the browser locates the web page or websites exists, it will request from the remote server to give it the specific file, in our case, index.html, since home pages for many sites use index as a naming convention
3. The server accepts and processes the request, and sends the browser the requested file
4. The browser then uses HTML to render the page to the visitor
5. Once the browser has finished rendering the web page or website, the visitor sees the following:



Browsers

In order to view a web page or website, you need a browser. Browsers are a software program which is capable of viewing different elements of a web page or website, such as text, images, tables, links, etc. All browsers today are free of charge and work well on Windows and Macintosh based computers. The most popular are listed below.

Microsoft Internet Explorer

If you have a Windows based computer, Internet Explorer is usually already installed. If not, you can download it at:

- <http://www.microsoft.com/windows/ie/default.msp>

Mozilla Firefox

If you want to support open source software initiatives and want a different taste for a browser, you may want to check out this browser at:

- <http://www.mozilla.com/en-US/firefox/>

Opera

If you want yet another choice for a browser, you may want to check out this browser at:

- <http://www.opera.com/>

In the end, it doesn't matter which browser you choose, since they all support HTML, CSS, JavaScript, etc. Matter of fact, as a web developer, you will probably use all three at some point.

DOCTYPES

Since the mass movement to CSS based designs has progressed in recent years, using a correct DOCTYPE ensures that your web page(s) are conforming to standards based code. DOCTYPE stands for Document Type Declaration which is a way to validate our code against the W3C specifications. Having this in places ensures you are properly creating, nesting and closing your tags. It also ensures that you are using correct CSS. Think of a DOCTYPE as our spell checker in Microsoft Word. In Word a spell checker will ensure all content is spelled right and grammatically formed. Likewise, our DOCTYPE ensures all HTML tags are spelled and marked up correctly. Without a correct DOCTYPE, your code and CSS will be rendered incorrectly to all modern browsers. A correct DOCTYPE declaration is:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
```

An incorrect DOCTYPE declaration is:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
```

Examples of various DOCTYPES can be found at the following address:

- <http://alistapart.com/stories/doctype/>

Hypertext Markup Language (HTML)

Hypertext Markup Language (HTML) is a markup language used to mark-up text, images, tables, lists, and links which will be rendered and viewed in a browser. This language is a client-side language which has no programming capabilities inherently available. It's used to provide structure and mark-up different elements (text, images, and tables) on our web page. In the following sections, we'll see how to create these various elements and common solutions to problem which you're likely to encounter.

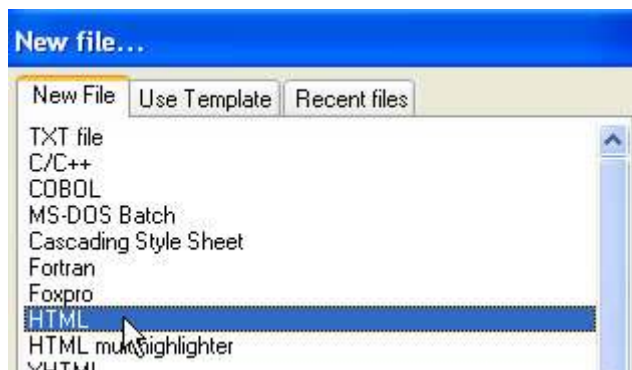
Create the web page

To make our learning experience a bit more intuitive we'll use PS Pad which is a free download. Follow these steps to download and install:

- Using your browser, type the following in the address bar:
 - <http://www.pspad.com/>
- Select English at the top
- From the left side, select Download
- Select the Installer file
- From the page you're re-directed to, left click the Download button
- Download to Local Disk (C:)
- Once the download is complete, double-click the installer in order to use the program

In PS Pad, close the initial document which is provided and follow these steps to create a new HTML document:

- From the main menu, select **File>New**
- In the New file window, select **HTML**:



- Click OK

Currently, PS Pad has our document in quirks mode and has unnecessary meta tags. Follow these steps to correct these problems.

Replacing the DOCTYPE Visit:

- <http://alistapart.com/stories/doctype/>
- Highlight HTML Loose
- Right click and select copy
- Highlight the DOCTYPE inside our file, right click and select Paste

Removing meta tags

Between the opening and closing <head> tag you see this:

```
<meta name="generator" content="PSPad editor, www.pspad.com">
```

- Highlight the code and press delete

Continuing, you see this meta tag:

```
<meta http-equiv="content-type" content="text/html; charset=windows-1250">
```

Highlight the code below

```
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
```

- Inside PS Pad, highlight the original meta tag, right click and select Paste

Your code should look like this:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<title></title>
</head>
<body>
</body>
</html>
```

Follow these steps to save your file:

- From the main menu, select **File>Save**
- In the Save As dialog, navigate to your folder on local disk (c:)
- In the File name text field type **index.html**
- Click **Save**

Create your HTML structure

Inside index.html you have the following markup:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
```

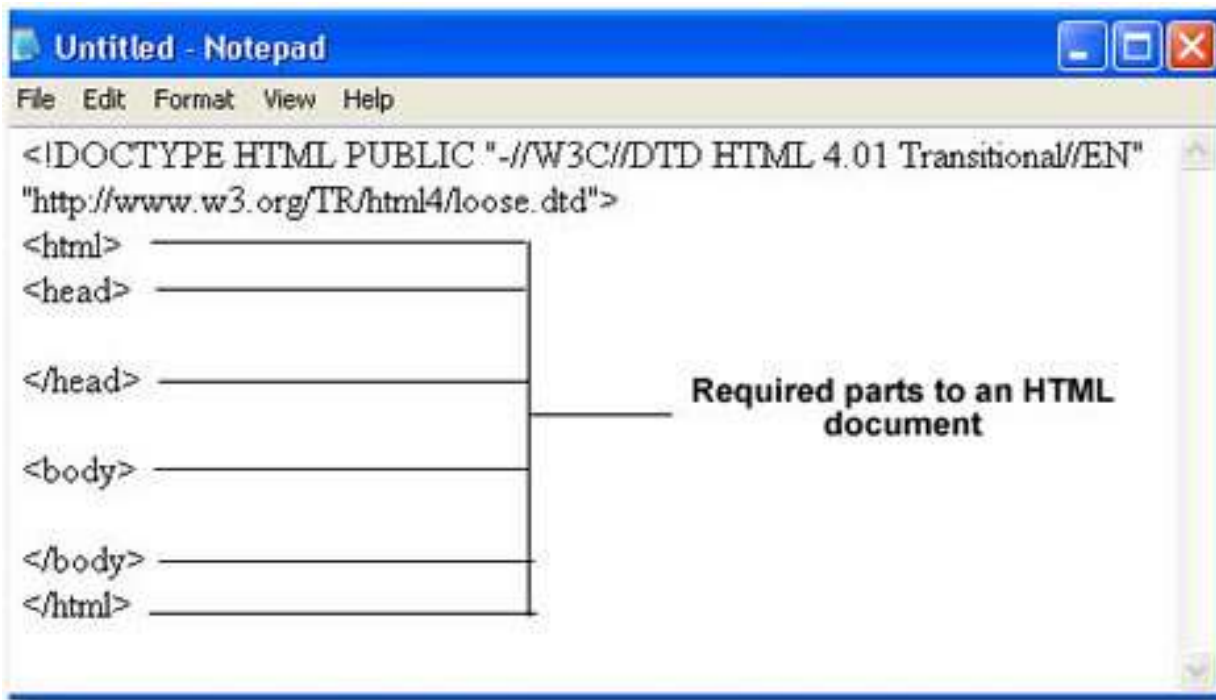
```
<title></title>
</head>
<body>
</html>
```

Note: The blank space between the opening and closing body tag is not necessary.

The basic structure required for all HTML files is:

- <html>
- <head>
- <title></title>
- </head>
- <body>
- </body>
- </html>

To help solidify the structure required, see the illustration below:



You will notice we have the DOCTYPE listed as well. Though the DOCTYPE is not required, it is helpful to think of this as being required, to ensure our code can be validated against the W3C specifications to ensure your web page is in compliant mode.

Mark-up tags

HTML is made up of mark-up tags which display text, images, lists and tables to the browser. These markup tags are created using the keys on your keyboard. Tags consist of an opening and closing tags, which are:

- <html>
 - Opening tag
- </html>
 - Closing tag

Title Tag

Giving your page a title is simple. Using the <title> tag, we'll name our web page **My first web page**:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<title>My first web page</title>
</head>
</body>
</html>
```

Save your file. Before you can view the title, you must open your web browser. Follow these steps to start your web browser:

- From the Desktop, select **Start>Programs>Microsoft Internet>Internet Explorer**
- From the main menu, select **File>Open**
- Browse dialog window shows
- Click **Browse**
- Navigate to your folder and select **index.html**
- Click **OK**
- You will see nothing in the window because we have no text, images, list or tables in our document

However, if you view the top left corner of your web page, you will see **My first web page**:



Body tag

Between the opening and closing body <body> tag is where content is displayed. This includes headings, paragraphs, tables, unordered & ordered lists, etc.

Heading Tags

HTML supports six headings, one through six, with one being the most prominent. The syntax for a heading three tag is:

```
<h3>Heading Three</h3>
```

Between the opening and closing <body> tag, place a heading three tag with your name as follows:

```
<h3>Your Name Page</h3>
```

Save your file and preview the results.

http://midwestwebdesign.net/tutorials/basics/index_finished_2.html

Heading three alignment

Heading tags support three basic alignments, which are:

- Left
 - default
- Center
- Right

In order to align our heading three in the center of the web page, between the opening heading three tags, press the space bar once and type the following:

```
<body>  
<h3 align="center">Ryan's Page</h3>  
</body>
```

Save your file and preview the results.

http://midwestwebdesign.net/tutorials/basics/index_finished_3.html

Paragraphs

Arguably the most used tag in HTML. The syntax for a paragraph tag is:

```
<body>  
<h3>Ryan's Page</h3>  
<p></p>  
</body>  
</html>
```

Paragraphs are not treated the same as they are with typical word processing documents. Paragraphs in HTML do not recognize word wrap the same as Word documents. For example, in your text editor, you can have one long line of text in a paragraph tag and it does not affect word wrap in a browser. The browser will wrap the text at a predefined width which is suitable for viewing. To create new paragraphs, you simply create additional paragraph tags. Using your current file, follow these steps to insert text into the opening and closing paragraph tags:

- Visit: <http://www.lipsum.com/>
- Highlight a sample paragraph of text, right click and select copy
- Between the opening and closing paragraph tags, right click and select paste

If you want word wrap in PS Pad, follow these steps:

- From the main menu, select **View>Word Wrap Lines**



Save your file and preview the results.

http://midwestwebdesign.net/tutorials/basics/index_finished_4.html

Alignment of paragraph tags work identically to that of heading tags.

Images

Web pages or websites wouldn't be very appealing if you didn't use images. The syntax for the image tag is:

```

```

The attribute/values are:

- Src
 - Informs the browser where the image is located
- Width & Height
 - Informs the browser of the physical dimensions of your graphic
- Alt
 - Informs assistant readers for blind people that an image is in your web page. As an assistant reader scans the above image, the voice command would say *"Dragon Blowing Smoke"*

If you don't know the dimensions of your graphic, follow these steps:

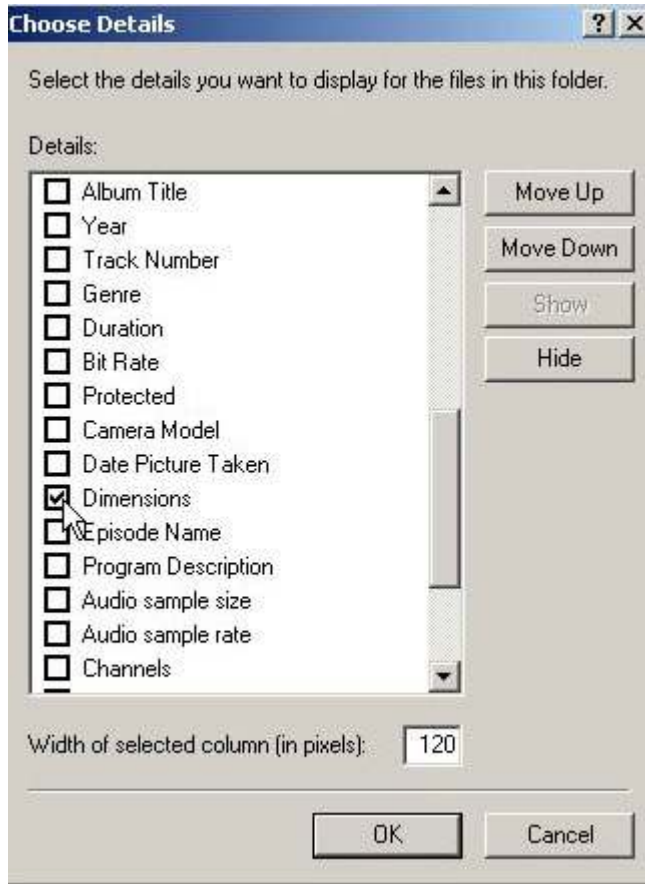
- From the Desktop, double-click **My Computer**
- Double click the local hard drive (C:)
- Right click just below the address bar:



- Select **More:**

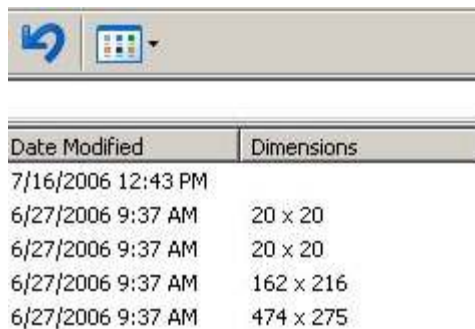


- A new window will show
- Scroll down to find **dimensions** and check it:



- Click OK

When you return to the original window, you will see the dimensions of your graphic:



Working with paths

When working with images, you have three choices regarding the path. The following paths are acceptable:

Absolute

```

```

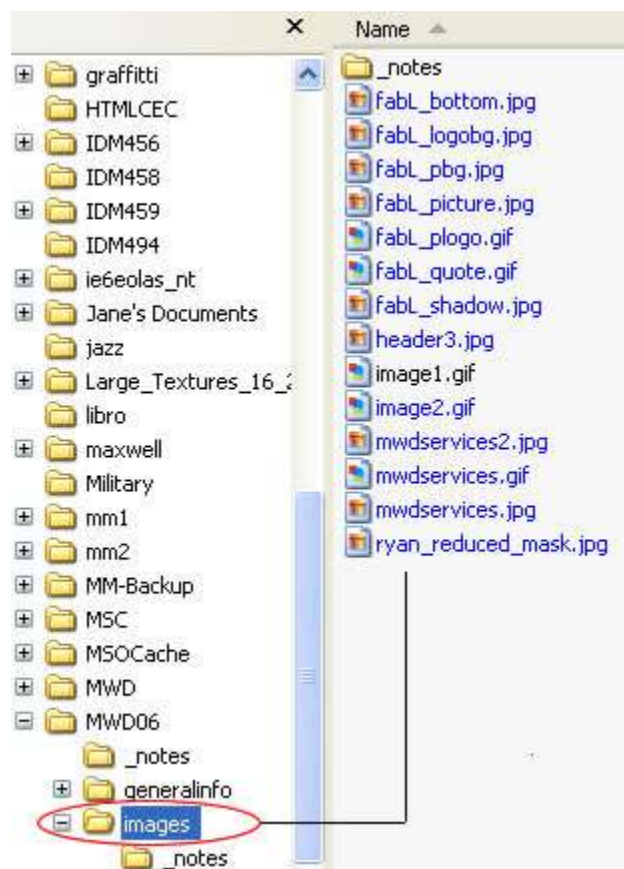
- The disadvantage to this method is that you have to be online to see your image because it's using the HTTP protocol.

Relative

```

```

- This method is typically used and recommended. In this example, inside your root folder (the folder you created earlier), you create an additional folder, images and place all images for your web page inside. The following illustration depicts this structure:



Site-Root Relative:

```

```

- This method is similar to absolute but uses a preceding forward slash before the images folder is referenced. The disadvantage with this method is you cannot view your images locally, since site root relative means the path will only work on web server.

Using the code from before, insert the image below our first paragraph tag:

```
<body>
<h3>Ryan's Page</h3>
<p>Lorem Ipsum is simply dummy text of the printing and typesetting
industry. Lorem Ipsum has been the industry's standard dummy text ever since
the 1500s, when an unknown printer took a galley of type and scrambled it to
make a type specimen book. It has survived not only five centuries, but also
the leap into electronic typesetting, remaining essentially unchanged. It
was popularised in the 1960s with the release of Letraset sheets containing
Lorem Ipsum passages, and more recently with desktop publishing software
like Aldus PageMaker including versions of Lorem Ipsum. </p>

</body>
```

Save your file and preview the results.

http://midwestwebdesign.net/tutorials/basics/index_finished_5.html

Common Problems with images

If you can't view your page on your computer or via your website, there are three potential problems:

- 1) Miss-spelled: One common problem is miss-spelling the word src. It's easy to spell it scr.
- 2) Local problems (your computer): If you receive a red x in your browser while viewing your page on your computer, there are three possibilities. You could have src spelled wrong, you could have the path to the image incorrect, or the image doesn't exist on your computer.
- 3) Remote problems: If you receive a red x in your browser while viewing your page on your website, there are three possibilities. You could have src spelled wrong, you could have the path to the image incorrect, or the image doesn't exist on your server.

Aligning the image

Aligning an image works a bit differently than headings or paragraphs. If we want to center our image in the web page, we add the following code:

```
<body>
<h3>Ryan's Page</h3>
<p>Lorem Ipsum is simply dummy text of the printing and typesetting
industry. Lorem Ipsum has been the industry's standard dummy text ever since
the 1500s, when an unknown printer took a galley of type and scrambled it to
make a type specimen book. It has survived not only five centuries, but also
the leap into electronic typesetting, remaining essentially unchanged. It
was popularised in the 1960s with the release of Letraset sheets containing
Lorem Ipsum passages, and more recently with desktop publishing software
like Aldus PageMaker including versions of Lorem Ipsum. </p>

<p align="center"></p>

</body>
```

By nesting the image in a paragraph, we make the image a block level element which can then be aligned in the same way as a paragraph tag. Save your file and preview the results.

http://midwestwebdesign.net/tutorials/basics/index_finished_5_centered.html

Wrapping the image around text

If we want the image to wrap around text, then we adjust the code as follows:

```
<h3>Ryan's Page</h3>
<p>

Lorem Ipsum is simply dummy text of the printing and typesetting industry.
Lorem Ipsum has been the industry's standard dummy text ever since the
1500s, when an unknown printer took a galley of type and scrambled it to
make a type specimen book. It has survived not only five centuries, but also
the leap into electronic typesetting, remaining essentially unchanged. It
was popularised in the 1960s with the release of Letraset sheets containing
Lorem Ipsum passages, and more recently with desktop publishing software
like Aldus PageMaker including versions of Lorem Ipsum.

</p>

</body>
```

We moved the image right after the opening paragraph, and then we set alignment to left. You will notice in the project file that there are additional paragraphs. Our reason is simple — to show the image wrapping around segments of text and then returning to the normal flow once content height is larger than image height. Save your file and preview the results.

http://midwestwebdesign.net/tutorials/basics/index_finished_5_imagewrap.html

Block level and inline elements

In HTML, you have two types of elements, block and inline. Some examples of block level elements are:

- Paragraphs
- Tables

Some examples of inline level elements are:

- Images
- Anchors (links)

Block level elements have an invisible box in which they occupy. This box contains space which is reserved for content, which can be text, images, and links. Inline elements have no invisible box, therefore, there's no space reserved for content. This is how we can nest an inline element (image) inside a block level element (paragraph) to simulate a wrapping affect. This is also how anchors (links) can reside inside a paragraph or table.

Lists

In HTML, you can create two types of lists, ordered and unordered. In a typical Word document an ordered list looks like this:

1. My name is Ryan Butler
2. I'm 25 years old

In HTML, the syntax for an ordered list is:

```
<ol>  
<li>list item 1</li>  
<li>list item 1</li>  
<li>list item 1</li>  
</ol>  
</body>
```

Let's examine the structure in greater detail.

- First, we create an ordered list by using ``
- Next, we create list items by using ``, followed by ``
- Finally, we finish the order list by using ``

You can have as many list items as you want. Save your file and preview the results.

http://midwestwebdesign.net/tutorials/basics/index_finished_6.html

Unordered list

The other type of list you can create is an unordered list. In a typical Word document an ordered list looks like this:

- List item one
- List item two

In HTML, the syntax for an unordered list is:

```
<ul>
<li>list item 1</li>
<li>list item 1</li>
<li>list item 1</li>
</ul>
</body>
```

Let's examine the structure in greater detail.

- First, we create an unordered list by using
- Next, we create list items by using , followed by
- Finally, we finish the unordered list by using

You can have as many list items as you want. Save your file and preview the results.

http://midwestwebdesign.net/tutorials/basics/index_finished_7.html

Do note, unordered lists by default, have indentation.

Change the starting number of an unordered list

To change the number at which an unordered list starts, inside the opening tag, we press the space bar once, then type start="3":

```
<ol start="3">
<li>list item 1</li>
<li>list item 1</li>
<li>list item 1</li>
</ol>
```

Save your file and preview the results.

http://midwestwebdesign.net/tutorials/basics/index_finished_7_orderedstart.html

If you want Roman numerals instead of conventional numbers, inside the opening tag, after start of 3, press the space bar once and type type="I".

Change the symbol of the bullet

To change the bullet character to a square, inside the opening tag, press the space bar once and type type="square":

```
<ul type="square">
<li>list item 1</li>
<li>list item 1</li>
<li>list item 1</li>
</ul>
```

Another possibility is circle. Save your file and preview the results.

http://midwestwebdesign.net/tutorials/basics/index_finished_7_unorderedbullet.html

Links

Web pages or websites would serve no purpose or have no real power if it weren't for the ability of linking to another page on your website or linking to another website such as Yahoo. The syntax for a link is:

```
<a href="http://www.yahoo.com">Yahoo!</a>
```

Let's examine this tag in greater detail.

- <a : stands for anchor
- href : is the parameter. Possible values are:
 - A separate website using the HTTP protocol
 - Yahoo!
 - A web page on your website
 - My Page
- Yahoo! : Descriptive text describing your link
- : closing your anchor tag

Using the code above, insert the link inside our web page below the unordered list:

```
</ul>
<a href="http://www.yahoo.com">Yahoo!</a>
</body>
```

Save your file and preview the results. Clicking the link should direct you to Yahoo!

http://midwestwebdesign.net/tutorials/basics/index_finished_8.html

Tables

Tables are ideal for *tabular data* and still should be used in this scenario. The syntax for a table is:

Let's examine the code in greater detail.

- `<table>`
 - starts a table
- `<tr>`
 - stands for table row and creates one
- `<td>`
 - stands for table data cell and create one
- `</td>`
 - closes a table data cell
- `</tr>`
 - closes a table row
- `</table>`
 - closes a table

Let's create a table which has three columns and three rows. The columns contain the following information:

1. Name
2. Title
3. Relationship

Add the following table code below our link:

```
<a href="http://www.yahoo.com">Yahoo!</a>
<table>
<tr>
<td>Name</td>
<td>Title</td>
<td>Relationship</td>
</tr>
</table>
```

Save your file and preview the results.

http://midwestwebdesign.net/tutorials/basics/index_finished_9.html

Table attributes and values

Currently, our table doesn't resemble a table. We add a border by pressing the space bar once and typing the following code:

```
<table border="1">
<tr>
<td>Name</td>
```

```
<td>Title</td>
<td>Relationship</td>
</tr>
</table>
```

Save your file and preview the results.

http://midwestwebdesign.net/tutorials/basics/index_finished_tableborder.html

In order to increase the entire width of our table, add a width:

```
<table border="1" width="400">
<tr>
<td>Name</td>
<td>Title</td>
<td>Relationship</td>
</tr>
</table>
```

Save your file and preview the results.

http://midwestwebdesign.net/tutorials/basics/index_finished_tablewidth.html

Aligning the table

To center our table in the browser, we add the align attribute inside the opening table tag:

```
<table border="1" width="400" align="center">
<tr>
<td>Name</td>
<td>Title</td>
<td>Relationship</td>
</tr>
</table>
```

Save your file and preview the results.

http://midwestwebdesign.net/tutorials/basics/index_finished_tablealign.html

To center our first three cells, we add the align attribute inside the opening table row tag:

```
<table border="1" width="400" align="center">
<tr align="center">
<td>Name</td>
<td>Title</td>
<td>Relationship</td>
</tr>
</table>
```

Technically, we could have placed the align attribute on each table cell, but it's easier to place it on the entire table row.

Adding more content

Let's make this table seem a little more practical, by adding two people, as shown below:

```
<table border="1" width="400" align="center">
<tr align="center">
<td>Name</td>
<td>Title</td>
<td>Relationship</td>
</tr>
<tr align="center">
<td>Ryan Butler</td>
<td>Web Development Technician</td>
<td>Husband</td>
</tr>
<tr align="center">
<td>A.J. Cook</td>
<td>Actress &#8212; Criminal Minds</td>
<td>Wife</td>
</tr>
</table>
```

We add two additional rows, along with corresponding table cells. Save your file and preview the results.

http://midwestwebdesign.net/tutorials/basics/index_finished_tablealigncontent.html

What is — on the last table cell?

In the web world, there are different ways of handling special characters, such as dashes, quotation marks, etc. We handle these special characters with an American Standard Code for Information Interchange (ASCII) standard. Since computers can only understand numbers, an ASCII code is a way to represent a special character. — is the numerical equivalent to a dash. For a complete list of ASCII codes, visit: <http://www.asciitable.com/>

Cell padding and cell spacing

The difference between these is subtle. Cell padding is the pixel width between the cell contents and the cell border. Cell spacing is the pixel width between the individual cell contents of the table. For example, to see the affect of cell padding, add the following to your table:

```
<table border="1" width="400" align="center" cellpadding="10">
<tr align="center">
<td>Name</td>
<td>Title</td>
<td>Relationship</td>
</tr>
<tr align="center">
<td>Ryan Butler</td>
```

```

<td>Web Development Technician</td>
<td>Husband</td>
</tr>
<tr align="center">
<td>A.J. Cook</td>
<td>Actress &#8212; Criminal Minds</td>
<td>Wife</td>
</tr>
</table>

```

Save your file and preview the results.

http://midwestwebdesign.net/tutorials/basics/index_finished_tablepadding.html

For example, to see the affect of cell spacing, add the following to your table:

```

<table border="1" width="400" align="center" cellspacing="10">
<tr align="center">
<td>Name</td>
<td>Title</td>
<td>Relationship</td>
</tr>
<tr align="center">
<td>Ryan Butler</td>
<td>Web Development Technician</td>
<td>Husband</td>
</tr>
<tr align="center">
<td>A.J. Cook</td>
<td>Actress &#8212; Criminal Minds</td>
<td>Wife</td>
</tr>

```

Save your file and preview the results.

http://midwestwebdesign.net/tutorials/basics/index_finished_tablespacing.html

Break tags

Currently, if we removed the following from the opening table tag:

- Cell spacing
 - 10
- Align
 - center

Your table will move back to the left side of the browser, which makes illustrating the next tag easier to see. You should notice the space between our link and table is very tight. One way of creating a carriage return (new line) in HTML is by using the break tag (
). For example, right after the closing anchor tag and before the opening table tag, insert two break tags:

```
<a href="http://www.yahoo.com">Yahoo!</a>
<br>
<br>
<table border="1" width="400">
```

By adding two breaking spaces we create a vertical gap (space) between the link and table. Save your file and preview the results.

http://midwestwebdesign.net/tutorials/basics/index_finished_brtag.html

Comment tags

Often times in web development when first creating your web page, you will need to keep mental reminders of where content, navigation and other elements will be placed in which you don't have content for yet. You obviously don't want these comments visible to visitors because it looks unprofessional and awkward. In HTML, you can hide these mental comments by using a comment tag. For example, right below the opening body tag, place the following comment:

```
<body>
  <!--this is a mental reminder-->
  <h3>Ryan's Page</h3>
```

HTML comments start with:

```
<!--
```

And end with:

```
-->
```

Any content inside will be ignored by the browser and not seen by any visitor. Save your file and preview the results.

http://midwestwebdesign.net/tutorials/basics/index_finished_comment.html

Formatting of text

You can change the default font families and colors used in your web page in two ways:

- Font tags
- CSS

Font tags should be avoided as they are being depreciated (non-supported) in future versions of HTML. The reasons they should be avoided are as follows:

- Bloats your markup

- Making a site wide font family or color change is a tedious task
- Support for a wide variety of formatting options is limited

While we'll save an in-depth discussion of CSS for another article, we'll cover the basics to allow us to change some basic formatting of our text and table.

Cascading Style Sheets

Cascading Style Sheets demystified is a language used by web developers to control the presentation of text, colors and positioning of elements on a web page. There are three basic ways to write CSS, which are:

- Inline
- Embedded
- External

Inline

This method should be avoided because it's similar to using font tags. The syntax for an inline style is:

```
<h3 style="font-family:Verdana, Helvetica, sans-serif;">Text here</h3>
```

Embedded

This method is better than inline but not as efficient as external. Embedded styles are written inside the opening and closing <head> tags and thus apply to only the page in which they reside. The syntax for an embed style is:

```
<style type="text/css" media="screen">
h3{
  font-family:Verdana, Helvetica, sans-serif;
}
</style>
```

Let's examine this code in greater detail. First, inside the <style> tag we have the following:

- type="text/css"
 - This informs the browser which language we're using. Another possibly value is:
 - type="text/javascript"
- media="screen"
 - This informs the browser which medium these styles are written. Possibly values include:
 - All
 - Screen
 - Print
 - TV
 - Projector

Next:

- h3
 - Is a selector in CSS and more importantly, our heading three is our HTML element which we select and style.
- Font-family
 - Is the property used to change a font family
- Verdana, Helvetica, sans-serif;
 - Is the value associated with font family

When the browser reads this value, if there was a voice command, it would say this: *“Look for Verdana font on the visitor’s computer. If it isn’t available, look for Helvetica, then use the nearest sans-serif font available.”* For a detailed discussion on sans-serif versus serif fonts, please visit:

<http://en.wikipedia.org/wiki/Serif>

In the past, you may have seen the embedded rule written like this:

```

<style type="text/css" media="screen">
<!--
h3{
  font-family:Verdana, Helvetica, sans-serif;
}
-->
</style>
```

You should notice the use of the HTML comment tag, which is split. The first part of the HTML comment is right before our CSS selector, h3. The last half of the comment is after the closing curly brace. You might be wondering why bother? In the past, when developing CSS for cross compatibility, including the HTML comment inside your embedded rules would hide these styles from Netscape Navigator 4. The CSS rendering engine for this browser was pathetic at best. Since most people are using Internet Explorer 6 and version 4 browsers are as common as a five-legged goat, in most practical instances, it’s safe to omit the HTML comment tag.

Changing our heading three tag

For our purposes, we’ll use embedded styles. Let’s change the font family, color, size and alignment of our heading three. Follow these steps:

Between the opening and closing <head> tag, below the <title> tag write this:

```

<style type="text/css" media="screen">
h3{
  font-family:Verdana, Arial, Helvetica, sans-serif;
  color:#990000;
  text-align:center;
  font-size: 1.5em;
}
</style>
```

Inside the opening and closing curly braces, we do the following:

- font-family:Verdana, Arial, Helvetica, sans-serif;

- Sets the font family
- color:#990000;
 - Sets the color to a dark red
- text-align:center;
 - Sets text alignment to center
- Font-size: 1.5em;
 - Sets font size to a proportional size

Save your file and preview the results.

http://midwestwebdesign.net/tutorials/basics/index_finished_h3style.html

Note that indentation of property-values in CSS is not required, though it makes your CSS easily readable.

Colors

When working with colors in web pages, you have 16 named colors and then a wide assortment of web safe colors which are usually in hexadecimal format. Hexadecimal numbers are always preceded with a hash mark (#) and then a six digit number. For example, here are common hexadecimal colors developer's uses:

- #000000
 - Black
- #FFFFFF
 - White
- #FF0000
 - Red
- Blue
 - #00FFFF

If the hexadecimal is a repeating pattern, then you can use short hand notation and specify the first three numbers. For example, with black and white above, those numbers could be shorted to:

- #000 & #FFF

Font sizes

When working with font sizes in web pages, you have two choices:

- Pixels
- Proportional

Pixels are a fixed width measurement and should generally be avoided with fonts. Proportional are adjustable width measurements. That is, if a visitor chooses to resize our heading three tags in their browser, it adjusts accordingly. To help in accessibility of your web page, it's recommended you always

use proportional font sizes in your web pages. In our example, we used a proportional measurement by using em, which is the size or height of the letter "m."

Changing our paragraph

Let's change all our paragraphs to a font family of Arial. Add this CSS rule right below our current h3:

```
p{
  font-family:Arial, Helvetica, sans-serif;
}
```

Save your file and preview the results.

http://midwestwebdesign.net/tutorials/basics/index_finished_pstyle.html

Span tag

Often in web pages, you want a specific section of text to stand out from the rest. Since the text is usually inside a paragraph which is block level, we can use another inline element, the span tag. We nest the span tag inside a segment of text which we want to highlight and then apply a CSS rule accordingly. Follow these steps to create a span tag: Inside the first paragraph, look for this segment of text:

```
It has survived not only five centuries, but also the leap into electronic
typesetting
```

Surround the text with a span tag as follows:

```
<span>It has survived not only five centuries, but also the leap into
electronic typesetting</span>
```

Next, inside the opening and closing <head> tags, add the following CSS rule below the paragraph selector:

```
span{
  font-family:"Times New Roman", Times, serif;
  font-size:1.2em;
  color:#CC9900;
  font-weight:bold;
}
```

Let's examine this CSS rule in greater detail.

- font-family:"Times New Roman", Times, serif;
 - Sets our font family
- font-size:1.2em;

- Sets a font size
- color:#CC9900;
 - Sets a color to a dark gold
- font-weight:bold;
 - Sets a font weight of bold

Save your file and preview the results.

http://midwestwebdesign.net/tutorials/basics/index_finished_span.html

Change the formatting of the table

Let's change the font family and size of our table by adding the following CSS rule right below the span:

```
table{
  font-family:Arial, Helvetica, sans-serif;
  font-size:.8em;
}
```

We set font family to Arial and a size to .8em, which is 80% of 100%, which is the default. Save your file and preview the results.

http://midwestwebdesign.net/tutorials/basics/index_finished_tablecss.html

Summary

Congratulations, you have built your very first web page that's viewable in all modern browsers! In this article, you learned the following:

- Create a local folder to store your files, such as your web page and graphics
- How web pages work
- Browsers
- DOCTYPEs
- Hypertext Markup Language (HTML)
 - Images
 - Paragraphs
 - Ordered and Unordered lists
 - Tables
- Formatting your text using CSS

Take with you the knowledge and concepts taught here to create the sites of your dreams!

If you have questions, please follow the link below.

<http://midwestwebdesign.net/tutorials/contact.aspx>

