**HTML - HyperText Markup Language**

Welcome to Tizag.com's HTML Tutorial! Here you will learn the basics of HyperText Markup Language (HTML), so that you may design your own web pages like the one you are viewing right now!

**[Advertise on Tizag.com](http://www.tizag.com/about/advertise.php)**

HTML is not a programming language, but rather a **markup** language. If you already know XML, HTML will be a snap for you to learn. We urge you not to attempt to blow through this tutorial in one sitting. Instead, we recommend that you spend 15 minutes to an hour a day practicing HTML and then take a break to let the information settle in. We aren't going anywhere!

**Preparation for HTML**

If you are new to HTML and haven't read through our [Beginner's Tutorial](http://www.tizag.com/beginnerT/), please take a few minutes to complete that tutorial before moving on.

Creating an HTML document is easy. To begin coding HTML, you need only two things: a **simple-text** editor, such as **Notepad**, and the dedication to follow this tutorial! Notepad is the most basic of simple-text editors, and you will probably code a fair amount of HTML with it in your early stages. Notepad++ is another popular favorite among web developers. These innovative text editors are specialized for writing simple code and they utilize color coding to help you write concise code.

**Brief HTML Background**

HTML hasn't been around for many years. The first web pages began in November 1990, and back then, there were little to no HTML standards to follow. As a result, a group called the [World Wide Web Consortium](http://www.w3.org/" \t "_blank) formed to set standards for coding HTML. We will base our teachings around these widely-accepted coding standards.

**Web Pages**

Here are some important facts about *why* web pages are so useful!

* They are a low-cost and easy way to spread information to a large audience.
* The provide yet another medium you can use to market your business!
* They serve as a platform to let the world know about you!

**Words to Know**

Throughout this tutorial, we will be using several terms that are unique to HTML. It is important for you to understand what these words mean, in the context of HTML.

* **Tag** - Used to tag or "mark-up" pieces of text. Once tagged, the text becomes HTML code to be interpreted by a web browser. Tags look like this: <tag>
* **Element** - A complete tag, having an opening <tag> and a closing </tag>.
* **Attribute** - Used to modify the value of the HTML element. Elements will often have multiple attributes.

For now, just understand that a **tag** a piece of code that acts as a label that a web browser interprets, an **element** is a complete tag with an opening and closing tag, and an **attribute** is a property value that customizes or modifies an HTML element.

# HTML - Elements

When you land on a website, all the items you see in front of you -- the paragraph texts, the page banners, and the navigation links are all ***elements*** of the web page. The term element is a just a name given to any piece of a web page. Many HTML elements are actually invisible to visitors and work quietly behind the scenes to provide web crawlers and search engines useful information about the site.

An element consists of three essential pieces: an opening tag, the content, and a closing tag.

1. **<p>** - opening paragraph tag
2. **Element Content** - "Once upon a time..."
3. **</p>** - closing tag

## A Complete HTML Element:

<p>Once upon a time...</p>

A single page can contain hundreds or thousands of elements, but when all is said and done, every HTML page should have a bare minimum of four critical elements: the ***HTML***, ***head***, ***title***, and ***body*** elements.

# HTML - <html> Element...</html>

<html> is the element that begins and ends each and every web page. Its sole purpose is to hold each web element nicely in position and serves the role of book cover; all other HTML elements are encapsulated within the <html> element. The tag also lets web browsers know, "Hey, I'm an HTML web page!", so that the browser knows how to render it. Remember to close your HTML documents with the corresponding </html> tag to signify the end of the HTML document.

If you haven't already, it is time to open up Notepad, Notepad++, or **[Crimson Editor](http://www.crimsoneditor.com" \t "_blank)** and have a new, blank document ready to go. Copy the following HTML code into your text editor.

## HTML Element Code:

<html>

</html>

Now save your file by selecting - **Menu** and then **Save**. Click on the **Save as Type** drop down box and select the option **All Files**. When you're asked to name your file, name it - *index.html*. Double-check that you did everything correctly and then press - **Save**. Now open your file in a new web browser so that you have the ability to *refresh* the page and see any new changes.

If you opened up your *index.html* document, you should be staring at your very first blank (white) web page!

# HTML - <head> Element

The <head> is usually the first element contained inside the <html> element. While it is also an element container that encapsulates other HTML elements, these elements are not directly displayed by a web browser. Instead they function behind the scenes, providing more descriptive information about the HTML document, like its page title and other meta data. Other elements used for scripting (JavaScript) and formatting (CSS) are also contained within the <head> element, and we will eventually introduce how to utilize those languages. For now, the head element will continue to lay empty except for the *title* element, which will be introduced next.

Here's a sample of the initial setup.

## HTML Head Element Code:

<html>

<head>

</head>

</html>

If you've made the code changes and refreshed the browser page, you will notice that we still have nothing happening on the page. So far, all we've done is add a couple of necessary elements that describe the web page document to the web browser. Content -- the stuff you can see -- will come next!

# HTML - <title> Element

The <title> element adds a title to a web page. Web page titles are displayed at the top of any browser window or at the top of browser tabs. They are probably the first thing seen by web surfers as pages are loaded, and web pages you bookmark are saved using the web pages' titles.

A proper title makes a good first impression, and any page caught without a title is considered unprofessional, at best.

## HTML Title Element Code:

<html>

<head>

<title>My Web Page!</title>

</head>

</html>

Save the file and refresh the browser. You should see "My Web Page!" in the upper-left bar of your browser window.

Name your webpage as you please. Just keep in mind that the best titles are brief and descriptive.

# HTML - <body> Element

The element that encapsulates all the visual elements (paragraphs, pictures, tables, etc.) of a web page is the <body> element. We will be looking at each of these elements in greater detail as the tutorial progresses, but for now, it's only important to understand that the body element is one of the four critical web page elements, and it contains all of the page's viewable content.

## HTML Body Element Code:

<html>

<head>

<title>My Web Page!</title>

</head>

<body>

<p>Once upon a time...</p>

</body>

</html>

Go ahead and view your first complete web page.

# HTML - Elements Reviewed

To recap quickly: we've laid out a set of four essential elements that are used to create a strong foundation and structure for your web page. The <html> element encapsulates **all** page content and elements, including two special elements: the <head> and <body> elements. The <head> element is a smaller container for elements that work behind the scenes of web pages, while the <body> element houses content elements such as web forms, text, images, and

# HTML - Tags

A web browser reads an HTML document from top to bottom, left to right. Each time the browser finds a tag, the tag is rendered accordingly. Paragraph tags render paragraph text, image tags render images, etc. By adding tags to an HTML document, you are not only coding HTML, but also signaling to the browser, "Hey look, this is paragraph text; now treat it as such!"

**[Advertise on Tizag.com](http://www.tizag.com/about/advertise.php)**

Do you recall that HTML elements are comprised of three major parts: the opening tag, the content, and the closing tag? As you will learn, there are infinite combinations of HTML tags/elements, including those used for forms, images, and lists. Everything displayed on an web page requires the use of a tag or two.

## HTML Tag Code:

<tag>Content</tag>

<p>This text will be rendered like a paragraph.</p>

Tags should always be written in lowercase letters if you plan on publishing the page online, as this is the web standard for XHTML and Dynamic HTML.

## HTML More Tag Code:

<body>Body Tag

<p>Paragraph Tag</p>

<h2>Heading Tag</h2>

<b>**Bold Tag**</b>

<i>*Italic Tag*</i>

</body>

# HTML - Elements Without Closing Tags

There are a few elements that do not require formal closing tags. In a way, they still have the 3 parts (opening/closing and content), but they are consolidated into one tag. The reason for this is that these tags do not really require any content to be placed within them, but sometimes may require attributes such as source URLs for images.

One prime, easy example of an HTML tag that does not require a closing tag is the line break tag.

## HTML Line Break Code:

<br />

To tell the browser we want to place a line break (carriage return) onto the site, it is not necessary to type <br>linebreak</br>. This would require a tremendous amount of effort,and if every line break tag needed all three components as other tags do, life would become redundant real quick. The better solution is to combine the opening and closing tags into a single format and shorten the number of characters needed to render a line break. Other tags, such as image tags and input tags, have also been modified in such a manner.

## HTML Code:

<img src="../mypic.jpg" /> -- Image Tag

<br /> -- Line Break Tag

<input type="text" size="12" /> -- Input Field

## Display:

--Line Break--  


As you can see from the above, the web browser is capable of interpreting the image tag as long as we inform the browser where the image file is located, using the *src* attribute. Attributes will be discussed in more detail throughout the remainder of the tutorial. For now, it's a good time to start thinking about what type of website you want to make. It is always easier to make content for a topic or to achieve a goal.

# HTML - Text

Text is the backbone of any web page. It plays an double role; it provides content for web surfers to enjoy as they skim through articles and articles of information, but it also gives Search Engines and Spiders keywords and meta data. It is because of text on web pages that we have a network of seemingly endless information available at our fingers.

**[Advertise on Tizag.com](http://www.tizag.com/about/advertise.php)**

HTML Text is probably the first element most HTML beginners learn to add to any web page, and this is generally achieved through a classic, "Hello, World!" example.

## HTML Text Code:

<html>

<head>

<title>My Web Page!</title>

</head>

<body>

Hello World!

</body>

</html>

# HTML - Paragraph Text

Any text containing more than a few lines (or sometimes even more) should exist inside of a paragraph tag <p>. This tag is reserved specifically for blocks of text, such as those you would expect to find inside your favorite novel.

## HTML <p> Tag Code:

<html>

<head>

<title>My Web Page!</title>

</head>

<body>

<p>Avoid losing floppy disks with important school...</p>

<p>For instance, let's say you had a HUGE school...</p>

</body>

</html>

## HTML Paragraph Text:

Well written HTML documents can gain popularity through Search Engine Optimization and careful coding of your HTML elements.

Precision is important when writing your code. Web spiders are a little forgiving when it comes to malformed HTML elements. For best results, do your best to ensure your code is complete and accurately constructed.

# HTML - Headings 1:6

HTML has heading tags which can be used as headers or subheaders. By placing text inside of an <h1> heading tag, for example, the text displays bold and the size of the text increases to a 24pt font size. Heading tags are numbered 1 through 6, and they change size depending on which tag you choose, with 1 being the largest font size at 24pt, and 6 being the smallest font size at 8pt.

## HTML Heading Element:

<body>

<h1>Headings</h1>

<h2>are</h2>

<h3>great</h3>

<h4>for</h4>

<h5>titles</h5>

<h6>and subtitles</h6>

</body>

Place these lines into your HTML file and you should see results similar to what you see below.

## HTML Heading Tags:

# Headings

## are

### great

#### for

##### titles

###### and subtitles

Notice that each heading has a line break (a blank line) rendered before and after it. This is a built-in attribute associated with the heading tag. Each time you place a heading tag, your web browser automatically places a line break in front of your beginning tag and after your ending tag, just like it does with <p> tags. This is expected behavior, and as a designer you need to take these line breaks into consideration when designing a layout. Later on, CSS code can be used to remove these extra line breaks or manipulate the amount of spacing, if needed.

# HTML - Formatting Text Elements w/ Tags

As you begin to place more and more text elements onto your website, you may find yourself wanting to incorporate **bold** or *italic* properties ing your text elements. HTML offers a handful of special tags that can be utilized to do just that:

**[Advertise on Tizag.com](http://www.tizag.com/about/advertise.php)**

## HTML Text Formatting Tags:

<p>An example of <b>Bold Text</b></p>

<p>An example of <em>Emphasized Text</em></p>

<p>An example of <strong>Strong Text</strong></p>

<p>An example of <i>Italic Text</i></p>

<p>An example of <sup>superscripted Text</sup></p>

<p>An example of <sub>subscripted Text</sub></p>

<p>An example of <del>struckthrough Text</del></p>

<p>An example of <code>Computer Code Text</code></p>

## HTML Formatting Text:

An example of **Bold Text**  
An example of *Emphasized Text*  
An example of **Strong Text**  
An example of *Italic Text*  
An example of superscripted Text  
An example of subscripted Text  
An example of   
An example of Computer Code Text

All of these tags add a pinch of flavor to HTML text elements, from paragraphs to lists and text links

# HTML - About Formatting Text Elements

Formatting elements such as <b> should be used sparingly, and what we mean by that is that you should only use them to bold or italicize one or two words at a time. If you wish to bold an entire paragraph, the better solution would involve Cascading Style Sheets (CSS) and adjust the element's font-weight property. You may consult how to do that in our [CSS Tutorial](http://www.tizag.com/cssT/). Ultimately, the decision is in the hands of the web designer, but generally, it's best to keep the use of these tags quick and sparse.

# HTML - Line Breaks

A line break is used in HTML text elements, and it is the equivalent of pressing **Enter** or **Return** on your keyboard. In short, a line break ends the line you are currently on and resumes on the next line. Earlier, we mentioned that each paragraph element begins and ends with a line break, which creates an empty space between the start of a paragraph element and the end of a paragraph element.

As we've mentioned, line break elements are a little different from most of the tags we have seen so far because they do not require a closing tag. Instead, their opening and closing tags are combined into a single line break element. Placing <br /> within the code is the same as pressing the return key in a word processor. Use the <br /> tag within other elements such as paragraphs (<p>).

## HTML Format Text:

<p>

Will Mateson<br />

Box 61<br />

Cleveland, Ohio<br />

</p>

## Address:

Will Mateson  
Box 61  
Cleveland, Ohio

We have created an address for a letterhead and used a line break <br /> tag inside of a paragraph element to add some line breaks to make this text appear more like an address. Here's another look as we add a signature element to the same letter.

## HTML Text Format:

<p>Sincerely,<br />

<br />

<br />

Kevin Sanders<br />

Vice President</p>

## Closing Letter:

Sincerely,  
  
  
Kevin Sanders  
*Vice President*

# HTML - <pre> Preformatting

A web browser interprets your HTML document as being one long line. Sure, you may have tabs and line breaks in Notepad which align your content so it's easier for you to read, but your browser ignores those tabs and line breaks.

**[Advertise on Tizag.com](http://www.tizag.com/about/advertise.php)**

We showed you that you can get around this problem by using the <br /> tag, but tabs and spacing are quite different from one another and can be absolutely annoying at times. One simpler solution might be to use the <pre> tag, which stands for previously formatted text.

Use the <pre> tag for any special circumstances where you wish to have the text appear exactly as it is typed. Spaces, tabs, and line breaks that exist in your actual code will be preserved with the <pre> tag.

## HTML Pre Code:

<pre>

Roses are Red,

Violets are blue,

I may sound crazy,

But I love you!

</pre>

## HTML Preformatted Text:

Roses are Red,

Violets are blue,

I may sound crazy,

But I love you!

# HTML - Attributes

Web page customization begins with HTML attributes. Attributes are like blue print schematics informing the browser how to render an HTML element. As an HTML tag is processed, the web browser looks to these attributes as guides for the construction of web elements. Without any attribute values specified, the browser will render the element using the default setting(s) (usually very boring).

**[Advertise on Tizag.com](http://www.tizag.com/about/advertise.php)**

HTML attributes are responsible for customizing web elements. As a web surfer, you've probably seen a vast assortment of color schemes, fonts, and styles -- all of which are brought to you by HTML and CSS element attributes.

# HTML - Title Attribute

The *title* attribute titles an HTML element and adds a tiny text pop-up to any HTML element, offering your web viewers a tool-tip mechanism where information can be found or where a better description of an HTML element can be seen.

Much like the tool tips found in word processing programs, this attribute can add spice to any page while offering the user priceless interactivity. As the mouse hovers over the element, a tool tip is displayed, giving the viewer just one extra piece of information.

## HTML Title Attribute:

<h2 title="Hello There!">Titled Heading Tag</h2>

Hover your mouse over the display heading and watch the *title* attribute in action!

## HTML Title Attribute:

## Titled Heading Tag

The *title* attribute is one that has not deprecated and should be used often. Many search engines are capable of identifying this attribute inside your HTML elements, granting increased search rankings based on the relevance of the *title* attribute text.

# HTML - Align Attribute

If you wish to change the horizontal alignment of your elements, you may do so using the *align* attribute. It allows you to align things left, right, or center. By default, most elements are automatically aligned left, unless otherwise specified.

## HTML Align Attribute:

<h2 align="center">Centered Heading</h2>

## HTML Align Attribute Display:

## Centered Heading

## HTML Align Attribute Code:

<h2 align="left">Left-aligned heading</h2>

<h2 align="center">Centered Heading</h2>

<h2 align="right">Right-aligned heading</h2>

## HTML Align Attribute Display:

## Left-aligned heading

## Centered heading

## Right-aligned heading

HTML attributes, including *align*, used to be the primary source for the customization of web elements, but they have now lost their market share to Cascading Style Sheets (CSS). Since most HTML attributes are now deprecated, they should ultimately be avoided in professional-level web design. Nonetheless, having an understanding of HTML attributes will prove to be a tremendous aid for anybody looking to move into professional web development using Cascading Style Sheets.

If you would like to get started now with Cascading Style Sheets, please feel free to move along to our [CSS Tutorial](http://www.tizag.com/cssT/index.php" \o "CSS Tutorial).

# HTML - Font

The <font> tag provides no real functionality by itself, but with the help of a few attributes, this tag is used to change the style, size, and color of HTML text elements. The *size*, *color*, and *face* attributes can be used all at once or individually, providing users with the ability to create dynamic font styles for any HTML element.

**[Advertise on Tizag.com](http://www.tizag.com/about/advertise.php)**

**Note:** The <font> and <basefont> tags are deprecated and should not be used for published work. Instead, use CSS styles to manipulate your font. See our [CSS Tutorial](http://www.tizag.com/cssT/) for more information.

# HTML - Font Size

Set the size of your font with *size*. The range of accepted values goes from 1 -- the smallest, to 7 -- the largest. The default size of a font is 3.

## HTML Font Size Code:

<p>

<font size="5">Here is a size 5 font</font>

</p>

## HTML Font Size Attribute:

Here is a size 5 font.

# HTML - Font Color

Set the color of your font with ***color***.

## HTML Font Color Code:

<font color="#990000">This text is hex color #990000</font>

<br />

<font color="red">This text is red</font>

## HTML Font Color Attribute:

This text is hex color #990000  
This text is red

# HTML - Font Face

Choose a different font face by specifying any font you have installed. Font face is synonymous with font type. As a web designer, be aware that if you specify a custom font type and users viewing the page don't have the exact same font installed, they will not be able to see it. Instead the chosen font will default to Times New Roman. To reduce the risk of running into this situation, you may provide a list of several fonts with the *face* attribute, such as outlined below.

## HTML Font Face Code:

<p>

<font face="Georgia, Arial, Garamond">This paragraph

has had its font...</font>

</p>

## HTML Font Face Attribute:

This paragraph has had its font formatted by the font tag!

In the example above, we have changed the font face (type) of a paragraph element and specified a list of similar fonts to apply to this element in the case that some of our viewers do not have these fonts installed.

# HTML Font - Attribute Review

## HTML Font Attributes:

|  |  |  |
| --- | --- | --- |
| **Attribute=** | **"Value"** | **Description** |
| size= | "Num. Value 1-7" | Size of your text, with 7 being biggest |
| color= | "rgb,name,or hexidecimal" | Font color |
| face= | "name of font" | Font type |

# Beautiful First Letter Style

Customize your fonts to achieve any desired look.

## HTML Code:

<p><font size="7" face="Georgia, Arial" color="maroon">C</font>ustomize

your font to achieve a desired look.</p>

## Beauty & Grace:

Customize your font to achieve a desired look.

# HTML - Text Links (Anchors)

The World Wide Web got its spidery name from the plentiful connections (links) that link websites together with the click of a button. What most people don't know is that HTML links are actually HTML anchors constructed using anchor tags (<a>).

**[Advertise on Tizag.com](http://www.tizag.com/about/advertise.php)**

## HTML Text Link:

<a>I am a text link!</a>

## HTML Text Link:

I am a text link!

While the example above appears and feels like a text link when viewed through a web browser, the element is incomplete as it is missing a vital attribute that references another web page called a Hypertext Reference (*href*).

# HTML - Hypertext Reference (href)

A Hypertext Reference (*href*) is an HTML attribute of an anchor (link) tag that requires a valid URL in order to properly direct a user to a different location. In other words, this Hypertext Reference is where users will navigate to if they do click on this link. Use the demonstration below as a reference.

## HTML Text Link Code:

<a href="http://www.tizag.com/" target="\_blank">Tizag Home</a>

<br />

<a href="http://www.espn.com/" target="\_blank">ESPN Home</a>

<br />

<a href="http://www.yahoo.com/" target="\_blank">Yahoo Home</a>

## HTML Text Links:

[Tizag Home](http://www.tizag.com" \t "_blank)   
[ESPN Home](http://www.espn.com" \t "_blank)   
[Yahoo Home](http://www.yahoo.com" \t "_blank)

The address of a website is called a Uniform Resource Locator (a URL), and it acts like a street address for a website as a user is directed from one site to another. There are different types of URLs, and each has a slightly different look. The examples above link to what are known as Global URLs, since they are external web addresses that do not reside on the Tizag.com domain. Here's a look at the different types of URLs.

## HTML Text Link Code:

Global - href="http://www.cnn.com/" Links to other domains outside your website domain.

Local - href="../internal/mypage2.html" Links to other pages within your website domain.

Internal - href="#anchorname" Links to anchors embedded in the current web page.

# HTML - Link Targets

The *target* attribute defines how each link will open when clicked. Will each one open in a new window, or will each one open in the current browser window? As the web designer, you call the shots as to how a user navigates from page to page, so long as you know how to handle the *target* attribute.

## Link Targets:

|  |  |
| --- | --- |
| **Target=** | **Description** |
| \_blank | Opens new page in a new browser window |
| \_self | Loads the new page in the current window |
| \_parent | Loads new page into a parent frame |
| \_top | Loads new page into the current browser window, cancelling all frames |

The two most important values are the top two, (target="\_blank" and target="\_self"). The *\_self* value is generally the default. It loads each new page in the current browser window, while *\_blank* opens up a new web browser window with each click of the text link.

The code below shows how you would link to ESPN.com, a popular sports website. The *target* attribute is added to allow the browser to open ESPN in a new window, so that the viewer can have a window that remains opened to our website. Here's the example.

## HTML Link Target Code:

<a href="http://www.ESPN.com" target="\_blank">ESPN.COM</a>

## \_blank Target:

**[ESPN.COM](http://www.espn.com" \t "_blank)**

Links are a big part of the user experience for any website. Always try to keep that in mind when working on a site's navigation. A web page that opens a new web browser window each time a user clicks a link is not the greatest way to entice users to stick around.

# HTML - Email Links

Creating an email link is simple. If you want people to mail you about your site, a good way to do it is place an email link with a subject line already filled out for them.

## HTML Email Link Code:

<a href="mailto:email@tizag.com?subject=Feedback" >Email@tizag.com</a>

## Email Links:

[Email@tizag.com](mailto:email@tizag.com?subject=Feedback)

In some circumstances, it may be necessary to fill in the body of the email for the user as well.

## HTML Email Link Code:

<a href="mailto:email@tizag.com?subject=Feedback&body=Sweet site!">

Email@tizag.com</a>

## Complete Email:

[Email@tizag.com](mailto:email@tizag.com?subject=Feedback&body=Sweet%20site%21)

# HTML - Download Links

Placing files available for download is done in exactly the same fashion as placing text links. However, things become complicated if we want to place image links for download. The best solution for images is to use a thumbnail links, which we will discuss in the next lesson.

## HTML Download Link Code:

<a href="http://www.tizag.com/pics/htmlT/blanktext.zip">Text Document</a>

## Download a Text Document:

[Text Document](http://www.tizag.com/pics/htmlT/blanktext.zip)

# HTML - Default Links; Base

Use the <base> tag in the *head* element to set a default URL for all links on a page to go to. It's always a good idea to set a base tag just in case your links become bugged somewhere down the line. Usually, you should set your base to your home page.

## HTML Base Link Code:

<head>

<base href="http://www.tizag.com/" />

</head>

# HTML - Comments

Comments are a great asset to new developers and a great way to place little notes to yourself reminding yourself what pieces of code are doing what. Comments are also great ways to troubleshoot bugs and code errors, as they give you the ability to comment out lines of code one at a time to search for the exact line causing problems.

**[Advertise on Tizag.com](http://www.tizag.com/about/advertise.php)**

As a sprouting young web developer, HTML code comments are your friends! A comment is a way to control which lines of code are to be ignored by the web browser and which lines of code to incorporate into your web page. There are three main reasons why you may want your code to be commented out or ignored.

* Comment out elements temporarily rather than removing them, especially if they've been left unfinished.
* Write notes or reminders to yourself inside your actual HTML documents.
* Create notes for other scripting languages like JavaScript which requires them.

## Comment Tags:

**<!--** Opening Comment Tag   
**-- >** Closing Comment Tag

As you can see, comments are also comprised of an opening and closing tag, (<!-- -->). Like other HTML elements, these tags can span across multiple lines of code, allowing you to comment out large blocks of HTML code. Any HTML elements that are encapsulated inside of the comment tags will be ignored by the web browser. This makes comment tags quite useful for debugging, as they allow the developer to temporarily comment out pieces of an HTML document without having to immediately delete the code from the HTML document.

## HTML Comment Code:

<!--Note to self: This is my banner image! Don't forget -->

<img src="http://www.tizag.com/pics/tizagSugar.jpg" height="100" width="400" />

## Comment to self:



Placing notes and reminders to yourself is a great way to remember your thoughts and to keep track of elements embedded inside the web page. Also, your code may exist for many, many years, and these notes to yourself are a great way to remember what was going on, since you may not remember five or more years down the road.

All combinations of text placed within the comment tags will be ignored by the web browser. This includes any HTML tags, scripting language(s), etc.

# HTML - <!-- Commenting Existing Code -->

As a web designer, you may sometimes have different websites in progress at once, and it might be easy to leave some HTML elements unfinished. In this case, you may comment out an element until it is 100% ready for the site. Below, we have commented out an input form element, since we are not quite ready to receive input from our users.

## HTML Code:

<!-- <input type="text" size="12" /> -- Input Field -->

Now when we are ready to show that element, we can simply remove the comment tags, and our browser will readily display the element!

## HTML Code:

<input type="text" size="12" />

## Input Field:



Comment out elements and bits of code that you may want to recall and use at a later date. Nothing is more frustrating than deleting bits of code only to turn around moments later and realize that you now need to recode them.

# HTML - Lists

HTML lists appear in web browsers as bulleted lines of text. There are actually three different types of HTML lists, including unordered lists (bullets), ordered lists (numbers), and definition lists (think: dictionaries). Each list type utilizes its own unique list tag, which we'll demonstrate below.

**[Advertise on Tizag.com](http://www.tizag.com/about/advertise.php)**

## HTML List Item Code:

<body>

<ul>

<li>I am a list item!>

<li>I am a list item too!>

<li>I am a list item also!>

</ul>

</body>

## HTML List Items:

* I am a list item!
* I am a list item too!
* I am a list item also!

The actual list tags themselves, such as <ul>, are nothing but container elements for list items (<li>). They work behind the scenes to identify the beginning and ending of an HTML list element.

# HTML - Unordered Lists

An unordered list (<ul>) signifies to a web browser that all list items contained inside the <ul> tag should be rendered with a bullet preceding the text. The default bullet type for most web browsers is a full disc (black circle), but this can be adjusted using an HTML attribute called *type*.

## HTML Default Bullet List Code:

<h4 align="center">Shopping List</h4>

<ul>

<li>Milk</li>

<li>Toilet Paper</li>

<li>Cereal</li>

<li>Bread</li>

</ul>

## HTML Default Disc Bullets:

#### Shopping List

* Milk
* Toilet Paper
* Cereal
* Bread

To render a list with a different bullet type, add a *type* attribute to the unordered list element. Using the same code in the example above, replace the <ul> line from the previous example with any of the lines listed below to change the bullet from disc shape to another shape.

## HTML Unordered List Type Code:

<ul type="square">

<ul type="disc">

<ul type="circle">

## HTML Unordered List Types:

|  |  |  |
| --- | --- | --- |
| **type="square"** | **type="disc"** | **type="circle"** |
| * Milk * Toilet Paper * Cereal * Bread | * Milk * Toilet Paper * Cereal * Bread | * Milk * Toilet Paper * Cereal * Bread |

# HTML - Ordered Lists

An ordered list is defined using the <ol> tag, and list items placed inside of an ordered list are preceded with numbers instead of bullets.

## HTML Numbered/Ordered List:

<h4 align="center">Goals</h4>

<ol>

<li>Find a Job</li>

<li>Get Money</li>

<li>Move Out</li>

</ol>

## HTML Numbered List:

#### Goals

1. Find a Job
2. Get Money
3. Move Out

The numbering of an HTML list can be changed to letters or Roman Numerals by once again adjusting the type attribute.

## HTML Code:

<ol type="a">

<ol type="A">

<ol type="i">

<ol type="I">

## Ordered List Types:

|  |  |  |  |
| --- | --- | --- | --- |
| **Lower-Case Letters** | **Upper-Case Letters** | **Lower-Case Numerals** | **Upper-Case Numerals** |
| 1. Find a Job 2. Get Money 3. Move Out | 1. Find a Job 2. Get Money 3. Move Out | 1. Find a Job 2. Get Money 3. Move Out | 1. Find a Job 2. Get Money 3. Move Out |

The *start* attribute allows you to further customize an HTML ordered list by setting a new starting digit for the ordered list element.

## HTML Numbered List Start Attribute:

<h4 align="center">Goals</h4>

<ol start="4" >

<li>Buy Food</li>

<li>Enroll in College</li>

<li>Get a Degree</li>

</ol>

## HTML Numbered List Start - 4:

#### Goals

1. Buy Food
2. Enroll in College
3. Get a Degree

# HTML - Definition Term Lists

HTML definition lists (<dl>) are list elements that have a unique array of tags and elements; the resulting listings are similar to those you'd see in a dictionary.

* <dl> - opening clause that defines the start of the list
* <dt> - list item that defines the definition term
* <dd> - definition of the list item

These lists displace the word term (<dt>) in such a way that it rests just above the definition element (<dd>) to offer a very unique look. For best, results we suggest bolding the definition terms with the bold tag <b>.

## HTML Definition List Code:

<dl>

<dt><b>Fromage</b></dt>

<dd>French word for cheese.</dd>

<dt><b>Voiture</b></dt>

<dd>French word for car.</dd>

</dt>

</dl>

## HTML Definition List Display:

**Fromage**  
French word for cheese.  
**Voiture**  
French word for car.

# HTML - Images & Pictures

Images are a staple of any web designer, so it is very important that you understand how to use them properly. In order to place an image onto a website, one needs to know where the image file is located within the file tree of the web server -- the URL (Unified Resource Locator).

**[Advertise on Tizag.com](http://www.tizag.com/about/advertise.php)**

Use the <img /> tag to place an image on your webpage. Like the <br /> tag, <img /> tag does not require a formal ending tag. Instead, all we need to do to close this tag out with a slash (/) placed just inside the ending bracket (/>).

## HTML Image Code:

<img src="http://www.tizag.com/pics/htmlT/sunset.gif" />

## HTML Image:

http://www.tizag.com/pics/htmlT/sunset.gif

So that you can follow along with us, we've provided a global URL of an image we have stored on Tizag's web server. If you have an active connection to the internet, you should be able to use the URL in the example to render this image on your own web page. If not, you will have to download (right click the image and **Save As**...on a PC, control-click and **Save** on a Mac). Once saved to your local computer, point the image *src* attribute to the recently downloaded image file. It might help to save the image file in the same folder as your web page.

# HTML - Image *src* Attribute

The source attribute (*src*) is what makes an image tag display an image. An image source is a URL value and should point to the directory location of an image file. Let's take one more look at the code from our first HTML image example.

## HTML Image Code:

<img src="http://www.tizag.com/pics/htmlT/sunset.gif" />

An image source value is essentially the URL of a picture file and tells the web browser where the image is located so that it can then display the image correctly.

# HTML - Source URLs

Image source URLs can be either local or global, meaning that the picture files you wish to display on your website can be either hosted locally on your machine (local) or hosted elsewhere on some other web site domain (global).

* **Global:** http://www.tizag.com/pics/htmlT/sunset.gif
* **Local:** pics/sunset.gif

Local URLs are relative to the file path of the web page itself. For example, if the picture file is placed inside the same directory as the web page, then the local URL for the image would simply be the name of the image, since it is residing in the same directory as the HTML page.

## Local URLs Explained:

|  |  |
| --- | --- |
| **Local Src** | **Location Description** |
| src="sunset.gif" | picture file resides in same directory as .html file |
| src="pics/sunset.gif" | picture file resides in the *pics* directory |
| src="../sunset.gif" | picture resides one folder "up" from the .html file |
| src="../pics/sunset.gif" | picture file resides in the *pics* directory, one folder "up" from the .html file. |

Pictures must reside on the same web host as your .html file in order for you to use local URLs. A URL cannot contain drive letters such as C:\, since a *src* URL is a relational interpretation based on the location of the.html file and the location of the picture file. Something like src="C:\www\web\pics\" will *not* work.

Each URL format has its pros and cons. Using the URL of pictures on other sites poses a problem if the other site happens to change the physical location of the picture file. Copying the file directly to your web server solves this problem. However, as you continue to upload picture files to your system, you may eventually run short on hard drive space. Use your best judgment based upon your situation.

# HTML - Image Height and Width Attributes

*Height* and *width* are HTML attributes that define an element's height and width properties. These values can either be percentage-based (%) or rely on pixel sizes.

## HTML Height and Width Attributes:

<img src="sunset.gif" height="50" width="100" />

## HTML Height and Width (Pixels):

http://www.tizag.com/pics/htmlT/sunset.gif

Above, we've used hard-coded pixel values for the height and width of the sunset image to ensure that this image will always render 50 pixels high by 100 pixels wide. By hard-coding these values, we are ensuring that the image will only display 50 pixels high by 100 pixels wide, even if the picture file itself happens to be much larger. If the dimensions of the picture are much larger, then we risk some severe skewing as the browser tries to shrink our image into our small box.

Height and width values can also be a percentage. Percentage values are relative to the parent HTML element (usually the body), which means if you have a parent element like a <body> element that is the whole screen (1024x768), then an image with a height and width of 100% will take up that entire body element (1024x768). In our example below, we have placed the image in a table element that is about 400 pixels wide by 200 pixels tall.

## HTML Height and Width Code:

<table height='200' width='400'>

<tr>

<td>

<img src="sunset.gif" height="100%" width="100%">

</td>

</tr>

</table>

## HTML Height and Width (Percentage):

http://www.tizag.com/pics/htmlT/sunset.gif

This image is now pixelated, having been rendered to fill the 400x200 table element. The reason we are seeing a low-quality, pixelated rendering of the image is because the actual image file is much smaller than 400x200 and has been stretched by the web browser because we coded it to do so.

Here's a few things to remember when trying to place images on your web page:

1. **Maintain the same height to width ratio.** The ratio is critical, and must be maintained to avoid skewing.
2. **Always scale down.** -- Larger images will always scale down nicely and continue to look sharp.

If no height or width attribute is specified inside the <img> tag, the browser will use the actual dimensions of the image file to render the image. This can cause problems with the page layout if the picture file is too large, as other HTML elements will be moved further down the page in the event of an over-sized image.

Another concept to keep in mind is that as a browser begins rendering HTML components, it handles them one after another in sequence. Before it can move from one element on to the next, the browser needs to know the size and shape of an element. If this information is provided in the tag, that's one less step required by the browser to render an image element and will result in the page loading faster for your users.

# HTML - Alternative Attribute

The *alt* attribute specifies alternate text to be displayed if for some reason the browser cannot find the image, or if a user has image files disabled in their web browser settings. Text-only browsers greatly depend on the *alt* attribute since they are not capable of displaying pictures.

## HTML Alternative Attribute (alt):

<img src="http://example.com/brokenlink/sunset.gif" alt="Beautiful Sunset" />

## HTML Alternative Text Attribute:

The *alt* attribute is also an attribute that search engines may look for when displaying images. The text value contained within this attribute must reflect the substance of the image in order to receive "credit" from a search engine.

# HTML - Horizontally Align Images

Images can be aligned horizontally to the right or to the left of other elements using the *align* attribute. Image elements are aligned to the left by default.

1. align
   * right
   * left

## HTML Align Attribute Code:

<img src="sunset.gif" align="right">

## HTML Image Align: Right:

http://www.tizag.com/pics/htmlT/sunset.gifAs you can see, the image's right edge has now been aligned with the right edge of the display box. Since the display box is the parent element, this is the desired behavior for the align attribute. If we take this example a step further, you can achieve some really great designs by embedding aligned images inside of paragraph <p> elements.

## HTML Image Align Code:

<p>This is paragraph 1, yes it is...</p>

<p>

<img src="sunset.gif" align="right">

The image will appear along the...isn't it?

</p>

<p>This is the third paragraph that appears...</p>

## Image Wrap Around:

This is paragraph 1, yes it is. I think this paragraph serves as a nice example to show how this image alignment works.

http://www.tizag.com/pics/htmlT/sunset.gifThe image will appear along the right-hand side of the paragraph. As you can see this is very nice for adding a little eye candy that relates to the specified paragraph. If we were talking about beautiful tropical sunsets, this picture would be perfect. But we aren't talking about that, so it's rather a waste, isn't it?

This is the third paragraph that appears below the paragraph with the image!

# HTML Gifs vs. Jpegs vs. PNGs

**Gifs** are best used for banners, clip art, and buttons. The main reason for this is that .gif files can have transparent backgrounds -- a priceless attribute when it comes to web design. On the down side, .gif files are limited to only 256 colors and any .gif image containing more than a few colors tends to have a larger file size than their .jpeg or .png counterparts. Large picture files are a plague of web design!

**Jpegs** have an unlimited color wheel and a high compression rate, which downsizes your load times and saves on hard drive space. Although .jpeg (or .jpg) files don't allow for transparent backgrounds, their size/quality ratio is outstanding. It's best to use .jpeg files for photo galleries or artwork. Avoid using .jpeg files for graphical designs, though; stick to using them for thumbnails, backgrounds, and photo galleries.

**PNG** image files are the best of both worlds. They have a large color wheel, low file size, and allow for transparencies like .gif images do. With a high-compression rate and transparent coloring, they might just be the best format for any web graphics.

When in doubt, try saving an image in multiple formats and decide which is better, keeping file size and quality in mind.

# HTML - Image Links

Image links are constructed as you might expect, by embedding an <img> tag inside of an anchor element <a>. Like HTML text links, image links require opening and closing anchor tags, but instead of placing text between these opening and closing tags, the developer needs to place an image tag -- with a valid source attribute value of course.

**[Advertise on Tizag.com](http://www.tizag.com/about/advertise.php)**

If you do not know how to use the image tag, please take a minute and review the [HTML Image](http://www.tizag.com/htmlT/images.php) lesson before reading further.

## HTML Image Link Code:

<a href="http://www.espn.com" target="\_blank">

<img src="ahman.gif" />

</a>

## HTML Image Link:

[http://www.tizag.com/pics/htmlT/ahman.gif](http://www.espn.com/)

By default, many browsers add a small border around image links. This default behavior is intended to give web viewers the ability to quickly decipher the difference between ordinary images and image links. Since this default is different from web browser to web browser, it may be best to squelch this ambiguity and set the *border* attribute of the image tag to 0 or 1.

## HTML Image Border Code:

<a href="http://www.espn.com" target="\_blank">

<img src="ahman.gif" border="0" />

</a>

## HTML Image Link; No Border:

[http://www.tizag.com/pics/htmlT/ahman.gif](http://www.espn.com/)

# HTML - Thumbnails

Thumbnails are by far the most common type of image link seen in today's world. To create a thumbnail, one must save a low-quality version of a picture with smaller dimensions. Then, one should link this low-quality picture to its higher-quality counterpart.

Thumbnails are intended to give your audience quick previews of images without them having to wait for the larger, higher-quality image to load. Photo galleries make heavy use of thumbnails, and they will allow you to display multiple pictures on one page with ease.

## HTML Thumbnail Code:

<a href="sunset.gif">

<img src="thmb\_sunset.gif" />

</a>

## HTML Thumbnails:

[http://www.tizag.com/pics/htmlT/thmb_sunset.gif](http://www.tizag.com/pics/htmlT/sunset.gif)

# HTML - Tables

An HTML table is an element comprised of table rows and columns, much like you'd see when working with an application such as Excel. Tables are container elements, and their sole purpose is to house other HTML elements and arrange them in a tabular fashion -- row by row, column by column.

**[Advertise on Tizag.com](http://www.tizag.com/about/advertise.php)**

Tables may seem difficult at first, but after working through this lesson, you'll see that they aren't so horrible. A table element consists of three different HTML tags including the <table> tag, <tr> (table rows), and the <td> (table columns) tags.

## HTML Table Code:

<table border="1">

<tr>

<td>Row 1 Cell 1</td>

<td>Row 1 Cell 2</td>

</tr>

<tr>

<td>Row 2 Cell 1</td>

<td>Row 2 Cell 2</td>

</tr>

</table>

## Basic HTML Table Layout:

|  |  |
| --- | --- |
| Row 1 Cell 1 | Row 1 Cell 2 |
| Row 2 Cell 1 | Row 2 Cell 2 |

We've adjusted the formatting of the code by adding additional spaces before some of the table elements, but this has no bearing on the rendering of the element. It simply helps keep track of each tag/element and helps us ensure we don't miss an opening or closing tag which would prevent our table element from rendering correctly. We've also added a *border* attribute to ensure the table cells/rows are more visible to our readers.

Content elements like HTML lists, images, and even other table elements can be placed inside each table cell. Doing so aligns the elements in a tabular fashion and provides structure.

## HTML Table Code:

<table border="1">

<tr>

<td width="50%">

<ul>

<li>List Item 1</li>

<li>List Item 2</li>

<li>List Item 3</li>

</ul>

</td>

<td>

<ul>

<li>List Item 4</li>

<li>List Item 5</li>

<li>List Item 6</li>

</ul>

</td>

</tr>

<tr>

<td>

<p>Avoid losing floppy disks with important school...</p>

</td>

<td>

<a href="http://www.espn.com" target="\_blank" rel="nofollow">

<img src="http://www.tizag.com/pics/htmlT/ahman.gif" class="linksESPN" />

</a>

</td>

</tr>

</table>

## HTML Table 2:

|  |  |
| --- | --- |
| * List Item 1 * List Item 2 * List Item 3 | * List Item 4 * List Item 5 * List Item 6 |
| Avoid losing floppy disks with important school... | [http://www.tizag.com/pics/htmlT/ahman.gif](http://www.espn.com/) |

HTML tables allow the web designer to align page content in a tabular fashion while spanning elements horizontally across the web page, rather than stacking them up one on top of another.

# HTML - Table Rows & Table Columns

A table can contain an infinite number of table rows. Each table row is essentially a table element itself, with an opening and closing tag (<tr> </tr>). Table columns are also considered child elements of HTML tables, and like table rows, an HTML table may contain an infinite number of table data cells (<td> <tr>).

Table rows and columns are container elements that house other HTML elements such as text links, images, and lists, as we've seen in previous examples. Below, we've applied a background color to the table example in order to help distinguish the different table elements.

## HTML Table Code:

<table border="1">

<tr title="You are looking at Row 1" bgcolor="silver">

<td>Row 1 Cell 1</td>

<td>Row 1 Cell 2</td>

</tr>

<tr title="You are looking at Row 2" bgcolor="aqua">

<td>Row 2 Cell 1</td>

<td>Row 2 Cell 2</td>

</tr>

</table>

## HTML Table Code Example:

|  |  |
| --- | --- |
| Row 1 Cell 1 | Row 1 Cell 2 |
| Row 2 Cell 1 | Row 2 Cell 2 |

# HTML - Tables: Spanning Multiple Rows and Cells

Use *rowspan* to span multiple rows merging together table rows and *colspan* to span across multiple columns.

## HTML Table Rowspan Attribute:

<table border="1">

<tr>

<td><b>Column 1</b></td>

<td><b>Column 2</b></td>

<td><b>Column 3</b></td>

</tr>

<tr>

<td rowspan="2">Row 1 Cell 1</td>

<td>Row 1 Cell 2</td>

<td>Row 1 Cell 3</td>

</tr>

<tr>

<td>Row 2 Cell 2</td>

<td>Row 2 Cell 3</td>

</tr>

<tr>

<td colspan="3">Row 3 Cell 1</td>

</tr>

</table>

## HTML Colspan and Rowspan Attributes:

|  |  |  |
| --- | --- | --- |
| **Column 1** | **Column 2** | **Column 3** |
| Row 1 Cell 1 | Row 1 Cell 2 | Row 1 Cell 3 |
| Row 2 Cell 2 | Row 2 Cell 3 |
| Row 3 Cell 1 | | |

# Cell Padding and Spacing

With the *cellpadding* and *cellspacing* attributes, you will be able to adjust the spacing between table cells. Setting the *cellpadding* attribute determines how much space will exist between a table cell border and the elements contained within it, whereas *cellspacing* determines how much space will exist between each table cell. Color has been added to the table below to emphasize these attributes.

## HTML Cellpadding/Cellspacing Code:

<table border="1" cellspacing="10" bgcolor="rgb(0,255,0)">

<tr>

<td><b>Column 1</b></td>

<td><b>Column 2</b></td>

</tr>

<tr>

<td>Row 1 Cell 1</td>

<td>Row 1 Cell 2</td>

</tr>

<tr>

<td>Row 2 Cell 1</td>

<td>Row 2 Cell 2</td>

</tr>

</table>

## HTML Cellspacing and Padding:

|  |  |
| --- | --- |
| **Column 1** | **Column 2** |
| Row 1 Cell 1 | Row 1 Cell 2 |
| Row 2 Cell 1 | Row 2 Cell 2 |

And now we will change the *cellpadding* of the table and remove the *cellspacing* from the previous example. This should clearly demonstrate the difference between *cellpadding* and *cellspacing.*

## HTML Code:

<table border="1" **cellpadding="10"** bgcolor="rgb(0,255,0)">

<tr>

<td><b>Column 1</b></td>

<td><b>Column 2</b></td>

</tr>

<tr>

<td>Row 1 Cell 1</td>

<td>Row 1 Cell 2</td>

</tr>

<tr>

<td>Row 2 Cell 1</td>

<td>Row 2 Cell 2</td>

</tr>

</table>

## HTML Cell Pads:

|  |  |
| --- | --- |
| **Column 1** | **Column 2** |
| Row 1 Cell 1 | Row 1 Cell 2 |
| Row 2 Cell 1 | Row 2 Cell 2 |

The value you specify for padding and spacing is interpreted by the browser as a pixel value. So a value of 10 is simply 10 pixels wide. Most HTML attributes that use numeric values for their measurements represent a pixel value.

# HTML - The bgcolor Attribute

The *bgcolor* attribute is used to set the background color of an HTML element. *Bgcolor* is one of those attributes that has become deprecated with the implementation of Cascading Style Sheets (see [CSS Backgrounds](http://www.tizag.com/cssT/background.php)). The reason we've included it in this tutorial is because it will give us an opportunity to introduce web colors and also add some life to our HTML web page as we continue to progress through this tutorial. It will serve as a visual aid for you as you are learning the mechanics of building a table.

**[Advertise on Tizag.com](http://www.tizag.com/about/advertise.php)**

Without much effort, we can bring that boring white web page to life by adding some color with the *bgcolor* attribute.

## HTML Bgcolor Code:

<body bgcolor="silver">

<p>This page now has a SILVER background!</p>

</body>

## HTML Bgcolor:

|  |
| --- |
| This page now has a SILVER background! |

# HTML - Web Colors

Our example uses the text value, which is one of three different types of color values that can be used with the *bgcolor* attribute. Below is a table of the 16 basic HTML color values that are available to HTML web designers.

## HTML Basic Colors:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | Black |  | Gray |  | Silver |  | White |
|  | Yellow |  | Lime |  | Aqua |  | Fuchsia |
|  | Red |  | Green |  | Blue |  | Purple |
|  | Maroon |  | Olive |  | Navy |  | Teal |

While the table above illustrates only 16 colors, 16 is surely not the limit to our color wheel. As we mentioned, HTML supports three different types of color values including *text values* (which we've pretty much covered above), *numeric*, (RGB) and *hexadecimal values*. We'll go into more detail regarding these values so just sit tight. This next example offers a sneak peak at what these values may look like.

## HTML Code:

<table bgcolor="#ff0000" border="1"><tr>

<td>A red colored table background using hexadecimal values "#FF0000".</td>

</tr></table>

<table bgcolor="rgb(0, 0, 255)" border="1"><tr>

<td>A blue colored table background using numeric, RGB values "rgb(0, 0, 255)".</td>

</tr></table>

<table bgcolor="lime" border="1"><tr>

<td>A lime colored table background using color names.</td>

</tr></table>

## Table Bgcolor Values:

|  |
| --- |
| A lime colored table background using color names. |

|  |
| --- |
| A red colored table background using hexadecimal values "#FF0000". |

|  |
| --- |
| A blue colored table background using numeric, RGB values "rgb(0, 0, 255)". |

*Hexadecimal* and *numeric* color values (RGB) allow HTML developers to expand the color wheel beyond 16 colors. Way beyond 16, in fact. Nonetheless, there's no need to memorize 256+ unique color combinations; instead, we can use numeric and hexadecimal values and calculate color shades. We'll show you how to use them in our [HTML Color Codes](http://www.tizag.com/htmlT/htmlcolorcodes.php" \o "HTML Color Codes) page.

# HTML - Coloring Fonts, Table Rows, & Table Columns

Here's a few common examples of *bgcolor* in action.

## HTML Bgcolor Code:

<table>

<tr bgcolor="#FFFF00"><td>This Row is Yellow!</td></tr>

<tr bgcolor="#AAAAAA"><td>This Row is Gray!</td></tr>

<tr bgcolor="#FFFF00"><td>This Row is Yellow!</td></tr>

<tr bgcolor="#AAAAAA"><td>This Row is Gray!</td></tr>

<tr bgcolor="#FFFF00"><td>This Row is Yellow!</td></tr>

<tr bgcolor="#AAAAAA"><td>This Row is Gray!</td></tr>

</table>

## Alternating Table Row Colors:

|  |
| --- |
| This Row is Yellow! |
| This Row is Gray! |
| This Row is Yellow! |
| This Row is Gray! |
| This Row is Yellow! |
| This Row is Gray! |

Check out this "Scoreboard" we made with the use of *font color* and *bgcolor*!

## HTML Code:

<table bgcolor="#000000">

<tr><td bgcolor="#009900" align="right">

<font color="#FFFF00">Green Bay</font></td>

<td><font color="#FFFFFF">13</font></td></tr>

<tr><td bgcolor="#0000FF" align="right">

<font color="#DDDDDD" >New England</font></td>

<td><font color="#FFFFFF">27</font></td>

</tr>

</table>

## Scoreboard:

|  |  |
| --- | --- |
| Green Bay | 13 |
| New England | 27 |

# HTML - Color Codes

Let's first review the 16 generic color values we mentioned previously before diving into the other, more complicated HTML coloring systems of *numeric* and *hexadecimal* values.

**[Advertise on Tizag.com](http://www.tizag.com/about/advertise.php)**

## HTML String Color Codes:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | Black |  | Gray |  | Silver |  | White |
|  | Yellow |  | Lime |  | Aqua |  | Fuchsia |
|  | Red |  | Green |  | Blue |  | Purple |
|  | Maroon |  | Olive |  | Navy |  | Teal |

Any of the string values listed above such as "teal", "black", or "gray" can be passed as a color value to the HTML *bgcolor* attribute.

## HTML Bgcolor Code Values:

bgcolor="purple"

# HTML - Colors: Numeric (RGB) Values

We do not recommend that you use RGB for safe web design because there are still a handful of internet browsers that do not support numeric color values. However, these values may pop up from time to time, and it is a good idea to have an understanding of how they work.

Colors shown on a computer monitor or any digital device are constructed using various amounts of red, green, and blue light. By blending together different amounts of each color of light, a computer monitor is able to display millions of different colors depending on the quality of the computer monitor. This concept is precisely what HTML numeric (RGB) values use. They specify the amount of each of the different colors of light to blend together (red, green, and blue light).

In a numeric color value the **RGB** stands for Red, Green, Blue (as in light) and the format for a RGB value is rgb(RED, GREEN, BLUE), just like the name implies. A numeric color value is essentially a comma-separated list of values ranging from 0 (none of that color) to 255 (fully that color) that are interpreted and then mixed together by the web browser and ultimately passed to the computer monitor for display.

## Red, Green, and Blue Values:

|  |  |
| --- | --- |
| bgcolor="rgb(255,0,0)" | Red |
| bgcolor="rgb(0,255,0)" | Green |
| bgcolor="rgb(0,0,255)" | Blue |
| bgcolor="rgb(0,0,0)" | Black |
| bgcolor="rgb(255,255,255)" | White |
| bgcolor="rgb(120,120,120)" | Grey |
| bgcolor="rgb(240,40,120)" | Pink |

# HTML Coloring System - Hexadecimal

The hexadecimal system is complex and difficult to understand at first. Rest assured that the system becomes much, MUCH easier with practice, and as a blossoming web developer, it is critical that you understand hexadecimals to be capable of using them in your own websites. They are far more reliable, and are more widely compatible with web browsers, and are currently the web standard for coloring HTML elements.

## HTML Color: Hexadecimal Code:

<!-- Hexadecimal Color Value -->

bgcolor="#004488"

A hexadecimal is a 6-digit numeric representation of a color, comprised of three different components (the red, the green, and the blue). The first two digits (00) represent a red value, the next two (44) are a green value, and the last (88) are the blue value. These three sets of values combined form the final color shade.

## Hexadecimal Formula: 2 digits at a time:

([first\_digit]\* 16) + ([second\_digit]) = primary color value

Let's apply this formula to an example and see how it works!

## HTML Color: Hexadecimal Example Code:

bgcolor="#004488"

We now know that this six digit value is actually three separate values working together to create a single shade. Let's separate each value and perform some calculations based on the formula we have listed above.

* **00** - represents the Red value.
* **44** - represents the Green value.
* **88** - represents the Blue value.

## Hexadecimal Formula: Calculated:

<!-- Red Value 00 -->

(0 \* 16) + (0) = 0 red value

<!-- Green Value 44 -->

(4 \* 16) + (4) = 68 green value

<!-- Blue Value 88 -->

(8\* 16) + (8) = 144 blue value

By applying this formula to each value pair, we now have a value that resembles the rgb(0,68,144) values we've seen before.

But now for the curve ball. Since hexadecimal values are limited to six single digits, you may assume that the value #999999 is the largest hexadecimal value that is possible. This is not the case. The hexadecimal system uses letters (A-F) to represent values greater than nine. *But why?* This is an excellent question.

It's probably easiest to understand by working through the calculation of our supposed maximum hexadecimal value #999999.

## Hexadecimal Formula: #999999 Calculated:

<!-- Red Value 99 -->

(9 \* 16) + (9) = 153 red value

<!-- Green Value 99 -->

(9 \* 16) + (9) = 153 green value

<!-- Blue Value FF -->

(9 \* 16) + (9) = 153 blue value

Based on the calculations above, we are seeing a maximum value of *rgb(153,153,153)*. This does not coincide with numeric values that have a maximum value of rgb(255,255,255) and it greatly limits what would be the available color spectrum. Therefore, letters are deployed to represent numbers and this is what makes it possible to extend the color wheel threshold to the max.

The table below identifies how each letter (A-F) corresponds to the hexadecimal numeric equivalent.

## Hexadecimal Color Values:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Decimal** | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
| **Hexadecimal** | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F |

Taking a look at the next few calculations, you should be able to see that, by using letters, we are able to extend the color wheel giving HTML developers a greater selection of color options for web page designers.

## HTML Color: Hexadecimal Example Code:

bgcolor="#AA93FF"

## Hexadecimal Formula: Calculated:

<!-- Red Value AA -->

(10 \* 16) + (10) = 170 red value

<!-- Green Value 93 -->

(9 \* 16) + (3) = 147 green value

<!-- Blue Value FF -->

(15 \* 16) + (15) = 255 blue value

## HTML Color: Hexadecimal Example #2:

bgcolor="#CC7005"

## Hexadecimal Formula: Example #2 Calculated:

<!-- Red Value CC -->

(12 \* 16) + (12) = 204 red value

<!-- Green Value 70 -->

(7 \* 16) + (0) = 112 green value

<!-- Blue Value 05 -->

(0 \* 16) + (5) = 5 blue value

# HTML - Color Chart

Below is the hexadecimal representation for an array of HTML background colors. The hexadecimal value that you see displayed in each box represents the value to get the background color of that cell.

**[Advertise on Tizag.com](http://www.tizag.com/about/advertise.php)**

If you would like more information about using HTML color, check out our [HTML Color Codes](http://www.tizag.com/htmlT/htmlcolorcodes.php) lesson.

## HTML Color Values:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **#000000** | **#000033** | **#000066** | **#000099** | **#0000CC** | **#0000FF** |
| **#003300** | **#003333** | **#003366** | **#003399** | **#0033CC** | **#0033FF** |
| **#006600** | **#006633** | **#006666** | **#006699** | **#0066CC** | **#0066FF** |
| **#009900** | **#009933** | **#009966** | **#009999** | **#0099CC** | **#0099FF** |
| **#00CC00** | **#00CC33** | **#00CC66** | **#00CC99** | **#00CCCC** | **#00CCFF** |
| **#00FF00** | **#00FF33** | **#00FF66** | **#00FF99** | **#00FFCC** | **#00FFFF** |
|  |  |  |  |  |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **#330000** | **#330033** | **#330066** | **#330099** | **#3300CC** | **#3300FF** |
| **#333300** | **#333333** | **#333366** | **#333399** | **#3333CC** | **#3333FF** |
| **#336600** | **#336633** | **#336666** | **#336699** | **#3366CC** | **#3366FF** |
| **#339900** | **#339933** | **#339966** | **#339999** | **#3399CC** | **#3399FF** |
| **#33CC00** | **#33CC33** | **#33CC66** | **#33CC99** | **#33CCCC** | **#33CCFF** |
| **#33FF00** | **#33FF33** | **#33FF66** | **#33FF99** | **#33FFCC** | **#33FFFF** |
|  |  |  |  |  |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **#660000** | **#660033** | **#660066** | **#660099** | **#6600CC** | **#6600FF** |
| **#663300** | **#663333** | **#663366** | **#663399** | **#6633CC** | **#6633FF** |
| **#666600** | **#666633** | **#666666** | **#666699** | **#6666CC** | **#6666FF** |
| **#669900** | **#669933** | **#669966** | **#669999** | **#6699CC** | **#6699FF** |
| **#66CC00** | **#66CC33** | **#66CC66** | **#66CC99** | **#66CCCC** | **#66CCFF** |
| **#66FF00** | **#66FF33** | **#66FF66** | **#66FF99** | **#66FFCC** | **#66FFFF** |
|  |  |  |  |  |  |

## HTML Color Values Continued:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **#990000** | **#990033** | **#990066** | **#990099** | **#9900CC** | **#9900FF** |
| **#993300** | **#993333** | **#993366** | **#993399** | **#9933CC** | **#9933FF** |
| **#996600** | **#996633** | **#996666** | **#996699** | **#9966CC** | **#9966FF** |
| **#999900** | **#999933** | **#999966** | **#999999** | **#9999CC** | **#9999FF** |
| **#99CC00** | **#99CC33** | **#99CC66** | **#99CC99** | **#99CCCC** | **#99CCFF** |
| **#99FF00** | **#99FF33** | **#99FF66** | **#99FF99** | **#99FFCC** | **#99FFFF** |
|  |  |  |  |  |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **#CC0000** | **#CC0033** | **#CC0066** | **#CC0099** | **#CC00CC** | **#CC00FF** |
| **#CC3300** | **#CC3333** | **#CC3366** | **#CC3399** | **#CC33CC** | **#CC33FF** |
| **#CC6600** | **#CC6633** | **#CC6666** | **#CC6699** | **#CC66CC** | **#CC66FF** |
| **#CC9900** | **#CC9933** | **#CC9966** | **#CC9999** | **#CC99CC** | **#CC99FF** |
| **#CCCC00** | **#CCCC33** | **#CCCC66** | **#CCCC99** | **#CCCCCC** | **#CCCCFF** |
| **#CCFF00** | **#CCFF33** | **#CCFF66** | **#CCFF99** | **#CCFFCC** | **#CCFFFF** |
|  |  |  |  |  |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **#FF0000** | **#FF0033** | **#FF0066** | **#FF0099** | **#FF00CC** | **#FF00FF** |
| **#FF3300** | **#FF3333** | **#FF3366** | **#FF3399** | **#FF33CC** | **#FF33FF** |
| **#FF6600** | **#FF6633** | **#FF6666** | **#FF6699** | **#FF66CC** | **#FF66FF** |
| **#FF9900** | **#FF9933** | **#FF9966** | **#FF9999** | **#FF99CC** | **#FF99FF** |
| **#FFCC00** | **#FFCC33** | **#FFCC66** | **#FFCC99** | **#FFCCCC** | **#FFCCFF** |
| **#FFFF00** | **#FFFF33** | **#FFFF66** | **#FFFF99** | **#FFFFCC** | **#FFFFFF** |

# HTML - Background

HTML *background* is the HTML attribute used to place pictures in the background of HTML elements. Like the *bgcolor* attribute, *background* is now deprecated and its use has been replaced by the use of CSS (see [CSS Background](http://www.tizag.com/cssT/background.php)). However, this lesson does cover some important aspects of background elements that do apply to both the deprecated HTML *background* and the new CSS backgrounds as well.

**[Advertise on Tizag.com](http://www.tizag.com/about/advertise.php)**

When we think of an HTML background, we generally have only two options: a solid background color or a background image. The backgrounds of both the background color and image grow or shrink dynamically with the growth and shrinkage of the HTML element they are contained in.

## HTML Background Image Code:

<table height="100" width="150"

background="http://www.tizag.com/pics/htmlT/background.jpg" >

<tr><td>This table has a background image</td></tr>

</table>

## Background Image:

|  |
| --- |
| This table has a background image |

# HTML - Background Repeat

In the first example, we specified predetermined *height* and *width* attributes that matched the dimensions of the image we used in the background of the table element. Everything looks great. But we will run into problems if we add more content to the table itself and this element's height increases in size. The image begins to repeat itself to fill in the extended regions, which may look awful.

## HTML Repeating Background Image:

<table height="200" width="150"

background="http://www.tizag.com/pics/htmlT/background.jpg" >

<tr><td>This table has a background image</td></tr>

</table>

## Repeating Background:

|  |
| --- |
| This table has a background image |

More often than not, this behavior causes more pain than pleasure, and as web designers, we have two options. We can either live with it (until learning how to squelch it with CSS code) or, use this behavior to our advantage by thinking outside the box.

# HTML - Transparent Background Images

In the [HTML Images](http://www.tizag.com/htmlT/images.php" \o "HTML Images) lesson, we mentioned that .gif and .png image types can be saved with transparencies and incorporated into HTML pages. By doing so, we can create HTML elements with a semi-transparent background colors, that can be used to create overlay effects for our web pages.

Transparency can be added to any image via photo editing software, and if you don't currently have such software installed, feel free to download our transparent image from this example via this link: [Transparent Backgrounds](http://www.tizag.com/files/html/htmltransparentbackground.zip" \o "HTML Transparent Background Images)

Let's now place another background image inside of a table as we did in the previous example, but this time, let's use an image that's a little bit better-suited for a background. This one has a small amount of transparency.

## Transparent Background Images:

<table height="163" width="480" background="http://www.tizag.com/files/html/htmltransparentbackground.png" cellspacing="0" cellpadding="0">

<tr>

<td>

</td>

</tr>

</table>

## Transparent Background Example:

|  |
| --- |
|  |

As the code suggests, we've created a semi-transparent (.png) file to use as the background of the table. Doing so allows the color from this web page to radiate through the image, tinting it blue.

# HTML - Background Repeat Revisited

Now that we are more familiar with transparent HTML backgrounds, we can take our previous example a step further and create a very nice effect called a gradient overlay. To do this, we're going to take advantage of the repeating behavior of HTML background images and create a transparent gradient image file that is the same width as the target image (480 pixels), but only 1 pixel tall.

If you do not wish to create your own transparent gradient then, download ours! - [HTML Transparent Background Images](http://www.tizag.com/files/html/htmltransparentbackground.zip" \o "HTML Transparent Background Images)

## Gradient Overlay Code:

<table height="163" width="480" background="http://www.tizag.com/files/html/htmltransparentbackground.png" cellspacing="0" cellpadding="0">

<tr>

<td>

<table height="163"width="480" background="http://www.tizag.com/files/html/htmltransparentgradient.png">

<tr>

<td></td>

</tr>

</table>

</td>

</tr>

</table>

## Gradient Overlay Example:

|  |  |
| --- | --- |
| |  | | --- | |  | |

The gradient image we have provided is 480 pixels wide and only 1 pixel tall. We've done this for two reasons. Although the image is only 1 pixel tall, the web browser continues to repeat this image indefinitely, or for the entire height specified using the *height* attribute (163 pixels in this case). Additionally, reducing the size of the image to 1 pixel in height drastically reduces the file size and will tremendously improve page performance.

# HTML - Web Forms

HTML web forms are a composition of buttons, checkboxes, and text input fields embedded inside of HTML documents with one goal in mind: to capture user input. By doing things such as providing fields for user data such as names, phone number, and email addresses, web forms give users the opportunity to interact directly with a webpage.

**[Advertise on Tizag.com](http://www.tizag.com/about/advertise.php)**

HTML forms are placed on a web page using the <form> tag. This tag should encapsulate a series of other form elements, identifying them as a single cohesive web form.

## HTML Form Element:

<formg name="myWebForm" action="myServerSideScript.php" method="post">

<input type="checkbox" /> Checkbox 1<br />

<input type="text" /> Text Field 1<br />

<input type="submit" value="SUBMIT" />

</form>

## HTML Web Form:

Top of Form

Checkbox 1  
Text Field 1

Bottom of Form

HTML form elements rely on *action* and *method* attributes to identify where to send the form data for processing (action) and how to process the data (method). In the code above, we've inserted some make-believe values to represent what a typical HTML form might look like behind the scenes.

Unfortunately, HTML alone is unable to process form data. A scripting language such as PHP, PERL, and/or JavaScript must be used with HTML forms to process data captured by HTML form elements. A complete form-processing example using PHP can be found here: [PHP Form Processing Example](http://www.tizag.com/phpT/examples/formex.php/" \o "PHP Form Processing Example).

For the purpose of following along, we can also adjust the *action* property slightly to have the form launch an email client instead of processing a make-believe server-side script. This will provide us with some form interactivity for us as we learn more about HTML forms.

## HTML Email Form Element:

<formg name="myWebForm" action="mailto:youremail@email.com" method="post">

<input type="checkbox" /> Checkbox 1<br />

<input type="text" /> Text Field 1<br />

<input type="submit" value="SUBMIT" />

</form>

## HTML Email Form:

Top of Form

Checkbox 1  
Text Field 1

Bottom of Form

Now when the **SUBMIT** button is clicked, the user should see their default email client launch.

HTML forms provide user interaction between visitors and the website while simultaneously collecting priceless user data from your users. They are a vital tool for any webmaster, and these days, it is common place to see form elements embedded in every web page.

# HTML - Input Element(s)

HTML input elements are form elements such as text fields, checkboxes, and buttons. The name comes from the <input> tag, which is the mark-up that identifies web form components. The <input> tag relies upon a few attributes to classify and name each form item, providing the web developer with a means to manipulate each element individually.

**[Advertise on Tizag.com](http://www.tizag.com/about/advertise.php)**

The *type* attribute determines what kind of input element to render to the screen. Options here include: *text, checkbox, radio, button, submit, reset, password,* and *hidden* form elements. Each has its own unique functionality and customizable presentation.

## HTML Input Element Code:

<formg name="myWebForm" action="mailto:youremail@email.com" method="post">

Check Me: <input type="checkbox" /><br />

Name: <input type="text" /><br />

Yes: <input type="radio" /> No: <input type="radio" /><br />

<input type="submit" value="SUBMIT" />

<input type="reset" value="RESET" />

</form>

## HTML Input Elements:

Top of Form

Check Me:   
Name:   
Yes: No: 

Bottom of Form

# HTML - Web Forms: Value Attribute

The *value* attribute plays a different role depending on the *type* of the input field. For example, when used with an HTML button, the *value* attribute defines the text inside of the button. When used with a text field, the *value* attribute populates the field with a default value.

## HTML Input Element Code:

<formg name="myWebForm" action="mailto:youremail@email.com" method="post">

Check Me: <input type="checkbox" /><br />

Name: <input type="text" value="David" /><br />

Yes: <input type="radio" /> No: <input type="radio" /><br />

<input type="submit" value="Send" />

<input type="reset" value="Clear" />

</form>

## HTML Input Elements:

Top of Form

Check Me:   
Name:   
Yes: No: 

Bottom of Form

# HTML - Web Forms: Name and ID Attributes

Setting the *name* and *id* attributes inside of form elements is a good habit. The element name and/or id will later serve as the link between your HTML form and any server-side script that you may deploy later on to process that data. Perhaps the best approach is to use both attributes in your code, since varying scripting languages demand one identifying attribute over the other.

## HTML Input Element Code:

<formg name="myWebForm" action="mailto:youremail@email.com" method="post">

Check Me: <input name="" id="" type="checkbox" /><br />

Name: <input name="userName" id="userName" type="text" /><br />

Yes: <input name="radioItem" id="radioItem" type="radio" /> No: <input name="radioItem" id="radioItem" type="radio" /><br />

<input name="submitForm" id="submitForm" type="submit" value="SUBMIT" />

<input name="resetForm" id="resetForm" type="reset" value="RESET" />

</form>

## HTML Input Elements:

Top of Form

Check Me:   
Name:   
Yes: No: 

Bottom of Form

# HTML - Text Fields

Text fields offer a small rectangular box that's always ready to receive information from viewers. Users will notice that when they click these fields, the cursor will change from the typical arrow to a pipe character ( | ), allowing for text entries to be typed inside each input field.

**[Advertise on Tizag.com](http://www.tizag.com/about/advertise.php)**

A text field is placed on a web page using the <input> tag, with the *type* attribute set with a value of "text".

## HTML Text Field Code:

<form name="myWebForm" action="mailto:youremail@email.com" method="post">

First: <input title="Please Enter Your First Name" id="first" name="first" type="text" /> Last: <input title="Please Enter Your Last Name" id="last" name="last" type="text" />

<input type="submit" value="SUBMIT" />

</form>

## HTML Text Fields:

Top of Form

First: Last: 

Bottom of Form

Text fields are designed to capture single words or phrases from the user. That information may then be processed through some kind of client/server side script (PHP, PERL, JavaScript). If you do plan on processing the data, be sure to include the *name* and *id* attributes. A descriptive *title* is also a great visual aid for providing a tool-tip display for your web elements.

# HTML - Text Fields: Size Attribute

To modify the visual presentation of a text field, one needs to pass an integer value to the *size* attribute. The value represents how many characters a text field can display within the text field window.

As the web designer, it is your job to analyze and predict the average length of characters that will be entered into each field by your users. First and last names may generally vary from 8-24 characters in length, while a typical email address may range from 12-36 digits.

## HTML Text Field Size:

<form name="myWebForm" action="mailto:youremail@email.com" method="post">

First: <input title="Please Enter Your First Name" id="first" name="first" type="text" size="12" /><br />

Last: <input title="Please Enter Your Last Name" id="last" name="last" type="text" size="18" /><br />

<input type="submit" value="SUBMIT" />

</form>

## HTML Text Field Size:

Top of Form

First:   
Last: 

Bottom of Form

If the user happens to enter more digits than the size attribute value, these characters will not be discarded; it just means that the user will not be able to see all of their input at once. Instead, they will be forced to scroll to the beginning and end of the input element, which tends to discourage user interaction.

# HTML - Text Fields: Maxlength Attribute

*Maxlength* is an optional attribute that accepts an integer value. It allows the developer to restrict the number of characters a user can type in a specific text field.

## HTML Text Field Maxlength:

<form name="myWebForm" action="mailto:youremail@email.com" method="post">

First: <input title="Please Enter Your First Name" id="first" name="first" type="text" size="12" maxlength="3" value="David" /><br />

Last: <input title="Please Enter Your Last Name" id="last" name="last" type="text" size="18" maxlength="3" value="Smith" /><br />

<input type="submit" value="SUBMIT" />

</form>

## HTML Text Field Maxlength:

Top of Form

First:   
Last: 

Bottom of Form

We've also called upon the *value* attribute to place some text inside the text fields for you!

# HTML - Password Fields

HTML password fields are designed to capture user input, but disguise each character with an asterisk (\*) instead of displaying the entered digits. They offer a user on-screen privacy while he or she is entering a password.

**[Advertise on Tizag.com](http://www.tizag.com/about/advertise.php)**

Password fields are placed on a website using the <input> tag and specify a value of "password" for the *type* attribute.

## HTML Password Field Code:

<form name="myWebForm" action="mailto:youremail@email.com" method="post">

Password: <input type="password" title="Please Enter Your Password" size="8" /><br />

<input type="submit" value="SUBMIT" />

</form>

## HTML Password Fields:

Top of Form

Password: 

Bottom of Form

# HTML - Password Fields: Attributes

Password form fields may be customized using the same attribute as outlined in the [HTML Text Fields](http://www.tizag.com/htmlT/htmltextfields.php" \o "HTML Text Fields) lesson.

## HTML Password Input Field Code:

<form name="myWebForm" action="mailto:youremail@email.com" method="post">

First: <input title="Please Enter Your First Name" id="first" name="first" type="text" size="12" maxlength="12" /> Last: <input title="Please Enter Your Last Name" id="last" name="last" type="text" size="18" maxlength="24" /><br />

Password: <input type="password" title="Please Enter Your Password" size="8" maxlength="8" /><br />

<input type="submit" value="SUBMIT" />

</form>

## HTML Password Input Fields:

Top of Form

First: Last:   
Password: 

Bottom of Form

Password fields offer a very thin layer of security by visually concealing passwords; they offer no security whatsoever against maintaining the integrity of the password data. From data is processed in plain text and can be readily sniffed by a hacker, unless ***HTTPS*** is used to encrypt the data.

# HTML - Reset Buttons

A reset button allows users to basically clear their web form. It wipes values from all fields by "resetting" the form to its default appearance.

**[Advertise on Tizag.com](http://www.tizag.com/about/advertise.php)**

Set the *type* attribute of the <input> tag to "reset" to incorporate a reset button into a web form.

## HTML Reset Button Code:

<input type="reset" value="Reset" />

<input type="reset" value="Start Over" />

## Two HTML Reset Buttons:

Placing a reset button inside of a form tag automatically associates the reset button with each form element and delivers a useful feature for your viewers.

## HTML Code:

<form action="myphp.php" method="post"> <input type="text" size="12" maxlength="12" />

<input type="text" size="24" maxlength="24" />

<input type="reset" value="Reset" /> </form>

## Reset Forms:

Top of Form

  


Bottom of Form

Fill out some information in the field boxes and press **reset** to experience a reset form!

# HTML - Submit Buttons

Submit buttons send form data to a back-end process or application. The back-end process then verifies and precesses the data, eventually passing the information into some database application.

**[Advertise on Tizag.com](http://www.tizag.com/about/advertise.php)**

Set the *type* attribute of the <input> tag to "submit" in order to place a submit button on a web page.

## HTML Submit Buttons:

<input type="submit" value="Submit" />

<br /> <input type="submit" value="Send" />

<br /> <input type="submit" value="Submit Form" /><br />

## Three HTML Submit Buttons:

Notice that in the above example, we also changed what was written on our button using the *value* attribute. This can be changed to any value you wish.

# HTML - Form Submission - Action

Submission buttons send form data to whatever action has been designated by the *action* attribute of the encapsulating <form> element. We learned about the *action* attribute in our [HTML Forms](http://www.tizag.com/htmlT/forms.php) lesson.

If you've been following along, we've also been using the deprecated *mailto* action to send form data to our default email client. This will allow us to get a sense of how form values are transferred to an action.

## HTML Code:

<form method="post" action="mailto:youremail@youremail.com" >

First:<input type="text" name="First" size="12" maxlength="12" />

Last:<input type="text" name="Last" size="24" maxlength="24" />

<input type="submit" value="Send Email" />

</form>

## Form Action:

Top of Form

First:  
Last:

Bottom of Form

Fill out the above form, and as your mail program opens, you can change the email address to a personal address and then send the results using the form.

# HTML - Checkbox Forms

Setting the *type* attribute of an <input> tag to *checkbox* places a checkbox element onto the web page.

**[Advertise on Tizag.com](http://www.tizag.com/about/advertise.php)**

Deploy checkbox elements in a situation when the user must check all boxes that apply (or none). A scripting language such as PHP will easily handle this form element, returning all elements the user has checked (check out our [PHP Form Example](http://www.tizag.com/phpT/examples/formex.php/" \o "PHP Form Example).)

## HTML Checkbox Code:

<form name="myWebForm" action="mailto:youremail@email.com" method="post">

<p>Please select every sport that you play.</p>

Soccer: <input type="checkbox" name="sports" value="soccer" /><br />

Football: <input type="checkbox" name="sports" value="football" /><br />

Baseball: <input type="checkbox" name="sports" value="baseball" /><br />

Basketball: <input type="checkbox" name="sports" value="basketball" />

</form>

## HTML Checkbox Form:

Please select every sport that you play.  
Soccer:   
Football:   
Baseball:   
Basketball: 

Checkboxes are used for instances where a user may wish to select multiple options, such as in the instance of a "check all that apply" question.

# HTML Checkboxes Selected

A checkbox element can be placed onto a web page in a pre-checked fashion by setting the *checked* attribute with a "yes" value. By doing so, this element will now default to a checked status each time the HTML page is loaded.

## HTML Checkbox Selected Code:

<form name="myWebForm" action="mailto:youremail@email.com" method="post">

<p>Please select every sport that you play.</p> Soccer: <input type="checkbox" checked="yes" name="sports" value="soccer" /> <br /> Football: <input type="checkbox" name="sports" value="football" /> <br /> Baseball: <input type="checkbox" name="sports" value="baseball" /> <br /> Basketball: <input type="checkbox" checked="yes" name="sports" value="basketball" />

</form>

## HTML Pre-Selected Checkboxes:

Please select every sport that you play.  
Soccer:   
Football:   
Baseball:   
Basketball: 

# HTML - Radio Forms

Radio form elements resemble the "Scan-Tron" sheets you may have used when you were in school to take a test. They basically allow the user to "bubble" in their choice and limit each question to only one selection per radio group.

**[Advertise on Tizag.com](http://www.tizag.com/about/advertise.php)**

Place a radio element on to your web page by setting the *type* attribute of the <input> tag to "radio".

## HTML Radio Input Code:

<form name="myWebForm" action="mailto:youremail@email.com" method="post">

<h4>Please select your favorite food category.</h4>

<input type="radio" name="food" /> : Italian<br />

<input type="radio" name="food" /> : Greek<br />

<input type="radio" name="food" /> : Chinese<br />

</form>

## HTML Radio Fields:

#### Please select your favorite food category.

: Italian  
: Greek  
: Chinese

By naming each field similarly with a type of cuisine, we have created a relation, or a "grouping," of radio elements. This is how we link each element together and assure that the user is able to select only one answer.

Let's now take a look at how we can group together different sets of radio elements and simulate capturing two pieces of user data: gender and favorite food.

## HTML Multiple Radios:

<form name="myWebForm" action="mailto:youremail@email.com" method="post">

<h4>Please select your favorite food category.</h4>

<input type="radio" name="food" /> : Italian<br />

<input type="radio" name="food" /> : Greek<br />

<input type="radio" name="food" /> : Chinese<br />

<h4>Please select your gender.</h4>

<input type="radio" name="gender" /> : Male<br />

<input type="radio" name="gender" /> : Female<br />

Lt44/form>

## HTML Multiple Radio Fields:

#### Please select your favorite food category.

: Italian  
: Greek  
: Chinese

#### Please select your gender.

: Male  
: Female

Words/values applied to the *value* attribute is the value or 'answer' passed to any server-side script language we may have in place to record the results.

# HTML - Radio: The Checked Attribute

By using the *checked* attribute, we adjust the form to load with a value already checked as the default setting.

## HTML Code:

<form name="myWebForm" action="mailto:youremail@email.com" method="post">

<h4>Please select your favorite food category.</h4>

<input type="radio" name="food" checked="yes" /> : Italian<br />

<input type="radio" name="food" /> : Greek<br />

<input type="radio" name="food" /> : Chinese<br />

</form>

## Default Italian:

Top of Form

: Italian  
: Greek  
: Chinese

Bottom of Form

Using either/or logic, radios provide a very efficient way to capture very specific data from visitors. Remember to use *radio* elements only when you'd like the viewer to select only a single value, just as you might expect to see when taking a multiple-choice test in school.

# HTML - Select Fields

HTML *select* fields provide essentially the same functionality as [HTML Checkbox Fields](http://www.tizag.com/htmlT/htmlselect.php). They allow the user to select one or more values from a pre-determined series of options.

**[Advertise on Tizag.com](http://www.tizag.com/about/advertise.php)**

Incorporating a select field into a web page is done using the <select> tag. List values are then added to the field using the <option> tag, similar to how list items <li> are added to ordered list elements (<ol>).

## HTML Drop Down List:

<select name="selectionField">

<option value="CA" >California -- CA </option>

<option value="CO" >Colorado -- CO</option>

<option value="CN" >Connecticut -- CN</option>

</select>

## HTML Drop Down List:

By default, select fields, popularly called drop down lists, only allow the user to choose a single value. This behavior and appearance may be changed by adjusting the *multiple* and *size* attributes as demonstrated below.

## HTML Selection Field Code:

<select size="3" name="selectionField" multiple="yes" >

<option value="CA" >California -- CA </option>

<option value="CO" >Colorado -- CO</option>

<option value="CN" >Connecticut -- CN</option>

</select>

## HTML Selection Element:

With the above settings, the user is now able to select multiple values by pressing and holding the **Control** (ctrl) key and clicking each value.

# HTML - Disabling Selection Fields

Disabling a selection field is achieved by setting the *disabled* attribute to "yes". But before doing that, you should set at least one of the values to be selected. Doing so renders a read-only selection field on the page that can inform your users of their selections without allowing them to alter the selection.

## HTML Read-Only Selection Field:

<select size="3" name="selectionField" multiple="yes" disabled="yes">

<option value="CA" >California -- CA </option>

<option selected value="CO" >Colorado -- CO</option>

<option value="CN" >Connecticut -- CN</option>

</select>

## HTML Read-Only Selections:

# HTML - Hidden Field

Hidden fields allow a coder to pass values to form elements in a subtle manner. An experienced web developer will utilize these fields to pass temporary, or session-based data, from one form to another or to store information that has already been entered in by the user.

**[Advertise on Tizag.com](http://www.tizag.com/about/advertise.php)**

Place a hidden input field into your web forms using the <input> tag and set the *type* attribute to "hidden". This field can be customized using any of the attributes discussed in the [HTML Input](http://www.tizag.com/htmlT/htmlinput.php" \o "HTML Input) and [HTML Text Fields](http://www.tizag.com/htmlT/htmltextfields.php" \o "HTML Text Fields) lessons.

## HTML Hidden Input Field:

<form name="myWebForm" action="mailto:youremail@email.com" method="post">

First: <input title="Please Enter Your First Name" id="first" name="first" type="text" size="12" maxlength="12" /> Last: <input title="Please Enter Your Last Name" id="last" name="last" type="text" size="18" maxlength="24" /><br />

Password: <input type="password" title="Please Enter Your Password" size="8" maxlength="8" /><br /><br />

<input type="hidden" name="orderNumber" id="orderNumber" value="0024" /><br />

<input type="submit" value="SUBMIT" />

<input type="reset" value="RESET" />

</form>

## HTML Hidden Fields:

Top of Form

First: Last:   
Password: 

Bottom of Form

It is important to note that HTML hidden fields do not offer any data security. Like all HTML form elements, the data is processed in plain text and is readily accessible by any novice hacker.

# HTML - Upload Forms

Upload fields provide the interface that allows users to select a local file and upload it to the web server. An upload field renders as two parts -- an empty text field and a **Browse** button that opens up a local window explorer on the user's computer. This allows them to quickly browse to the local file and automatically fills in the file path inside of the text field.

**[Advertise on Tizag.com](http://www.tizag.com/about/advertise.php)**

Setting the *type* attribute of the <input> to "file" places the upload element on a web page.

## HTML Upload Field Code:

<form name="myWebForm" action="mailto:youremail@email.com" method="post">

<input type="file" name="uploadField" />

</form>

## HTML Upload Field:

Top of Form

Bottom of Form

# HTML - Max File Size Field

File transferring across the internet is a complicated process and should include many layers of security. HTML alone cannot ensure safe and secure file transferring, but it can offer a first line of defense. Using a MAX\_FILE\_SIZE hidden field can limit the size of files that are transferred.

Placing a restraint on an HTML upload field is accomplished by using a hidden input field with the *name* attribute set to **MAX\_FILE\_SIZE**.

## HTML MAX\_FILE\_SIZE Code:

<form name="myWebForm" action="mailto:youremail@email.com" method="post">

<input type="hidden" name="MAX\_FILE\_SIZE" value="500" />

<input type="file" name="uploadField" />

</form>

## HTML MAX\_FILE\_SIZE:

Top of Form

Bottom of Form

The ***value*** attribute specifies the maximum allowable kilobytes (KB) for any file selected by the user.

# HTML - Textareas

An HTML textarea is an oversized [Text Field](http://www.tizag.com/htmlT/htmltextfields.php) capable of capturing "blurb" type information from a user. If you've ever posted on a forum or left feedback on your favorite blog, you probably do so using an HTML textarea.

**[Advertise on Tizag.com](http://www.tizag.com/about/advertise.php)**

Embed textareas in HTML documents using the <textarea> tag. Any text placed between the opening and closing textarea tags will be rendered inside the textarea element as the "default" text. This makes for a great way to give users an example or description of how to go about filling out the text area field. Something like, "Please limit your response to 100 characters," would be an ideal description.

## HTML Textarea Code:

<textarea name="myTextArea"cols="20" rows="10">Please limit your response to 100 characters.</textarea><br />

<textarea name="myTextArea" cols="40" rows="2">Please limit your response to 200 characters.</textarea><br />

<textarea name="myTextArea" cols="45" rows="5">Please limit your response to 500 characters.</textarea><br />

## HTML Textarea Form Element:

  
  


As you may have noticed, the attributes *cols* (columns) and *rows* control the rendered size of the textarea. These constraints only impact how the textarea is rendered visually, and in no way do they limit the maximum number of characters a user can place inside the textarea. In fact, if you fill up the fields above with text, the fields will just continue to grow as you type and you will be able to scroll up and down as you please. Limits must be set with JavaScript and/or a server-side scripting language such as PHP.

# HTML - Textarea Wrap

The *wrap* attribute refers to how the user input reacts when it reaches the end of each row in the text field. Wrapping can be defined using one of three values:

* soft
* hard
* off

"Soft" forces the words to wrap once inside the textarea but once the form is submitted, the words will no longer appear as such, and line breaks and spacing are not maintained.

"Hard" wraps the words inside the text box and places line breaks at the end of each line so that when the form is submitted the text will transfer as it appears in the field, including line breaks and spacing.

"Off" sets a textarea to ignore all wrapping and places the text into one ongoing line.

## HTML Text Area Word Wrap Code:

<textarea cols="20" rows="5" wrap="hard">

As you can see many times word wrapping is often the desired look for your textareas. Since it makes everything nice and easy to read and preserves line breaks.

</textarea>

## HTML Text Area Word Wrap:



Here's a textarea with no word wrapping at all!

## HTML Text Area No Word Wrap:

<textarea cols="20" rows="5" wrap="off">

As you can see many times word wrapping is often the desired look for your textareas. Since it makes everything nice and easy to read. </textarea>

## HTML Text Area No Word Wrap:



# HTML - Text Areas: Readonly

Setting a "yes" or "no" value for the *readonly* attribute determines whether or not a viewer has permission to manipulate the text inside the text field.

## HTML Readonly Attribute:

<textarea cols="20" rows="5" wrap="hard" readonly="yes">

As you can see many times word wrapping is often the desired look for your text areas. Since it makes everything nice and easy to read.

</textarea>

## HTML Read Only Text Areas:



This read-only behavior allows a web surfer to see and highlight the text inside the element, but he or she cannot alter it in any way. When highlighted, the user may also Copy (Ctrl + C on a PC, Ctrl-Click on a Mac) the text to local clipboard and paste it anywhere he/she pleases.

# HTML - Text Areas: Disabled

Disabling the textarea altogether prevents the surfer from highlighting, copying, or modifying the field in any way. To accomplish this, set the *disabled* property to "yes".

## HTML Code:

<textarea cols="20" rows="5" wrap="hard" disabled="yes">

As you can see many times word wrapping is often the desired look for your text areas. Since it makes everything nice and easy to read.  
</textarea>

## Disabled Textareas:



Keep in mind that just because users are unable to copy from the screen directly doesn't prevent them from taking a screen capture or extracting the data from the source code. Disabling the textarea offers no security whatsoever.

# HTML - Body

As we mentioned, the <body> tag serves as the element containing all the content for the website. Tables, lists, forms, paragraphs, and everything else must be placed within the body element to ensure each element is displayed on your site.

**[Advertise on Tizag.com](http://www.tizag.com/about/advertise.php)**

# HTML - Body Margins

**Unique Attributes**

* **leftmargin** - Sets a left hand margin for your body element.
* **topmargin** - Sets a margin along the top of your body element.

A unique set of margin attributes are available to the body tag. These attributes work much like those of a word processing program, allowing you set pixel value margins for the left, right, top, or bottom of your website. Setting these attributes means that all the content you place within your body tags will honor the preset margin.

## HTML Code:

<body topmargin="50">

<body leftmargin="50">

## Margin Example-1:

[Top Margin](http://www.tizag.com/pics/htmlT/topmargin.html" \t "_blank)

## Margin Example-2:

[Left Margin](http://www.tizag.com/pics/htmlT/leftmargin.html" \t "_blank)

# HTML - Base Text

The text attribute sets the text color of all text contained within the body tags. Think of it as a means to set the color of your text, unless otherwise noted. Basically, you use these tags to set a base color scheme, which you can later modify through additional tags inside of the body.

## HTML Code:

<body text="red" >

**or**

<body text="rgb(255,0,0)" >

# HTML - Base Links

Along the same lines, we may also specify base colors for visited or unvisited links. This method has deprecated, and we recommend that you use Cascading Style Sheets (CSS) instead.

## HTML Code:

<body link="white" vlink="black" >

**or**

<body link="rgb(255,255,255)" vlink="rgb(0,0,0)" >

Setting a baselink is a great way to ensure your viewers will not receive the annoying error message that occurs with broken links.

# HTML - Music Codes

Inserting music onto a web page is relatively easy these days. In the past, multiple tags had to be used because browsers did not have a uniform standard for embedded media files. However, we're happy to announce that this is a problem of the past, and you will now have a much easier time than webmasters of the past.

**[Advertise on Tizag.com](http://www.tizag.com/about/advertise.php)**

# HTML - Embed

Music is inserted onto a web page with the use of the *embed* tag. There are other ways to link to music, but embed is now considered the standard for inserting media. Below is an minimalist example of the embed tag using the *src* attribute to define the media file's location.

## HTML Embed Tag Code:

<embed src="beethoven.mid" />

<p>Above is an embedded media player. To stop the music press stop/pause.</p>

Depending on what kind of media software you or your visitor has installed, the above example will appear slightly different. To make your embedded player display properly, change the attributes associated with display.

# HTML - Embed Attributes - Related to Display

To customize the appearance of the embedded media player, be sure to set the following attributes.

* width - The width of the media player.
* height - The height of the media player.
* hidden - Determines if the media player is visible. If this value is set to "true", the media player will not be displayed. We recommend using this attribute only if you know that your visitors will not want the option to stop the music that is playing on your web page (The values are "true" or "false").

## HTML Code:

<embed src="beethoven.mid" width="360" height="165" />

## Embedded Music:

Usually, you do not want to mess with the width and height of the media player, as it can cause the media player to look rather distorted.

# HTML - Embed Attributes - Related to Functionality

To customize the functionality of the embedded media player, be sure to set the following attributes.

* autostart - Allows media player to start automaticcaly (values are "true" and "false")
* loop - Sets whether or not the media file will repeat (values are "true" and "false")
* volume - Sets the volume of the media file (values range from "0" to "100")

## HTML Code:

<embed src="beethoven.mid" autostart="false" loop="false"

volume="60" />

## Customize Your Code:

[Bookmark and Share](http://www.addthis.com/bookmark.php)

# HTML - Video Codes

Video files, including YouTube videos, are embedded into an HTML document using the <embed> tag. The *src* attribute defines what video file to embed into the page. The <embed> tag does not require a closing tag. Here is a look at the <embed> tag with a global URL. Notice that its controls, including Play, Stop, Pause, and volume, are already included.

**[Advertise on Tizag.com](http://www.tizag.com/about/advertise.php)**

## HTML Code:

<embed src="http://www.tizag.com/files/html/htmlexample.mpeg"

autostart="false" />

## Mpeg Movie:

You may start and stop your movie files by either pressing the buttons at the bottom of the object or by single-clicking on the object itself. The movie can be restarted by double-clicking your mouse.

# HTML - Video Media Types

Flash (.swf) and MOV (.mov) file types are also supported by the <embed> tag.

* **.swf** - Macromedia's Flash file types - very high compression, great for the web!.
* **.wmv** - Microsoft's Window's Media Video file types - good quality, variable compression.
* **.mov** - Apple's Quick Time Movie format - good quality, variable compression.
* **.mpeg** - the accepted standard for web movie files created by the Moving Pictures Expert Group - good quality, variable compression.

The list above outlines some of the most common "internet-ready" video files. Macromedia's .swf and .mpeg formats may be the best options for use with the web because the high compression rate of these file types reduces file size and expedites the download/buffering periods for your page visitors.

You may also simply place the URL of your media files into the *href* attribute of an anchor tag, much like the concept of "thumbnailing" images.

## HTML Code:

<a href="http://www.tizag.com/pics/flash/motiontween1easy.swf">

motiontween1easy.swf</a>

## Flash Media:

[motiontween1easy.swf](http://www.tizag.com/pics/flash/motiontween1easy.swf" \o "Flash Media" \t "_blank)

# HTML - YouTube Videos

YouTube videos can be included in HTML documents, and Google offers the code to do so right on the same page as the video itself!

The code offered by YouTube includes a small handful of parameters that help customize the embedded video object, and if you dive deep enough into the code, you will be able to identify the <embed> element and see the *src* attribute pointing to the URL of the media file.

## YouTube Video Code:

<object width="425" height="344"><param name="movie" value="http://www.youtube.com/v/opVb89Cmrtkamp;hl=enamp;fs=1">

</param><param name="allowFullScreen" value="true">

</param><param name="allowscriptaccess" value="always"></param>

<embed src="http://www.youtube.com/v/opVb89Cmrtk&hl=en&fs=1" type="application/x-shockwave-flash" allowscriptaccess="always" allowfullscreen="true" width="425" height="344">

</embed>

</object>

## Embed YouTube Video:

# HTML - Embed Attributes

To customize the functionality of the embedded media player, be sure to set any of the following attributes.

* **autostart** - Controls the media's ability to start without prompting (values are "true" or "false")
* **hidden** - Controls whether or not the play/stop/pause embedded object is hidden or not (values are "true" or "false"; hide your embeded media if you just want background noise)
* **loop** - Controls the ability of the media to continuously play (values are "true" and "false")
* **playcount** - Sets a playcount which means the media will repeat itself *x* number of times, instead of continuously as with the loop attribute above (a playcount of "2" will repeat the video twice)
* **volumn** - Sets a numeric value for the loudness of your media (values are "0" through "100")

# HTML - Meta Tags and Meta Data

Meta tags are used to supply information for search engines that will not be seen by the web surfer. These invisible units provide a flag for search engines to investigate and will then present that data to any potential users that stumble across your site through a search engine.

**[Advertise on Tizag.com](http://www.tizag.com/about/advertise.php)**

In the past, meta tags were *the* primary means for your site to be recognized by web spiders, but webmasters abused meta tags to improve their rankings in search engines. As a result, search engines have since modified their approach to keep results accurate. They now rely less on meta tags. Nevertheless, you should still include meta for those search bots that still do recognize them.

# HTML - Meta Tag Description

Search engines are the compasses of the web and help users navigate from site to site. Chances are, if you've used a search engine, you've probably seen the *description* meta tag in action.

Meta elements must be placed inside of the <head> element in order for them to be recognizable by web crawlers and bots. The <meta> tag generally requires the *name* and *content* attributes to be working together to present your web page in a good light.

## HTML Code:

<head>

<meta name="description" content="Tizag contains webmaster tutorials." />

</head>

The *description* meta element allows the developer to summarize the content that can be found on the page and is often the first chance you'll have to attract visitors. These brief narratives and hooks are often the only opportunity you'll have to generate a lasting first impression.

# HTML - Keyword Meta Tags

Keywords and/or phrases may be placed inside the *keyword* meta element. You should specify the most popular search terms you believe someone would use to reach your website. A few years back, you could spam this meta tag with any and every keyword possible to gain ranking on search engines. Now, however, repeated words, or words that do not pertain to the content of the site, will not benefit your search engine rankings.

## HTML Code:

<head>

<meta name="keywords" content="keyword, key keywords, etc" />

</head>

Separate each phrase/word with a comma to create large lists. An example of the keywords meta tag for Tizag.com would go something like this:

## HTML Code:

<head>

<meta name="keywords" content="HTML, XHTML, CSS, tutorials, tizag" />

</head>

Keep in mind that driving traffic and having your site listed high in the search engine rankings is not as easy as placing keywords inside your meta element. The phrase "Search Engine Optimization (SEO)" was coined to describe the rigorous process involved in achieving rankings in search engines. While meta tags do play a small role in this process, they are by no means a one-stop shop for your SEO needs.

# HTML - Refresh and Redirect Meta

Later down the road, you may need to redirect traffic to another domain. A common reason might be that you have just purchased a better domain name and would like to retain your old visitors, yet still use your new domain. With the *refresh* meta tag, you will be able to redirect visitors to the website of your choice or simply refresh your own page to update dynamic content automatically.

For the *refresh* meta tag, the *content* attribute accepts two arguments separated by a semicolon (;). The first argument specifies the number of seconds between refreshes or redirection and the 2nd argument is a URL of where the browser will relocate.

## HTML Redirect Meta Tag:

<head>

<meta http-equiv="refresh" content="10; url=http://www.tizag.com" />

</head>

The above code refreshes Tizag's home page every 10 seconds. A quick refresh may be necessary for news, stocks, or any other time-sensitive information. The most common use for this type of meta tag, however, is redirection. To redirect a viewer automatically, just change the URL to the new site, like shown below. This code will send your visitors to espn.com after landing at your site for five seconds.

## HTML Page Refresh Meta Tag:

<head>

<meta http-equiv="refresh" content="5; url=http://www.espn.com" />

</head>

# HTML - Revised Meta

The *revised* meta tag records when the last update was done to the site.

## HTML Code:

<head>

<meta name="revised" content="Happy New Year: 1/1/2003" />

</head>

Don't forget to get a little meta with your pages!

# HTML - Style Attribute

Understanding the HTML *style* attribute will provide you with a preview into the Cascading Style Sheet (CSS) world. In fact, the code we'll be using with *style* is indeed CSS code known as [Internal CSS](http://www.tizag.com/cssT/internal.php). CSS styling brings a whole new dimension to a website and offers endless customization of HTML elements and web page design.

**[Advertise on Tizag.com](http://www.tizag.com/about/advertise.php)**

When the *style* attribute was introduced into the HTML language along with CSS, a number of HTML attributes and tags became obsolete. Manipulation of the fonts and color of HTML elements is now accomplished through CSS styling, instead of stacking bulky formatting tags one inside the other.

## HTML Style: Inline CSS:

<p id="contentParagraph" style="color: #0900C4;">

Here we've changed the font color of this paragraph to blue.

</p>

## HTML Styling:

Here we've changed the font color of this paragraph to blue.

In the [HTML Font](http://www.tizag.com/htmlT/font.php" \o "HTML Font) lesson, we achieved similar results, but the code used to do so was cumbersome and inefficient.

# HTML - Styling

As we mentioned, the values passed to the *style* attribute are actually CSS code. This means that we can go ahead and pass a series of values at once, changing several properties in one go. Simply separate each CSS attribute with a semicolon (;).

## HTML Font Styling:

<p id="contentParagraph" style="font-family: Georgia ; font-size: 12pt; color: #0900C4;">

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

</p>

## HTML Font Styling:

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Inline CSS with the HTML *style* attribute offers a great way to improve the visual display of web elements and pages. With this new understanding of HTML and CSS, you're well on your way to mastering web design.

# HTML - Div Element(s)

The <div> tag is nothing more than a container unit that encapsulates other page elements and divides the HTML document into sections. Web developers use <div> elements to group together HTML elements and apply CSS styles to many elements at once. For instance, by wrapping a set of paragraph elements into a <div> element, the developer can take advantage of CSS styles and apply a font to all paragraphs at once by applying a font style to the <div> tag instead of coding the same style for each paragraph element.

**[Advertise on Tizag.com](http://www.tizag.com/about/advertise.php)**

Group together text elements within a <div> tag to slice up HTML documents.

## HTML Div Element Code:

<div id="myDiv" name="myDiv" title="Example Div Element">

<h5>Subtitle</h5>

<p>This paragraph would be your content paragraph...</p>

<p>Here's another content article right here.</p>

</div>

With these text elements now grouped together under a <div> element, we can alter the appearance of each underlying element collectively by applying a *style* attribute to the <div> tag.

## HTML Div Element Code:

<div id="myDiv" name="myDiv" title="Example Div Element" style="color: #0900C4; font: Helvetica 12pt;border: 1px solid black;">

<h5>Subtitle</h5>

<p>This paragraph would be your content paragraph...</p>

<p>Here's another content article right here.</p>

</div>

## HTML Div Element in Action:

|  |
| --- |
| Subtitle This paragraph would be your content paragraph...  Here's another content article right here. |

Elements housed within a <div> tag acquire any styles or properties applied to the master div element. Therefore the paragraph and heading elements should now be rendered *blue* in a *Helvetica* font. In addition, we've applied a border to the <div> element just to help visualize the grouping of elements together.

# HTML - Div inside of Div

Placing <div> elements inside of other <div> elements allows these elements to be further subdivided.

## HTML Div Subdivision:

<div id="myDiv" name="myDiv" title="Example Div Element" style="font-family: Helvetica; font-size: 12pt; border: 1px solid black;">

<div id="subDiv1" name="subDiv1" title="Subdivision Div Element" style="color: #FF0000; border: 1px dotted black;">

<h5>Section 1</h5>

<p>This paragraph would be your content paragraph...</p>

<p>Here's another content article right here.</p>

</div>

<br />

<div id="subDiv2" name="subDiv2" title="Subdivision Div Element" style="color: #FF00FF;border: 1px dashed black;">

<h5>Section 2</h5>

<p>This paragraph would be your content paragraph...</p>

<p>Here's another content article right here.</p>

</div>

</div>

## Divs Inside of Divs:

|  |  |
| --- | --- |
| |  | | --- | | Section 1 This paragraph would be your content paragraph...  Here's another content article right here. | |
| |  | | --- | | Section 2 This paragraph would be your content paragraph...  Here's another content article right here. | |

This concept is the foundation of which most web pages are now built. HTML documents that are properly divided and subdivided are easy to maintain and modify.

# HTML - Page Layouts and Templates

HTML layout is very basic. Not many options exist with the body tag alone. Tables, on the other hand, are the bread and butter of HTML layouts. Any element may be placed inside of a table, including tables themselves!

**[Advertise on Tizag.com](http://www.tizag.com/about/advertise.php)**

## HTML Code:

<table id="shell" bgcolor="black" border="1" height="200" width="300">

<tr>

<td>

<table id="inner" bgcolor="white" height="100" width="100">

<tr>

<td>Tables inside tables!</td>

</tr>

</table>

</td>

</tr>

</table>

## Tables inside tables:

|  |  |
| --- | --- |
| |  | | --- | | Tables inside tables! | |

The white table (identified as inner) exists inside of the (shell) table, the black one. A light bulb should be going off inside of your head as you explore how this system will allow for the creation of limitless layouts.

# HTML - Standard Layout

A fairly standard layout consists of a banner near the top, navigation, and your content or display box. These are the backbone to any great website.

## HTML Code:

<table cellspacing="1" cellpadding="0" border="0"

bgcolor="black" id="shell" height="250" width="400">

<tr height="50">

<td colspan="2" bgcolor="white">

<table title="Banner" id="banner" border="0">

<tr><td>Place a banner here</td></tr>

</table>

</td>

</tr>

<tr height="200">

<td bgcolor="white">

<table id="navigation" title="Navigation" border="0">

<tr><td>Links!</td></tr>

<tr><td>Links!</td></tr>

<tr><td>Links!</td></tr>

</table>

</td><td bgcolor="white">

<table title="Content" id="content" border="0">

<tr><td>Content goes here</td></tr>

</table>

</td>

</tr>

</table>

## Basic Layout:

|  |  |  |
| --- | --- | --- |
| |  | | --- | | Place a banner here | | |
| |  | | --- | | Links! | | Links! | | Links! | | |  | | --- | | Content goes here | |

This approach is basic, yet organized. The code becomes complex rather fast, so you will need to be sure to properly assign height and width values to your tables as well. The more specific you are about heights and widths, the less room there will be for error and debugging.

## HTML Code:

<table id="shell" title="Shell" height="250" width="400"

border="0" bgcolor="black" cellspacing="1" cellpadding="0">

<tr height="50">

<td bgcolor="white">

<table title="banner" id="banner">

<tr><td>Banner goes here</td></tr>

</table>

</td>

</tr>

<tr height="25">

<td bgcolor="white">

<table title="Navigation" id="navigation">

<tr><td>Links!</td>

<td>Links!</td>

<td>Links!</td></tr>

</table>

</td>

</tr>

<tr>

<td bgcolor="white">

<table title="Content" id="content">

<tr>

<td>Content goes here</td>

</tr>

</table>

</td>

</tr>

</table>

## Basic Layout 2:

|  |  |
| --- | --- |
| |  | | --- | | Banner goes here | |
| |  |  |  | | --- | --- | --- | | Links! | Links! | Links! | |
| |  | | --- | | Content goes here | |

The code is quite a lot to look at, SO break it up and organize it in your own way to make things easier for you.

# HTML - Frames

Frames allow for multiple .html documents to be displayed inside of one browser window at a time. This means that one page has no content on it, but rather tells the browser which web pages you would like to open. With the addition of [CSS](http://www.tizag.com/cssT/) and PHP, frames have become outdated, but if you wish to use them, read on.

**[Advertise on Tizag.com](http://www.tizag.com/about/advertise.php)**

# Frames - A Generic Frame Page

Frames are most typically used to have a menu in one frame, and content in another frame. When someone clicks a link on the menu, that link is then opened in the content page. Here is a classic example of a basic "index" frameset with a menu on the left and content on the right.

## HTML Code:

<html>

<body>

<frameset cols="30%,\*">

<frame src="menu.html">

<frame src="content.html">

</frameset>

</body>

</html>

## Frame Set:

Here's the example: [Frame Index](http://www.tizag.com/pics/htmlT/frameindex.html)

* **frameset** - The parent tag that defines the characteristics of this frames page. Individual frames are defined inside it.
* **frameset cols="#%, \*"** - The width that each frame will have. In the above example, we chose the menu (the 1st column) to be 30% of the total page and used a "\*", which means the content (the 2nd column) will use the remaining width for itself (70%).
* **frame src=""** - The URL of the web page to load into the frame.

A good rule of thumb is to call the page which contains this frame information "index.html", as that is typically a site's main page.

# Adding a Banner or Title Frame

Add a row to the top for a title and graphics with the code as follows:

## HTML Code:

<html>

<body>

<frameset rows="20%,\*">

<frame src="title.html">

<frameset cols="30%,\*">

<frame src="menu.html">

<frame src="content.html">

</frameset>

</frameset>

</body>

</html>

# FrameBorder and FrameSpacing

You've probably noticed those ugly gray lines that appear between the frames. It is possible to remove these and manipulate the spacing between frames with *frameborder* and *framespacing*. These attributes appear within the *frameset* tag.

Note: *Framespacing* and *border* are the same attribute, but some browsers only recognize one or the other, so use both, with the same value, to be safe.

* **frameborder="#"** - Determines whether there will be a border.
* **border="#"**- Modifies the border width.
* **framespacing="#"** -Modifies the border width, used by Internet Explorer.

Here's an example of the same frameset without the borders.

## HTML Code:

<html>

<body>

<frameset border="0" frameborder="0" framespacing="0" rows="20%,\*">

<frame src="title.html">

<frameset border="0" frameborder="0" framespacing="0" cols="30%,\*">

<frame src="menu.html">

<frame src="content.html">

</frameset>

</frameset>

</body>

</html>

## Frame Borders:

Here's a visual:[Visual](http://www.tizag.com/pics/htmlT/frameindex2.html)

# Frame Name and Frame Target

How nice would it be to make each menu link load into the content page? We do this by naming each frame and setting the correct *base target* inside "menu.html".

## HTML Code:

<html>

<body>

<frameset rows="20%,\*">

<frame name="title" src="title.html">

<frameset cols="30%,\*">

<frame name="menu" src="menu.html">

<name="content" src="content.html">

</frameset>

</frameset>

</body>

</html>

## HTML Code:

<html>

<head>

<base target="content">

</head>

<body>

<!-- Content Goes Here -->

</body>

</html>

## Frame Target:

Here's the Visual: [Visual](http://www.tizag.com/pics/htmlT/frameindex-target.html)

We first named the content frame "content" on our frame page, and then we set the base target inside "menu.html" to point to that frame. Our frame page is now a perfectly functional menu and content layout!

# Noresize and Scrolling

It's possible to further customize the <frame> tag using the *noresize* and *scrolling* attributes.

## HTML Code:

<html>

<body>

<frameset border="2" frameborder="1" framespacing="2" rows="20%,\*">

<frame src="title.html" noresize scrolling="no">

<frameset border="4" frameborder="1" framespacing="4" cols="30%,\*">

<frame src="menu.html" scrolling="auto" noresize>

<frame src="content.html" scrolling="yes" noresize>

</frameset>

</frameset>

</body>

</html>

## Noresize and Scrolling:

Here's the Visual: [Visual](http://www.tizag.com/pics/htmlT/frameindex3.html)

* **noresize** - Determines whether the frames can be resized by the visitor or not. (values "true" and "false")
* **scrolling** - Determines whether scrolling is allowed in the frame or not (values "true" and "false")

We set the scrolling for our content frame to "yes" to ensure our visitors will be able to scroll if the content goes off the screen. We also set the scrolling for our title banner to no, because it does not make sense to have a scrollbar appear in the title frame.

# HTML Character Entities

An "entity" is a fancy term for a symbol. Several symbols, such as copyright, trademark, or foreign cash symbols, exist on your standard keyboard, so you need to display these characters using a different method.

**[Advertise on Tizag.com](http://www.tizag.com/about/advertise.php)**

There are three parts to every entity.

* Each begins with a ampersand - &
* Then the entities name - copy
* And finally a semicolon - ;

## Copyright:

Combine &copy; to make - © - Copyright symbol.

Expect complications if you forget to include all three parts of an entity.

## More Entities:

|  |  |  |  |
| --- | --- | --- | --- |
| ¢ | Cent | ¢ | &cent; |
| £ | English Pound | £ | &pound; |
| ¤ | Currency | ¤ | &curren; |
| ¥ | Yen | ¥ | &yen; |
| ® | Registered Trademark | ® | &reg; |
| ° | Degree(s) | ° | &deg; |
| ± | Plus or Minus | ± | &plusmn; |
| ¼ | ¼ Fraction | ¼ | &frac14; |
| ½ | ½ Fraction | ½ | &frac12; |
| ¾ | ¾ Fraction | ¾ | &frac34; |

View a more complete list at: [Entities Table](http://www.tizag.com/htmlT/entreftable.php" \t "_blank)

# Additional Spaces and <>.

Regardless of how many spaces you place between words, your web browser will only render a single space. To get around this, use the *non-breaking space* character entity.

## HTML Code:

<p>Everything that goes up, must come &nbsp;&nbsp;&nbsp;  
&nbsp;&nbsp;&nbsp; down!</p>

## Spaces:

Everything that goes up, must come       down!

In HTML, we use less than and greater than characters to create tags, so to use them on your website you will need entities.

## HTML Code:

<p>

Less than - &lt; <br />

Greater than - &gt; <br />

Body tag - &lt;body&gt;

</p>

## Less than Greater than:

Less than - <  
Greater than - >   
Body tag - <body>

Take a few minutes to view and play with the symbols listed in the [Entities Table](http://www.tizag.com/htmlT/entreftable.php" \t "_blank).

# HTML - Scripts

There are two very popular scripting languages that are commonly used in HTML to make web pages come alive. JavaScript and VBScript are very useful scripting languages to know, if you have the time.

**[Advertise on Tizag.com](http://www.tizag.com/about/advertise.php)**

With HTML scripts, you can create dynamic web pages, make image rollovers for really cool menu effects, or even validate the data on your HTML forms before users submit their information. However, JavaScript and VBScript are very complicated compared to HTML. It may be simpler just to download someone else's scripting code and modify it for use on your web page (if they have given you permission to do so, of course!).

# HTML Javascript Code

If you want to insert JavaScript code into your HTML, you are going to use the <script> tag. If you would like to know more about JavaScript, check out our [JavaScript Tutorial](http://www.tizag.com/javascriptT/). Below is the correct code to insert embedded JavaScript code onto your site.

## HTML Code:

<script type="text/javascript">

<!--script

\*\*\*Some JavaScript code should go here\*\*\*

-->

</script>

For JavaScript, you set the *type* attribute equal to "text/javascript", which is similar to the process of [specifying CSS](http://www.tizag.com/cssT/). We can also include a comment around the JavaScript code. This will prevent browsers that do not support JavaScript or have had JavaScript disabled from displaying the JavaScript code in the web browser.

# HTML VBScript How To

To insert VBScript code onto your website, you must once again make use of the <script> tag. Below is the correct code to insert VBScript code onto your site.

## HTML Code:

<script type="text/vbscript">

<!--script

\*\*\*The VBScript code should go in this spot\*\*\*

-->

</script>

For VBScript, you set the *type* attribute equal to "text/vbscript", which is similar to [specifying CSS](http://www.tizag.com/cssT/). We also include a comment around the VBScript code. This will prevent browsers that do not support VBScript or have had VBScript disabled from displaying the VBScript code in the web browser.

# HTML - <!-- Commenting Scripts -->

Scripting languages such as JavaScript and VBScript must be commented out as well. You will learn that it is only once they are placed within the <script> tags that the browser executes the scripts without causing errors.

## HTML Code:

<script>

<!--

document.write("Hello World!")

//-->

</script>

With this example, we are jumping far ahead. Just be sure you understand when to use comments and where to look for them. They are a very useful tool for any large HTML project.

# HTML - Bold

Creating bold text can be accomplished through the use of the <b> bold tag.

**[Advertise on Tizag.com](http://www.tizag.com/about/advertise.php)**

## HTML Code:

<b>This text is entirely BOLD!</b>

## Bold:

**This text is entirely BOLD!**

Place the bold tag inside other elements to highlight important words and give feeling to your text.

## HTML Code:

<p><b>Don't</b> touch that!</p>

## More Bold:

**Don't** touch that!

You may also use it to separate words from their meaning in a dictionary fashion.

## HTML Code:

<p><b>Cardio:</b> Latin word meaning of the heart.</p>

## Dictionary:

**Cardio:** Latin word meaning of the heart.

The idea here is to use the bold tag in quick formatting situations. It is not a good idea to bold entire paragraphs or other elements simply because you want the text to be larger or fatter. Use Cascading Style Sheets for font styles and sizes.