**CSS Introduction**

It is time to take your web designing skills to the next level with **C**ascading **S**tyle **S**heets (CSS). They are a way to control the look and feel of your HTML documents in an organized and efficient manner. With CSS you will be able to:

* Add new looks to your old HTML
* Completely restyle a web site with only a few changes to your CSS code
* Use the "style" you create on any webpage you wish!

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

A stylesheet can, and should be, completely separate from your HTML documents. When you have mastered CSS and HTML, you will be able to separate your web site's design and formatting (CSS) from the content (HTML).

**Intended Audience**

Before you begin the CSS Tutorial we suggest that you check to see you meet the following recommendations:

* You have used HTML in the past
* You know the basic HTML tags and vocabulary.
* You want to be a better web designer!

If you said no to one of the above, we recommend that you check out our [HTML Tutorial](http://www.tizag.com/htmlT/) before taking on CSS.

When you are ready, continue the tutorial to learn about the basic form of CSS and where you should place your CSS code.

**CSS Selector**

CSS selectors are the heart and soul of CSS. They define which HTML elements you are going to be manipulating with CSS code and you should have a solid understanding of them when you are finished with this tutorial. Luckily for you, they are pretty simple to comprehend!

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

**CSS Selector: Where It Fits In**

In a typical CSS statement you have the following:

* SELECTOR { PROPERTY: VALUE }

"Property" is the CSS element you wish to manipulate and "VALUE" represents the value of the specified property.

**CSS Selector Name**

The selector name creates a direct relationship with the HTML tag you want to edit. If you wanted to change the way a paragraph tag behaved, the CSS code would look like:

* p { PROPERTY: VALUE }

The above example is a template that you can use whenever you are manipulating the paragraph HTML element. In the next lessons, we will not only teach where to place your CSS, but why and where you should use CSS as well.

# Internal CSS

Cascading Style Sheets come in three flavors: internal, external, and inline. We will cover internal and external, as they are the only flavors a designer should utilize. In this lesson, we cover the basics of the easier type, internal. When using internal CSS, you must add a new tag, <style>, inside the <head> tag. The HTML code below contains an example of <style>'s usage.

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

## CSS Code:

<html>

<head>

<style type="text/css">

</style>

</head>

<body>

<p>Your page's content!</p>

</body>

</html>

This doesn't actually do anything visually. The code style tag just tells the browser that we will be defining some CSS to be used on this page.

# Creating Internal CSS Code

CSS code is not written the same way as HTML code is. This makes sense because CSS is not HTML, but rather a way of manipulating existing HTML. Below is an example of some simple, yet fully functional, CSS code.

## CSS Code:

<html>

<head>

<style type="text/css">

p {color: white; }

body {background-color: black; }

</style>

</head>

<body>

<p>White text on a black background!</p>

</body>

</html>

## Display:

White text on a black background!

You probably noticed that in our CSS code we were altering the <body> and <p> HTML tags. The great thing about CSS is that it is an intuitive language. Once you understand the general format for CSS code, you are pretty much set.   
**General CSS Format**:

* "HTML tag" **{** "CSS Property" **:** "Value" **; }**

Back in our code example, we manipulated <p> and <body>, both well known HTML tags. To clarify, here is a step-by-step process of what is going on in that first line of CSS code where we played around with "p".

* We chose the HTML element we wanted to manipulate. - **p{ : ; }**
* Then we chose the CSS attribute color. - p { **color:** ; **}**
* Next we choose the font color to be white. - p { color: **white**; **}**

Now all text within a paragraph tag will show up as white! Now an explanation of the CSS code that altered the <body>'s background:

* We choose the HTML element Body - **body { : ; }**
* Then we chose the CSS attribute. - body { **background-color:** ; **}**
* Next we chose the background color to be black. - body { background-color: **black**; **}**

Until you become accustomed to using CSS code, you will probably find your CSS code not working as you expected. A leading cause of this might be an out of place **:**, **;**, **{**, or **}** or it might be that you forgot to use a **:**, **;**, **{**, or **}** when it was required. Be sure to check back here if you ever have issues with the correct format for CSS.

# Internal CSS Gotta Knows

* Place your CSS Code between <style> and </style>
* Be sure you know the correct format(syntax) of CSS code.
* CSS will literally save you hours of time... after you spend a few getting the hang of it.

# External CSS

When using CSS it is preferable to keep the CSS separate from your HTML. Placing CSS in a separate file allows the web designer to completely differentiate between content (HTML) and design (CSS). External CSS is a file that contains only CSS code and is saved with a ".css" file extension. This CSS file is then referenced in your HTML using the <link> **instead** of <style>. If you're confused, don't worry. We are going to walk you through the whole process.

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

# File Creation

Let us get started by making that external CSS file. Open up notepad.exe, or any other plain text editor and type the following CSS code.

## CSS Code:

body{ background-color: gray;}

p { color: blue; }

h3{ color: white; }

Now save the file as a CSS (.css) file. Make sure that you are not saving it as a text (.txt) file, as notepad likes to do by default. Name the file "test.css" (without the quotes). Now create a new HTML file and fill it with the following code.

## HTML Code:

<html>

<head>

<link rel="stylesheet" type="text/css" href="test.css" />

</head>

<body>

<h3> A White Header </h3>

<p> This paragraph has a blue font.

The background color of this page is gray because

we changed it with CSS! </p>

</body>

</html>

Then save this file as "index.html" (without the quotes) in the same directory as your CSS file. Now open your HTML file in your web browser and it should look something like this..

## Display:

### A White Header

This paragraph has a blue font. The background color of this page is gray because we changed it with CSS!

Congratulations! You just made your first website that uses External CSS! Now, let us move on to the fun stuff.

# Why Use External CSS?

* It keeps your website design and content separate.
* It's much easier to reuse your CSS code if you have it in a separate file. Instead of typing the same CSS code on every web page you have, simply have many pages refer to a single CSS file with the "link" tag.
* You can make drastic changes to your web pages with just a few changes in a single CSS file.

# CSS Inline

Thus far, we have only shown you how to use CSS the way it was meant to be used -- separated from the HTML. However, it is possible to place CSS right in the thick of your HTML code, and this method of CSS usage is referred to as *inline css*.

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

Inline CSS has the highest priority out of the three ways you can use CSS: external, internal, and inline. This means that you can override styles that are defined in external or internal by using inline CSS. However, inline CSS detracts from the true purpose of CSS, to separate design from content, so please use it sparingly.

# CSS Inline - An HTML Attribute

Believe it or not, CSS is built in to every HTML tag. If you want to add a style inside an HTML element all you have to do is specify the desired CSS properties with the *style* HTML attribute. Let's add some style to a paragraph tag.

## CSS Code:

<p style="background: blue; color: white;">A new background and

font color with inline CSS</p>

## Display:

A new background and font color with inline CSS

If you have read through the beginning of this CSS tutorial, you probably have a good idea of what is going on. Below is the general form for setting inline CSS in any HTML element.

## Pseudo Code:

<htmltag style="cssproperty1: value; cssproperty2: value;"> </htmltag>

The normal rules of CSS apply inside the style attribute. Each CSS statement must be separated with a semicolon ";" and colons appear between the CSS property and its value.

# Common Inline CSS Mistakes

When using CSS inline you must be sure not to use quotations within your inline CSS. If you use quotations the browser will interpret this as the end of your style value. Instead, copy our form as we have displayed below.

## CSS Code:

<p style="background: url("yellow\_rock.gif");">

This is broken</p>

<p style="background: url(yellow\_rock.gif);">  
This is workin'</p>

## Display:

This is broken

This is workin'

# CSS Classes

You may be wondering if it is possible to give an HTML element multiple looks with CSS. Say for example that sometimes you want the font to be large and white, while other times you would prefer the font to be small and black. CSS would not be very useful if it did not allow you to have many different types of formats for a single HTML tag. Well, you are in luck! CSS allows you to do just that with the use of classes.

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

# The Format of Classes

Using classes is simple. You just need to add an extension to the typical CSS code and make sure you specify this extension in your HTML. Let's try this with an example of making two paragraphs that behave differently. First, we begin with the CSS code, note the red text.

## CSS Code:

p.first{ color: blue; }

p.second{ color: red; }

## HTML Code:

<html>

<body>

<p>This is a normal paragraph.</p>

<p class="first">This is a paragraph that uses the p.first CSS code!</p>

<p class="second">This is a paragraph that uses the p.second CSS code!</p>

...

## Display:

This is a normal paragraph.

This is a paragraph that uses the p.first CSS code!

This is a paragraph that uses the p.second CSS code!

You can use CSS classes with any HTML element! However, what happens if we had already defined a value for the default <p> tag, would this cause any problems for classes of the paragraph tag?

Well, when this happens the CSS class for any <p> tag will override the default <p> CSS. If the CSS class uses a CSS attribute already defined by the default CSS, then the formatting defined by the class will be the value that is used.

It may be easier to imagine that the CSS for a generic HTML element is the starting point and the only way to change that look is to overwrite the attributes using CSS classes. Please see the example below for a visual of this tricky topic.

## CSS Code:

p{ color: red; font-size: 20px; }

p.test1{ color: blue; }

p.test2{ font-size: 12px; }

## HTML Code:

<html>

<body>

<p>This is a normal paragraph.</p>

<p class="test1">This is a paragraph that uses the p.test1 CSS code!</p>

<p class="test2">This is a paragraph that uses the p.test2 CSS code!</p>

...

## Display:

This is a normal paragraph.

This is a paragraph that uses the p.test1 CSS code! The p.test1 paragraph remained the same size, but its color changed.

This is a paragraph that uses the p.test2 CSS code! The p.test2 paragraph remained the same color, but its size changed.

**Remember**, CSS code in classes will override the general CSS code for that element. **p.test1** overrides **p**!

# CSS Background

The background of your website is very important, so please spend some time with this tutorial. If you are aiming for a professional website, a good rule of thumb is to use a light background with dark text. However, if you're just making a website for pleasure, then any kind of color combination is acceptable.

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

With CSS, you are able to set the background color or image of any CSS element. In addition, you have control over how the background image is displayed. You may choose to have it repeat horizontally, vertically, or in neither direction. You may also choose to have the background remain in a fixed position, or have it scroll as it does normally. The following examples will show you how to implement all of these options.

# CSS Background Color

As you have seen throughout Tizag Tutorials, many different background colors are present. These varying backgrounds were obtained without using tables! Below are a couple examples of CSS backgrounds.

## CSS Code:

h4 { background-color: white; }

p { background-color: #1078E1; }

ul { background-color: rgb( 149, 206, 145); }

## Display:

#### This is a <h4> with a white background

This is a <p> with a background using the hexadecimal value of #1078E1

* This is an unordered list
* with an RGB value of 149, 206, 145

In the above example we used three different formats for defining a color: a color name, hexadecimal values, and RGB. Check out the [the list of supported color names](http://www.utoronto.ca/webdocs/HTMLdocs/Book/Book-3ed/appf/color2.html" \t "_blank). Hexadecimal form is a pound sign (#) followed by, at most, 6 hex values (0-F). RGB defines the individual values for Red, Green, and Blue. Example form: rgb(Red, Green, Blue); with the range of 0-255 for each value.

# CSS Background Image

Need an image to repeat left-to-right, like the gradient background that appears at the top of Tizag.com? Or maybe you would like to have an image that remains fixed when the user scrolls down your page. This can be done quite easily with CSS and more, including:

* choosing if a background will repeat and which directions to repeat in.
* precision positioning
* scrolling/static images

Let's begin with a default CSS background image.

## CSS Code:

p { background-image: url(smallPic.jpg); }

h4{ background-image: url(http://www.tizag.com/pics/cssT/smallPic.jpg); }

## Display:

This <p> has a background image using a relative path to the picture.

#### This <h4> has a background image using the complete url path.

# Background Image Repeat

You can have a background image repeat vertically (y-axis), horizontally (x-axis), in both directions, or in neither direction.

## CSS Code:

p {

background-image: url(smallPic.jpg);

background-repeat: repeat; }

h4 {

background-image: url(smallPic.jpg);

background-repeat: repeat-y;}

ol {

background-image: url(smallPic.jpg);

background-repeat: repeat-x;}

ul {

background-image: url(smallPic.jpg);

background-repeat: no-repeat;}

## Display:

This <p> has a background image repeating in both directions (default repeat). If you use this option, make sure that your image was designed to be repeated.

#### This <h4> has a background image repeating vertically (y). You could this to add a style to the side of your element.

1. This is an ordered list
2. With a background that repeats
3. Horizontally (x)

* This is an unordered list
* With a background that repeats
* in neither direction.

# CSS Fixed Background Image

You may choose to have your background scroll naturally, or to have it in a fixed position. The default value is fixed, so you only need to worry about this if you would like your body's background to scroll. Note: This should only be used with backgrounds fixed to the <body>.

## CSS Code:

body.noScroll {

background-image: url(smallPic.jpg);

background-attachment: fixed;

}

body{

background-image: url(smallPic.jpg);

background-attachment: scroll;}

# CSS Background Image Positioning

If you would like to define where exactly an image appears within an HTML element, you may use CSS's background-position. Please take note that there are three different ways of defining position: length, percentages, and keywords. We recommending using lengths -- specifically, pixels.

## CSS Code:

p {

background-image: url(smallPic.jpg);

background-position: 20px 10px;

}

h4 {

background-image: url(smallPic.jpg);

background-position: 30% 30%;

}

ol {

background-image: url(smallPic.jpg);

background-position: top center;

}

## Display:

This <p> has a background image positioned with pixel values.

#### This <h4> has a background image positioned with percentages.

1. This is an ordered list
2. With a background that was positioned
3. using keywords.

**Note:** When using pixels, the location of the image will be (A)px from the left of the screen and (B)px from the top of the screen, where A and B are integers. **Note:** When using percentages, the location of the image will be (A)% from the left of the screen and (B)% from the top of the screen, where A and B are integers. **Note:** Available positioning keywords are: top, right, bottom, left, and center.

# CSS Gradient Background

If you would like to create a gradient background like the one that appears at the top of Tizag.com, you must first create an image inside a painting program (Photoshop, Draw, etc) like the one you see below.

## Necessary Image:

http://www.tizag.com/pics/BannerTileBackground.gif

Notice that the image is very slim. We are going to be tiling the image horizontally, so you can make the image skinny as possible. As long as the image is 1 pixel or wider, you will be fine.

Using the repeat attribute, we set the value to repeat-x which causes the image to span left to right across the specified element. This example adds a gradient background to the paragraph element.

## CSS Code:

p {

background-image: url(http://www.example.com/gradient.gif);

background-repeat: repeat-x;

}

## Display:

This paragraph has a gradient background. The gradient background was first made in a painting program and then repeated horizontally across the paragraph element. It makes for a neat effect that also loads quickly! Because the image is very skinny, the file size is also very small. You could also create a gradient that changes color left to right and repeat it in the vertical direction to have a gradient on the side of your web page.

# CSS Font

CSS gives you great control over the way your text is displayed. You can change the text size, color, style, and more. You probably already knew how to make text bold or underlined, but did you know you could resize your font using percentages? Let us begin the lesson with an easy and important font attribute, color!

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

# CSS Font Color

Although the color of the text seems like it would be part of CSS Font, it actually is a standalone attribute in CSS. This could be for many reasons, including the fact that it will be used a great deal, so why make the coder type out "font-color", when they could just type out "color" instead? Here's an example of changing the color of your font.

## CSS Code:

h4 { color: red; }

h5 { color: #9000A1; }

h6 { color: rgb(0, 220, 98); }

## Display:

#### This is a red h4 header.

##### This is a hexadecimal #9000A1 h5 header.

###### This is an rgb(0, 220, 98) h6 header.

In the above example we used three different formats for defining a color: a color name, hexadecimal values, and RGB. Check out [the list of supported color names](http://www.utoronto.ca/webdocs/HTMLdocs/Book/Book-3ed/appf/color2.html" \t "_blank). Hexadecimal form is a pound sign (#) followed by at most 6 hex values (0-F). RGB defines the individual values for Red, Green, and Blue.

Example form: rgb(Red, Green, Blue); with the range of 0-255 for each value.

# CSS Font Family

Font families can be divided into two groups: serif and sans-serif. A sans-serif font does not include the small lines at the end of characters, while a serif font does include these small lines. When choosing which kind you prefer, remember that studies have shown that sans-serif fonts are much easier to read on a computer monitor than serif fonts.

## CSS Code:

h4 { font-family: sans-serif; }

h5 { font-family: serif; }

h6 { font-family: arial; }

## Display:

#### This is a header with sans-serif font

##### This is a header with a serif font

###### This is a header with an arial font

As you probably noticed throughout Tizag.com, we do not use serif fonts, except in special cases, like for the titles of the Code and Display boxes.

# CSS Font Size

You can manipulate the size of your fonts by using values, percentages, or key terms. Using values are useful if you do not want the user to be able to increase the size of the font because your site will look incorrect if they did so. Percentages are great when you want to change the default font, but do not want to set a static value. Note: Some browsers now have a "Zoom" feature, so it will allow users to make your website bigger or smaller and static values will grow larger/smaller as well.

## CSS Code:

p { font-size: 120%; }

ol{ font-size: 10px; }

ul{ font-size: x-large; }

## Display:

This is a font size of 120%

1. This is a font size of 10px

* This is a font size of "x-large"

Though key terms are not very useful, the common terms are: xx-large, x-large, large, medium, small, x-small, and xx-small.

# CSS Font Style

CSS Font-Style is where you define if your font will be italic or not. Possible key terms are the following: italic, oblique, and normal.

## CSS Code:

p { font-style: italic; }

h4{ font-style: oblique; }

## Display:

This is an italic font

#### This is an oblique font

# CSS Font Weight

If you want to control the weight of your font (its thickness), using font weight is the best way to go about it. We suggest that you only use font-weight in multiples of 100 (e.g. 200, 300, etc) because any less and you probably will not see any difference. The values range from 100 (thin)-900 (thick).

## CSS Code:

p { font-weight: 100; }

ul{ font-weight: bolder; }

## Display:

**This is a font with a weight of 100**

* This is a font with
* a "bolder" weight

Available key terms for font-weight: bold, bolder, and normal.

# CSS Font Variant

CSS Font Variant allows you to convert your font to all small caps. Note: not every font supports CSS Font Variant, so be sure to test before you publish.

## CSS Code:

p { font-variant: small-caps; }

## Display:

This text was written normally and converted to small-caps

# CSS Text

While CSS Font covers most of the traditional ways to format your text, CSS Text allows you to control the spacing, decoration, and alignment of your text.

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

# Text Decoration

Have you ever wondered how a website removed the underline that usually accompanies a link's text? This is done by removing text-decoration from the link. To learn how to create these types of links, please check out our [CSS Links tutorial](http://www.tizag.com/cssT/pclass.php). Besides the utility with links, text-decoration allows you to add horizontal lines above, below, or through your text.

## CSS Code:

h4{ text-decoration: line-through; }

h5{ text-decoration: overline; }

h6{ text-decoration: underline; }

a { text-decoration: none; }

## Display:

#### This header has a line through the middle

##### This header has an overline

###### This header has an underline

**[This is a link without an underline](http://www.tizag.com/cssT/text.php)** - See our [CSS Link tutorial](http://www.tizag.com/cssT/pclass.php) for more information

# Text Indent

CSS text-indent is a great way to indent your paragraphs without having to use preformatted HTML tags, (<pre>), or inserting spaces manually (&nbsp;). You may define your indentation with exact values or percentages. We recommend using exact values.

## CSS Code:

p { text-indent: 20px; }

h5 { text-indent: 30%; }

## Display:

This is a paragraph that uses a text indentation with the value of 20px. This is the recommended usage of text indentation.

##### This is a header that uses a text indentation of 30%. Tizag does not recommend indenting your text with percentages.

# Text Align

By default, text on your website is aligned to the left, like most literature and other forms of media you read. However, sometimes you may require a different alignment and it can be specified using the text-align attribute.

## CSS Code:

p { text-align: right; }

h5{ text-align: justify; }

## Display:

This paragraph is aligned to the right side of the HTML element. If you ever find a use for using right justify, please let us know. Just kidding, we don't really want to know.

##### This header is justified. We recommend that you either align your text to the left, which is the default setting, or justify your text. But feel free to experiment with all the available alignment options that are at your disposal.

# Text Transform

Text-transform is a quick way to modify the capitalization of your text.

## CSS Code:

p { text-transform: capitalize; }

h5{ text-transform: uppercase; }

h6{ text-transform: lowercase; }

## Display:

Hi, I am happy to see you.

#### Hi, I am happy to see you.

##### Hi, I am happy to see you.

Note: All the above sentences originally were, "Hi, I am happy to see you." With the use of the text-transform CSS attribute we were able to modify the capitalization.

# CSS White Space

The white-space attribute allows you to prevent text from wrapping until you place a break <br /> into your text.

## CSS Code:

p { white-space: nowrap; }

## Display:

This paragraph will not wrap until I tell it to with a break tag. As you can see, it makes the page look   
quite ugly.

In the above paragraph the page break occurred after "... page look", which caused the text to resume on the following line.

**Note:** We set a CSS overflow property, above, so that the example could be shown more readily.

# CSS Word Spacing

With the CSS attribute word-spacing you are able to specify the exact value of the spacing between your words. Word-spacing should be defined with exact values.

## CSS Code:

p { word-spacing: 10px; }

## Display:

This paragraph has a word-spacing value of 10px.

# CSS Letter Spacing

With the CSS attribute letter-spacing you are able to specify the exact value of the spacing between your letters. Letter-spacing should be defined with exact values.

## CSS Code:

p { letter-spacing: 3px; }

## Display:

This is a paragraph with letter-spacing of 3px.

# CSS Padding

With CSS Padding you will be able to change the default padding that appears inside various HTML elements (paragraphs, tables, etc). But first, let us make sure we understand the definition of padding. A padding is the space between an element's border and the content within it.

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

Please see the example below for a visual representation. **Note:** The border has been made visible, for each element, so you may more readily see the effects of padding.

## CSS Code:

p {padding: 15px; border: 1px solid black; }

h5{padding: 0px; border: 1px solid red;}

## Display:

This is a paragraph that has a padding of 15 pixels on every side: top, right, bottom, and left.

##### This header has no padding, which places the text right against the border!

There are several ways to go about defining the CSS Padding attribute. We will show you every possible way and let you know which ways are the best.

# CSS Padding: 1 Value

As you saw in the example above, padding can be uniform inside an element. Specifying one value will create a uniform padding on all sides: top, right, bottom, left. In addition to using exact values, you may also define the padding with the use of percentages.

## CSS Code:

p {padding: 2%; border: 1px solid black; }

h5{padding: 0px; border: 1px solid red;}

## Display:

This is a paragraph that has a padding of 2% on every side: left, top, right, and bottom.

##### This header has no padding. It is only spaced away from the paragraph because the paragraph has a padding of 5 pixels!

# CSS Padding: padding-(direction):

Each HTML element actually has 4 different paddings: top, right, bottom, and left. It is possible to define these individual paddings simply by adding a direction suffix to the padding attribute. Example form: padding-(direction). Defining only one direction will leave the other 3 default paddings untouched.

## CSS Code:

p { padding-left: 5px; border: 1px solid black; }

h5{

padding-top: 0px;

padding-right: 2px;

padding-bottom: 13px;

padding-left: 21px;

border: 1px solid red;

}

## Display:

This paragraph had one padding specified(left), using directional declaration.

##### This header had each padding specified separately, using directional declaration.

# CSS Padding: 2 & 4 Values

Four padding values can be declared at once by either specifying two or four values. When only using two values, the first will define the padding on the top and bottom, while the second will define the padding on the left and right.

When using the four value padding specification, the corresponding directions are: top, right, bottom, left. To help you remember what the order is, just remember that it starts at the top and then moves clockwise until it reaches the left. The examples below shows partial (2) and complete (4) padding usage.

## CSS Code:

p {

padding: 5px 15px;

border: 1px solid black;

}

h5{

padding: 0px 5px 10px 3px;

border: 1px solid red;

}

## Display:

This paragraph has a top and bottom padding of 5 pixels and a right and left padding of 15 pixels.

##### This header has a top padding of 0 pixels, a right padding of 5 pixels, a bottom padding of 10 pixels, and a left padding of 3 pixels.

# CSS Border

CSS Border, our personal favorite CSS attribute, allow you to completely customize the borders that appear around HTML elements. With HTML, it used to be impossible to place a border around an element, except for the table. CSS Borders let you create crisp, customized border styles with very little work, compared to the antiquated methods of HTML.

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

# Border Style Types

There are numerous types of border styles at your disposal. We recommend that you experiment with many color/border-style combinations to get an idea of all the different looks you can create. **Note:** We have used [CSS Classes](http://www.tizag.com/cssT/class.php) below, so check out the [CSS Classes](http://www.tizag.com/cssT/class.php) lesson if you do not understand.

## CSS Code:

p.solid {border-style: solid; }

p.double {border-style: double; }

p.groove {border-style: groove; }

p.dotted {border-style: dotted; }

p.dashed {border-style: dashed; }

p.inset {border-style: inset; }

p.outset {border-style: outset; }

p.ridge {border-style: ridge; }

p.hidden {border-style: hidden; }

## Display:

This is a solid border

This is a double border

This is a grooved border

This is a dotted border

This is a dashed border

This is an inset border

This is an outset border

This is a ridged border

This is a hidden border

This is probably obvious, but the default border-style setting for an element is hidden.

# Border Width

To alter the thickness of your border use the border-width attribute. You may use key terms or exact values to define the border width. **Note:** You must define a border-style for the border to show up. Available terms: thin, medium, thick.

## CSS Code:

table { border-width: 7px;

border-style: outset; }

td { border-width: medium;

border-style: outset; }

p { border-width: thick;

border-style: solid; }

## Display:

|  |  |
| --- | --- |
| This table has an outset border | with a width of 7 pixels. |
| Each cell has an outset border | with a "medium" width. |

This paragraph has a solid border with a "thick" width.

# Border Color

Now for the creative aspect of CSS Borders! With the use of the border-color attribute, you will be able to create customized borders to fit the flow and layout of your website. Border colors can be any color defined by RGB, hexadecimal, or key terms. Below is an example of each of these types.

## CSS Code:

table { border-color: rgb( 100, 100, 255);

border-style: dashed; }

td { border-color: #FFBD32;

border-style: ridge; }

p { border-color: blue;

border-style: solid; }

## Display:

|  |  |
| --- | --- |
| This table has a dashed border | with the RGB value ( 100, 100, 255). |
| Each cell has a ridged border | with a hexadecimal color of #FFBD32. |

This paragraph has a solid border with a color of "blue".

# Border: border-(direction)

If you would like to place a border on only one side of an HTML element, or maybe have a unique look for each side of the border, then use border-(direction). The direction choices are of course: top, right, bottom, and left. CSS allows you to treat each side of a border separately from the other three sides. Each side can have its own color, width, and style set, as shown below.

## CSS Code:

p { border-bottom-style: dashed ;

border-bottom-color: yellow;

border-bottom-width: 5px; }

h4 { border-top-style: double;

border-top-color: purple;

border-top-width: thick; }

h5 { border-left-style: groove;

border-left-color: green;

border-left-width: 15px;

border-bottom-style: ridge;

border-bottom-color: yellow;

border-bottom-width: 25px; }

## Display:

This only has a bottom border

#### This header has a top only border.

##### This header has a left and bottom border.

# Border: All in One

While it is nice that CSS allows a web developer to be very specific in creating a customized border, sometimes it is just easier and less of a headache to create a uniform border, all in single line of CSS code. Most of the borders you see on Tizag are created in this manner.

## CSS Code:

p { border: 20px outset blue ;}

h4{ border: 5px solid; }

h5{ border: dotted; }

## Display:

This blue, outset border was defined in 1 line of CSS code.

#### We did not set the border-color for this header, so the default value is used.

##### This header only had the style defined.

# CSS Lists

Lists come in two basic flavors: unordered and ordered. However, CSS allows for more list customization than HTML -- to the extent that even images can be used as bullet points for unordered lists!.

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

# CSS List Style Type

If you want to use something different from the default numbering of ordered lists, or the bullets/discs of unordered lists, then all you have to do is choose a different style for your lists. CSS allows you to select from a wide variety of different list item shapes.

* Unordered list styles: square, circle, disc (default), and none
* Ordered list styles: upper-alpha, lower-alpha, upper-roman, lower-roman, decimal (default), and none

## CSS Code:

ol { list-style-type: upper-roman; }

ul { list-style-type: circle; }

## Display:

Here is an ordered list:

* This list is
* using roman
* numerals
* with CSS!

and now an unordered list:

1. This list is
2. using circle types
3. with CSS!

# CSS Lists With Images

As we stated in the introduction, CSS lists allow you to insert an image in place of the normal bullets. A good choice for a bullet image would be one that is smaller than the height of your text and is a relatively simple/plain graphic.

## CSS Code:

ul { list-style-image: url("listArrow.gif"); }

ol { list-style-image: url("listArrow2.gif"); }

## Display:

Here is an unordered list:

* This list is
* using a picture with CSS!

and now an ordered list:

1. This list is
2. using a picture
3. with CSS!

As you can see, it does not matter if the lists are <ul> or <ol> when using images. Nevertheless, it is a good coding practice to only use images for an unordered list. Ordered lists usually have an incremented (growing) value that appears next to each list item and you can't do this with just one image.

# CSS List Position

With CSS, it is possible to alter the indentation that takes place with your list items. See the example below for the trick of indenting your lists.

## CSS Code:

ul { list-style-position: inside; }

ol { list-style-position: outside; }

## Display:

* This list is using inside positioning so that means the bullets will move inside with the text and it's really easy to see how it's different with multiple lines.
* The bullets
* are inside.

and now an ordered list:

1. This list is using outside positioning and is the default position of bullets.
2. These lines
3. are just normal.

**Note:** "Outside" is actually the default setting for indentation.

# List: All in One

It is possible to combine the above CSS techniques into one line of CSS. This is useful if you would like to have a list-style-type take the place of your list image, if the image is not able to be loaded.

## CSS Code:

ul { list-style: upper-roman inside url("http://www.example.com/notHere.gif");}

## Display:

* This list's image did
* NOT load correctly
* But because we chose to include
* A CSS list type, we avoided a bland list!

# CSS Links ( Pseudo-classes )

Probably the coolest thing about CSS is that it gives you the ability to add effects to your anchor tags, otherwise known as links. In HTML, the only way to add this effect would be to use JavaScript, but with the addition of CSS, JavaScript links can be forgotten.

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

# CSS Anchor/Link States

You may not know it, but a link has four different states that it can be in. CSS allows you to customize each state. Please refer to the following keywords that each correspond to one specific state:

* **link** - this is a link that has not been used, nor is a mouse pointer hovering over it
* **visited** - this is a link that has been used before, but has no mouse on it
* **hover** - this is a link currently has a mouse pointer hovering over it/on it
* **active** - this is a link that is in the process of being clicked

Using CSS you can make a different look for each one of these states, but at the end of this lesson we will suggest a good practice for CSS Links.

# CSS Pseudo-Classes

The format for CSS Links is a little different from what you've seen in this tutorial so far. To modify these four states, you have to use the following CSS code formatting:

## CSS Code:

a:(STATE'S NAME) { attribute: value; }

The state's name is the "pseudo class" that defines how the HTML element should appear, depending on which state it is in. Below is an example of changing the "link", "visited", and "hover" state. Note the order that they are defined, as it is the proper ordering to make a functioning CSS link.

## CSS Code:

a:link { color: red; }

a:visited { color: red; }

a:hover { color: blue; }

## HTML Code:

<a href="">This is a special CSS Link</a>!

## Display:

[This is a special CSS Link](http://www.tizag.com/cssT/pclass.php)!

The states must be defined in the correct order. Here is the order, starting with the one you must define first:

1. link
2. visited
3. hover
4. active

# Removing the Default Underline

Throughout Tizag.com you probably have noticed the different styles that we have for certain links. Our menu's do not have an underline, unless you are hovering, while the links in our main content do have underlines. To remove the underline from certain states of a link, use text-decoration: none.

## CSS Code:

a:link { color: red; text-decoration: none; }

a:visited { color: red; text-decoration: none; }

a:hover { color: blue; }

## Display:

[This is a link that only has an underline if you hover your mouse over it...](http://www.tizag.com/cssT/pclass.php)

Be careful when removing the underline from your links. Sometimes, the removal of the underline may cause the link to be indistinguishable from normal text. There is a very small chance the common visitor will be able to notice that it is a link, if this is the case. So if you choose to remove the underline, be sure you do something else to the link to make it stand out.

# A Couple of Examples

Below are two examples that use many forms of CSS to manipulate the states of a hyperlink.

## CSS Code:

a:link {

color: white;

background-color: black;

text-decoration: none;

border: 2px solid white;

}

a:visited {

color: white;

background-color: black;

text-decoration: none;

border: 2px solid white;

}

a:hover {

color: black;

background-color: white;

text-decoration: none;

border: 2px solid black;

}

## Display:

[This link is a almost over-the-top](http://www.tizag.com/cssT/pclass.php)

## CSS Code:

a:link {

color: blue;

background-color: red;

font-size: 26px;

border: 10px outset blue;

font-family: sans-serif;

text-transform: lowercase;

text-decoration: none;

}

a:visited {

color: blue;

background-color: red;

font-size: 26px;

border: 10px outset blue;

font-family: sans-serif;

text-transform: lowercase;

text-decoration: none;

}

a:hover{

color: blue;

background-color: red;

font-size: 27px;

border: 10px inset blue;

font-family: serif;

text-transform: uppercase;

text-decoration: line-through;

letter-spacing: 3px;

word-spacing: 6px;

font-weight: normal;

}

## Display:

[The Ugly Linkling](http://www.tizag.com/cssT/pclass.php)

# CSS Mouse Cursor

When using Windows, Linux, or a Macintosh you will have undoubtedly seen many different mouse cursor icons. There is the normal mouse cursor icon that looks like a skewed arrow; the "I" looking icon when selecting text, and some sort of animated logo when the computer is thinking (usually an hourglass).

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

With CSS you can make it so different mouse icons appear when people visit your site. **NOTE:** You should know that some people find web pages that change mouse cursor icons annoying, so consider that when playing around with this CSS property!

# CSS Cursor Icons

In this lesson we will show how to change the mouse cursor icon for a few different HTML elements. Below is a list of the most commonly accepted cursors:

* **default** - Display the normal mouse cursor icon
* **wait** - The mouse icon to represent the computer "thinking"
* **crosshair** - A cross hair reticle
* **text** - An "I" shaped icon that is displayed when selecting text
* **pointer** - A hand icon that you see when you hover over an HTML link
* **help** - A question mark (usually)

# CSS Cursor Code

Now let's try these puppies out. Here are a few cursor code examples that should help you customize your site.

## CSS Code:

p { cursor: wait }

h4 { cursor: help }

h5 { cursor: crosshair }

## Display:

Please Wait While this Web Page Loads

#### How May I Help You?

##### Stick Your Hands in the Air!

Mouse over the lines of text and see which icon your cursor changes to! Sometimes you can add a little bit of excitement to a web page with a well-placed cursor icon change. However, if you make the icons confusing, or change them too often, people will probably leave your site with a poor impression of your web design talent!

**CSS Properties**

In HTML, the tags (i.e. <b>, <body>, <a>, etc) are the meat and potatoes of the HTML language. In CSS, there are many properties (i.e. Font, Text, Box, and Color) that you have probably seen if you've read through this tutorial.

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

CSS has grouped all the CSS properties into logical groups to give the massive amount of properties some order, unlike HTML. This lesson will review these areas and give a brief description of what they are for. For a quick reference, check out our CSS Properties Reference.

**CSS Font Properties**

The CSS font properties control all aspects of your text graphical representation. From the thickness of your font (font-weight) to font type (font-family) of your choice. Here are all the font properties at your disposal:

* font
* font-family
* font-size
* font-style
* font-weight
* font-variant

**CSS Text Properties**

The CSS text properties control the spacing, alignment, decoration, and other miscellaneous aspects of the text. Here is a list of all the CSS text properties. Remember to check out our [CSS Properties Reference](http://www.tizag.com/cssT/reference.php) for a description and example of all of the properties mentioned in this lesson.

* letter-spacing
* word-spacing
* text-decoration
* vertical-align
* text-transform
* text-align
* text-indent
* line-height

**CSS Box Properties**

The CSS box properties are used to define the spacing in and around HTML elements, their borders, and other positioning aspects. Border, Margin, and Padding all have four properties each: top, right, bottom, and left.

* Margin
* Padding
* Border
* Border-width
* Border-color
* Border-style
* Width
* Height
* Float
* Clear

**CSS Color Properties**

The CSS color property defines what color the text inside the specified HTML element will have. Use classes or identifiers to have multiple colors for one type of HTML element.

* Color

**CSS Background Properties**

The CSS background properties control things like if the background is a single color or maybe an image. If it's an image you can set the position of the image and tell it whether or not you want the image to repeat left-to-right and/or top-to-bottom.

* Background
* Background Color
* Background Image
* Background Repeat
* Background Attachment
* Background Position

**CSS Classification Properties**

We think of the classification properties as having the list element and all the leftover elements that would not fit into any other category. Check out our [CSS Properties Reference](http://www.tizag.com/cssT/reference.php) for a an example of all the properties mentioned here.

* Display
* Whitespace
* List Style
* List Style Type
* List Style Image
* List Style Position

# CSS Position

With the knowledge of CSS Positioning you will be able to manipulate the exact position of your HTML elements. Designs that previously required the use of JavaScript or HTML image maps may now be done entirely in CSS. Not only is it easier to code, but it also loads much quicker!

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

# Position Relative

Relative positioning changes the position of the HTML element relative to where it normally appears. If we had a header that appears at the top of our page, we could use relative positioning to move it a bit to the right and down a couple of pixels. Below is an example.

## CSS Code:

h3 {

position: relative;

top: 15px;

left: 150px;

}

p {

position: relative;

left: -10px;

}

You probably noticed that you define the four possible directions (left, right, up, and down) using only two (left and top). Here's a quick reference when moving HTML elements in CSS.

* Move Left - Use a negative value for *left*.
* Move Right - Use a positive value for *left*.
* Move Up - Use a negative value for *top*.
* Move Down - Use a positive value for *top*.

## Display:

[Before CSS](http://www.tizag.com/cssT/position-ex1.php" \t "_blank) [After Positioning](http://www.tizag.com/cssT/position-ex1css.php" \t "_blank)

Remember, relative positioning moves stuff from where it would normally be. So if you had a paragraph in the middle of a page and you made both the top and left values negative 50, then the paragraph would move up and to the left 50 pixels from its normal location.

# Position Absolute

With absolute positioning, you define the exact pixel value where the specified HTML element will appear. The point of origin is the top-left of the parent element (that's the HTML element that it is inside of), so be sure you are measuring from that point. For example, if you had a bold tag inside of a paragraph tag, the parent of the bold tag would be the paragraph

Since the paragraph tag is our parent element, we need to decide where want our bold tag to appear in regards to the top left of the paragraph. Let's have it appear 10 pixels down and 30 pixels to the right.

## CSS Code:

b{

position: absolute;

top: 10px;

left: 30px;

}

## Display:

[Before CSS](http://www.tizag.com/cssT/position-ex3.php" \t "_blank)   
[After Positioning](http://www.tizag.com/cssT/position-ex3css.php" \t "_blank)

Specifying a direction with absolute positioning works the same as with relative positioning.

# CSS Layers

After learning how to position HTML elements, you may have noticed how this can lead to HTML elements being on top of one another. CSS allows you to control which item will appear on top with the use of layers.

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

In CSS, each element is given a priority. HTML elements that appear later in the source code than others will have a higher priority by default. If there are two overlapping CSS positioned elements, the element with the higher priority will appear on top of the other.

To manually define a priority, set the *z-index* value. The larger the value, the higher the priority the element will have.

## CSS Code:

h4{

position: relative;

top: 30px;

left: 50px;

z-index: 2;

background-color: #336699;

}

p {

position: relative;

z-index: 1;

background-color: #FFCCCC;

}

## display:

#### Header Z-Index = 2

You probably can't read this part, so I will fill it in with useless text for the time being. This paragraph has a z-index of 1, which is less than the header. As you can see, the header has been moved down, using positioning, and is now resting on top of this paragraph. If we had not defined the z-index, by default the paragraph would have been on top of the header because it appears later in our HTML code.

Other ways to utilize layers might be to place a few images on top of each other to create a beautiful collage, have your menu slightly overlap you content pane, or anything your imagination can come up with. Just remember to keep your web site user-friendly!

# CSS Float

Floating is often used to push an image to one side or another, while having the text of a paragraph wrap around it. This type of usage is often referred to as text wrapping and resembles what you might see in many magazines that have articles which wrap around images of various shapes and sizes.

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

# Float Image

Wrapping text around an image is easy when using the CSS Float attribute. You have a choice to either float the picture to the left or to the right and the rest is done for you. Below is an example of an image that is floated to different sides of a paragraph.

## CSS Code:

img.floatLeft {

float: left;

margin: 4px;

}

img.floatRight {

float: right;

margin: 4px;

}

## HTML Code:

<body>

<img src="sunset.gif" class="floatLeft">

<p>The images are contained with...</p>

<img src="sunset.gif" class="floatRight">

<p>This second paragraph has an...</p>

</body>

## Display:

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

The images are contained within the paragraph tag. The image has floated to the left, and also to the right because we have used both types of image floating in this example. Notice how the text wraps around the images quite nicely. The images are contained within the paragraph tag. The image has floated to the left, and also to the right because we have used both types of image floating in this example. Notice how the text wraps around the images quite nicely.

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

This second paragraph has an image that is floated to the right. It should be noted that a margin should be added to images so that the text does not get too close to the image. There should always be a few pixels between words and borders, images, and other content. This second paragraph has an image that is floated to the right. It should be noted that a margin should be added to images so that the text does not get too close to the image. There should always be a few pixels between words and borders, images, and other content.

# Floating Multiple Images

If you were to simply float three images to the right, they would appear alongside one another. If you wish to have the next image start below the end of the previous image, then use the CSS Clear attribute.

## CSS Code:

img.floatRightClear {

float: right;

clear: right;

margin: 4px;

}

## HTML Code:

<body>

<img src="sunset.gif" class="floatRightClear">

<img src="sunset.gif" class="floatRightClear">

<p>The images are appearing...</p>

<p>If we had specified...</p>

<p>The number of paragraphs...</p>

</body>

## Display:

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

The images are appearing below one another because the CSS clear attribute was used with the value of "right". This forces all right floating items to appear below any previous right floating items.

If we had specified clear:left in our CSS code, then there would be no effect on the images, and they would appear in their default pattern, next to each other, all on one line.

The number of paragraphs, and the size of the paragraphs, will not effect how these images will appear.

# CSS Classes vs ID

There is often confusion about when it is appropriate to use CSS IDs and when CSS Classes should be used instead. This lesson is geared to display the differences, as well as provide more information about CSS IDs.

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

# What is an ID Selector?

The W3C defines class ID as "a unique identifier to an element". But what does that actually mean? Hopefully you have already read our [CSS Classes lesson](http://www.tizag.com/cssT/class.php), if not, we recommend that you do so.

CSS IDs are similar to classes in that they define a special case for an element. Below is an example of a CSS ID.

## CSS Code:

p#exampleID1 { background-color: white; }

p#exampleID2 { text-transform: uppercase; }

## HTML Code:

<p id="ExampleID1">This paragraph has an ID name of

"exampleID1" and has a white CSS defined background</p>

<p id="ExampleID2">This paragraph has an ID name of

"exampleID2" and has had its text transformed to uppercase letters. </p>

## Display:

This paragraph has an ID name of "exampleID1" and has a white CSS defined background.

This paragraph has an ID name of "exampleID2" and has had its text transformed to uppercase letters.

Notice that an ID's CSS is an HTML element, followed by a "#", and finally ID's name. The end result looks something like "element#idname". Also, be sure to absorb the fact that when an ID is used in HTML, we must use "id=name" instead of "class=name" to reference it!

# When to Use Classes?

A class can be used several times, while an ID can only be used once, so you should use classes for items that you know you're going to use a lot. An example would be if you wanted to give all the paragraphs on your webpage the same styling, you would use classes.

## CSS Code:

p.exampleID3 { background-color: #013370; color: white;}

## HTML Code:

<p class='exampleID3'>These paragraphs all have the same styling applied to them and we used classes because we wanted to reuse our styling!</p>

<p class='exampleID3'>These paragraphs all have the same styling applied to them and we used classes because we wanted to reuse our styling!</p>

<p class='exampleID3'>These paragraphs all have the same styling applied to them and we used classes because we wanted to reuse our styling!</p>

## Display:

These paragraphs all have the same styling applied to them and we used classes because we wanted to reuse our styling!

These paragraphs all have the same styling applied to them and we used classes because we wanted to reuse our styling!

These paragraphs all have the same styling applied to them and we used classes because we wanted to reuse our styling!

# Why Did They Choose Those Names??

* ID = A person's Identification (ID) is **unique** to one person.
* Class = There are **many** people in a class.

# ID for Layout and Uniqueness

Standards specify that any given ID name can only be referenced once within a page or document. From our experience, IDs are most commonly used correctly in CSS layouts. This makes sense because there are usually only one menu per page, one banner, and usually only one content pane.

In Tizag.com [CSS Layout Examples](http://www.tizag.com/cssT/liveExampleCss/) we have used IDs for the unique items mentioned above. [View the CSS Code](http://www.tizag.com/cssT/liveExampleCss/style1Css.php) for our first layout example. Below are the unique IDs in our code.

* Menu - div#menuPane
* Content - div#content

# Answer: Classes vs IDs

Use IDs when there is only one occurence per page. Use classes when there are one or more occurences per page.

# CSS Display

One of the most difficult aspects of creating a page without the use of tables is learning how to control the line breaks. Up to this point we haven't taught you how to use CSS to simulate a <br /> after the use of an element. Additionally, we have not shown how to remove line breaks that automatically occur with some elements (headers, list elements, paragraphs, etc).

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

All these issues and more are addressed with the display property. In this lesson you will find a brief overview and example for the most commonly used CSS Display values.

# Display Block and Inline

As you have seen in our CSS Examples, we were able to create many looks for our menus. A few of the examples have no line breaks in the HTML, but we made each anchor tag have a break on specific examples. These line breaks are created with the *block* value.

## CSS Code:

a { display: block; }

p { display: inline; }

## HTML Code:

<a href="http://www.tizag.com/" target="\_blank">Tizag.com - Learn to Whip the Web

</a>

...

<a href="http://www.tizag.com/" target="\_blank">Tizag.com - Learn to Whip the Web

</a>

<br />

<p>These paragraph </p>

<p>elements</p>

<p>have been </p>

<p>inlined.</p>

## Display:

[Tizag.com - Learn to Whip the Web](http://www.tizag.com/" \t "_blank) [Tizag.com - Learn to Whip the Web](http://www.tizag.com/" \t "_blank) [Tizag.com - Learn to Whip the Web](http://www.tizag.com/" \t "_blank) [Tizag.com - Learn to Whip the Web](http://www.tizag.com/" \t "_blank) [Tizag.com - Learn to Whip the Web](http://www.tizag.com/" \t "_blank)

These paragraph

elements

have been

inlined.

# Display None (Hidden)

At times you may want to hide pieces of content, while at other times you would wish to show it. With the use of JavaScript, you can create collapsible menus. This topic is beyond the scope of this lesson, but feel free to check out [O'Reilly's - Hierarchical Menus](http://www.oreillynet.com/pub/a/javascript/2002/02/22/hierarchical_menus.html" \t "_blank). Below is a simple example of how to hide an element.

## CSS Code:

p.show { display: block }

p.hide { display: none; }

## HTML Code:

<p class="show">This paragraph is displayed correctly because

it has a display value of "block".</p>

<p class="hide">This paragraph is hidden because

it has a display value of "none". Why am I even

writing anything in here?</p>

## Display:

This paragraph is displayed correctly because it has a display value of "block".

Using display correctly is key to CSS-heavy website designs and once you've mastered using it in your HTML your websites will be much more flexible than you ever imagined!