C1

What You Should Already Know  
Before you continue you should have a basic understanding of the following:  
  
HTML / XHTML  
If you want to study these subjects first, find the tutorials on our Home page.  
  
What is CSS?  
CSS stands for Cascading Style Sheets  
Styles define how to display HTML elements  
Styles were added to HTML 4.0 to solve a problem  
External Style Sheets can save a lot of work  
External Style Sheets are stored in CSS files  
CSS Demo  
An HTML document can be displayed with different styles: See how it works  
  
Styles Solved a Big Problem  
HTML was never intended to contain tags for formatting a document.  
  
HTML was intended to define the content of a document, like:  
  
<h1>This is a heading</h1>  
  
<p>This is a paragraph.</p>  
  
When tags like <font>, and color attributes were added to the HTML 3.2 specification, it started a nightmare for web developers. Development of large web sites, where fonts and color information were added to every single page, became a long and expensive process.  
  
To solve this problem, the World Wide Web Consortium (W3C) created CSS.  
  
In HTML 4.0, all formatting could be removed from the HTML document, and stored in a separate CSS file.  
  
All browsers support CSS today.  
  
CSS Saves a Lot of Work!  
CSS defines HOW HTML elements are to be displayed.  
  
Styles are normally saved in external .css files. External style sheets enable you to change the appearance and layout of all the pages in a Web site, just by editing one single file!

C2

CSS Syntax  
A CSS rule set consists of a selector and a declaration block:  
  
CSS selector  
  
The selector points to the HTML element you want to style.  
  
The declaration block contains one or more declarations separated by semicolons.  
  
Each declaration includes a property name and a value, separated by a colon.  
  
CSS Example  
A CSS declaration always ends with a semicolon, and declaration groups are surrounded by curly braces:  
  
p {color:red;text-align:center;}  
To make the CSS code more readable, you can put one declaration on each line, like this:  
  
Example  
  
p {  
 color:red;  
 text-align:center;  
}  
  
<!DOCTYPE html>  
<html>  
<head>  
<style>  
p {  
 color:red;  
 text-align:center;  
}  
</style>  
</head>  
<body>  
  
<p>Hello World!</p>  
<p>This paragraph is styled with CSS.</p>  
  
</body>  
</html>  
  
  
CSS Comments  
Comments are used to explain your code, and may help you when you edit the source code at a later date. Comments are ignored by browsers.  
  
A CSS comment starts with /\* and ends with \*/. Comments can also span multiple lines:  
  
Example  
  
p {  
    color: red;  
    /\* This is a single-line comment \*/  
    text-align: center;  
}  
  
/\* This is  
a multi-line  
comment \*/  
  
<!DOCTYPE html>  
<html>  
<head>  
<style>  
p {  
color: red;  
    /\* This is a single-line comment \*/  
text-align: center;  
}  
  
/\* This is  
a multi-line  
comment \*/  
</style>  
</head>  
<body>  
  
<p>Hello World!</p>  
<p>This paragraph is styled with CSS.</p>  
<p>CSS comments are not shown in the output.</p>  
  
</body>  
</html>

C3

CSS Selectors  
CSS selectors allow you to select and manipulate HTML element(s).  
  
CSS selectors are used to "find" (or select) HTML elements based on their id, classes, types, attributes, values of attributes and much more.  
  
The element Selector  
The element selector selects elements based on the element name.  
  
You can select all <p> elements on a page like this: (all <p> elements will be center-aligned, with a red text color)  
  
Example  
  
p {  
    text-align: center;  
    color: red;  
}  
  
<!DOCTYPE html>  
<html>  
<head>  
<style>  
p {  
    text-align: center;  
    color: red;  
}  
</style>  
</head>  
<body>  
  
<p>Every paragraph will be affected by the style.</p>  
<p id="para1">Me too!</p>  
<p>And me!</p>  
  
</body>  
</html>  
  
  
The id Selector  
The id selector uses the id attribute of an HTML tag to find the specific element.  
  
An id should be unique within a page, so you should use the id selector when you want to find a single, unique element.  
  
To find an element with a specific id, write a hash character, followed by the id of the element.  
  
The style rule below will be applied to the HTML element with id="para1":  
  
Example  
  
#para1 {  
    text-align: center;  
    color: red;  
}  
  
<!DOCTYPE html>  
<html>  
<head>  
<style>  
#para1 {  
    text-align: center;  
    color: red;  
}  
</style>  
</head>  
<body>  
  
<p id="para1">Hello World!</p>  
<p>This paragraph is not affected by the style.</p>  
  
</body>  
</html>  
  
  
Note    Do NOT start an ID name with a number!  
  
The class Selector  
The class selector finds elements with the specific class.  
  
The class selector uses the HTML class attribute.  
  
To find elements with a specific class, write a period character, followed by the name of the class:  
  
In the example below, all HTML elements with class="center" will be center-aligned:  
  
Example  
  
.center {  
    text-align: center;  
    color: red;  
}  
  
<!DOCTYPE html>  
<html>  
<head>  
<style>  
.center {  
    text-align: center;  
    color: red;  
}  
</style>  
</head>  
<body>  
  
<h1 class="center">Red and center-aligned heading</h1>  
<p class="center">Red and center-aligned paragraph.</p>  
  
</body>  
</html>  
  
You can also specify that only specific HTML elements should be affected by a class.  
  
In the example below, all p elements with class="center" will be center-aligned:  
  
Example  
  
p.center {  
    text-align:center;  
    color:red;  
}  
  
<!DOCTYPE html>  
<html>  
<head>  
<style>  
p.center {  
    text-align: center;  
    color: red;  
}  
</style>  
</head>  
<body>  
  
<h1 class="center">This heading will not be affected</h1>  
<p class="center">This paragraph will be red and center-aligned.</p>  
  
</body>  
</html>  
  
  
Note    Do NOT start a class name with a number!  
  
Grouping Selectors  
In style sheets there are often elements with the same style:  
  
h1 {  
    text-align: center;  
    color: red;  
}  
  
h2 {  
    text-align: center;  
    color: red;  
}  
  
p {  
    text-align: center;  
    color: red;  
}  
To minimize the code, you can group selectors.  
  
To group selectors, separate each selector with a comma.  
  
In the example below we have grouped the selectors from the code above:  
  
Example  
  
h1, h2, p {  
    text-align: center;  
    color: red;  
}  
  
<!DOCTYPE html>  
<html>  
<head>  
<style>  
h1, h2, p {  
    text-align: center;  
    color: red;  
}  
</style>  
</head>  
<body>  
  
<h1>Hello World!</h1>  
<h2>Smaller heading!</h2>  
<p>This is a paragraph.</p>  
  
</body>  
</html>

C4

When a browser reads a style sheet, it will format the document according to the information in the style sheet.  
  
Three Ways to Insert CSS  
There are three ways of inserting a style sheet:  
  
External style sheet  
Internal style sheet  
Inline style  
External Style Sheet  
An external style sheet is ideal when the style is applied to many pages. With an external style sheet, you can change the look of an entire Web site by changing just one file.  
  
Each page must include a link to the style sheet with the <link> tag. The <link> tag goes inside the head section:  
  
<head>  
<link rel="stylesheet" type="text/css" href="mystyle.css">  
</head>  
An external style sheet can be written in any text editor. The file should not contain any html tags. The style sheet file must be saved with a .css extension. An example of a style sheet file is shown below:  
  
"myStyle.css":  
  
hr {color: sienna;}  
p {margin-left: 20px;}  
body {background-image: url("images/background.gif");}  
  
Note    Do not add a space between the property value and the unit (such as margin-left: 20 px;). The correct way is: margin-left: 20px;  
  
Internal Style Sheet  
An internal style sheet should be used when a single document has a unique style. You define internal styles in the head section of an HTML page, inside the <style> tag, like this:  
  
<head>  
<style>  
hr {color: sienna;}  
p {margin-left: 20px;}  
body {background-image: url("images/background.gif");}  
</style>  
</head>  
  
Inline Styles  
An inline style loses many of the advantages of a style sheet (by mixing content with presentation). Use this method sparingly!  
  
To use inline styles, add the style attribute to the relevant tag. The style attribute can contain any CSS property. The example shows how to change the color and the left margin of a paragraph:  
  
<p style="color:sienna;margin-left:20px;">This is a paragraph.</p>  
  
Multiple Style Sheets  
If some properties have been set for the same selector in different style sheets, the values will be inherited from the more specific style sheet.  
  
For example, assume that an external style sheet has the following properties for the h3 selector:  
  
h3 {  
    color: red;  
    text-align: left;  
    font-size: 8pt;  
}  
then, assume that an internal style sheet also has the following properties for the h3 selector:  
  
h3 {  
    text-align: right;  
    font-size: 20pt;  
}  
If the page with the internal style sheet also links to the external style sheet the properties for the h3 element will be:  
  
color: red;  
text-align: right;  
font-size: 20pt;  
The color is inherited from the external style sheet and the text-alignment and the font-size is replaced by the internal style sheet.  
  
Multiple Styles Will Cascade into One  
Styles can be specified:  
  
inside an HTML element  
inside the head section of an HTML page  
in an external CSS file  
Tip: Even multiple external style sheets can be referenced inside a single HTML document.  
  
Cascading order  
  
What style will be used when there is more than one style specified for an HTML element?  
  
Generally speaking we can say that all the styles will "cascade" into a new "virtual" style sheet by the following rules, where number four has the highest priority:  
  
Browser default  
External style sheet  
Internal style sheet (in the head section)  
Inline style (inside an HTML element)  
So, an inline style (inside an HTML element) has the highest priority, which means that it will override a style defined inside the <head> tag, or in an external style sheet, or in a browser (a default value).  
  
Note    Note: If the link to the external style sheet is placed after the internal style sheet in HTML <head>, the external style sheet will override the internal style sheet!

C5

CSS background properties are used to define the background effects of an element.  
  
CSS properties used for background effects:  
  
background-color  
background-image  
background-repeat  
background-attachment  
background-position  
  
Background Color  
The background-color property specifies the background color of an element.  
  
The background color of a page is defined in the body selector:  
  
Example  
  
body {  
    background-color: #b0c4de;  
}  
  
<!DOCTYPE html>  
<html>  
<head>  
<style>  
body {  
    background-color: #b0c4de;  
}  
</style>  
</head>  
<body>  
  
<h1>My CSS web page!</h1>  
<p>Hello world! This is a W3Schools.com example.</p>  
  
</body>  
</html>  
  
  
With CSS, a color is most often specified by:  
  
a HEX value - like "#ff0000"  
an RGB value - like "rgb(255,0,0)"  
a color name - like "red"  
Look at CSS Color Values for a complete list of possible color values.  
  
In the example below, the h1, p, and div elements have different background colors:  
  
Example  
  
h1 {  
    background-color: #6495ed;  
}  
  
p {  
    background-color: #e0ffff;  
}  
  
div {  
    background-color: #b0c4de;  
}  
  
<!DOCTYPE html>  
<html>  
<head>  
<style>  
h1 {  
    background-color: #6495ed;  
}  
  
p {  
    background-color: #e0ffff;  
}  
  
div {  
    background-color: #b0c4de;  
}  
</style>  
</head>  
<body>  
  
<h1>CSS background-color example!</h1>  
<div>  
This is a text inside a div element.  
<p>This paragraph has its own background color.</p>  
We are still in the div element.  
</div>  
  
</body>  
</html>  
  
  
Background Image  
The background-image property specifies an image to use as the background of an element.  
  
By default, the image is repeated so it covers the entire element.  
  
The background image for a page can be set like this:  
  
Example  
  
body {  
    background-image: url("paper.gif");  
}  
  
<!DOCTYPE html>  
<html>  
<head>  
<style>  
body {  
    background-image: url("paper.gif");  
}  
</style>  
</head>  
<body>  
  
<h1>Hello World!</h1>  
  
</body>  
</html>  
  
Below is an example of a bad combination of text and background image. The text is almost not readable:  
  
Example  
  
body {  
    background-image: url("bgdesert.jpg");  
}  
  
<!DOCTYPE html>  
<html>  
<head>  
<style>  
body {  
    background-image: url("bgdesert.jpg");  
}  
</style>  
</head>  
<body>  
  
<h1>Hello World!</h1>  
<p>This text is not easy to read on this background image.</p>  
  
</body>  
</html>  
  
  
Background Image - Repeat Horizontally or Vertically  
By default, the background-image property repeats an image both horizontally and vertically.  
  
Some images should be repeated only horizontally or vertically, or they will look strange, like this:  
  
Example  
  
body {  
    background-image: url("gradient.png");  
}  
  
<!DOCTYPE html>  
<html>  
<head>  
<style>  
body {  
    background-image: url("gradient.png");  
}  
</style>  
</head>  
<body>  
  
<h1>Hello World!</h1>  
  
</body>  
</html>  
  
If the image is repeated only horizontally (repeat-x), the background will look better:  
  
Example  
  
body {  
    background-image: url("gradient.png");  
    background-repeat: repeat-x;  
}  
  
<!DOCTYPE html>  
<html>  
<head>  
<style>  
body {  
    background-image: url("gradient.png");  
    background-repeat: repeat-x;  
}  
</style>  
</head>  
<body>  
  
<h1>Hello World!</h1>  
  
</body>  
</html>  
  
  
Background Image - Set position and no-repeat  
Note    Note: When using a background image, use an image that does not disturb the text.  
Showing the image only once is specified by the background-repeat property:  
  
Example  
  
body {  
    background-image: url("img\_tree.png");  
    background-repeat: no-repeat;  
}  
  
<!DOCTYPE html>  
<html>  
<head>  
<style>  
body {  
    background-image: url("img\_tree.png");  
    background-repeat: no-repeat;  
}  
</style>  
</head>  
<body>  
  
<h1>Hello World!</h1>  
<p>W3Schools background image example.</p>  
<p>The background image is only showing once, but it is disturbing the reader!</p>  
  
</body>  
</html>  
  
In the example above, the background image is shown in the same place as the text. We want to change the position of the image, so that it does not disturb the text too much.  
  
The position of the image is specified by the background-position property:  
  
Example  
  
body {  
    background-image: url("img\_tree.png");  
    background-repeat: no-repeat;  
    background-position: right top;  
}  
  
<!DOCTYPE html>  
<html>  
<head>  
<style>  
body {  
    background-image: url("img\_tree.png");  
    background-repeat: no-repeat;  
    background-position: right top;  
    margin-right: 200px;  
}  
</style>  
</head>  
<body>  
  
<h1>Hello World!</h1>  
<p>W3Schools background no-repeat, set position example.</p>  
<p>Now the background image is only shown once, and positioned away from the text.</p>  
<p>In this example we have also added a margin on the right side, so the background image will never disturb the text.</p>  
  
</body>  
</html>  
  
  
Background - Shorthand property  
As you can see from the examples above, there are many properties to consider when dealing with backgrounds.  
  
To shorten the code, it is also possible to specify all the properties in one single property. This is called a shorthand property.  
  
The shorthand property for background is simply "background":  
  
Example  
  
body {  
    background: #ffffff url("img\_tree.png") no-repeat right top;  
}  
  
<!DOCTYPE html>  
<html>  
<head>  
<style>  
body {  
    background: #ffffff url("img\_tree.png") no-repeat right top;  
    margin-right: 200px;  
}  
</style>  
</head>  
<body>  
  
<h1>Hello World!</h1>  
<p>Now the background image is only shown once, and it is also positioned away from the text.</p>  
<p>In this example we have also added a margin on the right side, so that the background image will not disturb the text.</p>  
  
</body>  
</html>  
  
  
When using the shorthand property the order of the property values is:  
  
background-color  
background-image  
background-repeat  
background-attachment  
background-position  
It does not matter if one of the property values is missing, as long as the ones that are present are in this order.  
  
This example uses more advanced CSS. Take a look: Advanced example  
  
Examples  
More Examples  
How to set a fixed background image  
<!DOCTYPE html>  
<html>  
<head>  
<style>  
body {  
    background-image: url("w3css.gif");  
    background-repeat: no-repeat;  
    background-attachment: fixed;  
}  
</style>  
</head>  
  
<body>  
<p>The background-image is fixed. Try to scroll down the page.</p>  
<p>The background-image is fixed. Try to scroll down the page.</p>  
<p>The background-image is fixed. Try to scroll down the page.</p>  
<p>The background-image is fixed. Try to scroll down the page.</p>  
<p>The background-image is fixed. Try to scroll down the page.</p>  
<p>The background-image is fixed. Try to scroll down the page.</p>  
<p>The background-image is fixed. Try to scroll down the page.</p>  
<p>The background-image is fixed. Try to scroll down the page.</p>  
<p>The background-image is fixed. Try to scroll down the page.</p>  
<p>The background-image is fixed. Try to scroll down the page.</p>  
<p>The background-image is fixed. Try to scroll down the page.</p>  
<p>The background-image is fixed. Try to scroll down the page.</p>  
<p>The background-image is fixed. Try to scroll down the page.</p>  
<p>The background-image is fixed. Try to scroll down the page.</p>  
<p>The background-image is fixed. Try to scroll down the page.</p>  
<p>If you do not see any scrollbars, try to resize the browser window.</p>  
</body>  
  
</html>  
  
This example demonstrates how to set a fixed background image. The image will not scroll with the rest of the page.  
  
All CSS Background Properties  
Property    Description  
background    Sets all the background properties in one declaration  
background-attachment    Sets whether a background image is fixed or scrolls with the rest of the page  
background-color    Sets the background color of an element  
background-image    Sets the background image for an element  
background-position    Sets the starting position of a background image  
background-repeat    Sets how a background image will be repeated

C6

Text Color  
The color property is used to set the color of the text.  
  
With CSS, a color is most often specified by:  
  
a HEX value - like "#ff0000"  
an RGB value - like "rgb(255,0,0)"  
a color name - like "red"  
Look at CSS Color Values for a complete list of possible color values.  
  
The default color for a page is defined in the body selector.  
  
Example  
  
body {  
    color: blue;  
}  
  
h1 {  
    color: #00ff00;  
}  
  
h2 {  
    color: rgb(255,0,0);  
}  
  
<!DOCTYPE html>  
<html>  
<head>  
<style>  
body {  
    color: red;  
}  
  
h1 {  
    color: #00ff00;  
}  
  
p.ex {  
    color: rgb(0,0,255);  
}  
</style>  
</head>  
<body>  
  
<h1>This is heading 1</h1>  
<p>This is an ordinary paragraph. Notice that this text is red. The default text-color for a page is defined in the body selector.</p>  
<p class="ex">This is a paragraph with class="ex". This text is blue.</p>  
  
</body>  
</html>  
  
  
Note    Note: For W3C compliant CSS: If you define the color property, you must also define the background-color property.  
  
Text Alignment  
The text-align property is used to set the horizontal alignment of a text.  
  
Text can be centered, or aligned to the left or right, or justified.  
  
When text-align is set to "justify", each line is stretched so that every line has equal width, and the left and right margins are straight (like in magazines and newspapers).  
  
Example  
  
h1 {  
    text-align: center;  
}  
  
p.date {  
    text-align: right;  
}  
  
p.main {  
    text-align: justify;  
}  
  
<!DOCTYPE html>  
<html>  
<head>  
<style>  
h1 {  
    text-align: center;  
}  
  
p.date {  
    text-align: right;  
}  
  
p.main {  
    text-align: justify;  
}  
</style>  
</head>  
<body>  
  
<h1>CSS text-align Example</h1>  
<p class="date">May, 2009</p>  
<p class="main">In my younger and more vulnerable years my father gave me some advice that I've been turning over in my mind ever since. 'Whenever you feel like criticizing anyone,' he told me,  
'just remember that all the people in this world haven't had the advantages that you've had.'</p>  
<p><b>Note:</b> Resize the browser window to see how the value "justify" works.</p>  
  
</body>  
</html>  
  
  
Text Decoration  
The text-decoration property is used to set or remove decorations from text.  
  
The text-decoration property is mostly used to remove underlines from links for design purposes:  
  
Example  
  
a {  
    text-decoration: none;  
}  
  
<!DOCTYPE html>  
<html>  
<head>  
<style>  
a {  
    text-decoration: none;  
}  
</style>  
</head>  
<body>  
  
<p>Link to: <a href="[http://www.w3schools.com](http://www.w3schools.com/)">W3Schools.com</a></p>  
  
</body>  
</html>  
  
It can also be used to decorate text:  
  
Example  
  
h1 {  
    text-decoration: overline;  
}  
  
h2 {  
    text-decoration: line-through;  
}  
  
h3 {  
    text-decoration: underline;  
}  
  
<!DOCTYPE html>  
<html>  
<head>  
<style>  
h1 {  
    text-decoration: overline;  
}  
  
h2 {  
    text-decoration: line-through;  
}  
  
h3 {  
    text-decoration: underline;  
}  
</style>  
</head>  
<body>  
  
<h1>This is heading 1</h1>  
<h2>This is heading 2</h2>  
<h3>This is heading 3</h3>  
  
</body>  
</html>  
  
  
Note    Note: It is not recommended to underline text that is not a link, as this often confuses users.  
  
Text Transformation  
The text-transform property is used to specify uppercase and lowercase letters in a text.  
  
It can be used to turn everything into uppercase or lowercase letters, or capitalize the first letter of each word.  
  
Example  
  
p.uppercase {  
    text-transform: uppercase;  
}  
  
p.lowercase {  
    text-transform: lowercase;  
}  
  
p.capitalize {  
    text-transform: capitalize;  
}  
  
<!DOCTYPE html>  
<html>  
<head>  
<style>  
p.uppercase {  
    text-transform: uppercase;  
}  
  
p.lowercase {  
    text-transform: lowercase;  
}  
  
p.capitalize {  
    text-transform: capitalize;  
}  
</style>  
</head>  
<body>  
  
<p class="uppercase">This is some text.</p>  
<p class="lowercase">This is some text.</p>  
<p class="capitalize">This is some text.</p>  
  
</body>  
</html>  
  
  
Text Indentation  
The text-indent property is used to specify the indentation of the first line of a text.  
  
Example  
  
p {  
    text-indent: 50px;  
}  
  
<!DOCTYPE html>  
<html>  
<head>  
<style>  
p {  
    text-indent: 50px;  
}  
</style>  
</head>  
<body>  
  
<p>In my younger and more vulnerable years my father gave me some advice that I've been turning over in my mind ever since. 'Whenever you feel like criticizing anyone,' he told me, 'just remember that all the people in this world haven't had the advantages that you've had.'</p>  
  
</body>  
</html>  
  
  
Examples  
More Examples  
Specify the space between characters  
<!DOCTYPE html>  
<html>  
<head>  
<style>  
h1 {  
    letter-spacing: 3px;  
}  
  
h2 {  
    letter-spacing: -3px;  
}  
</style>  
</head>  
<body>  
  
<h1>This is heading 1</h1>  
<h2>This is heading 2</h2>  
  
</body>  
</html>  
  
This example demonstrates how to increase or decrease the space between characters.  
  
Specify the space between lines  
<!DOCTYPE html>  
<html>  
<head>  
<style>  
p.small {  
    line-height: 70%;  
}  
  
p.big {  
    line-height: 200%;  
}  
</style>  
</head>  
<body>  
  
<p>  
This is a paragraph with a standard line-height.<br>  
The default line height in most browsers is about 110% to 120%.<br>  
</p>  
  
<p class="small">  
This is a paragraph with a smaller line-height.<br>  
This is a paragraph with a smaller line-height.<br>  
</p>  
  
<p class="big">  
This is a paragraph with a bigger line-height.<br>  
This is a paragraph with a bigger line-height.<br>  
</p>  
  
</body>  
</html>  
  
This example demonstrates how to specify the space between the lines in a paragraph.  
  
Set the text direction of an element  
<!DOCTYPE html>  
<html>  
<head>  
<style>  
div.ex1 {  
    direction: rtl;  
    unicode-bidi: bidi-override;  
}  
</style>  
</head>  
<body>  
  
<div>This is default text direction.</div>  
<div class="ex1">This is right-to-left text direction.</div>  
  
</body>  
</html>  
  
This example demonstrates how to change the text direction of an element.  
  
Increase the white space between words  
<!DOCTYPE html>  
<html>  
<head>  
<style>  
p {  
    word-spacing: 30px;  
}  
</style>  
</head>  
<body>  
  
<p>This is some text in a paragraph.</p>  
  
</body>  
</html>  
  
This example demonstrates how to increase the white space between words in a paragraph.  
  
Disable text wrapping inside an element  
<!DOCTYPE html>  
<html>  
<head>  
<style>  
p {  
    white-space: nowrap;  
}  
</style>  
</head>  
<body>  
  
<p>  
This is some text. This is some text. This is some text.  
This is some text. This is some text. This is some text.  
This is some text. This is some text. This is some text.  
This is some text. This is some text. This is some text.  
</p>  
  
</body>  
</html>  
  
This example demonstrates how to disable text wrapping inside an element.  
  
Vertical alignment of an image  
<!DOCTYPE html>  
<html>  
<head>  
<style>  
img.top {  
    vertical-align: text-top;  
}  
  
img.bottom {  
    vertical-align: text-bottom;  
}  
</style>  
</head>  
<body>  
  
<p>An <img src="w3schools\_logo.gif" alt="W3Schools" width="270" height="50"> image with a default alignment.</p>  
<p>An <img class="top" src="w3schools\_logo.gif" alt="W3Schools" width="270" height="50"> image with a text-top alignment.</p>  
<p>An <img class="bottom" src="w3schools\_logo.gif" alt="W3Schools" width="270" height="50"> image with a text-bottom alignment.</p>  
  
</body>  
</html>  
  
This example demonstrates how to set the vertical align of an image in a text.  
  
Add shadow to text  
<!DOCTYPE html>  
<html>  
<head>  
<style>  
h1 {  
    text-shadow: 2px 2px #FF0000;  
}  
</style>  
</head>  
<body>  
  
<h1>Text-shadow effect</h1>  
<p><b>Note:</b> Internet Explorer 9 and earlier do not support the text-shadow property.</p>  
  
</body>  
</html>  
  
This example demonstrates how to add shadow to text.  
  
All CSS Text Properties  
Property    Description  
color    Sets the color of text  
direction    Specifies the text direction/writing direction  
letter-spacing    Increases or decreases the space between characters in a text  
line-height    Sets the line height  
text-align    Specifies the horizontal alignment of text  
text-decoration    Specifies the decoration added to text  
text-indent    Specifies the indentation of the first line in a text-block  
text-shadow    Specifies the shadow effect added to text  
text-transform    Controls the capitalization of text  
unicode-bidi    Used together with the direction property to set or return whether the text should be overridden to support multiple languages in the same document  
vertical-align    Sets the vertical alignment of an element  
white-space    Specifies how white-space inside an element is handled  
word-spacing    Increases or decreases the space between words in a tekst

C7

CSS Font Families  
In CSS, there are two types of font family names:  
  
generic family - a group of font families with a similar look (like "Serif" or "Monospace")  
font family - a specific font family (like "Times New Roman" or "Arial")  
Generic family    Font family    Description  
Serif    Times New Roman  
Georgia    Serif fonts have small lines at the ends on some characters  
Sans-serif    Arial  
Verdana    "Sans" means without - these fonts do not have the lines at the ends of characters  
Monospace    Courier New  
Lucida Console    All monospace characters have the same width  
  
Note    Note: On computer screens, sans-serif fonts are considered easier to read than serif fonts.  
  
Font Family  
The font family of a text is set with the font-family property.  
  
The font-family property should hold several font names as a "fallback" system. If the browser does not support the first font, it tries the next font.  
  
Start with the font you want, and end with a generic family, to let the browser pick a similar font in the generic family, if no other fonts are available.  
  
Note: If the name of a font family is more than one word, it must be in quotation marks, like: "Times New Roman".  
  
More than one font family is specified in a comma-separated list:  
  
Example  
  
p {  
    font-family: "Times New Roman", Times, serif;  
}  
  
<!DOCTYPE html>  
<html>  
<head>  
<style>  
p.serif {  
    font-family: "Times New Roman", Times, serif;  
}  
  
p.sansserif {  
    font-family: Arial, Helvetica, sans-serif;  
}  
</style>  
</head>  
<body>  
  
<h1>CSS font-family</h1>  
<p class="serif">This is a paragraph, shown in the Times New Roman font.</p>  
<p class="sansserif">This is a paragraph, shown in the Arial font.</p>  
  
</body>  
</html>  
  
For more commonly used font combinations, look at our Web Safe Font Combinations.  
  
Font Style  
The font-style property is mostly used to specify italic text.  
  
This property has three values:  
  
normal - The text is shown normally  
italic - The text is shown in italics  
oblique - The text is "leaning" (oblique is very similar to italic, but less supported)  
Example  
  
p.normal {  
    font-style: normal;  
}  
  
p.italic {  
    font-style: italic;  
}  
  
p.oblique {  
    font-style: oblique;  
}  
  
<!DOCTYPE html>  
<html>  
<head>  
<style>  
p.normal {  
    font-style: normal;  
}  
  
p.italic {  
    font-style: italic;  
}  
  
p.oblique {  
    font-style: oblique;  
}  
</style>  
</head>  
<body>  
  
<p class="normal">This is a paragraph in normal style.</p>  
<p class="italic">This is a paragraph in italic style.</p>  
<p class="oblique">This is a paragraph in oblique style.</p>  
  
</body>  
</html>  
  
  
Font Size  
The font-size property sets the size of the text.  
  
Being able to manage the text size is important in web design. However, you should not use font size adjustments to make paragraphs look like headings, or headings look like paragraphs.  
  
Always use the proper HTML tags, like <h1> - <h6> for headings and <p> for paragraphs.  
  
The font-size value can be an absolute, or relative size.  
  
Absolute size:  
  
Sets the text to a specified size  
Does not allow a user to change the text size in all browsers (bad for accessibility reasons)  
Absolute size is useful when the physical size of the output is known  
Relative size:  
  
Sets the size relative to surrounding elements  
Allows a user to change the text size in browsers  
  
Note    Note: If you do not specify a font size, the default size for normal text, like paragraphs, is 16px (16px=1em).  
  
Set Font Size With Pixels  
Setting the text size with pixels gives you full control over the text size:  
  
Example  
  
h1 {  
    font-size: 40px;  
}  
  
h2 {  
    font-size: 30px;  
}  
  
p {  
    font-size: 14px;  
}  
  
<!DOCTYPE html>  
<html>  
<head>  
<style>  
h1 {  
    font-size: 40px;  
}  
  
h2 {  
    font-size: 30px;  
}  
  
p {  
    font-size: 14px;  
}  
</style>  
</head>  
<body>  
  
<h1>This is heading 1</h1>  
<h2>This is heading 2</h2>  
<p>This is a paragraph.</p>  
<p>Specifying the font-size in px allows Internet Explorer 9, Firefox, Chrome, Opera, and Safari to resize the text.</p>  
<p><b>Note:</b> This example does not work in IE, prior version 9.</p>  
  
</body>  
</html>  
  
The example above allows Internet Explorer 9, Firefox, Chrome, Opera, and Safari to resize the text.  
  
Note: The example above does not work in IE, prior version 9.  
  
The text can be resized in all browsers using the zoom tool (however, this resizes the entire page, not just the text).  
  
Set Font Size With Em  
To avoid the resizing problem with older versions of Internet Explorer, many developers use em instead of pixels.  
  
The em size unit is recommended by the W3C.  
  
1em is equal to the current font size. The default text size in browsers is 16px. So, the default size of 1em is 16px.  
  
The size can be calculated from pixels to em using this formula: pixels/16=em  
  
Example  
  
h1 {  
    font-size: 2.5em; /\* 40px/16=2.5em \*/  
}  
  
h2 {  
    font-size: 1.875em; /\* 30px/16=1.875em \*/  
}  
  
p {  
    font-size: 0.875em; /\* 14px/16=0.875em \*/  
}  
  
<!DOCTYPE html>  
<html>  
<head>  
<style>  
h1 {  
    font-size: 2.5em; /\* 40px/16=2.5em \*/  
}  
  
h2 {  
    font-size: 1.875em; /\* 30px/16=1.875em \*/  
 }  
  
p {  
    font-size: 0.875em; /\* 14px/16=0.875em \*/  
}  
</style>  
</head>  
<body>  
  
<h1>This is heading 1</h1>  
<h2>This is heading 2</h2>  
<p>This is a paragraph.</p>  
<p>Specifying the font-size in em allows all major browsers to resize the text.  
Unfortunately, there is still a problem with older versions of IE. When resizing the text, it becomes larger/smaller than it should.</p>  
  
</body>  
</html>  
  
In the example above, the text size in em is the same as the previous example in pixels. However, with the em size, it is possible to adjust the text size in all browsers.  
  
Unfortunately, there is still a problem with older versions of IE. The text becomes larger than it should when made larger, and smaller than it should when made smaller.  
  
Use a Combination of Percent and Em  
The solution that works in all browsers, is to set a default font-size in percent for the <body> element:  
  
Example  
  
body {  
    font-size: 100%;  
}  
  
h1 {  
    font-size: 2.5em;  
}  
  
h2 {  
    font-size: 1.875em;  
}  
  
p {  
    font-size: 0.875em;  
}  
  
<!DOCTYPE html>  
<html>  
<head>  
<style>  
body {  
    font-size: 100%;  
}  
  
h1 {  
    font-size: 2.5em;  
}  
  
h2 {  
    font-size: 1.875em;  
}  
  
p {  
    font-size: 0.875em;  
}  
</style>  
</head>  
<body>  
  
<h1>This is heading 1</h1>  
<h2>This is heading 2</h2>  
<p>This is a paragraph.</p>  
<p>Specifying the font-size in percent and em displays the same size in all major browsers, and allows all browsers to resize the text!</p>  
  
</body>  
</html>  
  
Our code now works great! It shows the same text size in all browsers, and allows all browsers to zoom or resize the text!  
  
Examples  
More Examples  
Set the boldness of the font  
<!DOCTYPE html>  
<html>  
<head>  
<style>  
p.normal {  
    font-weight: normal;  
}  
  
p.light {  
    font-weight: lighter;  
}  
  
p.thick {  
    font-weight: bold;  
}  
  
p.thicker {  
    font-weight: 900;  
}  
</style>  
</head>  
<body>  
  
<p class="normal">This is a paragraph.</p>  
<p class="light">This is a paragraph.</p>  
<p class="thick">This is a paragraph.</p>  
<p class="thicker">This is a paragraph.</p>  
  
</body>  
</html>  
  
This example demonstrates how to set the boldness of a font.  
  
Set the variant of the font  
<!DOCTYPE html>  
<html>  
<head>  
<style>  
p.normal {  
    font-variant: normal;  
}  
  
p.small {  
    font-variant: small-caps;  
}  
</style>  
</head>  
<body>  
  
<p class="normal">My name is Hege Refsnes.</p>  
<p class="small">My name is Hege Refsnes.</p>  
  
</body>  
</html>  
  
This example demonstrates how to set the variant of a font.  
  
All the font properties in one declaration  
<!DOCTYPE html>  
<html>  
<head>  
<style>  
p.ex1 {  
    font: 15px arial, sans-serif;  
}  
  
p.ex2 {  
    font:italic bold 12px/30px Georgia, serif;  
}  
</style>  
</head>  
<body>  
  
<p class="ex1">This is a paragraph. This is a paragraph. This is a paragraph. This is a paragraph. This is a paragraph. This is a paragraph. This is a paragraph. This is a paragraph.</p>  
  
<p class="ex2">This is a paragraph. This is a paragraph. This is a paragraph. This is a paragraph. This is a paragraph. This is a paragraph. This is a paragraph. This is a paragraph.</p>  
  
</body>  
</html>  
  
This example demonstrates how to use the shorthand property for setting all of the font properties in one declaration.  
  
All CSS Font Properties  
Property    Description  
font    Sets all the font properties in one declaration  
font-family    Specifies the font family for text  
font-size    Specifies the font size of text  
font-style    Specifies the font style for text  
font-variant    Specifies whether or not a text should be displayed in a small-caps font  
font-weight    Specifies the weight of a font

C8

Links can be styled in different ways.  
  
Styling Links  
Links can be styled with any CSS property (e.g. color, font-family, background, etc.).  
  
In addition, links can be styled differently depending on what state they are in.  
  
The four links states are:  
  
a:link - a normal, unvisited link  
a:visited - a link the user has visited  
a:hover - a link when the user mouses over it  
a:active - a link the moment it is clicked  
Example  
  
/\* unvisited link \*/  
a:link {  
    color: #FF0000;  
}  
  
/\* visited link \*/  
a:visited {  
    color: #00FF00;  
}  
  
/\* mouse over link \*/  
a:hover {  
    color: #FF00FF;  
}  
  
/\* selected link \*/  
a:active {  
    color: #0000FF;  
}  
  
<!DOCTYPE html>  
<html>  
<head>  
<style>  
/\* unvisited link \*/  
a:link {  
    color: #FF0000;  
}  
  
/\* visited link \*/  
a:visited {  
    color: #00FF00;  
}  
  
/\* mouse over link \*/  
a:hover {  
    color: #FF00FF;  
}  
  
/\* selected link \*/  
a:active {  
    color: #0000FF;  
}  
</style>  
</head>  
<body>  
  
<p><b><a href="default.asp" target="\_blank">This is a link</a></b></p>  
<p><b>Note:</b> a:hover MUST come after a:link and a:visited in the CSS definition in order to be effective.</p>  
<p><b>Note:</b> a:active MUST come after a:hover in the CSS definition in order to be effective.</p>  
  
</body>  
</html>  
  
When setting the style for several link states, there are some order rules:  
  
a:hover MUST come after a:link and a:visited  
a:active MUST come after a:hover  
Common Link Styles  
In the example above the link changes color depending on what state it is in.  
  
Lets go through some of the other common ways to style links:  
  
Text Decoration  
The text-decoration property is mostly used to remove underlines from links:  
  
Example  
  
a:link {  
    text-decoration: none;  
}  
  
a:visited {  
    text-decoration: none;  
}  
  
a:hover {  
    text-decoration: underline;  
}  
  
a:active {  
    text-decoration: underline;  
}  
  
<!DOCTYPE html>  
<html>  
<head>  
<style>  
a:link {  
    text-decoration: none;  
}  
  
a:visited {  
    text-decoration: none;  
}  
  
a:hover {  
    text-decoration: underline;  
}  
  
a:active {  
    text-decoration: underline;  
}  
</style>  
</head>  
<body>  
  
<p><b><a href="default.asp" target="\_blank">This is a link</a></b></p>  
<p><b>Note:</b> a:hover MUST come after a:link and a:visited in the CSS definition in order to be effective.</p>  
<p><b>Note:</b> a:active MUST come after a:hover in the CSS definition in order to be effective.</p>  
  
</body>  
</html>  
  
  
Background Color  
The background-color property specifies the background color for links:  
  
Example  
  
a:link {  
    background-color: #B2FF99;  
}  
  
a:visited {  
    background-color: #FFFF85;  
}  
  
a:hover {  
    background-color: #FF704D;  
}  
  
a:active {  
    background-color: #FF704D;  
}  
  
<!DOCTYPE html>  
<html>  
<head>  
<style>  
a:link {  
    background-color: #B2FF99;  
}  
  
a:visited {  
    background-color: #FFFF85;  
}  
  
a:hover {  
    background-color: #FF704D;  
}  
  
a:active {  
    background-color: #FF704D;  
}  
</style>  
</head>  
<body>  
  
<p><b><a href="default.asp" target="\_blank">This is a link</a></b></p>  
<p><b>Note:</b> a:hover MUST come after a:link and a:visited in the CSS definition in order to be effective.</p>  
<p><b>Note:</b> a:active MUST come after a:hover in the CSS definition in order to be effective.</p>  
  
</body>  
</html>  
  
  
Examples  
More Examples  
Add different styles to hyperlinks  
<!DOCTYPE html>  
<html>  
<head>  
<style>  
a.one:link {color:#ff0000;}  
a.one:visited {color:#0000ff;}  
a.one:hover {color:#ffcc00;}  
  
a.two:link {color:#ff0000;}  
a.two:visited {color:#0000ff;}  
a.two:hover {font-size:150%;}  
  
a.three:link {color:#ff0000;}  
a.three:visited {color:#0000ff;}  
a.three:hover {background:#66ff66;}  
  
a.four:link {color:#ff0000;}  
a.four:visited {color:#0000ff;}  
a.four:hover {font-family:monospace;}  
  
a.five:link {color:#ff0000;text-decoration:none;}  
a.five:visited {color:#0000ff;text-decoration:none;}  
a.five:hover {text-decoration:underline;}  
</style>  
</head>  
<body>  
  
<p>Mouse over the links and watch them change layout:</p>  
  
<p><b><a class="one" href="default.asp" target="\_blank">This link changes color</a></b></p>  
<p><b><a class="two" href="default.asp" target="\_blank">This link changes font-size</a></b></p>  
<p><b><a class="three" href="default.asp" target="\_blank">This link changes background-color</a></b></p>  
<p><b><a class="four" href="default.asp" target="\_blank">This link changes font-family</a></b></p>  
<p><b><a class="five" href="default.asp" target="\_blank">This link changes text-decoration</a></b></p>  
  
</body>  
</html>  
  
This example demonstrates how to add other styles to hyperlinks.  
  
Advanced - Create link boxes  
<!DOCTYPE html>  
<html>  
<head>  
<style>  
a:link, a:visited {  
    display: block;  
    font-weight: bold;  
    color: #ffffff;  
    background-color: #98bf21;  
    width: 120px;  
    text-align: center;  
    padding: 4px;  
    text-decoration: none;  
}  
  
a:hover, a:active {  
    background-color: #7A991A;  
}  
</style>  
</head>  
<body>  
  
<a href="default.asp" target="\_blank">This is a link</a>  
  
</body>  
</html>  
  
This example demonstrates a more advanced example where we combine several CSS properties to display links as boxes.

C9

The CSS list properties allow you to:  
  
Set different list item markers for ordered lists  
Set different list item markers for unordered lists  
Set an image as the list item marker  
  
List  
In HTML, there are two types of lists:  
  
unordered lists - the list items are marked with bullets  
ordered lists - the list items are marked with numbers or letters  
With CSS, lists can be styled further, and images can be used as the list item marker.  
  
Different List Item Markers  
The type of list item marker is specified with the list-style-type property:  
  
Example  
  
ul.a {  
    list-style-type: circle;  
}  
  
ul.b {  
    list-style-type: square;  
}  
  
ol.c {  
    list-style-type: upper-roman;  
}  
  
ol.d {  
    list-style-type: lower-alpha;  
}  
  
<!DOCTYPE html>  
<html>  
<head>  
<style>  
ul.a {  
    list-style-type: circle;  
}  
  
ul.b {  
    list-style-type: square;  
}  
  
ol.c {  
    list-style-type: upper-roman;  
}  
  
ol.d {  
    list-style-type: lower-alpha;  
}  
</style>  
</head>  
<body>  
  
<p>Example of unordered lists:</p>  
<ul class="a">  
  <li>Coffee</li>  
  <li>Tea</li>  
  <li>Coca Cola</li>  
</ul>  
  
<ul class="b">  
  <li>Coffee</li>  
  <li>Tea</li>  
  <li>Coca Cola</li>  
</ul>  
  
<p>Example of ordered lists:</p>  
<ol class="c">  
  <li>Coffee</li>  
  <li>Tea</li>  
  <li>Coca Cola</li>  
</ol>  
  
<ol class="d">  
  <li>Coffee</li>  
  <li>Tea</li>  
  <li>Coca Cola</li>  
</ol>  
  
</body>  
</html>  
  
Some of the values are for unordered lists, and some for ordered lists.  
  
An Image as The List Item Marker  
To specify an image as the list item marker, use the list-style-image property:  
  
Example  
  
ul {  
   list-style-image: url('sqpurple.gif');  
}  
  
<!DOCTYPE html>  
<html>  
<head>  
<style>  
ul {  
    list-style-image: url('sqpurple.gif');  
}  
</style>  
</head>  
<body>  
  
<ul>  
  <li>Coffee</li>  
  <li>Tea</li>  
  <li>Coca Cola</li>  
</ul>  
  
</body>  
</html>  
  
The example above does not display equally in all browsers. IE and Opera will display the image-marker a little bit higher than Firefox, Chrome, and Safari.  
  
If you want the image-marker to be placed equally in all browsers, a crossbrowser solution is explained below.  
  
Crossbrowser Solution  
The following example displays the image-marker equally in all browsers:  
  
Example  
  
ul {  
    list-style-type: none;  
    padding: 0px;  
    margin: 0px;  
}  
  
ul li {  
    background-image: url(sqpurple.gif);  
    background-repeat: no-repeat;  
    background-position: 0px 5px;  
    padding-left: 14px;  
}  
  
<!DOCTYPE html>  
<html>  
<head>  
<style>  
ul {  
    list-style-type: none;  
    padding: 0px;  
    margin: 0px;  
}  
  
ul li {  
    background-image: url(sqpurple.gif);  
    background-repeat: no-repeat;  
    background-position: 0px 5px;  
    padding-left: 14px;  
}  
</style>  
</head>  
<body>  
  
<ul>  
  <li>Coffee</li>  
  <li>Tea</li>  
  <li>Coca Cola</li>  
</ul>  
  
</body>  
</html>  
  
Example explained:  
  
For ul:  
Set the list-style-type to none to remove the list item marker  
Set both padding and margin to 0px (for cross-browser compatibility)  
For all li in ul:  
Set the URL of the image, and show it only once (no-repeat)  
Position the image where you want it (left 0px and down 5px)  
Position the text in the list with padding-left  
List - Shorthand property  
It is also possible to specify all the list properties in one, single property. This is called a shorthand property.  
  
The shorthand property used for lists, is the list-style property:  
  
Example  
  
ul {  
    list-style: square url("sqpurple.gif");  
}  
  
<!DOCTYPE html>  
<html>  
<head>  
<style>  
ul {  
    list-style: square url("sqpurple.gif");  
}  
</style>  
</head>  
<body>  
  
<ul>  
  <li>Coffee</li>  
  <li>Tea</li>  
  <li>Coca Cola</li>  
</ul>  
  
</body>  
</html>  
  
When using the shorthand property, the order of the values are:  
  
list-style-type  
list-style-position (for a description, see the CSS properties table below)  
list-style-image  
It does not matter if one of the values above are missing, as long as the rest are in the specified order.  
  
Examples  
More Examples  
All the different list-item markers for lists  
<!DOCTYPE html>  
<html>  
<head>  
<style>  
ul.a {list-style-type: circle;}  
ul.b {list-style-type: disc;}  
ul.c {list-style-type: square;}  
  
ol.d {list-style-type: armenian;}  
ol.e {list-style-type: cjk-ideographic;}  
ol.f {list-style-type: decimal;}  
ol.g {list-style-type: decimal-leading-zero;}  
ol.h {list-style-type: georgian;}  
ol.i {list-style-type: hebrew;}  
ol.j {list-style-type: hiragana;}  
ol.k {list-style-type: hiragana-iroha;}  
ol.l {list-style-type: katakana;}  
ol.m {list-style-type: katakana-iroha;}  
ol.n {list-style-type: lower-alpha;}  
ol.o {list-style-type: lower-greek;}  
ol.p {list-style-type: lower-latin;}  
ol.q {list-style-type: lower-roman;}  
ol.r {list-style-type: upper-alpha;}  
ol.s {list-style-type: upper-latin;}  
ol.t {list-style-type: upper-roman;}  
ol.u {list-style-type: none;}  
ol.v {list-style-type: inherit;}  
</style>  
</head>  
<body>  
  
<ul class="a">  
  <li>Circle type</li>  
  <li>Tea</li>  
  <li>Coca Cola</li>  
</ul>  
  
<ul class="b">  
  <li>Disc type</li>  
  <li>Tea</li>  
  <li>Coca Cola</li>  
</ul>  
  
<ul class="c">  
  <li>Square type</li>  
  <li>Tea</li>  
  <li>Coca Cola</li>  
</ul>  
  
<ol class="d">  
  <li>Armenian type</li>  
  <li>Tea</li>  
  <li>Coca Cola</li>  
</ol>  
  
<ol class="e">  
  <li>Cjk-ideographic type</li>  
  <li>Tea</li>  
  <li>Coca Cola</li>  
</ol>  
  
<ol class="f">  
  <li>Decimal type</li>  
  <li>Tea</li>  
  <li>Coca Cola</li>  
</ol>  
  
<ol class="g">  
  <li>Decimal-leading-zero type</li>  
  <li>Tea</li>  
  <li>Coca Cola</li>  
</ol>  
  
<ol class="h">  
  <li>Georgian type</li>  
  <li>Tea</li>  
  <li>Coca Cola</li>  
</ol>  
  
<ol class="i">  
  <li>Hebrew type</li>  
  <li>Tea</li>  
  <li>Coca Cola</li>  
</ol>  
  
<ol class="j">  
  <li>Hiragana type</li>  
  <li>Tea</li>  
  <li>Coca Cola</li>  
</ol>  
  
<ol class="k">  
  <li>Hiragana-iroha type</li>  
  <li>Tea</li>  
  <li>Coca Cola</li>  
</ol>  
  
<ol class="l">  
  <li>Katakana type</li>  
  <li>Tea</li>  
  <li>Coca Cola</li>  
</ol>  
  
<ol class="m">  
  <li>Katakana-iroha type</li>  
  <li>Tea</li>  
  <li>Coca Cola</li>  
</ol>  
  
<ol class="n">  
  <li>Lower-alpha type</li>  
  <li>Tea</li>  
  <li>Coca Cola</li>  
</ol>  
  
<ol class="o">  
  <li>Lower-greek type</li>  
  <li>Tea</li>  
  <li>Coca Cola</li>  
</ol>  
  
<ol class="p">  
  <li>Lower-latin type</li>  
  <li>Tea</li>  
  <li>Coca Cola</li>  
</ol>  
  
<ol class="q">  
  <li>Lower-roman type</li>  
  <li>Tea</li>  
  <li>Coca Cola</li>  
</ol>  
  
<ol class="r">  
  <li>Upper-alpha type</li>  
  <li>Tea</li>  
  <li>Coca Cola</li>  
</ol>  
  
<ol class="s">  
  <li>Upper-latin type</li>  
  <li>Tea</li>  
  <li>Coca Cola</li>  
</ol>  
  
<ol class="t">  
  <li>Upper-roman type</li>  
  <li>Tea</li>  
  <li>Coca Cola</li>  
</ol>  
  
<ol class="u">  
  <li>None type</li>  
  <li>Tea</li>  
  <li>Coca Cola</li>  
</ol>  
  
<ol class="v">  
  <li>inherit type</li>  
  <li>Tea</li>  
  <li>Coca Cola</li>  
</ol>  
  
</body>  
</html>  
  
This example demonstrates all the different list-item markers in CSS.  
  
All CSS List Properties  
Property    Description  
list-style    Sets all the properties for a list in one declaration  
list-style-image    Specifies an image as the list-item marker  
list-style-position    Specifies if the list-item markers should appear inside or outside the content flow  
list-style-type    Specifies the type of list-item marker

C10

The look of an HTML table can be greatly improved with CSS:  
  
Company    Contact    Country  
Alfreds Futterkiste    Maria Anders    Germany  
Berglunds snabbköp    Christina Berglund    Sweden  
Centro comercial Moctezuma    Francisco Chang    Mexico  
Ernst Handel    Roland Mendel    Austria  
Island Trading    Helen Bennett    UK  
Königlich Essen    Philip Cramer    Germany  
Laughing Bacchus Winecellars    Yoshi Tannamuri    Canada  
Magazzini Alimentari Riuniti    Giovanni Rovelli    Italy  
North/South    Simon Crowther    UK  
Paris spécialités    Marie Bertrand    France  
The Big Cheese    Liz Nixon    USA  
Vaffeljernet    Palle Ibsen    Denmark  
  
Table Borders  
To specify table borders in CSS, use the border property.  
  
The example below specifies a black border for table, th, and td elements:  
  
Example  
  
table, th, td {  
   border: 1px solid black;  
}  
  
<!DOCTYPE html>  
<html>  
<head>  
<style>  
table, th, td {  
    border: 1px solid black;  
}  
</style>  
</head>  
<body>  
  
<table>  
  <tr>  
    <th>Firstname</th>  
    <th>Lastname</th>  
  </tr>  
  <tr>  
    <td>Peter</td>  
    <td>Griffin</td>  
  </tr>  
  <tr>  
    <td>Lois</td>  
    <td>Griffin</td>  
  </tr>  
</table>  
  
</body>  
</html>  
  
Notice that the table in the example above has double borders. This is because both the table and the th/td elements have separate borders.  
  
To display a single border for the table, use the border-collapse property.  
  
Collapse Borders  
The border-collapse property sets whether the table borders are collapsed into a single border or separated:  
  
Example  
  
table {  
    border-collapse: collapse;  
}  
  
table, th, td {  
    border: 1px solid black;  
}  
  
<!DOCTYPE html>  
<html>  
<head>  
<style>  
table {  
    border-collapse: collapse;  
}  
  
table, td, th {  
    border: 1px solid black;  
}  
</style>  
</head>  
<body>  
  
<table>  
  <tr>  
    <th>Firstname</th>  
    <th>Lastname</th>  
  </tr>  
  <tr>  
    <td>Peter</td>  
    <td>Griffin</td>  
  </tr>  
  <tr>  
    <td>Lois</td>  
    <td>Griffin</td>  
  </tr>  
</table>  
  
<p><b>Note:</b> If a !DOCTYPE is not specified, the border-collapse property can produce unexpected results  
in IE8 and earlier versions.</p>  
  
</body>  
</html>  
  
  
Table Width and Height  
Width and height of a table is defined by the width and height properties.  
  
The example below sets the width of the table to 100%, and the height of the th elements to 50px:  
  
Example  
  
table {  
    width: 100%;  
}  
  
th {  
    height: 50px;  
}  
  
Try it yourself »  
  
Table Text Alignment  
The text in a table is aligned with the text-align and vertical-align properties.  
  
The text-align property sets the horizontal alignment, like left, right, or center:  
  
Example  
  
td {  
    text-align: right;  
}  
  
<!DOCTYPE html>  
<html>  
<head>  
<style>  
table, td, th {  
    border: 1px solid black;  
}  
  
td {  
    text-align: right;  
}  
</style>  
</head>  
<body>  
  
<table>  
  <tr>  
    <th>Firstname</th>  
    <th>Lastname</th>  
    <th>Savings</th>  
  </tr>  
  <tr>  
    <td>Peter</td>  
    <td>Griffin</td>  
    <td>$100</td>  
  </tr>  
  <tr>  
    <td>Lois</td>  
    <td>Griffin</td>  
    <td>$150</td>  
  </tr>  
  <tr>  
    <td>Joe</td>  
    <td>Swanson</td>  
    <td>$300</td>  
  </tr>  
  <tr>  
    <td>Cleveland</td>  
    <td>Brown</td>  
    <td>$250</td>  
</tr>  
</table>  
  
</body>  
</html>  
  
The vertical-align property sets the vertical alignment, like top, bottom, or middle:  
  
Example  
  
td {  
    height: 50px;  
    vertical-align: bottom;  
}  
  
<!DOCTYPE html>  
<html>  
<head>  
<style>  
table, td, th {  
    border: 1px solid black;  
}  
  
td {  
    height: 50px;  
    vertical-align: bottom;  
}  
</style>  
</head>  
<body>  
  
<table>  
  <tr>  
    <th>Firstname</th>  
    <th>Lastname</th>  
    <th>Savings</th>  
  </tr>  
  <tr>  
    <td>Peter</td>  
    <td>Griffin</td>  
    <td>$100</td>  
  </tr>  
  <tr>  
    <td>Lois</td>  
    <td>Griffin</td>  
    <td>$150</td>  
  </tr>  
  <tr>  
    <td>Joe</td>  
    <td>Swanson</td>  
    <td>$300</td>  
  </tr>  
  <tr>  
    <td>Cleveland</td>  
    <td>Brown</td>  
    <td>$250</td>  
</tr>  
</table>  
  
</body>  
</html>  
  
  
Table Padding  
To control the space between the border and content in a table, use the padding property on td and th elements:  
  
Example  
  
td {  
    padding: 15px;  
}  
  
<!DOCTYPE html>  
<html>  
<head>  
<style>  
table, td, th {  
    border: 1px solid black;  
}  
  
td {  
    padding: 15px;  
}  
</style>  
</head>  
<body>  
  
<table>  
  <tr>  
    <th>Firstname</th>  
    <th>Lastname</th>  
    <th>Savings</th>  
  </tr>  
  <tr>  
    <td>Peter</td>  
    <td>Griffin</td>  
    <td>$100</td>  
  </tr>  
  <tr>  
    <td>Lois</td>  
    <td>Griffin</td>  
    <td>$150</td>  
  </tr>  
  <tr>  
    <td>Joe</td>  
    <td>Swanson</td>  
    <td>$300</td>  
  </tr>  
  <tr>  
    <td>Cleveland</td>  
    <td>Brown</td>  
    <td>$250</td>  
</tr>  
</table>  
  
</body>  
</html>  
  
  
Table Color  
The example below specifies the color of the borders, and the text and background color of th elements:  
  
Example  
  
table, td, th {  
    border: 1px solid green;  
}  
  
th {  
    background-color: green;  
    color: white;  
}  
  
<!DOCTYPE html>  
<html>  
<head>  
<style>  
table, td, th {  
    border: 1px solid green;  
}  
  
th {  
    background-color: green;  
    color: white;  
}  
</style>  
</head>  
  
<body>  
<table>  
  <tr>  
    <th>Firstname</th>  
    <th>Lastname</th>  
    <th>Savings</th>  
  </tr>  
  <tr>  
    <td>Peter</td>  
    <td>Griffin</td>  
    <td>$100</td>  
  </tr>  
  <tr>  
    <td>Lois</td>  
    <td>Griffin</td>  
    <td>$150</td>  
  </tr>  
  <tr>  
    <td>Joe</td>  
    <td>Swanson</td>  
    <td>$300</td>  
  </tr>  
  <tr>  
    <td>Cleveland</td>  
    <td>Brown</td>  
    <td>$250</td>  
</tr>  
</table>  
  
</body>  
</html>  
  
  
Examples  
More Examples  
Make a fancy table  
<!DOCTYPE html>  
<html>  
<head>  
<style>  
#customers {  
    font-family: "Trebuchet MS", Arial, Helvetica, sans-serif;  
    width: 100%;  
    border-collapse: collapse;  
}  
  
#customers td, #customers th {  
    font-size: 1em;  
    border: 1px solid #98bf21;  
    padding: 3px 7px 2px 7px;  
}  
  
#customers th {  
    font-size: 1.1em;  
    text-align: left;  
    padding-top: 5px;  
    padding-bottom: 4px;  
    background-color: #A7C942;  
    color: #ffffff;  
}  
  
#customers tr.alt td {  
    color: #000000;  
    background-color: #EAF2D3;  
}  
</style>  
</head>  
<body>  
  
<table id="customers">  
  <tr>  
    <th>Company</th>  
    <th>Contact</th>  
    <th>Country</th>  
  </tr>  
  <tr>  
    <td>Alfreds Futterkiste</td>  
    <td>Maria Anders</td>  
    <td>Germany</td>  
  </tr>  
  <tr class="alt">  
    <td>Berglunds snabbk鰌</td>  
    <td>Christina Berglund</td>  
    <td>Sweden</td>  
  </tr>  
  <tr>  
    <td>Centro comercial Moctezuma</td>  
    <td>Francisco Chang</td>  
    <td>Mexico</td>  
  </tr>  
  <tr class="alt">  
    <td>Ernst Handel</td>  
    <td>Roland Mendel</td>  
    <td>Austria</td>  
  </tr>  
  <tr>  
    <td>Island Trading</td>  
    <td>Helen Bennett</td>  
    <td>UK</td>  
  </tr>  
  <tr class="alt">  
    <td>K鰊iglich Essen</td>  
    <td>Philip Cramer</td>  
    <td>Germany</td>  
  </tr>  
  <tr>  
    <td>Laughing Bacchus Winecellars</td>  
    <td>Yoshi Tannamuri</td>  
    <td>Canada</td>  
  </tr>  
  <tr class="alt">  
    <td>Magazzini Alimentari Riuniti</td>  
    <td>Giovanni Rovelli</td>  
    <td>Italy</td>  
  </tr>  
  <tr>  
    <td>North/South</td>  
    <td>Simon Crowther</td>  
    <td>UK</td>  
  </tr>  
  <tr class="alt">  
    <td>Paris sp閏ialit閟</td>  
    <td>Marie Bertrand</td>  
    <td>France</td>  
  </tr>  
</table>  
  
</body>  
</html>  
  
This example demonstrates how to create a fancy table.  
  
Set the position of the table caption  
<!DOCTYPE html>  
<html>  
<head>  
<style>  
table, td, th {  
    border: 1px solid black;  
}  
  
caption {  
    caption-side: bottom;  
}  
</style>  
</head>  
<body>  
  
<table>  
<caption>Table 1.1 Customers</caption>  
  <tr>  
    <th>Company</th>  
    <th>Contact</th>  
    <th>Country</th>  
  </tr>  
  <tr>  
    <td>Alfreds Futterkiste</td>  
    <td>Maria Anders</td>  
    <td>Germany</td>  
  </tr>  
  <tr class="alt">  
    <td>Berglunds snabbk鰌</td>  
    <td>Christina Berglund</td>  
    <td>Sweden</td>  
  </tr>  
  <tr>  
    <td>Centro comercial Moctezuma</td>  
    <td>Francisco Chang</td>  
    <td>Mexico</td>  
  </tr>  
  <tr class="alt">  
    <td>Ernst Handel</td>  
    <td>Roland Mendel</td>  
    <td>Austria</td>  
  </tr>  
  <tr>  
    <td>Island Trading</td>  
    <td>Helen Bennett</td>  
    <td>UK</td>  
  </tr>  
</table>  
  
<p><b>Note:</b> IE8 supports the caption-side property if a !DOCTYPE is specified.</p>  
  
</body>  
</html>  
  
This example demonstrates how to position the table caption.

C11

The CSS Box Model  
All HTML elements can be considered as boxes. In CSS, the term "box model" is used when talking about design and layout.  
  
The CSS box model is essentially a box that wraps around HTML elements, and it consists of: margins, borders, padding, and the actual content.  
  
The box model allows us to place a border around elements and space elements in relation to other elements.  
  
The image below illustrates the box model:  
  
  
CSS box-model  
Explanation of the different parts:  
  
Margin - Clears an area around the border. The margin does not have a background color, it is completely transparent  
Border - A border that goes around the padding and content. The border is inherited from the color property of the box  
Padding - Clears an area around the content. The padding is affected by the background color of the box  
Content - The content of the box, where text and images appear  
In order to set the width and height of an element correctly in all browsers, you need to know how the box model works.  
  
Width and Height of an Element  
Note    Important: When you set the width and height properties of an element with CSS, you just set the width and height of the content area. To calculate the full size of an element, you must also add the padding, borders and margins.  
The total width of the element in the example below is 300px:  
  
width: 250px;  
padding: 10px;  
border: 5px solid gray;  
margin: 10px;  
Let's do the math:  
250px (width)  
+ 20px (left + right padding)  
+ 10px (left + right border)  
+ 20px (left + right margin)  
= 300px  
  
Assume that you had only 250px of space. Let's make an element with a total width of 250px:  
  
Example  
  
div {  
    width: 220px;  
    padding: 10px;  
    border: 5px solid gray;  
    margin: 0px;  
}  
  
<!DOCTYPE html>  
<html>  
<head>  
<style>  
div.ex {  
    width: 220px;  
    padding: 10px;  
    border: 5px solid gray;  
    margin: 0px;  
}  
</style>  
</head>  
<body>  
  
<img src="w3css.gif" width="250" height="250">  
<div class="ex">The picture above is 250px wide. The total width of this element is also 250px.</div>  
  
</body>  
</html>  
  
The total width of an element should be calculated like this:  
  
Total element width = width + left padding + right padding + left border + right border + left margin + right margin  
  
The total height of an element should be calculated like this:  
  
Total element height = height + top padding + bottom padding + top border + bottom border + top margin + bottom margin  
  
Browsers Compatibility Issue  
IE8 and earlier versions of IE, included padding and border in the width property.  
  
To fix this problem, add a <!DOCTYPE html> to the HTML page.

C12

Border Style  
The border-style property specifies what kind of border to display.  
  
Note    Note: None of the border properties will have ANY effect unless the border-style property is set!  
border-style values:  
none: Defines no border  
  
dotted: Defines a dotted border  
  
dashed: Defines a dashed border  
  
solid: Defines a solid border  
  
double: Defines two borders. The width of the two borders are the same as the border-width value  
  
groove: Defines a 3D grooved border. The effect depends on the border-color value  
  
ridge: Defines a 3D ridged border. The effect depends on the border-color value  
  
inset: Defines a 3D inset border. The effect depends on the border-color value  
  
outset: Defines a 3D outset border. The effect depends on the border-color value  
  
Try it yourself: Set the style of the border  
  
Border Width  
The border-width property is used to set the width of the border.  
  
The width is set in pixels, or by using one of the three pre-defined values: thin, medium, or thick.  
  
Note: The "border-width" property does not work if it is used alone. Use the "border-style" property to set the borders first.  
  
Example  
  
p.one {  
    border-style: solid;  
    border-width: 5px;  
}  
  
p.two {  
    border-style: solid;  
    border-width: medium;  
}  
  
<!DOCTYPE html>  
<html>  
<head>  
<style>  
p.one {  
    border-style: solid;  
    border-width: 5px;  
}  
  
p.two {  
    border-style: solid;  
    border-width: medium;  
}  
  
p.three {  
    border-style: solid;  
    border-width: 1px;  
}  
</style>  
</head>  
<body>  
  
<p class="one">Some text.</p>  
<p class="two">Some text.</p>  
<p class="three">Some text.</p>  
<p><b>Note:</b> The "border-width" property does not work if it is used alone. You must add the "border-style" property to set the borders first.</p>  
  
</body>  
</html>  
  
  
Border Color  
The border-color property is used to set the color of the border. The color can be set by:  
  
name - specify a color name, like "red"  
RGB - specify a RGB value, like "rgb(255,0,0)"  
Hex - specify a hex value, like "#ff0000"  
You can also set the border color to "transparent".  
  
Note: The "border-color" property does not work if it is used alone. Use the "border-style" property to set the borders first.  
  
Example  
  
p.one {  
    border-style: solid;  
    border-color: red;  
}  
  
p.two {  
    border-style: solid;  
    border-color: #98bf21;  
}  
  
<!DOCTYPE html>  
<html>  
<head>  
<style>  
p.one {  
    border-style: solid;  
    border-color: red;  
}  
  
p.two {  
    border-style: solid;  
    border-color: #98bf21;  
}  
</style>  
</head>  
<body>  
  
<p class="one">A solid red border</p>  
<p class="two">A solid green border</p>  
<p><b>Note:</b> The "border-color" property does not work if it is used alone. Use the "border-style" property to set the borders first.</p>  
  
</body>  
</html>  
  
  
Border - Individual sides  
In CSS it is possible to specify different borders for different sides:  
  
Example  
  
p {  
    border-top-style: dotted;  
    border-right-style: solid;  
    border-bottom-style: dotted;  
    border-left-style: solid;  
}  
  
<!DOCTYPE html>  
<html>  
<head>  
<style>  
p {  
    border-top-style: dotted;  
    border-right-style: solid;  
    border-bottom-style: dotted;  
    border-left-style: solid;  
}  
</style>  
</head>  
<body>  
  
<p>2 different border styles.</p>  
  
</body>  
</html>  
  
The example above can also be set with a single property:  
  
Example  
  
p {  
    border-style: dotted solid;  
}  
  
<!DOCTYPE html>  
<html>  
<head>  
<style>  
p {  
    border-style: dotted solid;  
}  
</style>  
</head>  
<body>  
  
<p>2 different border styles.</p>  
  
</body>  
</html>  
  
The border-style property can have from one to four values.  
  
border-style: dotted solid double dashed;  
top border is dotted  
right border is solid  
bottom border is double  
left border is dashed  
  
border-style: dotted solid double;  
top border is dotted  
right and left borders are solid  
bottom border is double  
  
border-style: dotted solid;  
top and bottom borders are dotted  
right and left borders are solid  
  
border-style: dotted;  
all four borders are dotted  
The border-style property is used in the example above. However, it also works with border-width and border-color.  
  
Border - Shorthand property  
As you can see from the examples above, there are many properties to consider when dealing with borders.  
  
To shorten the code, it is also possible to specify all the individual border properties in one property. This is called a shorthand property.  
  
The border property is a shorthand for the following individual border properties:  
  
border-width  
border-style (required)  
border-color  
Example  
  
p {  
    border: 5px solid red;  
}  
  
<!DOCTYPE html>  
<html>  
<head>  
<style>  
p {  
    border: 5px solid red;  
}  
</style>  
</head>  
<body>  
  
<p>This is some text in a paragraph.</p>  
  
</body>  
</html>  
  
  
Examples  
More Examples  
All the top border properties in one declaration  
<!DOCTYPE html>  
<html>  
<head>  
<style>  
p {  
    border-style: solid;  
    border-top: thick double #ff0000;  
}  
</style>  
</head>  
<body>  
  
<p>This is some text in a paragraph.</p>  
  
</body>  
</html>  
  
This example demonstrates a shorthand property for setting all of the properties for the top border in one declaration.  
  
Set the style of the bottom border  
<!DOCTYPE html>  
<html>  
<head>  
<style>  
p {border-style: solid;}  
p.none {border-bottom-style: none;}  
p.dotted {border-bottom-style: dotted;}  
p.dashed {border-bottom-style: dashed;}  
p.solid {border-bottom-style: solid;}  
p.double {border-bottom-style: double;}  
p.groove {border-bottom-style: groove;}  
p.ridge {border-bottom-style: ridge;}  
p.inset {border-bottom-style: inset;}  
p.outset {border-bottom-style: outset;}  
</style>  
</head>  
<body>  
  
<p class="none">No bottom border.</p>  
<p class="dotted">A dotted bottom border.</p>  
<p class="dashed">A dashed bottom border.</p>  
<p class="solid">A solid bottom border.</p>  
<p class="double">A double bottom border.</p>  
<p class="groove">A groove bottom border.</p>  
<p class="ridge">A ridge bottom border.</p>  
<p class="inset">An inset bottom border.</p>  
<p class="outset">An outset bottom border.</p>  
  
</body>  
</html>  
  
This example demonstrates how to set the style of the bottom border.  
  
Set the width of the left border  
<!DOCTYPE html>  
<html>  
<head>  
<style>  
p {  
    border-style: solid;  
    border-left-width: 15px;  
}  
</style>  
</head>  
<body>  
  
<p><b>Note:</b> The "border-left-width" property does not work if it is used alone. Use the "border-style" property to set the borders first.</p>  
  
</body>  
</html>  
  
This example demonstrates how to set the width of the left border.  
  
Set the color of the four borders  
<!DOCTYPE html>  
<html>  
<head>  
<style>  
p.one {  
    border-style: solid;  
    border-color: #0000ff;  
}  
  
p.two {  
    border-style: solid;  
    border-color: #ff0000 #0000ff;  
}  
  
p.three {  
    border-style: solid;  
    border-color: #ff0000 #00ff00 #0000ff;  
}  
  
p.four {  
    border-style: solid;  
    border-color: #ff0000 #00ff00 #0000ff rgb(250,0,255);  
}  
</style>  
</head>  
<body>  
  
<p class="one">One-colored border!</p>  
<p class="two">Two-colored border!</p>  
<p class="three">Three-colored border!</p>  
<p class="four">Four-colored border!</p>  
<p><b>Note:</b> The "border-color" property does not work if it is used alone. Use the "border-style" property to set the borders first.</p>  
  
</body>  
</html>  
  
This example demonstrates how to set the color of the four borders. It can have from one to four colors.  
  
Set the color of the right border  
<!DOCTYPE html>  
<html>  
<head>  
<style>  
p {  
    border-style: solid;  
    border-right-color: #ff0000;  
}  
</style>  
</head>  
<body>  
  
<p>This is some text in a paragraph.</p>  
  
</body>  
</html>  
  
This example demonstrates how to set the color of the right border.  
  
All CSS Border Properties  
Property    Description  
border    Sets all the border properties in one declaration  
border-bottom    Sets all the bottom border properties in one declaration  
border-bottom-color    Sets the color of the bottom border  
border-bottom-style    Sets the style of the bottom border  
border-bottom-width    Sets the width of the bottom border  
border-color    Sets the color of the four borders  
border-left    Sets all the left border properties in one declaration  
border-left-color    Sets the color of the left border  
border-left-style    Sets the style of the left border  
border-left-width    Sets the width of the left border  
border-right    Sets all the right border properties in one declaration  
border-right-color    Sets the color of the right border  
border-right-style    Sets the style of the right border  
border-right-width    Sets the width of the right border  
border-style    Sets the style of the four borders  
border-top    Sets all the top border properties in one declaration  
border-top-color    Sets the color of the top border  
border-top-style    Sets the style of the top border  
border-top-width    Sets the width of the top border  
border-width    Sets the width of the four borders

C13

An outline is a line that is drawn around elements (outside the borders) to make the element "stand out".  
  
The outline properties specify the style, color, and width of an outline.  
  
Examples  
Draw a line around an element (outline)  
<!DOCTYPE html>  
<html>  
<head>  
<style>  
p {  
    border: 1px solid red;  
    outline: green dotted thick;  
}  
</style>  
</head>  
<body>  
  
<p><b>Note:</b> IE8 supports the outline properties only if a !DOCTYPE is specified.</p>  
  
</body>  
</html>  
  
This example demonstrates how to draw a line around an element, outside the border edge.  
  
Set the style of an outline  
<!DOCTYPE html>  
<html>  
<head>  
<style>  
p {border: 1px solid red;}  
p.dotted {outline-style: dotted;}  
p.dashed {outline-style: dashed;}  
p.solid {outline-style: solid;}  
p.double {outline-style: double;}  
p.groove {outline-style: groove;}  
p.ridge {outline-style: ridge;}  
p.inset {outline-style: inset;}  
p.outset {outline-style: outset;}  
</style>  
</head>  
<body>  
  
<p class="dotted">A dotted outline</p>  
<p class="dashed">A dashed outline</p>  
<p class="solid">A solid outline</p>  
<p class="double">A double outline</p>  
<p class="groove">A groove outline</p>  
<p class="ridge">A ridge outline</p>  
<p class="inset">An inset outline</p>  
<p class="outset">An outset outline</p>  
<b>Note:</b> IE8 supports the outline properties only if a !DOCTYPE is specified.  
  
</body>  
</html>  
  
This example demonstrates how to set the style of an outline.  
  
Set the color of an outline  
<!DOCTYPE html>  
<html>  
<head>  
<style>  
p {  
    border: 1px solid red;  
    outline-style: dotted;  
    outline-color: #00ff00;  
}  
</style>  
</head>  
<body>  
  
<p><b>Note:</b> IE8 supports the outline properties only if a !DOCTYPE is specified.</p>  
  
</body>  
</html>  
  
This example demonstrates how to set the color of an outline.  
  
Set the width of an outline  
<!DOCTYPE html>  
<html>  
<head>  
<style>  
p.one {  
    border: 1px solid red;  
    outline-style: solid;  
    outline-width: thin;  
}  
  
p.two {  
    border: 1px solid red;  
    outline-style: dotted;  
    outline-width: 3px;  
}  
</style>  
</head>  
<body>  
  
<p class="one">This is some text in a paragraph.</p>  
<p class="two">This is some text in a paragraph.</p>  
<p><b>Note:</b> IE8 supports the outline properties only if a !DOCTYPE is specified.</p>  
  
</body>  
</html>  
  
This example demonstrates how to set the width of an outline.  
  
CSS Outline  
An outline is a line that is drawn around elements (outside the borders) to make the element "stand out".  
  
However, the outline property is different from the border property.  
  
The outline is not a part of an element's dimensions; the element's total width and height is not affected by the width of the outline.  
  
Outline  
  
All CSS Outline Properties  
Property    Description    Values  
outline    Sets all the outline properties in one declaration    outline-color  
outline-style  
outline-width  
inherit  
outline-color    Sets the color of an outline    color\_name  
hex\_number  
rgb\_number  
invert  
inherit  
outline-style    Sets the style of an outline    none  
dotted  
dashed  
solid  
double  
groove  
ridge  
inset  
outset  
inherit  
outline-width    Sets the width of an outline    thin  
medium  
thick  
length  
inherit

C14

The CSS margin properties define the space around elements.  
  
Margin  
The margin clears an area around an element (outside the border). The margin does not have a background color, and is completely transparent.  
  
The top, right, bottom, and left margin can be changed independently using separate properties. A shorthand margin property can also be used, to change all margins at once.  
  
Possible Values  
Value    Description  
auto    The browser calculates a margin  
length    Specifies a margin in px, pt, cm, etc. Default value is 0px  
%    Specifies a margin in percent of the width of the containing element  
inherit    Specifies that the margin should be inherited from the parent element  
  
Note    Note: It is also possible to use negative values, to overlap content.  
  
Margin - Individual sides  
In CSS, it is possible to specify different margins for different sides of an element:  
  
Example  
  
p {  
    margin-top: 100px;  
    margin-bottom: 100px;  
    margin-right: 150px;  
    margin-left: 50px;  
}  
  
<!DOCTYPE html>  
<html>  
<head>  
<style>  
p {  
    background-color: yellow;  
}  
  
p.ex {  
    margin-top: 100px;  
    margin-bottom: 100px;  
    margin-right: 150px;  
    margin-left: 50px;  
}  
</style>  
</head>  
<body>  
  
<p>This is a paragraph with no specified margins.</p>  
<p class="ex">This is a paragraph with specified margins.</p>  
  
</body>  
</html>  
  
  
Margin - Shorthand property  
To shorten the code, it is possible to specify all the margin properties in one property. This is called a shorthand property.  
  
The shorthand property for all the margin properties is "margin":  
  
Example  
  
p {  
    margin: 100px 50px;  
}  
  
<!DOCTYPE html>  
<html>  
<head>  
<style>  
p {  
    background-color: yellow;  
}  
  
p.ex {  
    margin: 100px 50px;  
}  
</style>  
</head>  
<body>  
  
<p>This is a paragraph with no specified margins.</p>  
<p class="ex">This is a paragraph with specified margins.</p>  
  
</body>  
</html>  
  
The margin property can have from one to four values.  
  
margin: 25px 50px 75px 100px;  
top margin is 25px  
right margin is 50px  
bottom margin is 75px  
left margin is 100px  
  
margin: 25px 50px 75px;  
top margin is 25px  
right and left margins are 50px  
bottom margin is 75px  
  
margin: 25px 50px;  
top and bottom margins are 25px  
right and left margins are 50px  
  
margin: 25px;  
all four margins are 25px  
Examples  
More Examples  
Set the top margin of a text using a cm value  
<!DOCTYPE html>  
<html>  
<head>  
<style>  
p.ex1 {  
    margin-top: 2cm;  
}  
</style>  
</head>  
<body>  
  
<p>A paragraph with no margins specified.</p>  
<p class="ex1">A paragraph with a 2cm top margin.</p>  
<p>A paragraph with no margins specified.</p>  
  
</body>  
</html>  
  
This example demonstrates how to set the top margin of a text using a cm value.  
  
Set the bottom margin of a text using a percent value  
<!DOCTYPE html>  
<html>  
<head>  
<style>  
p.bottommargin {  
    margin-bottom: 25%;  
}  
</style>  
</head>  
<body>  
  
<p>This is a paragraph with no margin specified.</p>  
<p class="bottommargin">This is a paragraph with a specified bottom margin.</p>  
<p>This is a paragraph with no margin specified.</p>  
  
</body>  
</html>  
  
This example demonstrates how to set the bottom margin in percent, relative to the width of the containing element.  
  
All CSS Margin Properties  
Property    Description  
margin    A shorthand property for setting the margin properties in one declaration  
margin-bottom    Sets the bottom margin of an element  
margin-left    Sets the left margin of an element  
margin-right    Sets the right margin of an element  
margin-top    Sets the top margin of an element

C15

The CSS padding properties define the space between the element border and the element content.  
  
Padding  
The padding clears an area around the content (inside the border) of an element. The padding is affected by the background color of the element.  
  
The top, right, bottom, and left padding can be changed independently using separate properties. A shorthand padding property can also be used, to change all paddings at once.  
  
Possible Values  
Value    Description  
length    Defines a fixed padding (in pixels, pt, em, etc.)  
%    Defines a padding in % of the containing element  
  
Padding - Individual sides  
In CSS, it is possible to specify different padding for different sides:  
  
Example  
  
p {  
    padding-top: 25px;  
    padding-bottom: 25px;  
    padding-right: 50px;  
    padding-left: 50px;  
}  
  
<!DOCTYPE html>  
<html>  
<head>  
<style>  
p {  
    background-color: yellow;  
}  
  
p.padding {  
    padding-top: 25px;  
    padding-bottom: 25px;  
    padding-right: 50px;  
    padding-left: 50px;  
}  
</style>  
</head>  
<body>  
  
<p>This is a paragraph with no specified padding.</p>  
<p class="padding">This is a paragraph with specified paddings.</p>  
  
</body>  
</html>  
  
  
Padding - Shorthand property  
To shorten the code, it is possible to specify all the padding properties in one property. This is called a shorthand property.  
  
The shorthand property for all the padding properties is "padding":  
  
Example  
  
p {  
    padding: 25px 50px;  
}  
  
<!DOCTYPE html>  
<html>  
<head>  
<style>  
p {  
    background-color: yellow;  
}  
  
p.padding {  
    padding: 25px 50px;  
}  
</style>  
</head>  
<body>  
  
<p>This is a paragraph with no specified padding.</p>  
<p class="padding">This is a paragraph with specified paddings.</p>  
  
</body>  
</html>  
  
The padding property can have from one to four values.  
  
padding: 25px 50px 75px 100px;  
top padding is 25px  
right padding is 50px  
bottom padding is 75px  
left padding is 100px  
  
padding: 25px 50px 75px;  
top padding is 25px  
right and left paddings are 50px  
bottom padding is 75px  
  
padding: 25px 50px;  
top and bottom paddings are 25px  
right and left paddings are 50px  
  
padding: 25px;  
all four paddings are 25px  
Examples  
More Examples  
All the padding properties in one declaration  
<!DOCTYPE html>  
<html>  
<head>  
<style>  
p.ex1 {  
    padding: 2cm;  
}  
  
p.ex2 {  
    padding: 0.5cm 3cm;  
}  
</style>  
</head>  
<body>  
  
<p class="ex1">This text has equal padding on each side. The padding on each side is 2cm.</p>  
<p class="ex2">This text has a top and bottom padding of 0.5cm and a left and right padding of 3cm.</p>  
  
</body>  
</html>  
  
This example demonstrates a shorthand property for setting all of the padding properties in one declaration, can have from one to four values.  
  
Set the left padding  
<!DOCTYPE html>  
<html>  
<head>  
<style>  
p.padding {  
    padding-left: 2cm;  
}  
p.padding2 {  
    padding-left: 50%;  
}  
</style>  
</head>  
<body>  
  
<p>This is a text with no left padding.</p>  
<p class="padding">This text has a left padding of 2 cm.</p>  
<p class="padding2">This text has a left padding of 50%.</p>  
  
</body>  
</html>  
  
This example demonstrates how to set the left padding of a p element.  
  
Set the right padding  
<!DOCTYPE html>  
<html>  
<head>  
<style>  
p.padding {  
    padding-right: 2cm;  
}  
  
p.padding2 {  
    padding-right: 50%;  
}  
</style>  
</head>  
<body>  
  
<p>This is a text with no right padding. This is a text with no right padding. This is a text with no right padding.</p>  
<p class="padding">This text has a right padding of 2 cm. This text has a right padding of 2 cm. This text has a right padding of 2 cm.</p>  
<p class="padding2">This text has a right padding of 50%. This text has a right padding of 50%. This text has a right padding of 50%.</p>  
  
</body>  
</html>  
  
This example demonstrates how to set the right padding of a p element.  
  
Set the top padding  
<!DOCTYPE html>  
<html>  
<head>  
<style>  
p.padding {  
    padding-top: 2cm;  
}  
  
p.padding2 {  
    padding-top: 50%;  
}  
</style>  
</head>  
<body>  
  
<p>This is a text with no top padding. This is a text with no top padding. This is a text with no top padding.</p>  
<p class="padding">This text has a top padding of 2 cm. This text has a top padding of 2 cm. This text has a top padding of 2 cm.</p>  
<p class="padding2">This text has a top padding of 50%. This text has a top padding of 50%. This text has a top padding of 50%.</p>  
  
</body>  
</html>  
  
This example demonstrates how to set the top padding of a p element.  
  
Set the bottom padding  
<!DOCTYPE html>  
<html>  
<head>  
<style>  
p.padding {  
    padding-bottom:2cm;  
}  
  
p.padding2 {  
    padding-bottom:50%;  
}  
</style>  
</head>  
<body>  
  
<p>This is a text with no bottom padding. This is a text with no bottom padding. This is a text with no bottom padding.</p>  
<p class="padding">This text has a bottom padding of 2 cm. This text has a bottom padding of 2 cm. This text has a bottom padding of 2 cm.</p>  
<p class="padding2">This text has a bottom padding of 50%. This text has a bottom padding of 50%. This text has a bottom padding of 50%.</p>  
  
</body>  
</html>  
  
This example demonstrates how to set the bottom padding of a p element.  
  
All CSS Padding Properties  
Property    Description  
padding    A shorthand property for setting all the padding properties in one declaration  
padding-bottom    Sets the bottom padding of an element  
padding-left    Sets the left padding of an element  
padding-right    Sets the right padding of an element  
padding-top    Sets the top padding of an element

C16

The CSS dimension properties allow you to control the height and width of an element.  
  
Examples  
Try it Yourself - Examples  
Set the height of elements  
<!DOCTYPE html>  
<html>  
<head>  
<style>  
img.normal {  
    height: auto;  
}  
  
img.big {  
    height: 120px;  
}  
  
p.ex {  
    height: 100px;  
    width: 100px;  
}  
</style>  
</head>  
<body>  
  
<img class="normal" src="smiley.gif"><br>  
<img class="big" src="smiley.gif">  
<p class="ex">The height and width of this paragraph is 100px.</p>  
<p>This is a paragraph with no height and width specified.</p>  
<p>This is also a paragraph with no height and width specified.</p>  
  
</body>  
</html>  
  
This example demonstrates how to set the height of different elements.  
  
Set the height of an image using percent  
<!DOCTYPE html>  
<html>  
<head>  
<style>  
html, body {  
    height: 100%;  
}  
  
img.normal {  
    height: auto;  
}  
  
img.big {  
    height: 10%;  
}  
</style>  
</head>  
<body>  
  
<img class="normal" src="smiley.gif" width="32" height="32"><br>  
<img class="big" src="smiley.gif" width="32" height="32">  
  
</body>  
</html>  
  
This example demonstrates how to set the height of an element using a percent value.  
  
Set the width of an element using a pixel value  
<!DOCTYPE html>  
<html>  
<head>  
<style>  
img {  
    width: 200px;  
}  
</style>  
</head>  
<body>  
  
<img src="smiley.gif" width="32" height="32">  
  
</body>  
</html>  
  
This example demonstrates how to set the width of an element using a pixel value.  
  
Set the maximum height of an element  
<!DOCTYPE html>  
<html>  
<head>  
<style>  
p {  
    max-height: 50px;  
    background-color: yellow;  
}  
</style>  
</head>  
<body>  
  
<p>The maximum height of this paragraph is set to 50px. The maximum height of this paragraph is set to 50px. The maximum height of this paragraph is set to 50px. The maximum height of this paragraph is set to 50px. The maximum height of this paragraph is set to 50px. The maximum height of this paragraph is set to 50px. The maximum height of this paragraph is set to 50px. The maximum height of this paragraph is set to 50px. The maximum height of this paragraph is set to 50px. The maximum height of this paragraph is set to 50px.</p>  
  
</body>  
</html>  
  
This example demonstrates how to set the maximum height of an element.  
  
Set the maximum width of an element using percent  
<!DOCTYPE html>  
<html>  
<head>  
<style>  
p {  
    max-width: 20%;  
    background-color: yellow;  
}  
</style>  
</head>  
<body>  
  
<p>This is some text. This is some text. This is some text.  
This is some text. This is some text. This is some text.  
This is some text. This is some text. This is some text.</p>  
  
</body>  
</html>  
  
This example demonstrates how to set the maximum width of an element using a percent value.  
  
Set the minimum height of an element  
<!DOCTYPE html>  
<html>  
<head>  
<style>  
p {  
    min-height: 100px;  
    background-color: yellow;  
}  
</style>  
</head>  
<body>  
  
<p>The minimum height of this paragraph is set to 100px.</p>  
  
</body>  
</html>  
  
This example demonstrates how to set the minimum height of an element.  
  
Set the minimum width of an element using a pixel value  
<!DOCTYPE html>  
<html>  
<head>  
<style>  
p {  
    min-width: 150px;  
    background-color: yellow;  
}  
</style>  
</head>  
<body>  
  
<p>The minimum width of this paragraph is set to 150px.</p>  
  
</body>  
</html>  
  
This example demonstrates how to set the minimum width of an element using a pixel value.  
  
All CSS Dimension Properties  
Property    Description    Values  
height    Sets the height of an element    auto  
length  
%  
inherit  
max-height    Sets the maximum height of an element    none  
length  
%  
inherit  
max-width    Sets the maximum width of an element    none  
length  
%  
inherit  
min-height    Sets the minimum height of an element    length  
%  
inherit  
min-width    Sets the minimum width of an element    length  
%  
inherit  
width    Sets the width of an element    auto  
length  
%  
inherit

C17

Hiding an Element - display:none or visibility:hidden  
Hiding an element can be done by setting the display property to "none" or the visibility property to "hidden". However, notice that these two methods produce different results:  
  
visibility:hidden hides an element, but it will still take up the same space as before. The element will be hidden, but still affect the layout.  
  
Example  
  
h1.hidden {  
    visibility: hidden;  
}  
  
<!DOCTYPE html>  
<html>  
<head>  
<style>  
h1.hidden {  
    visibility: hidden;  
}  
</style>  
</head>  
<body>  
  
c  
</body>  
</html>  
  
  
display:none hides an element, and it will not take up any space. The element will be hidden, and the page will be displayed as if the element is not there:  
  
Example  
  
h1.hidden {  
    display: none;  
}  
  
<!DOCTYPE html>  
<html>  
<head>  
<style>  
h1.hidden {  
    display: none;  
}  
</style>  
</head>  
<body>  
  
<h1>This is a visible heading</h1>  
<h1 class="hidden">This is a hidden heading</h1>  
<p>Notice that the hidden heading does not take up space.</p>  
  
</body>  
</html>  
  
  
CSS Display - Block and Inline Elements  
A block element is an element that takes up the full width available, and has a line break before and after it.  
  
Examples of block elements:  
  
<h1>  
<p>  
<li>  
<div>  
An inline element only takes up as much width as necessary, and does not force line breaks.  
  
Examples of inline elements:  
  
<span>  
<a>  
Changing How an Element is Displayed  
Changing an inline element to a block element, or vice versa, can be useful for making the page look a specific way, and still follow web standards.  
  
The following example displays <li> elements as inline elements:  
  
Example  
  
li {  
    display: inline;  
}  
  
<!DOCTYPE html>  
<html>  
<head>  
<style>  
li {  
    display: inline;  
}  
</style>  
</head>  
<body>  
  
<p>Display a list of links as a horizontal menu:</p>  
<ul>  
<li><a href="/html/default.asp" target="\_blank">HTML</a></li>  
<li><a href="/css/default.asp" target="\_blank">CSS</a></li>  
<li><a href="/js/default.asp" target="\_blank">JavaScript</a></li>  
</ul>  
  
</body>  
</html>  
  
  
The following example displays <span> elements as block elements:  
  
Example  
  
span {  
    display: block;  
}  
  
<!DOCTYPE html>  
<html>  
<head>  
<style>  
span {  
    display: block;  
}  
</style>  
</head>  
<body>  
  
<span>A display property with a value of "block" results in</span> <span>a line break between the two elements.</span>  
  
</body>  
</html>  
  
  
Note    Note: Setting the display property of an element only changes how the element is displayed, NOT what kind of element it is. So, an inline element with display:block is not allowed to have other block elements inside of it.  
  
Examples  
More Examples  
How to display an element as an inline element.  
<!DOCTYPE html>  
<html>  
<head>  
<style>  
p {  
    display: inline;  
}  
</style>  
</head>  
<body>  
  
<p>A display property with a value of "inline" results in</p>  
<p>no distance between two elements.</p>  
  
</body>  
</html>  
  
This example demonstrates how to display an element as an inline element.  
  
How to make a table element collapse  
<!DOCTYPE html>  
<html>  
<head>  
<style>  
table, th, td {  
    border: 1px solid black;  
}  
  
tr.collapse {  
    visibility: collapse;  
}  
</style>  
</head>  
<body>  
  
<table>  
  <tr>  
    <td>Peter</td>  
    <td>Griffin</td>  
  </tr>  
  <tr class="collapse">  
    <td>Lois</td>  
    <td>Griffin</td>  
  </tr>  
</table>  
  
<p><b>Note:</b> IE8 and earlier support visibility:collapse only if a !DOCTYPE is specified.</p>  
  
</body>  
</html>  
  
This example demonstrates how to make a table element collapse.

C18

Positioning  
The CSS positioning properties allow you to position an element. It can also place an element behind another, and specify what should happen when an element's content is too big.  
  
Elements can be positioned using the top, bottom, left, and right properties. However, these properties will not work unless the position property is set first. They also work differently depending on the positioning method.  
  
There are four different positioning methods.  
  
Static Positioning  
HTML elements are positioned static by default. A static positioned element is always positioned according to the normal flow of the page.  
  
Static positioned elements are not affected by the top, bottom, left, and right properties.  
  
Fixed Positioning  
An element with fixed position is positioned relative to the browser window.  
  
It will not move even if the window is scrolled:  
  
Example  
  
p.pos\_fixed {  
    position: fixed;  
    top: 30px;  
    right: 5px;  
}  
  
<!DOCTYPE html>  
<html>  
<head>  
<style>  
p.pos\_fixed {  
    position: fixed;  
    top: 30px;  
    right: 5px;  
    color: red;  
}  
</style>  
</head>  
<body>  
  
<p><b>Note:</b> IE7 and IE8 supports the fixed value only if a !DOCTYPE is specified.</p>  
<p>Some text</p><p>Some text</p><p>Some text</p><p>Some text</p><p>Some text</p><p>Some text</p><p>Some text</p><p>Some text</p><p>Some text</p><p>Some text</p><p>Some text</p><p>Some text</p><p>Some text</p><p>Some text</p><p>Some text</p><p>Some text</p>  
<p class="pos\_fixed">Some positioned text.</p>  
</body>  
</html>  
  
  
  
Note    Note: IE7 and IE8 support the fixed value only if a !DOCTYPE is specified.  
Fixed positioned elements are removed from the normal flow. The document and other elements behave like the fixed positioned element does not exist.  
  
Fixed positioned elements can overlap other elements.  
  
Relative Positioning  
A relative positioned element is positioned relative to its normal position.  
  
Example  
  
h2.pos\_left {  
    position: relative;  
    left: -20px;  
}  
  
h2.pos\_right {  
    position: relative;  
    left: 20px;  
}  
  
<!DOCTYPE html>  
<html>  
<head>  
<style>  
h2.pos\_left {  
    position: relative;  
    left: -20px;  
}  
  
h2.pos\_right {  
    position: relative;  
    left: 20px;  
}  
</style>  
</head>  
<body>  
  
<h2>Heading with no position</h2>  
<h2 class="pos\_left">This heading is moved left according to its normal position</h2>  
<h2 class="pos\_right">This heading is moved right according to its normal position</h2>  
<p>Relative positioning moves an element RELATIVE to its original position.</p>  
<p>The style "left:-20px" subtracts 20 pixels from the element's original left position.</p>  
<p>The style "left:20px" adds 20 pixels to the element's original left position.</p>  
  
</body>  
</html>  
  
The content of relatively positioned elements can be moved and overlap other elements, but the reserved space for the element is still preserved in the normal flow.  
  
Example  
  
h2.pos\_top {  
    position: relative;  
    top: -50px;  
}  
  
<!DOCTYPE html>  
<html>  
<head>  
<style>  
h2.pos\_top {  
    position: relative;  
    top: -30px;  
}  
</style>  
</head>  
<body>  
  
<h2>Heading with no position</h2>  
<h2 class="pos\_top">This heading is moved upwards according to its normal position</h2>  
<p><b>Note:</b> Even if the content of the relatively positioned element is moved, the reserved space for the element is still preserved in the normal flow.</p>  
  
</body>  
</html>  
  
Relatively positioned elements are often used as container blocks for absolutely positioned elements.  
  
Absolute Positioning  
An absolute position element is positioned relative to the first parent element that has a position other than static. If no such element is found, the containing block is <html>:  
  
Example  
  
h2 {  
    position: absolute;  
    left: 100px;  
    top: 150px;  
}  
  
<!DOCTYPE html>  
<html>  
<head>  
<style>  
h2 {  
    position: absolute;  
    left: 100px;  
    top: 150px;  
}  
</style>  
</head>  
<body>  
  
<h2>This heading has an absolute position</h2>  
<p>With absolute positioning, an element can be placed anywhere on a page. The heading below is placed 100px from the left of the page and 150px from the top of the page.</p>  
  
</body>  
</html>  
  
Absolutely positioned elements are removed from the normal flow. The document and other elements behave like the absolutely positioned element does not exist.  
  
Absolutely positioned elements can overlap other elements.  
  
Overlapping Elements  
When elements are positioned outside the normal flow, they can overlap other elements.  
  
The z-index property specifies the stack order of an element (which element should be placed in front of, or behind, the others).  
  
An element can have a positive or negative stack order:  
  
Example  
  
img {  
    position: absolute;  
    left: 0px;  
    top: 0px;  
    z-index: -1;  
}  
  
<!DOCTYPE html>  
<html>  
<head>  
<style>  
img {  
    position: absolute;  
    left: 0px;  
    top: 0px;  
    z-index: -1;  
}  
</style>  
</head>  
<body>  
  
<h1>This is a heading</h1>  
<img src="w3css.gif" width="100" height="140">  
<p>Because the image has a z-index of -1, it will be placed behind the text.</p>  
  
</body>  
</html>  
  
An element with greater stack order is always in front of an element with a lower stack order.  
  
Note    Note: If two positioned elements overlap without a z-index specified, the element positioned last in the HTML code will be shown on top.  
  
Examples  
More Examples  
Set the shape of an element  
<!DOCTYPE html>  
<html>  
<head>  
<style>  
img {  
    position: absolute;  
    clip: rect(0px,60px,200px,0px);  
}  
</style>  
</head>  
<body>  
  
<img src="w3css.gif" width="100" height="140">  
  
</body>  
</html>  
  
This example demonstrates how to set the shape of an element. The element is clipped into this shape, and displayed.  
  
How to show overflow in an element using scroll  
<!DOCTYPE html>  
<html>  
<head>  
<style>  
div.scroll {  
    background-color: #00FFFF;  
    width: 100px;  
    height: 100px;  
    overflow: scroll;  
}  
  
div.hidden {  
    background-color: #00FF00;  
    width: 100px;  
    height: 100px;  
    overflow: hidden;  
}  
</style>  
</head>  
<body>  
  
<p>The overflow property specifies what to do if the content of an element exceeds the size of the element's box.</p>  
  
<p>Result with overflow:scroll</p>  
<div class="scroll">You can use the overflow property when you want to have better control of the layout. The default value is visible.</div>  
  
<p>Result with overflow:hidden</p>  
<div class="hidden">You can use the overflow property when you want to have better control of the layout. The default value is visible.</div>  
  
</body>  
</html>  
  
This example demonstrates how to set the overflow property to create a scroll bar when an element's content is too big to fit in a specified area.  
  
How to set the browser to automatically handle overflow  
<!DOCTYPE html>  
<html>  
<head>  
<style>  
div {  
    background-color: #00FFFF;  
    width: 150px;  
    height: 150px;  
    overflow: auto;  
}  
</style>  
</head>  
<body>  
  
<p>The overflow property decides what to do if the content inside an element exceeds the given width and height properties.</p>  
<div>You can use the overflow property when you want to have better control of the layout. Try to change the overflow property to: visible, hidden, scroll, or inherit and see what happens. The default value is visible.</div>  
  
</body>  
</html>  
  
This example demonstrates how to set the browser to automatically handle overflow.  
  
Change the cursor  
<!DOCTYPE html>  
<html>  
<body>  
<p>Mouse over the words to change the cursor.</p>  
<span style="cursor:auto">auto</span><br>  
<span style="cursor:crosshair">crosshair</span><br>  
<span style="cursor:default">default</span><br>  
<span style="cursor:e-resize">e-resize</span><br>  
<span style="cursor:help">help</span><br>  
<span style="cursor:move">move</span><br>  
<span style="cursor:n-resize">n-resize</span><br>  
<span style="cursor:ne-resize">ne-resize</span><br>  
<span style="cursor:nw-resize">nw-resize</span><br>  
<span style="cursor:pointer">pointer</span><br>  
<span style="cursor:progress">progress</span><br>  
<span style="cursor:s-resize">s-resize</span><br>  
<span style="cursor:se-resize">se-resize</span><br>  
<span style="cursor:sw-resize">sw-resize</span><br>  
<span style="cursor:text">text</span><br>  
<span style="cursor:w-resize">w-resize</span><br>  
<span style="cursor:wait">wait</span><br>  
</body>  
</html>  
  
This example demonstrates how to change the cursor.  
  
All CSS Positioning Properties  
Property    Description    Values  
bottom    Sets the bottom margin edge for a positioned box    auto  
length  
%  
inherit  
clip    Clips an absolutely positioned element    shape  
auto  
inherit  
cursor    Specifies the type of cursor to be displayed    url  
auto  
crosshair  
default  
pointer  
move  
e-resize  
ne-resize  
nw-resize  
n-resize  
se-resize  
sw-resize  
s-resize  
w-resize  
text  
wait  
help  
left    Sets the left margin edge for a positioned box    auto  
length  
%  
inherit  
overflow  
Specifies what happens if content overflows an element's box    auto  
hidden  
scroll  
visible  
inherit  
position    Specifies the type of positioning for an element    absolute  
fixed  
relative  
static  
inherit  
right    Sets the right margin edge for a positioned box    auto  
length  
%  
inherit  
top    Sets the top margin edge for a positioned box    auto  
length  
%  
inherit  
z-index    Sets the stack order of an element    number  
auto  
inherit

C19

How Elements Float  
Elements are floated horizontally, this means that an element can only be floated left or right, not up or down.  
  
A floated element will move as far to the left or right as it can. Usually this means all the way to the left or right of the containing element.  
  
The elements after the floating element will flow around it.  
  
The elements before the floating element will not be affected.  
  
If an image is floated to the right, a following text flows around it, to the left:  
  
Example  
  
img {  
    float: right;  
}  
  
<!DOCTYPE html>  
<html>  
<head>  
<style>  
img {  
    float: right;  
}  
</style>  
</head>  
<body>  
  
<p>In the paragraph below, we have added an image with style <b>float:right</b>. The result is that the image will float to the right in the paragraph.</p>  
<p><img src="w3css.gif" width="100" height="140">  
This is some text. This is some text. This is some text.  
This is some text. This is some text. This is some text.  
This is some text. This is some text. This is some text.  
This is some text. This is some text. This is some text.  
This is some text. This is some text. This is some text.  
This is some text. This is some text. This is some text.  
This is some text. This is some text. This is some text.  
This is some text. This is some text. This is some text.  
This is some text. This is some text. This is some text.  
This is some text. This is some text. This is some text.  
This is some text. This is some text. This is some text.  
This is some text. This is some text. This is some text.  
This is some text. This is some text. This is some text.  
</p>  
  
</body>  
</html>  
  
  
Floating Elements Next to Each Other  
If you place several floating elements after each other, they will float next to each other if there is room.  
  
Here we have made an image gallery using the float property:  
  
Example  
  
.thumbnail {  
    float: left;  
    width: 110px;  
    height: 90px;  
    margin: 5px;  
}  
  
<!DOCTYPE html>  
<html>  
<head>  
<style>  
.thumbnail {  
    float: left;  
    width: 110px;  
    height: 90px;  
    margin: 5px;  
}  
</style>  
</head>  
<body>  
  
<h3>Image Gallery</h3>  
<p>Try to resize the browser-window to see what happens when the images do not have enough room.</p>  
<img class="thumbnail" src="klematis\_small.jpg" width="107" height="90">  
<img class="thumbnail" src="klematis2\_small.jpg" width="107" height="80">  
<img class="thumbnail" src="klematis3\_small.jpg" width="116" height="90">  
<img class="thumbnail" src="klematis4\_small.jpg" width="120" height="90">  
<img class="thumbnail" src="klematis\_small.jpg" width="107" height="90">  
<img class="thumbnail" src="klematis2\_small.jpg" width="107" height="80">  
<img class="thumbnail" src="klematis3\_small.jpg" width="116" height="90">  
<img class="thumbnail" src="klematis4\_small.jpg" width="120" height="90">  
  
</body>  
</html>  
  
  
Turning off Float - Using Clear  
Elements after the floating element will flow around it. To avoid this, use the clear property.  
  
The clear property specifies which sides of an element other floating elements are not allowed.  
  
Add a text line into the image gallery, using the clear property:  
  
Example  
  
.text\_line {  
    clear: both;  
}  
  
<!DOCTYPE html>  
<html>  
<head>  
<style>  
.thumbnail {  
    float: left;  
    width: 110px;  
    height: 90px;  
    margin: 5px;  
}  
  
.text\_line {  
    clear: both;  
    margin-bottom: 2px;  
}  
</style>  
</head>  
<body>  
  
<h3>Image Gallery</h3>  
<p>Try to resize the browser-window to see what happens when the images does not have enough room.</p>  
<img class="thumbnail" src="klematis\_small.jpg" width="107" height="90">  
<img class="thumbnail" src="klematis2\_small.jpg" width="107" height="80">  
<img class="thumbnail" src="klematis3\_small.jpg" width="116" height="90">  
<img class="thumbnail" src="klematis4\_small.jpg" width="120" height="90">  
  
<h3 class="text\_line">Second row</h3>  
<img class="thumbnail" src="klematis\_small.jpg" width="107" height="90">  
<img class="thumbnail" src="klematis2\_small.jpg" width="107" height="80">  
<img class="thumbnail" src="klematis3\_small.jpg" width="116" height="90">  
<img class="thumbnail" src="klematis4\_small.jpg" width="120" height="90">  
  
</body>  
</html>  
  
  
Examples  
More Examples  
An image with border and margins that floats to the right in a paragraph  
<!DOCTYPE html>  
<html>  
<head>  
<style>  
img {  
    float: right;  
    border: 1px dotted black;  
    margin: 0px 0px 15px 20px;  
}  
</style>  
</head>  
<body>  
  
<p>In the paragraph below, the image will float to the right. A dotted black border is added to the image.  
We have also added margins to the image to push the text away from the image:  
0 px margin on the top and right side, 15 px margin on the bottom, and 20 px margin on the left side of the image.  
</p>  
  
<p><img src="w3css.gif" width="100" height="140">  
This is some text. This is some text. This is some text.  
This is some text. This is some text. This is some text.  
This is some text. This is some text. This is some text.  
This is some text. This is some text. This is some text.  
This is some text. This is some text. This is some text.  
This is some text. This is some text. This is some text.  
This is some text. This is some text. This is some text.  
This is some text. This is some text. This is some text.  
This is some text. This is some text. This is some text.  
This is some text. This is some text. This is some text.  
This is some text. This is some text. This is some text.  
This is some text. This is some text. This is some text.  
This is some text. This is some text. This is some text.  
</p>  
  
</body>  
</html>  
  
Let an image float to the right in a paragraph. Add border and margins to the image.  
  
An image with a caption that floats to the right  
<!DOCTYPE html>  
<html>  
<head>  
<style>  
div {  
    float: right;  
    width: 120px;  
    margin: 0 0 15px 20px;  
    padding: 15px;  
    border: 1px solid black;  
    text-align: center;  
}  
</style>  
</head>  
<body>  
  
<div>  
<img src="w3css.gif" width="100" height="140"><br>CSS is fun!  
</div>  
  
<p>  
This is some text. This is some text. This is some text.  
This is some text. This is some text. This is some text.  
This is some text. This is some text. This is some text.  
This is some text. This is some text. This is some text.  
This is some text. This is some text. This is some text.  
This is some text. This is some text. This is some text.  
This is some text. This is some text. This is some text.  
This is some text. This is some text. This is some text.  
This is some text. This is some text. This is some text.  
This is some text. This is some text. This is some text.  
This is some text. This is some text. This is some text.  
This is some text. This is some text. This is some text.  
This is some text. This is some text. This is some text.  
</p>  
  
<p>  
In the paragraph above, the div element is 120 pixels wide and it contains the image.  
The div element will float to the right. Margins are added to the div to push the text away from the div.  
Borders and padding are added to the div to frame in the picture and the caption.  
</p>  
  
</body>  
</html>  
  
Let an image with a caption float to the right.  
  
Let the first letter of a paragraph float to the left  
<!DOCTYPE html>  
<html>  
<head>  
<style>  
span {  
    float: left;  
    width: 0.7em;  
    font-size: 400%;  
    font-family: algerian, courier;  
    line-height: 80%;  
}  
</style>  
</head>  
<body>  
  
<p>  
<span>T</span>his is some text.  
This is some text. This is some text.  
This is some text. This is some text. This is some text.  
This is some text. This is some text. This is some text.  
This is some text. This is some text. This is some text.  
This is some text. This is some text. This is some text.  
This is some text. This is some text. This is some text.  
This is some text. This is some text. This is some text.  
</p>  
  
<p>  
In the paragraph above, the first letter of the text is embedded in a span element.  
The span element has a width that is 0.7 times the size of the current font.  
The font-size of the span element is 400% (quite large) and the line-height is 80%.  
The font of the letter in the span will be in "Algerian".  
</p>  
  
</body>  
</html>  
  
Let the first letter of a paragraph float to the left and style the letter.  
  
Creating a horizontal menu  
<!DOCTYPE html>  
<html>  
<head>  
<style>  
ul {  
    float: left;  
    width: 100%;  
    padding: 0;  
    margin: 0;  
    list-style-type: none;  
}  
  
a {  
    float: left;  
    width: 6em;  
    text-decoration: none;  
    color: white;  
    background-color: purple;  
    padding: 0.2em 0.6em;  
    border-right: 1px solid white;  
}  
  
a:hover {  
    background-color: fuchsia;  
}  
  
li {  
    display: inline;  
}  
</style>  
</head>  
<body>  
  
<ul>  
<li><a href="#">Link one</a></li>  
<li><a href="#">Link two</a></li>  
<li><a href="#">Link three</a></li>  
<li><a href="#">Link four</a></li>  
</ul>  
  
<p>In the example above, we let the ul element and the a element float to the left.  
li elements will be displayed as inline elements (no line break before or after the element). This forces the list to be on one line.  
The ul element has a width of 100% and each hyperlink in the list has a width of 6em (6 times the size of the current font).  
We add some colors and borders to make it more fancy.  
</p>  
  
</body>  
</html>  
  
Use float with a list of hyperlinks to create a horizontal menu.  
  
Creating a homepage without tables  
<!DOCTYPE html>  
<html>  
<head>  
<style>  
div.container {  
    width: 100%;  
    margin: 0px;  
    border: 1px solid gray;  
    line-height: 150%;  
}  
  
div.header, div.footer {  
    padding: 0.5em;  
    color: white;  
    background-color: gray;  
    clear: left;  
}  
  
h1.header {  
    padding: 0;  
    margin: 0;  
}  
  
div.left {  
    float: left;  
    width: 160px;  
    margin: 0;  
    padding: 1em;  
}  
  
div.content {  
    margin-left: 190px;  
    border-left: 1px solid gray;  
    padding: 1em;  
}  
</style>  
</head>  
<body>  
  
<div class="container">  
  <div class="header"><h1 class="header">W3Schools.com</h1></div>  
  <div class="left"><p>"Never increase, beyond what is necessary, the number of entities required to explain anything." William of Ockham (1285-1349)</p></div>  
  <div class="content">  
    <h2>Free Web Building Tutorials</h2>  
    <p>At W3Schools you will find all the Web-building tutorials you need.</p>  
    <p>W3Schools.com - The largest Web Developers' Site on the internet!</p>  
  </div>  
  <div class="footer">© Copyright by Refsnes Data.</div>  
</div>  
  
</body>  
</html>  
  
Use float to create a homepage with a header, footer, left content and main content.  
  
All CSS Float Properties  
Property    Description    Values  
clear    Specifies which sides of an element where other floating elements are not allowed    left  
right  
both  
none  
inherit  
float    Specifies whether or not a box should float    left  
right  
none  
inherit

C20

Aligning Block Elements  
A block element is an element that takes up the full width available, and has a line break before and after it.  
  
Examples of block elements:  
  
<h1>  
<p>  
<div>  
For aligning text, see the CSS Text chapter.  
  
In this chapter we will show you how to horizontally align block elements for layout purposes.  
  
Center Aligning Using the margin Property  
Block elements can be center-aligned by setting the left and right margins to "auto".  
  
Note    Note: Using margin:auto; will not work in IE8 and earlier, unless a !DOCTYPE is declared.  
Setting the left and right margins to auto specifies that they should split the available margin equally. The result is a centered element:  
  
Example  
  
.center {  
    margin-left: auto;  
    margin-right: auto;  
    width: 70%;  
    background-color: #b0e0e6;  
}  
  
<!DOCTYPE html>  
<html>  
<head>  
<style>  
.center {  
    margin: auto;  
    width: 70%;  
    background-color: #b0e0e6;  
}  
</style>  
</head>  
<body>  
  
<div class="center">  
  <p>In my younger and more vulnerable years my father gave me some advice that I've been turning over in my mind ever since.</p>  
  <p>'Whenever you feel like criticizing anyone,' he told me, 'just remember that all the people in this world haven't had the advantages that you've had.'</p>  
</div>  
  
<p><b>Note: </b>Using margin:auto will not work in IE8, unless a !DOCTYPE is declared.</p>  
  
</body>  
</html>  
  
Tip: Center-aligning has no effect if the width is 100%.  
  
Left and Right Aligning Using the position Property  
One method of aligning elements is to use absolute positioning:  
  
Example  
  
.right {  
    position: absolute;  
    right: 0px;  
    width: 300px;  
    background-color: #b0e0e6;  
}  
  
<!DOCTYPE html>  
<html>  
<head>  
<style>  
.right {  
    position: absolute;  
    right: 0px;  
    width: 300px;  
    background-color: #b0e0e6;  
}  
</style>  
</head>  
<body>  
  
<div class="right">  
  <p>In my younger and more vulnerable years my father gave me some advice that I've been turning over in my mind ever since.</p>  
  <p>'Whenever you feel like criticizing anyone,' he told me, 'just remember that all the people in this world haven't had the advantages that you've had.'</p>  
</div>  
  
</body>  
</html>  
  
Note: Absolute positioned elements are removed from the normal flow, and can overlap elements.  
  
Crossbrowser Compatibility Issues  
When aligning elements like this, it is always a good idea to predefine margin and padding for the <body> element. This is to avoid visual differences in different browsers.  
  
There is a problem with IE8 and earlier, when using the position property. If a container element (in our case <div class="container">) has a specified width, and the !DOCTYPE declaration is missing, IE8 and earlier versions will add a 17px margin on the right side. This seems to be space reserved for a scrollbar. Always set the !DOCTYPE declaration when using the position property:  
  
Example  
  
body {  
    margin: 0;  
    padding: 0;  
}  
  
.container {  
    position: relative;  
    width: 100%;  
}  
  
.right {  
    position: absolute;  
    right: 0px;  
    width: 300px;  
    background-color: #b0e0e6;  
}  
  
<!DOCTYPE html>  
<html>  
<head>  
<style>  
body {  
    margin: 0;  
    padding: 0;  
}  
  
.container {  
    position: relative;  
    width: 100%;  
}  
  
.right {  
    position: absolute;  
    right: 0px;  
    width: 300px;  
    background-color: #b0e0e6;  
}  
</style>  
</head>  
<body>  
  
<div class="container">  
  <div class="right">  
    <p><b>Note: </b>When aligning using the position property, always include the !DOCTYPE declaration! If missing, it can produce strange results in IE browsers.</p>  
  </div>  
</div>  
  
</body>  
</html>  
  
  
Left and Right Aligning Using the float Property  
One method of aligning elements is to use the float property:  
  
Example  
  
.right {  
    float: right;  
    width: 300px;  
    background-color: #b0e0e6;  
}  
  
<!DOCTYPE html>  
<html>  
<head>  
<style>  
.right {  
    float: right;  
    width: 300px;  
    background-color: #b0e0e6;  
}  
</style>  
</head>  
<body>  
  
<div class="right">  
  <p>In my younger and more vulnerable years my father gave me some advice that I've been turning over in my mind ever since.</p>  
  <p>'Whenever you feel like criticizing anyone,' he told me, 'just remember that all the people in this world haven't had the advantages that you've had.'</p>  
</div>  
  
</body>  
</html>  
  
  
Crossbrowser Compatibility Issues  
When aligning elements like this, it is always a good idea to predefine margin and padding for the <body> element. This is to avoid visual differences in different browsers.  
  
There is a problem with IE8 and earlier when using the float property. If the !DOCTYPE declaration is missing, IE8 and earlier versions will add a 17px margin on the right side. This seems to be space reserved for a scrollbar. Always set the !DOCTYPE declaration when using the float property:  
  
Example  
  
body {  
    margin: 0;  
    padding: 0;  
}  
  
.right {  
    float: right;  
    width: 300px;  
    background-color: #b0e0e6;  
}  
  
<!DOCTYPE html>  
<html>  
<head>  
<style>  
body {  
    margin: 0;  
    padding: 0;  
}  
  
.right {  
    float: right;  
    width: 300px;  
    background-color: #b0e0e6;  
}  
</style>  
</head>  
<body>  
  
<div class="right">  
  <p><b>Note: </b>When aligning using the float property, always include the !DOCTYPE declaration! If missing, it can produce strange results in IE browsers.</p>  
</div>  
  
</body>  
</html>

C21

A CSS selector can contain more than one simple selector. Between the simple selectors, we can include a combinator.  
  
There are four different combinators in CSS3:  
  
descendant selector  
child selector  
adjacent sibling selector  
general sibling selector  
Descendant Selector  
The descendant selector matches all element that are descendants of a specified element.  
  
The following example selects all <p> elements inside <div> elements:  
  
Example  
  
div p {  
    background-color: yellow;  
}  
  
<!DOCTYPE html>  
<html>  
<head>  
<style>  
div p {  
    background-color: yellow;  
}  
</style>  
</head>  
<body>  
  
<div>  
<p>Paragraph 1 in the div.</p>  
<p>Paragraph 2 in the div.</p>  
</div>  
  
<p>Paragraph 3. Not in a div.</p>  
<p>Paragraph 4. Not in a div.</p>  
  
</body>  
</html>  
  
  
Child Selector  
The child selector selects all elements that are the immediate children of a specified element.  
  
The following example selects all <p> elements that are immediate children of a <div> element:  
  
Example  
  
div > p {  
    background-color: yellow;  
}  
  
<!DOCTYPE html>  
<html>  
<head>  
<style>  
div > p {  
    background-color: yellow;  
}  
</style>  
</head>  
<body>  
  
<div>  
<p>Paragraph 1 in the div.</p>  
<p>Paragraph 2 in the div.</p>  
</div>  
  
<p>Paragraph 3. Not in a div.</p>  
<p>Paragraph 4. Not in a div.</p>  
  
</body>  
</html>  
  
  
Adjacent Sibling Selector  
The adjacent sibling selector selects all elements that are the adjacent siblings of a specified element.  
  
Sibling elements must have the same parent element, and "adjacent" means "immediately following".  
  
The following example selects all <p> elements that are placed immediately after <div> elements:  
  
Example  
  
div + p {  
    background-color: yellow;  
}  
  
<!DOCTYPE html>  
<html>  
<head>  
<style>  
div + p {  
    background-color: yellow;  
}  
</style>  
</head>  
<body>  
  
<div>  
<p>Paragraph 1 in the div.</p>  
<p>Paragraph 2 in the div.</p>  
</div>  
  
<p>Paragraph 3. Not in a div.</p>  
<p>Paragraph 4. Not in a div.</p>  
  
</body>  
</html>  
  
  
General Sibling Selector  
The general sibling selector selects all elements that are siblings of a specified element.  
  
The following example selects all <p> elements that are siblings of <div> elements:  
  
Example  
  
div ~ p {  
    background-color: yellow;  
}  
  
<!DOCTYPE html>  
<html>  
<head>  
<style>  
div ~ p {  
    background-color: yellow;  
}  
</style>  
</head>  
<body>  
  
<div>  
<p>Paragraph 1 in the div.</p>  
<p>Paragraph 2 in the div.</p>  
</div>  
  
<p>Paragraph 3. Not in a div.</p>  
<p>Paragraph 4. Not in a div.</p>  
  
</body>  
</html>

C22

CSS pseudo-classes are used to add special effects to some selectors.  
  
Syntax  
The syntax of pseudo-classes:  
  
selector:pseudo-class {  
    property:value;  
}  
CSS classes can also be used with pseudo-classes:  
  
selector.class:pseudo-class {  
    property:value;  
}  
  
Anchor Pseudo-classes  
Links can be displayed in different ways in a CSS-supporting browser:  
  
Example  
  
/\* unvisited link \*/  
a:link {  
    color: #FF0000;  
}  
  
/\* visited link \*/  
a:visited {  
    color: #00FF00;  
}  
  
/\* mouse over link \*/  
a:hover {  
    color: #FF00FF;  
}  
  
/\* selected link \*/  
a:active {  
    color: #0000FF;  
}  
  
<!DOCTYPE html>  
<html>  
<head>  
<style>  
/\* unvisited link \*/  
a:link {  
    color: #FF0000;  
}  
  
/\* visited link \*/  
a:visited {  
    color: #00FF00;  
}  
  
/\* mouse over link \*/  
a:hover {  
    color: #FF00FF;  
}  
  
/\* selected link \*/  
a:active {  
    color: #0000FF;  
}  
</style>  
</head>  
<body>  
  
<p><b><a href="default.asp" target="\_blank">This is a link</a></b></p>  
<p><b>Note:</b> a:hover MUST come after a:link and a:visited in the CSS definition in order to be effective.</p>  
<p><b>Note:</b> a:active MUST come after a:hover in the CSS definition in order to be effective.</p>  
  
</body>  
</html>  
  
  
Note    Note: a:hover MUST come after a:link and a:visited in the CSS definition in order to be effective!!  
a:active MUST come after a:hover in the CSS definition in order to be effective!!  
Pseudo-class names are not case-sensitive.  
  
Pseudo-classes and CSS Classes  
Pseudo-classes can be combined with CSS classes:  
  
CSS:  
  
a.red:visited {  
    color: #FF0000;  
}  
  
HTML:  
  
<a class="red" href="css\_syntax.asp">CSS Syntax</a>  
If the link in the example above has been visited, it will be displayed in red.  
  
CSS - The :first-child Pseudo-class  
The :first-child pseudo-class matches a specified element that is the first child of another element.  
  
Note    Note: For :first-child to work in IE8 and earlier, a <!DOCTYPE> must be declared.  
Match the first <p> element  
In the following example, the selector matches any <p> element that is the first child of any element:  
  
Example  
  
p:first-child {  
    color: blue;  
}  
  
<!DOCTYPE html>  
<html>  
<head>  
<style>  
p:first-child {  
    color: blue;  
}  
</style>  
</head>  
<body>  
  
<p>This is some text.</p>  
<p>This is some text.</p>  
<p><b>Note:</b> For :first-child to work in IE8 and earlier, a DOCTYPE must be declared.</p>  
  
</body>  
</html>  
  
  
Match the first <i> element in all <p> elements  
In the following example, the selector matches the first <i> element in all <p> elements:  
  
Example  
  
p > i:first-child {  
    color: blue;  
}  
  
<!DOCTYPE html>  
<html>  
<head>  
<style>  
p > i:first-child {  
    color: blue;  
}  
</style>  
</head>  
<body>  
  
<p>I am a <i>strong</i> man. I am a <i>strong</i> man.</p>  
<p>I am a <i>strong</i> man. I am a <i>strong</i> man.</p>  
<p><b>Note:</b> For :first-child to work in IE8 and earlier, a DOCTYPE must be declared.</p>  
  
</body>  
</html>  
  
  
Match all <i> elements in all first child <p> elements  
In the following example, the selector matches all <i> elements in <p> elements that are the first child of another element:  
  
Example  
  
p:first-child i {  
    color: blue;  
}  
  
<!DOCTYPE html>  
<html>  
<head>  
<style>  
p:first-child i {  
    color: blue;  
}  
</style>  
</head>  
<body>  
  
<p>I am a <i>strong</i> man. I am a <i>strong</i> man.</p>  
<p>I am a <i>strong</i> man. I am a <i>strong</i> man.</p>  
<p><b>Note:</b> For :first-child to work in IE8 and earlier, a DOCTYPE must be declared.</p>  
  
</body>  
</html>  
  
  
CSS - The :lang Pseudo-class  
The :lang pseudo-class allows you to define special rules for different languages.  
  
Note    Note: IE8 supports the :lang pseudo-class only if a <!DOCTYPE> is specified.  
In the example below, the :lang class defines the quotation marks for q elements with lang="no":  
  
Example  
  
<html>  
<head>  
<style>  
q:lang(no) {  
    quotes: "~" "~";  
}  
</style>  
</head>  
  
<body>  
<p>Some text <q lang="no">A quote in a paragraph</q> Some text.</p>  
</body>  
</html>  
  
  
Examples  
More Examples  
Add different styles to hyperlinks  
<!DOCTYPE html>  
<html>  
<head>  
<style>  
a.one:link {color:#ff0000;}  
a.one:visited {color:#0000ff;}  
a.one:hover {color:#ffcc00;}  
  
a.two:link {color:#ff0000;}  
a.two:visited {color:#0000ff;}  
a.two:hover {font-size:150%;}  
  
a.three:link {color:#ff0000;}  
a.three:visited {color:#0000ff;}  
a.three:hover {background:#66ff66;}  
  
a.four:link {color:#ff0000;}  
a.four:visited {color:#0000ff;}  
a.four:hover {font-family:monospace;}  
  
a.five:link {color:#ff0000;text-decoration:none;}  
a.five:visited {color:#0000ff;text-decoration:none;}  
a.five:hover {text-decoration:underline;}  
</style>  
</head>  
<body>  
  
<p>Mouse over the links and watch them change layout:</p>  
  
<p><b><a class="one" href="default.asp" target="\_blank">This link changes color</a></b></p>  
<p><b><a class="two" href="default.asp" target="\_blank">This link changes font-size</a></b></p>  
<p><b><a class="three" href="default.asp" target="\_blank">This link changes background-color</a></b></p>  
<p><b><a class="four" href="default.asp" target="\_blank">This link changes font-family</a></b></p>  
<p><b><a class="five" href="default.asp" target="\_blank">This link changes text-decoration</a></b></p>  
  
</body>  
</html>  
  
This example demonstrates how to add other styles to hyperlinks.  
  
Use of :focus  
<!DOCTYPE html>  
<html>  
<head>  
<style>  
input:focus {  
    background-color: yellow;  
}  
</style>  
</head>  
<body>  
  
<form action="form\_action.asp" method="get">  
  First name: <input type="text" name="fname"><br>  
  Last name: <input type="text" name="lname"><br>  
  <input type="submit" value="Submit">  
</form>  
  
<p><b>Note:</b> IE8 supports the :focus pseudo-class only if a !DOCTYPE is specified.</p>  
  
</body>  
</html>  
  
This example demonstrates how to use the :focus pseudo-class.  
  
All CSS Pseudo Classes/Elements  
Selector    Example    Example description  
:link    a:link    Selects all unvisited links  
:visited    a:visited    Selects all visited links  
:active    a:active    Selects the active link  
:hover    a:hover    Selects links on mouse over  
:focus    input:focus    Selects the input element which has focus  
::first-letter    p::first-letter    Selects the first letter of every <p> element  
::first-line    p::first-line    Selects the first line of every <p> element  
:first-child    p:first-child    Selects every <p> elements that is the first child of its parent  
::before    p::before    Insert content before every <p> element  
::after    p::after    Insert content after every <p> element  
:lang(language)    p:lang(it)    Selects every <p> element with a lang attribute value starting with "it"

C23

CSS pseudo-elements are used to add special effects to some selectors.  
  
Syntax  
The syntax of pseudo-elements:  
  
selector::pseudo-element {  
    property:value;  
}  
CSS classes can also be used with pseudo-elements:  
  
selector.class::pseudo-element {  
    property:value;  
}  
  
The ::first-line Pseudo-element  
The ::first-line pseudo-element is used to add a special style to the first line of a text.  
  
The ::first-line pseudo-element can only be applied to block-level elements.  
  
Example  
  
Format the first line of the text in p elements:  
  
p::first-line {  
    color: #ff0000;  
    font-variant: small-caps;  
}  
  
<!DOCTYPE html>  
<html>  
<head>  
<style>  
p::first-line {  
    color: #ff0000;  
    font-variant: small-caps;  
}  
</style>  
</head>  
<body>  
  
<p>You can use the ::first-line pseudo-element to add a special effect to the first line of a text.</p>  
  
</body>  
</html>  
  
  
The following properties apply to the ::first-line pseudo-element:  
  
font properties  
color properties  
background properties  
word-spacing  
letter-spacing  
text-decoration  
vertical-align  
text-transform  
line-height  
clear  
The ::first-letter Pseudo-element  
The ::first-letter pseudo-element is used to add a special style to the first letter of a text.  
  
The ::first-letter pseudo-element can only be applied to block-level elements.  
  
Example  
  
Format the first letter of the text in p elements:  
  
p::first-letter {  
    color: #ff0000;  
    font-size: xx-large;  
}  
  
<!DOCTYPE html>  
<html>  
<head>  
<style>  
p::first-letter {  
    color: #ff0000;  
    font-size: xx-large;  
}  
</style>  
</head>  
<body>  
  
<p>You can use the ::first-letter pseudo-element to add a special effect to the first character of a text!</p>  
  
</body>  
</html>  
  
The following properties apply to the ::first-letter pseudo- element:  
  
font properties  
color properties  
background properties  
margin properties  
padding properties  
border properties  
text-decoration  
vertical-align (only if "float" is "none")  
text-transform  
line-height  
float  
clear  
Pseudo-elements and CSS Classes  
Pseudo-elements can be combined with CSS classes:  
  
CSS:  
  
p.article::first-letter {color:#ff0000;}  
  
HTML:  
  
<p class="article">A paragraph in an article</p>  
The example above will display the first letter of all paragraphs with class="article", in red.  
  
Multiple Pseudo-elements  
Several pseudo-elements can also be combined.  
  
In the following example, the first letter of a paragraph will be red, in an xx-large font size. The rest of the first line will be blue, and in small-caps. The rest of the paragraph will be the default font size and color:  
  
Example  
  
p::first-letter {  
    color: #ff0000;  
    font-size: xx-large;  
}  
  
p::first-line {  
    color: #0000ff;  
    font-variant: small-caps;  
}  
  
<!DOCTYPE html>  
<html>  
<head>  
<style>  
p::first-letter {  
    color: #ff0000;  
    font-size: xx-large;  
}  
  
p::first-line {  
    color: #0000ff;  
    font-variant: small-caps;  
}  
</style>  
</head>  
<body>  
  
<p>You can combine the ::first-letter and ::first-line pseudo-elements to add a special effect to the first letter and the first line of a text!</p>  
  
</body>  
</html>  
  
CSS - The ::before Pseudo-element  
The ::before pseudo-element can be used to insert some content before the content of an element.  
  
The following example inserts an image before each <h1> element:  
  
Example  
  
h1::before {  
    content: url(smiley.gif);  
}  
  
<!DOCTYPE html>  
<html>  
<head>  
<style>  
h1::before {  
    content: url(smiley.gif);  
}  
</style>  
</head>  
<body>  
  
<h1>This is a heading</h1>  
<p>The ::before pseudo-element inserts content before an element.</p>  
  
<h1>This is a heading</h1>  
<p><b>Note:</b> IE8 supports the content property only if a !DOCTYPE is specified.</p>  
  
</body>  
</html>  
  
  
CSS - The ::after Pseudo-element  
The ::after pseudo-element can be used to insert some content after the content of an element.  
  
The following example inserts an image after each <h1> element:  
  
Example  
  
h1::after {  
    content: url(smiley.gif);  
}  
  
<!DOCTYPE html>  
<html>  
<head>  
<style>  
h1::after {  
    content: url(smiley.gif);  
}  
</style>  
</head>  
<body>  
  
<h1>This is a heading</h1>  
<p>The ::after pseudo-element inserts content after an element.</p>  
  
<h1>This is a heading</h1>  
<p><b>Note:</b> IE8 supports the content property only if a !DOCTYPE is specified.</p>  
  
</body>  
</html>  
  
  
All CSS Pseudo Classes/Elements  
Selector    Example    Example description  
:link    a:link    Selects all unvisited links  
:visited    a:visited    Selects all visited links  
:active    a:active    Selects the active link  
:hover    a:hover    Selects links on mouse over  
:focus    input:focus    Selects the input element which has focus  
::first-letter    p::first-letter    Selects the first letter of every <p> element  
::first-line    p::first-line    Selects the first line of every <p> element  
:first-child    p:first-child    Selects every <p> elements that is the first child of its parent  
::before    p::before    Insert content before every <p> element  
::after    p::after    Insert content after every <p> element  
:lang(language)    p:lang(it)    Selects every <p> element with a lang attribute value starting with "it"

C24

Navigation Bars  
Having easy-to-use navigation is important for any web site.  
  
With CSS you can transform boring HTML menus into good-looking navigation bars.  
  
Navigation Bar = List of Links  
A navigation bar needs standard HTML as a base.  
  
In our examples we will build the navigation bar from a standard HTML list.  
  
A navigation bar is basically a list of links, so using the <ul> and <li> elements makes perfect sense:  
  
Example  
  
<ul>  
  <li><a href="default.asp">Home</a></li>  
  <li><a href="news.asp">News</a></li>  
  <li><a href="contact.asp">Contact</a></li>  
  <li><a href="about.asp">About</a></li>  
</ul>  
  
<!DOCTYPE html>  
<html>  
<body>  
  
<ul>  
  <li><a href="#home">Home</a></li>  
  <li><a href="#news">News</a></li>  
  <li><a href="#contact">Contact</a></li>  
  <li><a href="#about">About</a></li>  
</ul>  
  
<p>Note: We use href="#" for test links. In a real web site this would be URLs.</p>  
  
</body>  
</html>  
  
Now let's remove the bullets and the margins and padding from the list:  
  
Example  
  
ul {  
    list-style-type: none;  
    margin: 0;  
    padding: 0;  
}  
  
<!DOCTYPE html>  
<html>  
<head>  
<style>  
ul {  
    list-style-type: none;  
    margin: 0;  
    padding: 0;  
}  
</style>  
</head>  
<body>  
  
<ul>  
  <li><a href="#home">Home</a></li>  
  <li><a href="#news">News</a></li>  
  <li><a href="#contact">Contact</a></li>  
  <li><a href="#about">About</a></li>  
</ul>  
  
</body>  
</html>  
  
Example explained:  
  
list-style-type: none - Removes the bullets. A navigation bar does not need list markers  
Setting margins and padding to 0 to remove browser default settings  
The code in the example above is the standard code used in both vertical, and horizontal navigation bars.  
  
Vertical Navigation Bar  
To build a vertical navigation bar we only need to style the <a> elements, in addition to the code above:  
  
Example  
  
a {  
    display: block;  
    width: 60px;  
}  
  
<!DOCTYPE html>  
<html>  
<head>  
<style>  
ul {  
    list-style-type: none;  
    margin: 0;  
    padding: 0;  
}  
  
a {  
    display: block;  
    width: 60px;  
    background-color: #dddddd;  
}  
</style>  
</head>  
<body>  
  
<ul>  
  <li><a href="#home">Home</a></li>  
  <li><a href="#news">News</a></li>  
  <li><a href="#contact">Contact</a></li>  
  <li><a href="#about">About</a></li>  
</ul>  
  
<p>A background color is added to the links to show the link area.</p>  
<p>Notice that the whole link area is clickable, not just the text.</p>  
  
</body>  
</html>  
  
Example explained:  
  
display: block - Displaying the links as block elements makes the whole link area clickable (not just the text), and it allows us to specify the width  
width: 60px - Block elements take up the full width available by default. We want to specify a 60 px width  
Tip: Also take a look at our fully styled vertical navigation bar example.  
  
Note    Note: Always specify the width for <a> elements in a vertical navigation bar. If you omit the width, IE6 can produce unexpected results.  
  
Horizontal Navigation Bar  
There are two ways to create a horizontal navigation bar. Using inline or floating list items.  
  
Both methods work fine, but if you want the links to be the same size, you have to use the floating method.  
  
Inline List Items  
One way to build a horizontal navigation bar is to specify the <li> elements as inline, in addition to the "standard" code above:  
  
Example  
  
li {  
    display: inline;  
}  
  
<!DOCTYPE html>  
<html>  
<head>  
<style>  
ul {  
    list-style-type: none;  
    margin: 0;  
    padding: 0;  
}  
  
li {  
    display: inline;  
}  
</style>  
</head>  
<body>  
  
<ul>  
  <li><a href="#home">Home</a></li>  
  <li><a href="#news">News</a></li>  
  <li><a href="#contact">Contact</a></li>  
  <li><a href="#about">About</a></li>  
</ul>  
  
</body>  
</html>  
  
Example explained:  
  
display: inline; - By default, <li> elements are block elements. Here, we remove the line breaks before and after each list item, to display them on one line  
Floating List Items  
In the example above the links have different widths.  
  
For all the links to have an equal width, float the <li> elements and specify a width for the <a> elements:  
  
Example  
  
li {  
    float: left;  
}  
  
a {  
    display: block;  
    width: 60px;  
}  
  
<!DOCTYPE html>  
<html>  
<head>  
<style>  
ul {  
    list-style-type: none;  
    margin: 0;  
    padding: 0;  
    overflow: hidden;  
}  
  
li {  
    float: left;  
}  
  
a {  
    display: block;  
    width: 60px;  
    background-color: #dddddd;  
}  
</style>  
</head>  
<body>  
  
<ul>  
  <li><a href="#home">Home</a></li>  
  <li><a href="#news">News</a></li>  
  <li><a href="#contact">Contact</a></li>  
  <li><a href="#about">About</a></li>  
</ul>  
  
<p><b>Note:</b> If a !DOCTYPE is not specified, floating items can produce unexpected results.</p>  
<p>A background color is added to the links to show the link area. The whole link area is clickable, not just the text.</p>  
<p><b>Note:</b> overflow:hidden is added to the ul element to prevent li elements from going outside of the list.</p>  
  
</body>  
</html>  
  
Example explained:  
  
float: left - use float to get block elements to slide next to each other  
display: block - Displaying the links as block elements makes the whole link area clickable (not just the text), and it allows us to specify the width  
width: 60px - Since block elements take up the full width available, they cannot float next to each other. We specify the width of the links to 60px

C25

Image Gallery  
The following image gallery is created with CSS:  
  
Example  
  
<html>  
<head>  
<style>  
div.img {  
    margin: 5px;  
    padding: 5px;  
    border: 1px solid #0000ff;  
    height: auto;  
    width: auto;  
    float: left;  
    text-align: center;  
}  
  
div.img img {  
    display: inline;  
    margin: 5px;  
    border: 1px solid #ffffff;  
}  
  
div.img a:hover img {  
    border:1px solid #0000ff;  
}  
  
div.desc {  
    text-align: center;  
    font-weight: normal;  
    width: 120px;  
    margin: 5px;  
}  
</style>  
</head>  
<body>  
  
<div class="img">  
  <a target="\_blank" href="klematis\_big.htm">  
    <img src="klematis\_small.jpg" alt="Klematis" width="110" height="90">  
  </a>  
  <div class="desc">Add a description of the image here</div>  
</div>  
<div class="img">  
  <a target="\_blank" href="klematis2\_big.htm">  
    <img src="klematis2\_small.jpg" alt="Klematis" width="110" height="90">  
  </a>  
  <div class="desc">Add a description of the image here</div>  
</div>  
<div class="img">  
  <a target="\_blank" href="klematis3\_big.htm">  
    <img src="klematis3\_small.jpg" alt="Klematis" width="110" height="90">  
  </a>  
  <div class="desc">Add a description of the image here</div>  
</div>  
<div class="img">  
  <a target="\_blank" href="klematis4\_big.htm">  
    <img src="klematis4\_small.jpg" alt="Klematis" width="110" height="90">  
  </a>  
  <div class="desc">Add a description of the image here</div>  
</div>  
  
</body>  
</html>

C26

Image Gallery  
The following image gallery is created with CSS:  
  
Example  
  
<html>  
<head>  
<style>  
div.img {  
    margin: 5px;  
    padding: 5px;  
    border: 1px solid #0000ff;  
    height: auto;  
    width: auto;  
    float: left;  
    text-align: center;  
}  
  
div.img img {  
    display: inline;  
    margin: 5px;  
    border: 1px solid #ffffff;  
}  
  
div.img a:hover img {  
    border:1px solid #0000ff;  
}  
  
div.desc {  
    text-align: center;  
    font-weight: normal;  
    width: 120px;  
    margin: 5px;  
}  
</style>  
</head>  
<body>  
  
<div class="img">  
  <a target="\_blank" href="klematis\_big.htm">  
    <img src="klematis\_small.jpg" alt="Klematis" width="110" height="90">  
  </a>  
  <div class="desc">Add a description of the image here</div>  
</div>  
<div class="img">  
  <a target="\_blank" href="klematis2\_big.htm">  
    <img src="klematis2\_small.jpg" alt="Klematis" width="110" height="90">  
  </a>  
  <div class="desc">Add a description of the image here</div>  
</div>  
<div class="img">  
  <a target="\_blank" href="klematis3\_big.htm">  
    <img src="klematis3\_small.jpg" alt="Klematis" width="110" height="90">  
  </a>  
  <div class="desc">Add a description of the image here</div>  
</div>  
<div class="img">  
  <a target="\_blank" href="klematis4\_big.htm">  
    <img src="klematis4\_small.jpg" alt="Klematis" width="110" height="90">  
  </a>  
  <div class="desc">Add a description of the image here</div>  
</div>  
  
</body>  
</html>

C27

Creating transparent images with CSS is easy.  
  
The CSS opacity property is a part of the W3C CSS3 recommendation.  
  
Example 1 - Creating a Transparent Image  
The CSS3 property for transparency is opacity.  
  
First we will show you how to create a transparent image with CSS.  
  
Regular image:  
  
klematis  
The same image with transparency:  
  
klematis  
Look at the following CSS:  
  
img {  
    opacity: 0.4;  
    filter: alpha(opacity=40); /\* For IE8 and earlier \*/  
}  
IE9, Firefox, Chrome, Opera, and Safari use the property opacity for transparency. The opacity property can take a value from 0.0 - 1.0. A lower value makes the element more transparent.  
  
IE8 and earlier use filter:alpha(opacity=x). The x can take a value from 0 - 100. A lower value makes the element more transparent.  
  
Example 2 - Image Transparency - Hover Effect  
Mouse over the images:  
  
klematis klematis  
The CSS looks like this:  
  
Example  
  
img {  
    opacity: 0.4;  
    filter: alpha(opacity=40); /\* For IE8 and earlier \*/  
}  
  
img:hover {  
    opacity: 1.0;  
    filter: alpha(opacity=100); /\* For IE8 and earlier \*/  
}  
  
<!DOCTYPE html>  
<html>  
<head>  
<style>  
img {  
    opacity: 0.4;  
    filter: alpha(opacity=40); /\* For IE8 and earlier \*/  
}  
  
img:hover {  
    opacity: 1.0;  
    filter: alpha(opacity=100); /\* For IE8 and earlier \*/  
}  
</style>  
</head>  
<body>  
  
<h1>Image Transparency</h1>  
<img src="klematis.jpg" width="150" height="113" alt="klematis">  
<img src="klematis2.jpg" width="150" height="113" alt="klematis">  
  
<p><b>Note:</b> In IE, a !DOCTYPE must be added for the :hover selector to work on other elements than the a element.</p>  
</body>  
</html>  
  
  
The first CSS block is similar to the code in Example 1. In addition, we have added what should happen when a user hover over one of the images. In this case we want the image to NOT be transparent when the user hover over it.  
  
The CSS for this is: opacity=1.  
  
IE8 and earlier: filter:alpha(opacity=100).  
  
When the mouse pointer moves away from the image, the image will be transparent again.  
  
Example 3 - Text in Transparent Box  
This is some text that is placed in the transparent box. This is some text that is placed in the transparent box. This is some text that is placed in the transparent box. This is some text that is placed in the transparent box. This is some text that is placed in the transparent box.  
  
The source code looks like this:  
  
Example  
  
<html>  
<head>  
<style>  
div.background {  
    width: 500px;  
    height: 250px;  
    background: url(klematis.jpg) repeat;  
    border: 2px solid black;  
}  
  
div.transbox {  
    width: 400px;  
    height: 180px;  
    margin: 30px 50px;  
    background-color: #ffffff;  
    border: 1px solid black;  
    opacity: 0.6;  
    filter: alpha(opacity=60); /\* For IE8 and earlier \*/  
}  
  
div.transbox p {  
    margin: 30px 40px;  
    font-weight: bold;  
    color: #000000;  
}  
</style>  
</head>  
<body>  
  
<div class="background">  
  <div class="transbox">  
    <p>This is some text that is placed in the transparent box.  
    This is some text that is placed in the transparent box.  
    This is some text that is placed in the transparent box.  
    This is some text that is placed in the transparent box.  
    This is some text that is placed in the transparent box.</p>  
  </div>  
</div>  
  
</body>  
</html>  
  
First, we create a div element (class="background") with a fixed height and width, a background image, and a border. Then we create a smaller div (class="transbox") inside the first div element. The "transbox" div have a fixed width, a background color, and a border - and it is transparent. Inside the transparent div, we add some text inside a p element.

C28

Image Sprites  
An image sprite is a collection of images put into a single image.  
  
A web page with many images can take a long time to load and generates multiple server requests.  
  
Using image sprites will reduce the number of server requests and save bandwidth.  
  
Image Sprites - Simple Example  
Instead of using three separate images, we use this single image ("img\_navsprites.gif"):  
  
navigation images  
  
With CSS, we can show just the part of the image we need.  
  
In the following example the CSS specifies which part of the "img\_navsprites.gif" image to show:  
  
Example  
  
img.home {  
    width: 46px;  
    height: 44px;  
    background: url(img\_navsprites.gif) 0 0;  
}  
  
<!DOCTYPE html>  
<html>  
<head>  
<style>  
img.home {  
    width: 46px;  
    height: 44px;  
    background: url(img\_navsprites.gif) 0 0;  
}  
  
img.next {  
    width: 43px;  
    height: 44px;  
    background: url(img\_navsprites.gif) -91px 0;  
}  
</style>  
</head>  
<body>  
  
<img class="home" src="img\_trans.gif"><br><br>  
<img class="next" src="img\_trans.gif">  
  
</body>  
</html>  
  
  
Example explained:  
  
<img class="home" src="img\_trans.gif"> - Only defines a small transparent image because the src attribute cannot be empty. The displayed image will be the background image we specify in CSS  
width: 46px; height: 44px; - Defines the portion of the image we want to use  
background: url(img\_navsprites.gif) 0 0; - Defines the background image and its position (left 0px, top 0px)  
This is the easiest way to use image sprites, now we want to expand it by using links and hover effects.  
  
Image Sprites - Create a Navigation List  
We want to use the sprite image ("img\_navsprites.gif") to create a navigation list.  
  
We will use an HTML list, because it can be a link and also supports a background image:  
  
Example  
  
#navlist {  
    position: relative;  
}  
  
#navlist li {  
    margin: 0;  
    padding: 0;  
    list-style: none;  
    position: absolute;  
    top: 0;  
}  
  
#navlist li, #navlist a {  
    height: 44px;  
    display: block;  
}  
  
#home {  
    left: 0px;  
    width: 46px;  
    background: url('img\_navsprites.gif') 0 0;  
}  
  
#prev {  
    left: 63px;  
    width: 43px;  
    background: url('img\_navsprites.gif') -47px 0;  
}  
  
#next {  
    left: 129px;  
    width: 43px;  
    background: url('img\_navsprites.gif') -91px 0;  
}  
  
<!DOCTYPE html>  
<html>  
<head>  
<style>  
#navlist {  
    position: relative;  
}  
  
#navlist li {  
    margin: 0;  
    padding: 0;  
    list-style: none;  
    position: absolute;  
    top: 0;  
}  
  
#navlist li, #navlist a {  
    height: 44px;  
    display: block;  
}  
  
#home {  
    left: 0px;  
    width: 46px;  
    background: url('img\_navsprites.gif') 0 0;  
}  
  
#prev {  
    left: 63px;  
    width: 43px;  
    background: url('img\_navsprites.gif') -47px 0;  
}  
  
#next {  
    left: 129px;  
    width: 43px;  
    background: url('img\_navsprites.gif') -91px 0;  
}  
</style>  
</head>  
<body>  
  
<ul id="navlist">  
  <li id="home"><a href="default.asp"></a></li>  
  <li id="prev"><a href="css\_intro.asp"></a></li>  
  <li id="next"><a href="css\_syntax.asp"></a></li>  
</ul>  
  
</body>  
</html>  
  
  
Example explained:  
  
#navlist {position:relative;} - position is set to relative to allow absolute positioning inside it  
#navlist li {margin:0;padding:0;list-style:none;position:absolute;top:0;} - margin and padding is set to 0, list-style is removed, and all list items are absolute positioned  
#navlist li, #navlist a {height:44px;display:block;} - the height of all the images are 44px  
Now start to position and style for each specific part:  
  
#home {left:0px;width:46px;} - Positioned all the way to the left, and the width of the image is 46px  
#home {background:url(img\_navsprites.gif) 0 0;} - Defines the background image and its position (left 0px, top 0px)  
#prev {left:63px;width:43px;} - Positioned 63px to the right (#home width 46px + some extra space between items), and the width is 43px.  
#prev {background:url('img\_navsprites.gif') -47px 0;} - Defines the background image 47px to the right (#home width 46px + 1px line divider)  
#next {left:129px;width:43px;}- Positioned 129px to the right (start of #prev is 63px + #prev width 43px + extra space), and the width is 43px.  
#next {background:url('img\_navsprites.gif') -91px 0;} - Defines the background image 91px to the right (#home width 46px + 1px line divider + #prev width 43px + 1px line divider )  
Image Sprites - Hover Effect  
Now we want to add a hover effect to our navigation list.  
  
Note    The :hover selector is used to select elements when you mouse over them.  
  
Tip: The :hover selector can be used on all elements, not only on links.  
Our new image ("img\_navsprites\_hover.gif") contains three navigation images and three images to use for hover effects:  
  
navigation images  
  
Because this is one single image, and not six separate files, there will be no loading delay when a user hovers over the image.  
  
We only add three lines of code to add the hover effect:  
  
Example  
  
#home a:hover {  
    background: url('img\_navsprites\_hover.gif') 0 -45px;  
}  
  
#prev a:hover {  
    background: url('img\_navsprites\_hover.gif') -47px -45px;  
}  
  
#next a:hover {  
    background: url('img\_navsprites\_hover.gif') -91px -45px;  
}  
  
<!DOCTYPE html>  
<html>  
<head>  
<style>  
#navlist {  
    position: relative;  
}  
  
#navlist li {  
    margin: 0;  
    padding: 0;  
    list-style: none;  
    position: absolute;  
    top: 0;  
}  
  
#navlist li, #navlist a {  
    height: 44px;  
    display: block;  
}  
  
#home {  
    left: 0px;  
    width: 46px;  
    background: url('img\_navsprites.gif') 0 0;  
}  
  
#prev {  
    left: 63px;  
    width: 43px;  
    background: url('img\_navsprites.gif') -47px 0;  
}  
  
#next {  
    left: 129px;  
    width: 43px;  
    background: url('img\_navsprites.gif') -91px 0;  
}  
  
#home a:hover {  
    background: url('img\_navsprites\_hover.gif') 0 -45px;  
}  
  
#prev a:hover {  
    background: url('img\_navsprites\_hover.gif') -47px -45px;  
}  
  
#next a:hover {  
    background: url('img\_navsprites\_hover.gif') -91px -45px;  
}  
</style>  
</head>  
<body>  
  
<ul id="navlist">  
  <li id="home"><a href="default.asp"></a></li>  
  <li id="prev"><a href="css\_intro.asp"></a></li>  
  <li id="next"><a href="css\_syntax.asp"></a></li>  
</ul>  
  
</body>  
</html>  
  
Example explained:  
  
#home a:hover {background: transparent url('img\_navsprites\_hover.gif') 0 -45px;} - For all three hover images we specify the same background position,    only 45px further down

C29

Media Types  
Some CSS properties are only designed for a certain media. For example the "voice-family" property is designed for aural user agents. Some other properties can be used for different media types. For example, the "font-size" property can be used for both screen and print media, but perhaps with different values. A document usually needs a larger font-size on a screen than on paper, and sans-serif fonts are easier to read on the screen, while serif fonts are easier to read on paper.  
  
The @media Rule  
The @media rule allows different style rules for different media in the same style sheet.  
  
The style in the example below tells the browser to display a 14 pixels Verdana font on the screen. But if the page is printed, it will be in a 20 pixels font, and in a red color:  
  
Example  
  
@media screen {  
    p {  
        font-family: verdana,sans-serif;  
        font-size: 14px;  
    }  
}  
  
@media print {  
    p {  
        font-size: 20px;  
        color: red;  
    }  
}  
  
<!DOCTYPE html>  
<html>  
<head>  
<style>  
@media screen {  
    p {  
        font-family: verdana,sans-serif;  
        font-size: 14px;  
    }  
}  
  
@media print {  
    p {  
        font-size: 20px;  
        color: red;  
    }  
}  
</style>  
</head>  
<body>  
  
<h2>The @media Rule</h2>  
<p>The @media rule allows different style rules for different media in the same style sheet.</p>  
<p>The style in this example tells the browser to display a 14 pixels Verdana font on the screen.  
However, if the page is printed, the text will be in 20 pixels Verdana font, and in a red color.</p>  
<p><b>See it yourself !</b> Print this page (or open Print Preview), and you will see  
that the text will be displayed in a larger font size, and in red color.</p>  
  
</body>  
</html>  
  
  
Other Media Types  
Media Type    Description  
all    Used for all media type devices  
aural    Used for speech and sound synthesizers  
braille    Used for braille tactile feedback devices  
embossed    Used for paged braille printers  
handheld    Used for small or handheld devices  
print    Used for printers  
projection    Used for projected presentations, like slides  
screen    Used for computer screens  
tty    Used for media using a fixed-pitch character grid, like teletypes and terminals  
tv    Used for television-type devices

C30

Style HTML Elements With Specific Attributes  
It is possible to style HTML elements that have specific attributes, not just class and id.  
  
Note    Note: IE7 and IE8 support attribute selectors only if a !DOCTYPE is specified.  
  
CSS [attribute] Selector  
The [attribute] selector is used to select elements with the specified attribute.  
  
The following example selects all <a> elements with a target attribute:  
  
Example  
  
a[target] {  
    background-color: yellow;  
}  
  
<!DOCTYPE html>  
<html>  
<head>  
<style>  
a[target] {  
    background-color: yellow;  
}  
</style>  
</head>  
<body>  
  
<p>The links with a target attribute gets a yellow background:</p>  
  
<a href="[http://www.w3schools.com](http://www.w3schools.com/)">[w3schools.com](http://w3schools.com/)</a>  
<a href="[http://www.disney.com](http://www.disney.com/)" target="\_blank">[disney.com](http://disney.com/)</a>  
<a href="[http://www.wikipedia.org](http://www.wikipedia.org/)" target="\_top">[wikipedia.org](http://wikipedia.org/)</a>  
  
<p><b>Note:</b> For [<i>attribute</i>] to work in IE8 and earlier, a DOCTYPE must be declared.</p>  
  
</body>  
</html>  
  
  
CSS [attribute=value] Selector  
The [attribute=value] selector is used to select elements with the specified attribute and value.  
  
The following example selects all <a> elements with a target="\_blank" attribute:  
  
Example  
  
a[target="\_blank"] {  
    background-color: yellow;  
}  
  
<!DOCTYPE html>  
<html>  
<head>  
<style>  
a[target=\_blank] {  
    background-color: yellow;  
}  
</style>  
</head>  
<body>  
  
<p>The link with target="\_blank" gets a yellow background:</p>  
  
<a href="[http://www.w3schools.com](http://www.w3schools.com/)">[w3schools.com](http://w3schools.com/)</a>  
<a href="[http://www.disney.com](http://www.disney.com/)" target="\_blank">[disney.com](http://disney.com/)</a>  
<a href="[http://www.wikipedia.org](http://www.wikipedia.org/)" target="\_top">[wikipedia.org](http://wikipedia.org/)</a>  
  
<p><b>Note:</b> For [<i>attribute</i>] to work in IE8 and earlier, a DOCTYPE must be declared.</p>  
</body>  
</html>  
  
  
CSS [attribute~=value] Selector  
The [attribute~=value] selector is used to select elements with an attribute value containing a specified word.  
  
The following example selects all elements with a title attribute that contains a space-separated list of words, one of which is "flower":  
  
Example  
  
[title~="flower"] {  
    border: 5px solid yellow;  
}  
  
<!DOCTYPE html>  
<html>  
<head>  
<style>  
[title~=flower] {  
    border: 5px solid yellow;  
}  
</style>  
</head>  
<body>  
  
<p>All images with the title attribute containing the word "flower" get a yellow border.</p>  
  
<img src="klematis.jpg" title="klematis flower" width="150" height="113">  
<img src="img\_flwr.gif" title="flower" width="224" height="162">  
<img src="img\_tree.gif" title="tree" width="200" height="358">  
  
<p><b>Note:</b> For [<i>attribute</i>~=<i>value</i>] to work in IE8 and earlier, a DOCTYPE must be declared.</p>  
  
</body>  
</html>  
  
The example above will match elements with title="flower", title="summer flower", and title="flower new", but not title="my-flower" or title="flowers".  
  
CSS [attribute|=value] Selector  
The [attribute|=value] selector is used to select elements with the specified attribute starting with the specified value.  
  
The following example selects all elements with a class attribute value that begins with "top":  
  
Note: The value has to be a whole word, either alone, like class="top", or followed by a hyphen( - ), like class="top-text"!  
  
Example  
  
[class|="top"] {  
    background: yellow;  
}  
  
<!DOCTYPE html>  
<html>  
<head>  
<style>  
[class|=top] {  
    background: yellow;  
}  
</style>  
</head>  
<body>  
  
<h1 class="top-header">Welcome</h1>  
<p class="top-text">Hello world!</p>  
<p class="topcontent">Are you learning CSS?</p>  
  
<p><b>Note:</b> For [<i>attribute</i>|=<i>value</i>] to work in IE8 and earlier, a DOCTYPE must be declared.</p>  
  
</body>  
</html>  
  
  
CSS [attribute^=value] Selector  
The [attribute^=value] selector is used to select elements whose attribute value begins with a specified value.  
  
The following example selects all elements with a class attribute value that begins with "top":  
  
Note: The value does not have to be a whole word!  
  
Example  
  
[class^="top"] {  
    background: yellow;  
}  
  
<!DOCTYPE html>  
<html>  
<head>  
<style>  
[class^="top"] {  
    background: yellow;  
}  
</style>  
</head>  
<body>  
  
<h1 class="top-header">Welcome</h1>  
<p class="top-text">Hello world!</p>  
<p class="topcontent">Are you learning CSS?</p>  
  
<p><b>Note:</b> For [<i>attribute</i>|=<i>value</i>] to work in IE8 and earlier, a DOCTYPE must be declared.</p>  
  
</body>  
</html>  
  
  
CSS [attribute$=value] Selector  
The [attribute$=value] selector is used to select elements whose attribute value ends with a specified value.  
  
The following example selects all elements with a class attribute value that ends with "test":  
  
Note: The value does not has to be a whole word!  
  
Example  
  
[class$="test"] {  
    background: yellow;  
}  
  
<!DOCTYPE html>  
<html>  
<head>  
<style>  
[class$="test"] {  
    background: yellow;  
}  
</style>  
</head>  
<body>  
  
<div class="first\_test">The first div element.</div>  
<div class="second">The second div element.</div>  
<div class="my-test">The third div element.</div>  
<p class="mytest">This is some text in a paragraph.</p>  
  
</body>  
</html>  
  
  
CSS [attribute\*=value] Selector  
The [attribute\*=value] selector is used to select elements whose attribute value contains a specified value.  
  
The following example selects all elements with a class attribute value that contains "te":  
  
Note: The value does not has to be a whole word!  
  
Example  
  
[class\*="te"] {  
    background: yellow;  
}  
  
<!DOCTYPE html>  
<html>  
<head>  
<style>  
[class\*="te"] {  
    background: yellow;  
}  
</style>  
</head>  
<body>  
  
<div class="first\_test">The first div element.</div>  
<div class="second">The second div element.</div>  
<div class="my-test">The third div element.</div>  
<p class="mytest">This is some text in a paragraph.</p>  
  
</body>  
</html>  
  
  
Styling Forms  
The attribute selectors can be useful for styling forms without class or ID:  
  
Example  
  
input[type="text"] {  
    width: 150px;  
    display: block;  
    margin-bottom: 10px;  
    background-color: yellow;  
}  
  
input[type="button"] {  
    width: 120px;  
    margin-left: 35px;  
    display: block;  
}  
  
<!DOCTYPE html>  
<html>  
<head>  
<style>  
input[type=text] {  
    width: 150px;  
    display: block;  
    margin-bottom: 10px;  
    background-color: yellow;  
}  
  
input[type=button] {  
    width: 120px;  
    margin-left: 35px;  
    display: block;  
}  
</style>  
</head>  
<body>  
  
<form name="input" action="" method="get">  
  Firstname:<input type="text" name="Name" value="Peter" size="20">  
  Lastname:<input type="text" name="Name" value="Griffin" size="20">  
  <input type="button" value="Example Button">  
</form>  
  
</body>  
</html>  
  
  
More Examples of CSS Selectors  
Use our CSS Selector Tester to demonstrate the different selectors.  
  
For a complete reference of all the CSS selectors, please go to our CSS Selectors Reference.