UsefulCharts.com *The Ultimate Quick Reference Guide for Web Designers* FREE

HTML / CSS Cheat Sheet

XHTML
CSS-3
HTML 5

HTML vs CSS

Although there is some overlap, HTML and CSS serve different functions:

HTML should be used for **structure** and **semantics**, ie. to order the content and to specify its intended use (e.g. - distinguishing between the main body of an article and the links to other sections)

CSS should be used for **presentation**, ie. to specify what the content should look like to a human user (e.g. - making the title appear red)

HTML Syntax

Most HTML elements have an opening and a closing tag that surround the content you want to mark: `<tag>...</tag>`

A few elements do not have a closing tag. In XHTML, these must be written `<tag />`

Elements with attributes are written like this: `<tag attribute="value">...</tag>`

Tags can go inside other tags like this: `<tag1><tag2>...</tag2></tag1>`

Please Note

This cheat sheet can be shared freely in both print and electronic formats. The originals (available in color and b/w) can be downloaded in PDF format for free at **usefulcharts.com**

A laminated version is also available for **$4.99** (Class sets are $29.99 for 10 sheets).

We welcome your suggestions for improving this cheat sheet. Send all correspondance to **matt@usefulcharts.com**

Anatomy of an HTML File

Aside from the `<!DOCTYPE>`, everything must go inside an `<html>` tag. From there, the file is divided into the `<head>` and `<body>` sections. Only a few elements are permitted inside the `<head>` tag, such as `<meta>`, `<link>`, `<script>`, and `<style>`. These give information about the file or link to other files needed to display it correctly. The page content itself goes inside the `<body>` tag.

```
<!DOCTYPE html>
<html>
<head>
<title>Page Title</title>
<meta http-equiv="content-type" content="text/html;charset=UTF-8" />
<meta name="description" content="Brief description of the page" />
<link href="styles.css" rel="stylesheet" type="text/css" />
</head>
<body>
Page content
</body>
</html>
```

Specifies which HTML version and standard you will be using

Specifies what type of characters you will be using (e.g. - Latin vs Chinese)

Points to your CSS file

This, along with the Page Title, is what shows up when a search engine finds the page.

Special Characters

SPACE	&nbsp;	‘	&lsquo;	
&	&amp;	’	&rsquo;	
<	&lt;	“	&ldquo;	
>	&gt;	”	&rdquo;	
¢	&cent;	•	&bull;	
£	&pound;	À	&Agrave;	*
€	&euro;	Á	&Aacute;	*
©	&copy;	Â	&Acirc;	*
®	&reg;	Ã	&Atilde;	*
™	&trade;	Ä	&Auml;	*
¼	&frac14;	à	&agrave;	*
½	&frac12;	á	&aacute;	*
¾	&frac34;	â	&acirc;	*
×	&times;	ã	&atilde;	*
÷	&divide;	ä	&auml;	*
–	&ndash;	ç	&ccedil;	
—	&mdash;	ñ	&ntilde;	

*Follow same pattern for E, e, I, i, O, o, U, u

HTML Block Elements

Block elements divide content into rectangular sections that can then be formatted by CSS. They are automatically followed by a line break. You can put block elements inside other block elements.

Element:	**Used for:**
`<div>`	basic block divisions
`<h1>`	the main heading
`<h2>` to `<h6>`	various levels of subheadings
`<p>`	paragraphs
`<address>`	addresses
`<blockquote>`	long quotations
`<hr />`	horizontal lines **no closing tag
`<pre>`	preformatted mono-space text

Block elements that must be used together:

Numbered List:

```
<ol>
<li>Item 1</li>
<li>Item 2</li>
</ol>
```

Bullet List:

```
<ul>
<li>Item 1</li>
<li>Item 2</li>
</ul>
```

Data Table:

```
<table>
<tr><th>Head</th><th>Head</th></tr>
<tr><td>Cell</td><td>Cell</td></tr>
<tr><td>Cell</td><td>Cell</td></tr>
</table>
```

Note: tables used to be used for page layout. This should now be avoided. Use CSS instead.

HTML Inline Elements

Inline elements must always go inside a block element. They are NOT followed by a line break. Inline elements can contain other inline elements but never a block element. They can be formatted by CSS.

Element:	**Used for:**
`<span>`	basic inline divisions
`<strong>`	strong emphasis (bold)
`<em>`	emphasis (italics)
`<del>`	deleted text (strikethrough)
`<ins>`	inserted text (underline)
`<q>`	short quotations (in quotes)
`<abbr>`	abbreviations (mouseover for more)
`<sup>`	superscript
`<sub>`	subscript
`<br />`	forced line breaks **no closing tag

Note: `<b>` and `<i>` can be used for bold and italics but carry no semantic value.

Inline elements requiring attributes:

Hypertext Links:

```
<a href="URL">Click</a>
```

To link within the same page, use these tags:

```
<a name="Anchor">Section B</a>
<a href="URL#Anchor">Go to B</a>
```

Images:

```
<img src="file.jpg" alt="Text" />
```

HTML Form Example

```
<form action="URL" method="post">
<div>
 Name: <input type="text"></input>
</div>
<select name="vote">
  <option>Yes</option>
  <option>No</option>
</select>
<div>
  <input type="submit"></input>
</div>
</form>
```

New HTML5 Elements

New block-level elements:

Element:	**Used for:**
`<article>`	main, independent content
`<section>`	a particular section of content
`<aside>`	a side point
`<header>`	page headers
`<footer>`	page footers
`<nav>`	menus and navigational items

New inline element: `<mark>` for highlighted text

Deprecated Elements

Avoid using the following out-of-date elements (use CSS instead): `<center>`, `<font>`, `<big>`, `<strike>`, `<s>`, `<u>`, and `<tt>`.

CSS Syntax

CSS can be added to any HTML tag like this:

```
<tag style="attribute:value;">
```

But most of the time it is placed in a separate file (or between `<style>...</style>` in the head of an HTML file) so that the styles can be applied to multiple tags at once. In this case, the syntax is:

```
selector {
  attribute: value;
  }
```

Selectors follow this format:

`div`	all div's
`div, p`	all div's and p's
`div p`	all p's within div's
`a:hover`	link with a mouse over it
`a:visited`	links that have been visited

Within HTML tags, the `id` and `class` attributes can be used to divide tags into groups for styling purposes. For example:

```
<tag id="foo"> <tag class="boo">
```

Note: use `id` when the formatting is used in only one place in a document, otherwise use `class`.

The names assigned can then be used as selectors in the CSS syntax:

`#foo`	the element with the id 'foo'
`div#foo`	the div with the id 'foo'
`.boo`	all elements with the class 'boo'
`div.boo`	all div's with the class 'boo'

The most common **unit of measurement** in CSS is pixels (`px`). 1px is equal to one dot on the computer screen. When dealing with text, `em` refers to the current font size (e.g. - if the font is 12px then 2em equals 24px).

Formatting Text in CSS

Font face, size, and/or color:

```
font-family: Arial;
font-size: 16px;
color: #ff0000;
```

Bold, italic, and/or underline:

```
font-weight: bold;
font-style: italic;
text-decoration: underline;
```

Align text in a box:

`text-align: center;` (or: `left`, `right`)
`vertical-align: middle;` (`top`, `bottom`)

Indent first line and double-space:

```
text-indent: 50px;
line-height: 2em;
```

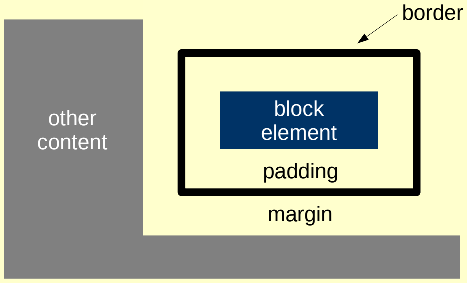
Other effects:

```
text-transform: uppercase;
font-variant: small-caps;
letter-spacing: 2px;
```

Bullets:

`list-style: disc;` (Also: `circle`, `square`, `decimal`, `lower-alpha`, `upper-alpha`, `lower-roman`, `upper-roman`)

CSS Box Model



To set the **dimensions** of a block element, use the following format (you can also use %):

```
width: 250px;
height: 50px;
```

(also available: `min-width`, `max-width`, etc.)

Margins and **Padding** can be set as follows (`margin` can be replaced with `padding`):

`margin: 1px;`
Sets all 4 margins to 1px

`margin: 1px 2px;`
Sets top/bottom to 1px and left/right to 2px

`margin: 1px 2px 3px 4px;`
Sets margins in this order: top; right, bottom, left

`margin-left: 1px;`
Sets only one margin (`left` can be replaced with `right`, `top`, or `bottom`)

Borders are set like this:

```
border: 1px solid #ff0000;
```

Or individually like this:

```
border-width: 1px;
border-style: solid;
border-color: #ff0000;
```

Or even `border-left-width`, etc.
Other border styles: `double`, `dashed`, `dotted`

Tables: If you want the cells in a table to share common border attributes rather than duplicate them, use the following for the table element:

```
border-collapse: collapse;
```

CSS Box Effects

Background:

```
background: #ff0000;
```

To use an image instead of a color:

```
url('file.jpg') no-repeat;
```

Rounded Corners:

```
border-radius: 12px;
```

To round only specific corners

`border-top-left-radius: 12px;` etc.

Shading:

`box-shadow: 10px 10px 5px #808080;`
(the first two values are the horizontal & vertical offsets and the third is the blur distance)

Transparency:

To make a box element (or image) partially transparent (useful for mouseovers), use:
`opacity: 0.3;` (gives 70% transparency)

Positioning in CSS

By default, block elements will appear one after each other with a single page break inbetween. However, there are several ways to alter this:

```
position: relative;
top: 10px;
left: 20px;
```

This will position the element 10px down from and 20px to the right of its usual position. Note: negative numbers are allowed.

Changing the position to `absolute` will place it 10px down and 20px over from the top-left of the page (or its not-static parent element).

Changing the position to `fixed` will place it 10px down and 20px over from the top-left of the browser window and fix it there so that it will not move even if the user scrolls down.

Floating:

`float: left;` or `float: right;`
This attribute will lift a box above the regular flow of the page and align it to either the parent element's left or right edge. Any block elements occurring after the floated box will be wrapped around it (unless there is in inadequate space).

Other Tricks:

In some cases, elements will overlap. To control the stack order, give each element a z-index:
`z-index: 1;`

Higher numbers will appear above lower numbers (negative numbers are allowed).

To make an inline element (such as `img`) behave like a block element, use:
`display: block;`

Basic Colors

maroon #800000	red #ff0000	orange #ffa500
olive #808000	yellow #ffff00	white #ffffff
green #008000	lime #00ff00	silver #c0c0c0
teal #008080	aqua #00ffff	gray #808080
navy #000080	blue #0000ff	black #000000
purple #800080	fuchsia #ff00ff	

The six-digit hex code or the color name can be used or this format: `RGB (255,0,0)` etc. `#ff0000` can be shortened to `#f00` etc.

Layout & text by Matt Baker
Chart v. 1.0 (Apr 20, 2012)
Send all correspondence to:
matt@usefulcharts.com