Mr. Giansante

Website Design Using HTML

February 2019

Table of Contents

PRINTING

Please consider the environment before printing anything from this document.

Pages that need to be printed are marked with a printer icon.

Learning Outcomes

The student will ...

General

- Demonstrate proper care of all computer equipment
- Demonstrate positive attitudes and work habits
- Demonstrate interpersonal and organizational skills
- Demonstrate Problem Solving Skills
- Make Productive Use of Time
- Demonstrate independence skills and only seek help when necessary

HTML Basics and Tags

- Know and understand the "required elements" of every HTML page (html, head, body)
- Create websites which are compatible with different browsers and platforms (ie. not use browser-specific codes)

Visual Design of Page

- Be able to change background, text and link color
- Develop a color scheme for a website
- Maintain a consistent look and feel throughout a website

Formatting Text

- Be able to create paragraphs of text (left, right, center, blockquote)
- Be able to format text (bold, italics, underline, font size, font color)
- Use lists of text to format text

Graphics

- Be able to insert images into their web pages
- Be able to convert a graphic into a web compatible format
- Create graphics for a webpage (using Paint, PhotoShop or ButtonStudio)
- Create graphics buttons as navigational aids

Hyperlinks

- Create hyperlinks to other pages in the current folder
- Create hyperlinks to other pages in a sub-folder of the current folder
- Create hyperlinks to other pages on the internet
- Create hyperlinks within the same page
- Create an "email" hyperlink
- Create hyperlinks to a file to be downloaded

Advanced Layout

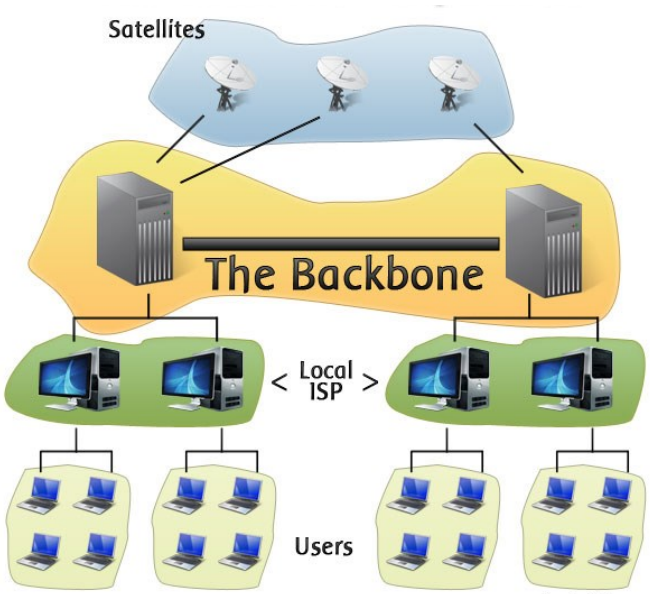
- Use tables to enhance web page layout
- Use frames to enhance web page layout

Trouble-shooting and Proofreading

- Be able to trouble-shoot their website when things go wrong
- Understand the need to create clean and easy-to-read code for the purpose of future updates and trouble-shooting

The Internet and the Web

How it Works



The Internet is a global network of interconnected computers, enabling users to share information along multiple channels.

ISP - Internet Service Provider

Users generally "connect" to the Internet using a modem (modulator-demodulator).

Modem Types
- Dial-up (phone lines)
- Cable
- Wireless

Domain Extentions

The domain extention often lets you know what type of website you are dealing with.

.com	Usually represents a business
.edu	Reserved for educational institutions
.gov	Reserved for governement institutions
.org	Organizations
.net	
.info	

World Wide Web

The World Wide Web (or simply "The Web") is a collection of documents on the Internet which are linked together using **HyperLinks**.

In 1989, Tim Berners-Lee, while working at the European Laboratory for Particle Physics (CERN) in Geneva, Switzerland, jotted down some ideas on a method for restructuring the organization's internal information systems. He though of the information as a "multiply connected web" rather than a ladder-like hierarchy.

He passed on those ideas, contained in a 15-page document entitled "Information Management: A Proposal", to his boss who reviewed it, then returned it to him with a note scrawled atop: "Vague but exciting."

It would be another year before Mr. Berners-Lee would write the HyperText Transfer Protocol (HTTP), HyperText Markup Language (HTML), and the software that would become known as a "browser".

(adapted from an article in the National Post written by David Akin, May 13, 1999)

Web Browsers

A Web Browser is a program that allows you to view web pages. Some examples include:

- Internet Explorer
- FireFox
- Opera
- Safari (Mac)
- Google Chrome

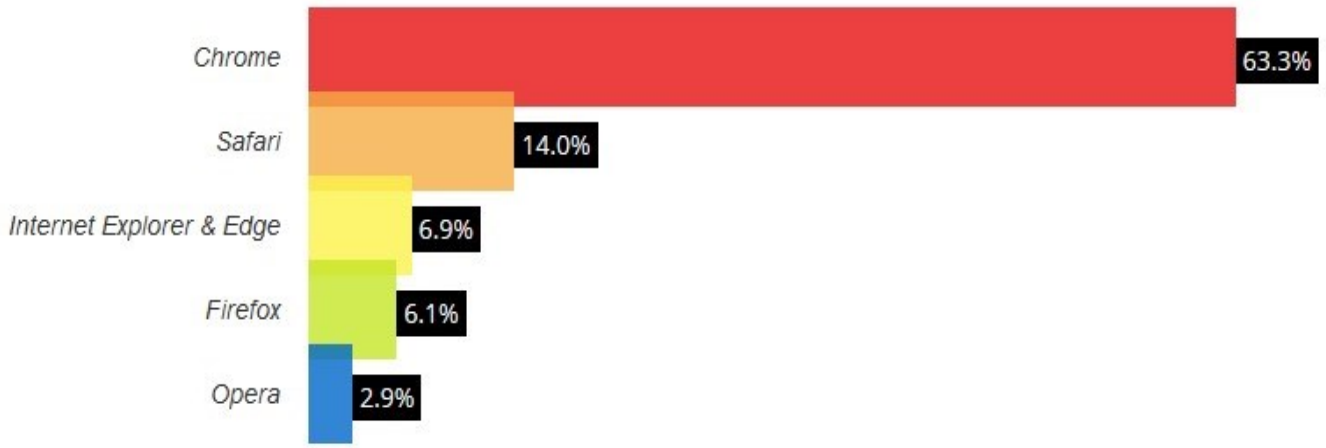
In 1993, the first graphical Web Browser, Mosaic™, was developed by the National Center for Supercomputing Applications (NCSA) at the University of Illinois.

One of the first people to realize the potential of the web was Marc Andressen. He had worked on the Mosiac™ browser but left NCSA to form his own company. In 1993 Andressen released a version of his new, handsome, point-and-click graphical browser for the Web, designed to run on Unix machines. The program, later adopted for other platforms such as the PC and Macintosh, was Netscape Navigator™.

The introduction of this free browser began a huge groundswell of popular interest in the World Wide Web. Within twenty-four months, the Web would go from being unknown to absolutely ubiquitous.

Source: A Brief History of Cyberspace,
by Mark Pesce, ZDNet, October 15, 1995

Market Share of Web Browsers - January 2019



Source: https://www.w3counter.com/globalstats.php

Steps to Designing a Website

When creating a website, most students want to start with writing code. In reality, writing code is one of the last steps in creating a professional website.

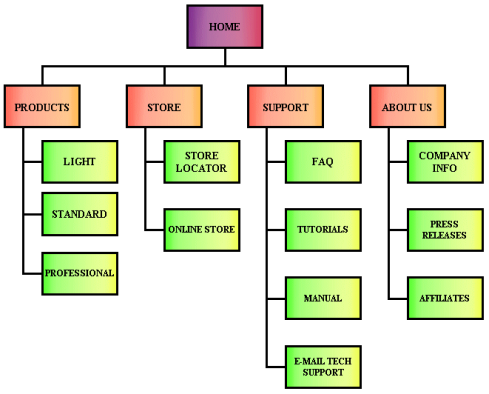
- Determine the purpose(s), goal(s) and audience for the website
- Create a site plan (on paper or using a flowcharting program)
- Choose an appropriate colour scheme and consistent look
- Create the site using proper and consistent methods
- Check the individual pages using an HTML tag checking program
- Ensure that there are no "dead" links or broken pictures
- Optimize the graphics
- Proofread and spellcheck the site
- Maintain and update the website as required

Planning a Website

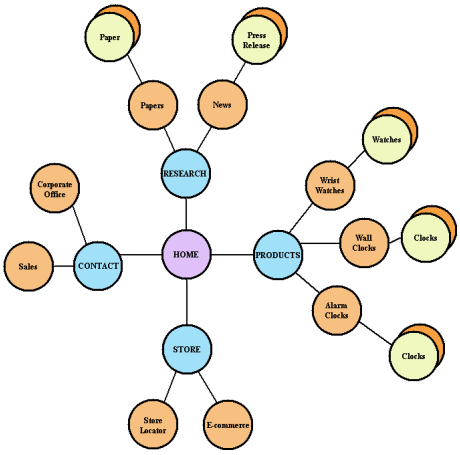
Before you begin coding your site, you should plan it out on paper or a computer program. This is sometimes called making a **storyboard** or **flowcharting**.

There are many ways to graphically represent your web site. Some types include:

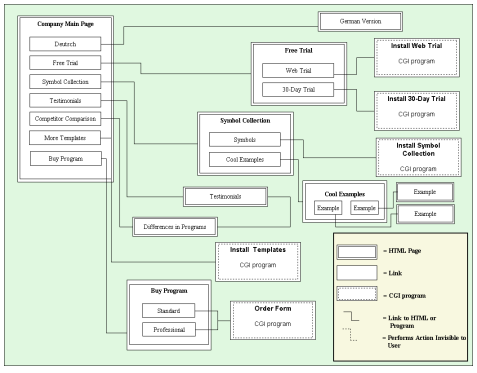
Hierarchical Map



Radial Tree Map



Flow Chart



Making a Wireframe

A wireframe helps you organize and simplify the elements and content within a website and is an essential tool in the development process.

The wireframe acts as a prototype that shows the placement of page features, such as header, footer, content, sidebars, and navigation.

Here are several important things to keep in mind when developing a wireframe:

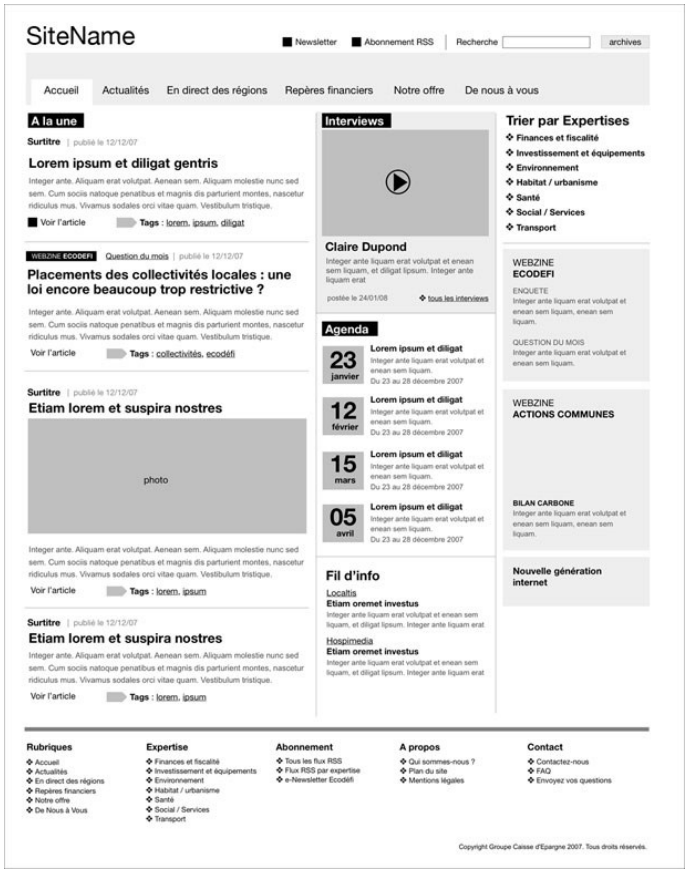
Simplicity
The key is to keep it simple enough to be clear to the client and to be flexible for the designer, but detailed enough to guide the programmer.

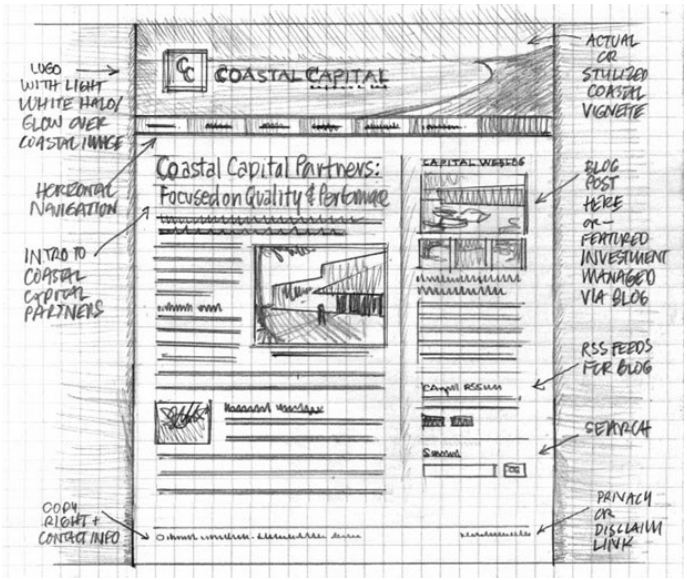
Work in Grayscale
When creating elements for a wireframe, it's best to work in grayscale so that you can focus on the layout without being distracted by the design.

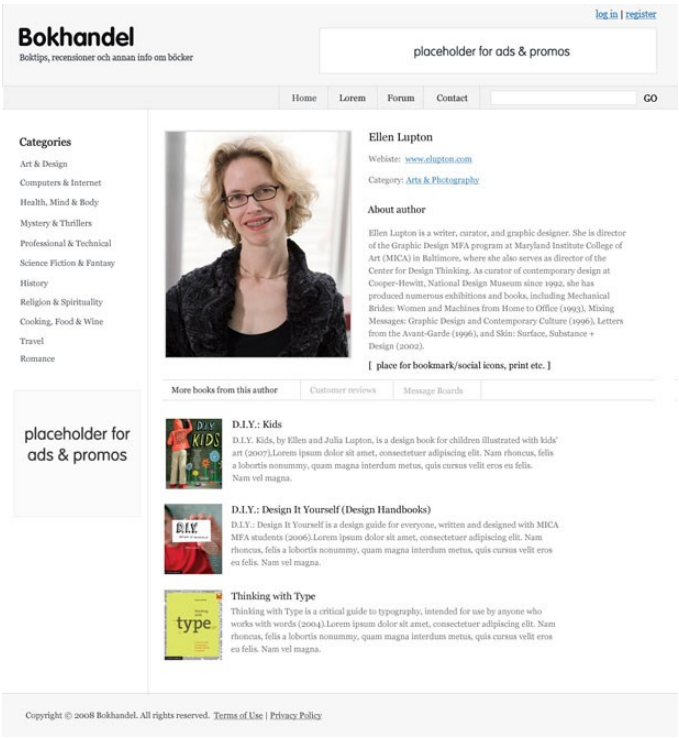
Avoid Too much happening on the page
Leave ample white space so that the design doesn't appear too busy or cluttered.

Avoid Too much detail
You can always add more detail later, but if you include too much in the begining, the client may confuse the wireframe for the final mockup.

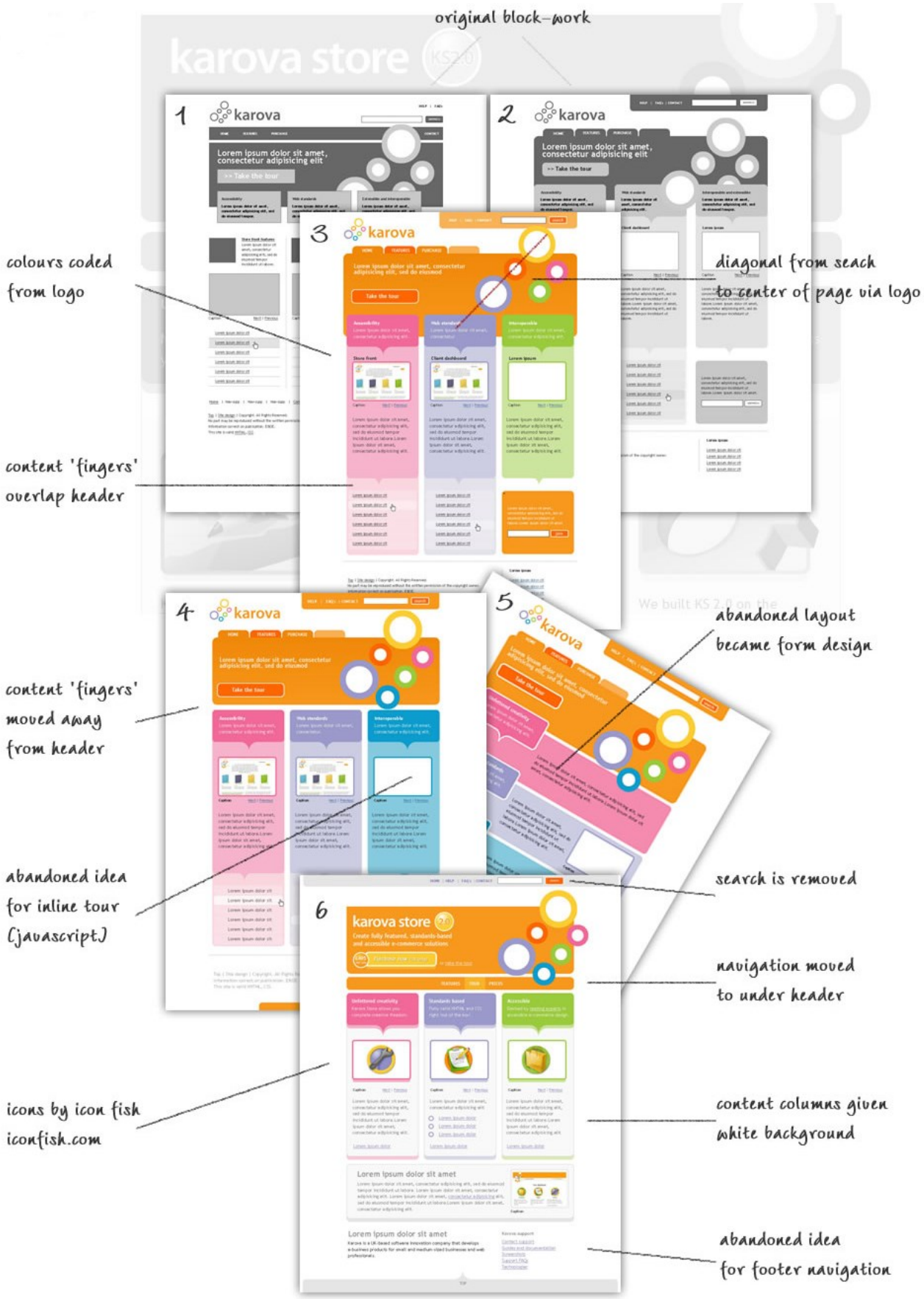
Adapted from an article by Eric Shafer (www.webdesignerdepot.com)







Evolution of a Website's Design



HyperText Markup Language (HTML)

Web Documents are authored in a specific format which is known as **HyperText MarkUp Language** (HTML).

HTML is a fairly straight-forward set of rules which define the use of tags which are embedded in a regular text document. These tags are interpreted by a Web Browser and the information is displayed in a graphical format.

Like any programming language, HTML is constantly evolving. Each new level includes the previously defined tags plus new ones.

Unlike most computer programming languages, tags which the browser does not understand do not cause errors - however, the web page will not look exactly as you intended.

Start and End Tags

In general, the HTML tag format is as follows:

<command>	This tag represents the start of an element.
</command>	This tag represents the end of an element.

Note: Not all elements require an end tag.

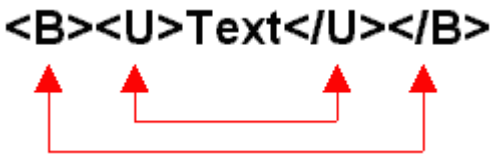
The BOLD Tag

For example, the HTML command for bold text is the letter "b". Therefore, the tag indicates where bold text begins and the tag indicates where bold text ends.

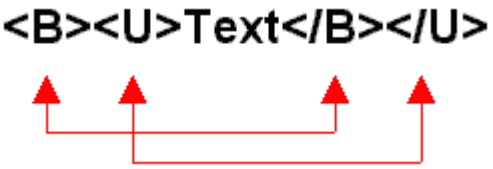
Crossing Tags

You should avoid crossing tags. For example, if you want to make some text bold and underlined ...

The correct way to do it ...



The wrong way to do it ...



Uppercase or Lowercase Tags?

According to the official HTML 4 standard, it doesn't matter whether tags and attributes are in uppercase or lowercase letters. However, the newer XHTML standard will require tags and attributes to be lowercase, so it is a good idea to make all your HTML tags lowercase to ensure future compatability.

Required Elements

A valid HTML document must follow certain guidelines.

As a general rule, the structure of HTML documents looks as follows:

```
<html>
<head>
<title> Title of Document </title>
    Meta Tag information (for search engines) goes here
    CSS (Cascading Style Sheet) information can go here
    (or be in a separate file)
</head>
<body>
    The text and Images of your Web Document go here
</body>
</html>
```

Guidelines for Naming Files and Folders

HTML filenames should end in the suffix ".html"

The file name should be no more than 32 characters, including the ".html" suffix.

It is better to use a slightly longer filename that will have an easier URL to remember than a jumble of unpronounceable letters. For example: Which of these would be easier to remember and give to a friend as a URL: hurricanes.html or hrrcns.html ?

File names and folder names should contain only letters, digits, and underscores - no spaces, punctuation, or special characters.

The first character of the file name should be a letter.

Use all lower case letters.

The first page should always be called **index.html** as this is the file that the web browser automatically looks for if none is specified. In many cases, index.html only contains frame information.

HTML Cheat Sheet

Required Elements

```
<html>

<head>
<title> title of document </title>
</head>

<body>

... web document goes here ...

</body>

</html>
```

Graphics

```
<img src="bob.gif" hspace="0" vspace="0"
align="left">
```

Special Characters

©	&#169;	Non-breaking space
é	&#233;	
à	&#224;	

Hyperlink

```
<a href="http://www.server.com/
index.htm">Click</a>
```

Email Hyperlink

```
<a href="mailto:bob@it.com">Email to
bob@it.com</a>
```

Common RGB Color Codes

Black	000000	Light Blue	C0D9D9
White	FFFFFF	Orange	EEA600
Red	FF0000	Gold	FFD700
Green	00FF00	Gray	C0C0C0
Blue	0000FF	Light Gray	A8A8A8
Cyan	00FFFF	Dark Gray	888888
Yellow	FFFF00	Bright Gold	D9D919
Magenta	FF00FF	Brown	A62A2A

The Paragraph Tag - CSS

You can insert multiple elements listed below in the `style` attributes.

Alignment	`text-align:left;`
	`text-align:right;`
	`text-align:center;`
	`text-align:justify;`

Indenting	`text-indent:50px;` Specify amount in pixels.
Font Size	`font-size:12pt;` Specify size in points.
Font Name	`font-family:"Arial";`

Formatting Text

Bold	
Italics	
Underline	<u></u>
Preformatted	<pre></pre>
Subscript (ex. CO_2)	
Superscript (ex. x^2)	
Non-breaking Text	<nobr></nobr>
Line Break	
Multiple Line Breaks	<br clear="all" />

List Example

```
<ul type="disc">
<li> Item 1 </li>
<li> Item 2 </li>
<li> Item 3 </li>
</ul>
```

Use `start="n"` to start the list at a number other than 1

type choices:
Disc
Circle
Square
"1" gives 1, 2, 3, 4 …
"I" gives I, II, III, IV …
"i" gives i, ii, iii, iv …
"A" gives A, B, C, D …
"a" gives a, b, c, d, e

Graphics File Formats

Graphics Interface Format (GIF)

GIF (pronounced jiff) files are best for line art and images with solid colours. There are two GIF standards: GIF87 and GIF89a. The GIF89a standard supports transparent images as well as animation (called Animated GIFs).

Joint Photographics Experts Group (JPG)

JPG or JPEG (pronounced jay-peg) files are best for complex visual images with many colors, such as photo-realistic images. JPG does not support transparency.

Portable Network Graphics (PNG)

PNG (pronounced ping) is expected to someday become the standard for web graphics. PNG supports transparent images.

Troubleshooting Broken Pictures

When you see the icons below in your web page, it means something has gone wrong with your pictures (ie. `<img>` tag).

Here is a list of things for you to check in order to find the problem ...

Is the name of the file in the `<img>` tag **exactly** the same as the name of the picture file? Sometimes a file ends in .GIF and you typed .JPG instead.

Is the picture file in the same directory as the html file which is calling it? If the file is in a subdirectory, include the name of the directory in the `src` attribute. For example, if you have a picture named bob.gif which is in a subdirectory called pictures, the `src` attibute tag should look like: `src="pictures/bob.gif"`.

Is the file in .GIF or .JPG format? If you are using Paint, saving the file with a .GIF extension is not sufficient ... you must also make sure you chose the GIF option in the "Save as" type dropdown box. If you are using a program such as PhotoShop, make sure you export the file rather than simply saving it.

Did you mistakenly put two single quotes (') side by side instead of using a double quote (") around the name of the picture file?

Minimizing Graphic File Size

It is important to keep the size of graphics files to a minimum so that the web pages load faster.

If a page contains many large pictures, consider including thumbnails (small representations of the picture). Users can click on the thumbnails to view the larger images.

Large images can be cut up into smaller pieces. In may cases the sum of the sizes of the smaller parts is less than the size of the original. In cases where the size of the parts is the same as the large image, the user will at least get the impression that the page is loading faster.

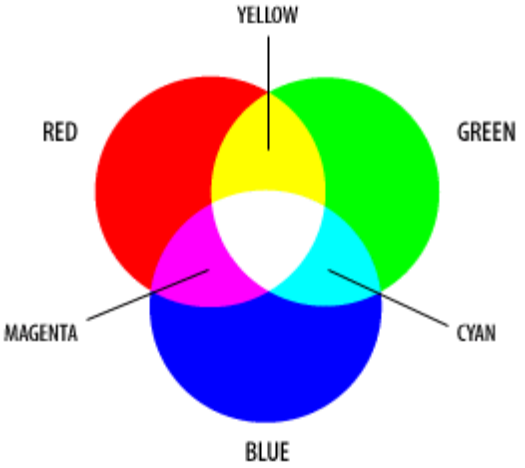
Crop images to remove unnecessary information.

Original Image (size: 26 kb)

Cropped Image (size: 5 kb)

RGB Color

When programming with HTML, colors are specified using a six-digit hexadecimal number which describes the amount of **Red**, **Green** and **Blue** a certain color contains. The six digits of the RGB (Red-Green-Blue) number are broken down as follows: RRGGBB.



A value of 00 is the lowest possible and FF is the highest. Using this system, there are approx. 16.7 million color choices available to control background and text.

FF	100% brightness
CC	80% brightness
99	60% brightness
66	40% brightness
33	20% brightness
00	0% brightness

White Background?

While it has been shown that text is most legible on a contrasting background, that doesn't mean you should always resort to white when using black text.

The white that a browser displays (normally #FFFFFF) is often a bit overwhelming to the eye - especially after you've been surfing for 32 hours straight. To make things a bit easier on your viewer's eyes, consider using a variation of white, something like #FFFFCC or #CCCCFF.

Once you add the other elements to the page, the viewer won't notice any difference, other than the page being more pleasing.

Source: Element K Journals' Inside Web Design Tips

RGB Color Code Chart

000000	330000	660000	990000	CC0000	FF0000
003300	333300	663300	993300	CC3300	FF3300
006600	336600	666600	996600	CC6600	FF6600
009900	339900	669900	999900	CC9900	FF9900
00CC00	33CC00	66CC00	99CC00	CCCC00	FFCC00
00FF00	33FF00	66FF00	99FF00	CCFF00	FFFF00
000033	330033	660033	990033	CC0033	FF0033
003333	333333	663333	993333	CC3333	FF3333
006633	336633	666633	996633	CC6633	FF6633
009933	339933	669933	999933	CC9933	FF9933
00CC33	33CC33	66CC33	99CC33	CCCC33	FFCC33
00FF33	33FF33	66FF33	99FF33	CCFF33	FFFF33
000066	330066	660066	990066	CC0066	FF0066
003366	333366	663366	993366	CC3366	FF3366
006666	336666	666666	996666	CC6666	FF6666
009966	339966	669966	999966	CC9966	FF9966
00CC66	33CC66	66CC66	99CC66	CCCC66	FFCC66
00FF66	33FF66	66FF66	99FF66	CCFF66	FFFF66
000099	330099	660099	990099	CC0099	FF0099
003399	333399	663399	993399	CC3399	FF3399
006699	336699	666699	996699	CC6699	FF6699
009999	339999	669999	999999	CC9999	FF9999
00CC99	33CC99	66CC99	99CC99	CCCC99	FFCC99
00FF99	33FF99	66FF99	99FF99	CCFF99	FFFF99
0000CC	3300CC	6600CC	9900CC	CC00CC	FF00CC
0033CC	3333CC	6633CC	9933CC	CC33CC	FF33CC
0066CC	3366CC	6666CC	9966CC	CC66CC	FF66CC
0099CC	3399CC	6699CC	9999CC	CC99CC	FF99CC
00CCCC	33CCCC	66CCCC	99CCCC	CCCCCC	FFCCCC
00FFCC	33FFCC	66FFCC	99FFCC	CCFFCC	FFFFCC
0000FF	3300FF	6600FF	9900FF	CC00FF	FF00FF
0033FF	3333FF	6633FF	9933FF	CC33FF	FF33FF
0066FF	3366FF	6666FF	9966FF	CC66FF	FF66FF
0099FF	3399FF	6699FF	9999FF	CC99FF	FF99FF
00CCFF	33CCFF	66CCFF	99CCFF	CCCCFF	FFCCFF
00FFFF	33FFFF	66FFFF	99FFFF	CCFFFF	FFFFFF

Common RGB Color Codes

Black	000000
White	FFFFFF
Red	FF0000
Green	00FF00
Blue	0000FF
Cyan	00FFFF
Yellow	FFFF00
Fuchsia	FF00FF
Light Blue	C0D9D9
Orange	EEA600
Gold	FFD700
Gray	C0C0C0
Light Gray	A8A8A8
Dark Gray	888888
Bright Gold	D9D919
Brown	A62A2A

Frames

This example creates a website with two frames. The frames are set up to be like columns. The frame on the left will be 250 pixels wide and the one on the right will fill up the rest of the space (the "*" character is used to achieve this)

The frame on the left will be named leftframe and will display menu.html

The frame on the right will be named rightframe and will display start.html

```
<html>

<head>
<title>Frame Example</title>
</head>

<frameset cols="250,*" border="0">
   <frame src="menu.html" name="leftframe">
   <frame src="start.html" name="rightframe">
</frameset>

</html>
```

Important: Note that there is no <body> tag.

The "noframes" Attribute

Not all browsers support frames and so you should include information in the <noframes> tag

```
<noframes>Your browser does not support
frames and therefore this page will not
display as it was intended. You should
consider upgrading your browser.</
noframes>
```

Frameset Tag Attributes

To specify if you want a 3D border or not:

```
frameborder="yes"
frameborder="no"
```

To specify the border thickness:

`border="n"`	where n is pixels
`framespacing="n"`	where n is pixels

Note: You must use **both** since one is for FireFox and one is for Internet Explorer

`bordercolor="#RRGGBB"`

Frame Tag Attributes

`marginwidth="n"`	where n is pixels
`marginheight="n"`	where n is pixels

To specify if the frame will have scroll bars:

```
scrolling="yes"
scrolling="no"
scrolling="auto"
```

To specify that the frame cannot be resized, include:

```
noresize
```

The Target Attribute

Often, the frame on the left simply contains links which will change what is being displayed in the frame on the right. This can be accomplished using the target attribute of the hyperlinks. The following code goes in menu.htm ...

```
<a href="info.html"
target="rightframe">Information</a>
```

The window name specified by a target attribute must begin with an alpha-numeric character to be valid. All other window names will be ignored.

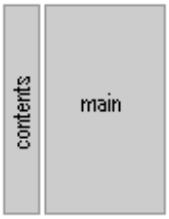
Special Values for Target

`target="_blank"`	This target will cause the link to always be loaded in a new blank window. This window is not named.
`target="_self"`	This target causes the link to load in the same window the anchor was clicked in.
`target="_parent"`	This target makes the link load in the immediate frameset parent of this document.
`target="_top"`	This target makes the link load in the full body of the window. This defaults to acting like "_self" if the document is already at the top. It is useful for breaking out of an arbitrarily deep frame nesting.

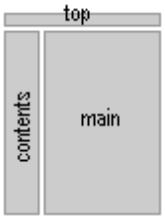
Frame Examples


```
<frameset rows="45,*">
   <frame name="top" src="menu.html" scrolling="no" noresize>
   <frame name="main" src="start.html">
</frameset>
```


```
<frameset rows="90%,10%">
   <frame name="main" src="start.html" scrolling="auto">
   <frame name="footer" src="footer.html" scrolling="auto" noresize>
</frameset>
```

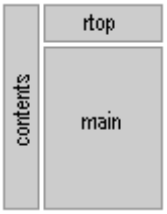



```
<frameset cols="150,*">
   <frame name="contents" src="menu.html" noresize>
   <frame name="main" src="start.html">
</frameset>
```

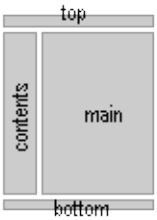



```
<frameset rows="64,*">
    <frame name="top" src="topmenu.html" scrolling="no" noresize>

    <frameset cols="150,*">
        <frame name="contents" src="sidemenu.html">
        <frame name="main" src="start.html">
    </frameset>
</frameset>
```




```
<frameset cols="150,*">
    <frame name="contents" scrolling="no" noresize src="menu.htm">
    <frameset rows="20%,*">
        <frame name="rtop" src="header.htm">
        <frame name="main" src="start.htm">
    </frameset>
</frameset>
```




```
<frameset rows="64,*,64">
    <frame name="top" src="topmenu.html" scrolling="no" noresize>
    <frameset cols="150,*">
        <frame name="contents" src="leftmenu.html">
        <frame name="main" src="start.html">
    </frameset>
    <frame name="bottom" src="footer.html" scrolling="no" noresize >
</frameset>
```

Special Characters

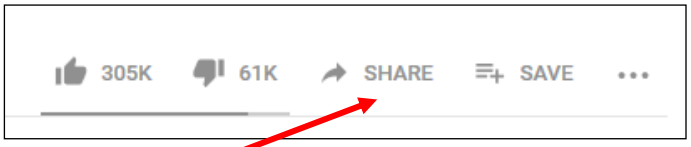
CHARACTERS		
&#32;		Space
&#33;	!	Exclamation mark
&#34;	"	Quotation mark
&#35;	#	Pound symbol
&#36;	$	Dollar sign
&#37;	%	Percent sign
&#38;	&	Ampersand
&#39;	`	Apostrophe
&#40;	(	Opening bracket
&#41;	)	Closing bracket
&#42;	*	Asterisk
&#43;	+	Plus
&#44;	,	Comma
&#45;	-	Hyphen
&#46;	.	Period
&#47;	/	Forward slash
&#48;	0	Zero
&#49;	1	One
&#50;	2	Two
&#51;	3	Three
&#52;	4	Four
&#53;	5	Five
&#54;	6	Six
&#55;	7	Seven
&#56;	8	Eight
&#57;	9	Nine
&#58;	:	Colon
&#59;	;	Semicolon
&#60;	<	Less than
&#61;	=	Equals sign
&#62;	>	Greater than
&#63;	?	Question mark
&#64;	@	"At" symbol
&#65;	A	Upper case A
&#66;	B	Upper case B
&#67;	C	Upper case C
&#68;	D	Upper case D
&#69;	E	Upper case E
&#70;	F	Upper case F
&#71;	G	Upper case G
&#72;	H	Upper case H
&#73;	I	Upper case I
&#74;	J	Upper case J
&#75;	K	Upper case K
&#76;	L	Upper case L
&#77;	M	Upper case M
&#78;	N	Upper case N
&#79;	O	Upper case O
&#80;	P	Upper case P

CHARACTERS		
&#81;	Q	Upper case Q
&#82;	R	Upper case R
&#83;	S	Upper case S
&#84;	T	Upper case T
&#85;	U	Upper case U
&#86;	V	Upper case V
&#87;	W	Upper case W
&#88;	X	Upper case X
&#89;	Y	Upper case Y
&#90;	Z	Upper case Z
&#91;	[	Opening square bracket
&#92;	\	Backslash
&#93;	]	Closing square bracket
&#94;	^	Caret
&#95;	_	Underscore
&#96;	`	Single quote
&#97;	a	Lower case a
&#98;	b	Lower case b
&#99;	c	Lower case c
&#100;	d	Lower case d
&#101;	e	Lower case e
&#102;	f	Lower case f
&#103;	g	Lower case g
&#104;	h	Lower case h
&#105;	i	Lower case i
&#106;	j	Lower case j
&#107;	k	Lower case k
&#108;	l	Lower case l
&#109;	m	Lower case m
&#110;	n	Lower case n
&#111;	o	Lower case o
&#112;	p	Lower case p
&#113;	q	Lower case q
&#114;	r	Lower case r
&#115;	s	Lower case s
&#116;	t	Lower case t
&#117;	u	Lower case u
&#118;	v	Lower case v
&#119;	w	Lower case w
&#120;	x	Lower case x
&#121;	y	Lower case y
&#122;	z	Lower case z
&#123;	{	Opening curly brace
&#124;	|	Vertical line
&#125;	}	Closing curly brace
&#126;	~	Tilde
&#127;		Delete

CHARACTERS (EXTENDED)		
		Non-breaking space
&#161;	¡	Inverted exclamation mark
&#162;	¢	Cent symbol
&#163;	£	Pound symbol
&#164;	¤	Currency symbol
&#165;	¥	Yen symbol
&#166;	¦	Broken vertical bar
&#167;	§	Section symbol
&#168;	¨	Umlaut
&#169;	©	Copyright
&#170;	ª	Feminine ordinal
&#171;	«	Double-left arrow
&#172;	¬	"Not" symbol
&#173;		Soft hyphen
&#174;	®	Registered
&#175;	¯	Overline
&#176;	°	Degree symbol
&#177;	±	Plus-or-minus
&#178;	²	Squared
&#179;	³	Cubed
&#180;	´	Acute accent
&#181;	µ	Micro symbol
&#182;	¶	Paragraph symbol
&#183;	·	Middle dot
&#184;	¸	Cedilla
&#185;	¹	Superscript "1"
&#186;	º	Masculine ordinal
&#187;	»	Double-right arrow
&#188;	¼	One quarter
&#189;	½	One half
&#190;	¾	Three quarters
&#191;	¿	Inverted question mark
&#192;	À	A with grave
&#193;	Á	A with acute
&#194;	Â	A with circumflex
&#195;	Ã	A with tilde
&#196;	Ä	A with umlaut
&#197;	Å	A with ring
&#198;	Æ	AE
&#199;	Ç	C with cedilla
&#200;	È	E with grave
&#201;	É	E with acute
&#202;	Ê	E with circumflex
&#203;	Ë	E with umlaut
&#204;	Ì	I with grave
&#205;	Í	I with acute
&#206;	Î	I with circumflex

CHARACTERS (EXTENDED)		
&#207;	Ï	I with umlaut
&#208;	Ð	ETH
&#209;	Ñ	N with tilde
&#210;	Ò	O with grave
&#211;	Ó	O with acute
&#212;	Ô	O with circumflex
&#213;	Õ	O with tilde
&#214;	Ö	O with umlaut
&#215;	×	Multiply symbol
&#216;	Ø	O with slash
&#217;	Ù	U with grave
&#218;	Ú	U with acute
&#219;	Û	U with circumflex
&#220;	Ü	U with umlaut
&#221;	Ý	Y with acute
&#222;	Þ	THORN
&#223;	ß	Sharp S
&#224;	à	A with grave
&#225;	á	A with acute
&#226;	â	A with circumflex
&#227;	ã	A with tilde
&#228;	ä	A with umlaut
&#229;	å	A with ring
&#230;	æ	AE
&#231;	ç	C with cedilla
&#232;	è	E with grave
&#233;	é	E with acute
&#234;	ê	E with circumflex
&#235;	ë	E with umlaut
&#236;	ì	I with grave
&#237;	í	I with acute
&#238;	î	I with circumflex
&#239;	ï	I with umlaut
&#240;	ð	ETH
&#241;	ñ	N with tilde
&#242;	ò	O with grave
&#243;	ó	O with acute
&#244;	ô	O with circumflex
&#245;	õ	O with tilde
&#246;	ö	O with umlaut
&#247;	÷	Divide symbol
&#248;	ø	O with slash
&#249;	ù	U with grave
&#250;	ú	U with acute
&#251;	û	U with circumflex
&#252;	ü	U with umlaut
&#253;	ý	Y with acute
&#254;	þ	THORN
&#255;	ÿ	Y with umlaut

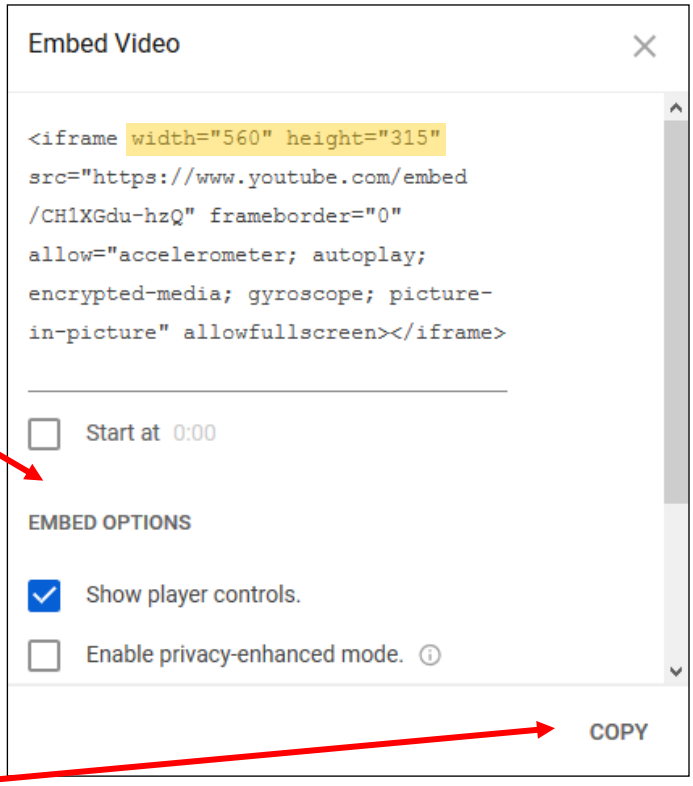
Embedding a YouTube Video



1. Click the "Share" button below the video.

2. Click the "Embed" button.

3. Select your options.



4. The code for embedding the video is displayed. Simply copy it.

Cascading Style Sheets (CSS)

Cascading Style Sheets (CSS) is a style sheet language used to describe the presentation semantics (the look and formatting) of a document written in a markup language. Its most common application is to style web pages written in HTML. (Source: WikiPedia.org)

CSS Formatting can be inserted directly into the <head> section of a website, or can be a separate file that is referenced in the <head> section.

Common CSS Properties

```
font-size:          xx-small
                    x-small
                    small
                    medium
                    large
                    x-large
                    xx-large

font-size:          larger
                    smaller

font-size:          10px
                    20pt
                    0.5in

font-style:         normal
                    italic
                    oblique

font-weight:        normal
                    bold
                    bolder
                    lighter

font-family:        Arial
                    Helvetica
                    sans-serif

text-decoration:    underline
                    overline
                    line-through

text-align:         left
                    right
                    center
                    justify
```

Using CSS within HTML Code

Define an inline style to make the current paragraph with blue foreground:

```
<p style="color:blue">blue paragraph</p>
```

Using CSS within <style> Tag in <head> Section

Make Header 1 with blue foreground:

```
h1 {color:blue}
```

Make Header 1 with yellow foreground and black background

```
h1 {color: yellow; background-color: black}
```

Make Header 2, Header 3, and Bold with green foreground:

```
h2,h3,b {color:green}
```

Make paragraph bold and red:

```
p {color:red; font-weight:bold}
```

Make bold that occurs within a paragraph blue:

```
p b {color:blue}
```

Make two paragraph classes, one with black font and one with red font:

```
p.normal {color: black}
p.error {color: red}
```

To use in HTML code:

```
<p class="normal">normal paragraph</p>
<p class="error">error paragraph</p>
```

Make a class with red and bold font, without attaching to a specific element

```
.error1 {color:red; font-weight:bold }
```

To use in HTML code:

```
<div class="error1">hello</div>
```

Cascading Style Sheets Example

The following example demonstrates CSS in a separate file that is referenced in the <head> tag.

HTML Code

```
<html>
<head>
<title>Website Title</title>
<link rel="stylesheet" href="page.css">
</head>
<body>
    ... contents of your page go here ...
</body>
</html>
```

CSS File

In this example, the CSS file is named "page.css". It is a plain text document that can be created with a text editor such as Window's NotePad.

```
body{
background-color: FFFFFF;
color: 000000;
margin: 10px;
font-size: 12;
font-family: "Arial"}

A:link{
text-decoration: none;
font-weight: bold;
font-size: 12;
color: 0000FF}

A:visited{
text-decoration: none;
font-weight: bold;
font-size: 12;
color: 0000FF}

A:active{
text-decoration: none;
font-weight: bold;
font-size: 12;
color: 0000FF}

A:hover{
text-decoration: none;
font-weight: bold;
font-size: 12;
color: 000000}
```

If you want to specify a background image instead of a color, use the following (replace `paper.gif` with the name of the image you want to use):

```
background-image:url('paper.gif');
```

Cascading Style Sheets Cheat Sheet

Selectors

*	All elements
div	
div *	All elements within
div span	within
div, span	and
div > span	with parent
div + span	preceded by
.class	Elements of class "class"
div.class	of class "class"
#itemid	Element with id "itemid"
div#itemid	with id "itemid"
a[attr]	<a> with attribute "attr"
a[attr='x']	<a> when "attr" is "x"
a[class~='x']	<a> when class is a list containing 'x'
a[lang|='en']	<a> when lang begins "en"

Pseudo-Selectors and Pseudo-Classes

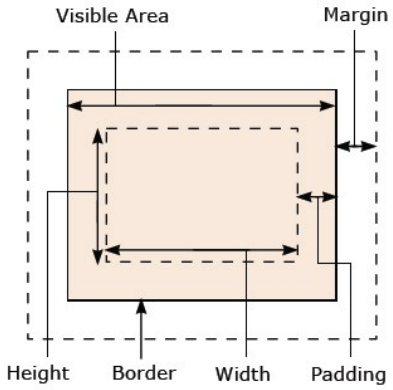
:first-child	First child element
:first-line	First line of element
:first-letter	First letter of element
:hover	Element with mouse over
:active	Active element
:focus	Element with focus
:link	Unvisited links
:visited	Visited links
:lang(var)	Element with language "var"
:before	Before element
:after	After element

Sizes and Colours

0	0 requires no unit
Relative Sizes	
em	1em equal to font size of parent (same as 100%)
ex	Height of lower case "x"
%	Percentage
Absolute Sizes	
px	Pixels
cm	Centimeters
mm	Millimeters
in	Inches
pt	1pt = 1/72in
pc	1pc = 12pt
Colours	
#789abc	RGB Hex Notation
#acf	Equates to "#aaccff"
rgb(0,25,50)	Value of each of red, green, and blue. 0 to 255, may be swapped for percentages.

Note Shorthand properties are marked x
Properties that inherit are marked +

Box Model



Positioning

display	clear
position	z-index
top	direction +
right	unicode-bidi
bottom	overflow
left	clip
float	visibility

Dimensions

width	min-height
min-width	max-height
max-width	vertical-align
height	

Color / Background

color +	background-repeat
background x	background-image
background-color	background-position
background-attachment	

Text

text-indent +	word-spacing +
text-align +	text-transform +
text-decoration	white-space +
text-shadow	line-height +
letter-spacing +	

Fonts

font + x	font-weight +
font-family +	font-stretch +
font-style +	font-size +
font-variant +	font-size-adjust +

Available free from www.AddedBytes.com

Boxes

margin x	border-color x
margin-top	border-top-color
margin-right	border-right-color
margin-bottom	border-bottom-color
margin-left	border-left-color
padding x	border-style x
padding-top	border-top-style
padding-right	border-right-style
padding-bottom	border-bottom-style
padding-left	border-left-style
border x	border-width x
border-top x	border-top-width
border-bottom x	border-right-width
border-right x	border-bottom-width
border-left x	border-left-width

Tables

caption-side +	border-spacing +
table-layout	empty-cells +
border-collapse +	speak-header +

Paging

size	page-break-inside +
marks	page +
page-break-before	orphans +
page-break-after	widows +

Interface

cursor +	outline-style
outline x	outline-color
outline-width	

Aural

volume +	elevation
speak +	speech-rate
pause x	voice-family
pause-before	pitch
pause-after	pitch-range
cue x	stress
cue-before	richness
cue-after	speak-punctuation
play-during	speak-numeral
azimuth +	

Miscellaneous

content	list-style-type +
quotes +	list-style-image +
counter-reset	list-style-position +
counter-increment	marker-offset
list-style + x	

25 Point Usability Checklist

Accessibility	Rating	Comments
1. Site load-time is reasonable	✓ ✓ ×	
2. Adequate text-to-background contrast	✓ ✓ ×	
3. Font size/spacing is easy to read	✓ ✓ ×	
4. Flash & add-ons are used sparingly	✓ ✓ ×	
5. Images have appropriate ALT tags	✓ ✓ ×	
6. Site has custom not-found/404 page	✓ ✓ ×	
Identity		
7. Company logo is prominently placed	✓ ✓ ×	
8. Tagline makes company's purpose clear	✓ ✓ ×	
9. Home-page is digestible in 5 seconds	✓ ✓ ×	
10. Clear path to company information	✓ ✓ ×	
11. Clear path to contact information	✓ ✓ ×	
Navigation		
12. Main navigation is easily identifiable	✓ ✓ ×	
13. Navigation labels are clear & concise	✓ ✓ ×	
14. Number of buttons/links is reasonable	✓ ✓ ×	
15. Company logo is linked to home-page	✓ ✓ ×	
16. Links are consistent & easy to identify	✓ ✓ ×	
17. Site search is easy to access	✓ ✓ ×	
Content		
18. Major headings are clear & descriptive	✓ ✓ ×	
19. Critical content is above the "fold"	✓ ✓ ×	
20. Styles & colors are consistent	✓ ✓ ×	
21. Emphasis (bold, etc.) is used sparingly	✓ ✓ ×	
22. Ads & pop-ups are unobtrusive	✓ ✓ ×	
23. Main copy is concise & explanatory	✓ ✓ ×	
24. URLs are meaningful & user-friendly	✓ ✓ ×	
25. HTML page titles are explanatory	✓ ✓ ×	