Appendix B

Cascading Style Sheets Beginners Reference

One goal of the W3C in developing XHTML was to separate content from style. That means that—if you haven't already—you need to begin learning how to work with Cascading Style Sheets. The following charts are not exhaustive, but they will give you a basic toolbox for learning to work with CSS. You might also want to check out the CSS tutorials and reference material at the following Web sites: www.htmlhelp.com, www.westciv.com, and www.blooberry.com.

TIP *If you are interested in learning more about CSS, you can download the CSS2 specification free of charge at the W3C's Web site at www.w3.org.*

Basic CSS Syntax

The following table lists the syntax for creating inline, embedded, and linked style sheets. It also includes information on how to create some of the most commonly used selectors. Of course, CSS syntax can become more complicated than this, but you will find that what is listed in this table will be sufficient for most of your style needs.

To Do This…	Use This…
Apply an inline style sheet	`style=" "` (Inside the element you want to modify)
Embed a style sheet	`<head><style type="text/css">`
Link to a style sheet	`<head><link rel="stylesheet" type="text/css" href="filename.css" /> </head>`
Apply the same style declarations to more than one selector	`h1, h2, h3, h4, h5, h6 {color: navy; font-style: serif; }` (Selectors should be separated by commas.)

Use more than one declaration in an embedded or external style sheet	`selector  {property: value;`
	`property: value;`

Use more than one declaration in an inline style sheet	`<element style="property: value; property: value;" >`
Create a class selector	`selector.classname {property: value;}` *Example:* `p.green {color: green;}`
Create an ID selector	`#IDname {property: value;}` *Example:* `#Myname1 {font-family:cursive; font-size: 8pt; color: maroon;}`
Create a descendent selector	`selector1 selector2 {property: value;}` (With descendent selectors, the selectors should not be separated by commas.)

Understand Measurement Units

Learning to apply measurement units in CSS can be tricky. One thing to keep in mind, particularly when working with length units, is the difference between *absolute* and *relative* measurements. Absolute measurements (inches, for example) are fixed and cannot scale or adjust from medium to medium. Relative measurements (ems, for example) are based on the size of the font being used. Whenever possible, use relative measurements. CSS measurement units are listed in the following table:

This Unit	Can Be Specified This Way

Color	`name: (red, blue, black, and so on)`
	`hex code: #ffffff`
	`hex shorthand: #ff0`
	`rgb numeric values: rgb(0, 255, 0)`
	`rgb percentages: rgb(20%, 45%, 96%)`

Percentages	`200%, -14%, +35%` (Some selectors cannot take negative values.)
Lengths (relative measurements)	px (pixels) em (width of the letter *m* for the font in use) ex (height of the letter *x* for the font in use)
Lengths (absolute measurements)	pt (points, as in a 10- or 12-point font) pc (picas; 12 points equals 1 pica) in (inches) mm (millimeters) cm (centimeters)

Use Font Properties

The font properties enable you to control how fonts display on your page. Font properties give you much greater freedom in your page design than the deprecated <font> element. The font properties are listed for you in the following table:

To Do This	Use This Property	With These Values	Example
Specify a particular font family	font-family	You can use a specific font or a generic: serif, sans-serif, cursive, fantasy, or monospace	`p {font-family: Arial, sans-serif;}`
Choose between normal and italic	font-style	normal | italic | oblique	`h1 {font-style: italic;}`

To Do This	Use This Property	With These Values	Example
Alternate between normal or small caps	font-variant	normal | small caps	`h2 {font-variant: small caps;}`
Make a font bolder or lighter	font-weight	normal | bold | bolder | lighter | 100 –900	`h3 {font-weight: 500;}`
Change a font's size	font-size	Absolute: xx-small | x-small | small | medium | large | x-large | xx-large Relative: larger | smaller (relative to the parent element) Length: em | ex Percentage	`h3 {font-size: smaller;)`

Apply Text Properties

Text properties greatly increase your ability to specify how the text, word spacing, line spacing, and so on will display when your page is viewed. Using relative measurements when you specify these properties allows your page to adjust dynamically. CSS text properties are listed in the following table:

To Do This	Use This Property	With These Values	Example
Adjust space between words	word-spacing	normal | by length	`h5 {word-spacing: .75 em;}`
Adjust space between letters	letter-spacing	normal | by length	`div {letter-spacing: 2mm;}`
Add and remove underlines or other text "decoration"	text-decoration	none | underline | overline | line-through | blink	`a:link {text-decoration: none;}`
Control the vertical positioning of an element	vertical-align	top | text-top | middle | bottom | text-bottom | baseline | sub | super | (percentage)	`p {vertical-align: super;}`
Convert text from uppercase to lowercase	text-transform	none | uppercase | lowercase | capitalize	`span {text-transform: uppercase;}`
Justify, center, or align text	text-align	justify | left | right | center	`h4 {text-align: center;}`
Indent text	text-indent	by length or percentage	`p {text-indent: .50in;}`
Adjust line spacing (relative to font size)	line-height	normal | numeric | by length or percentage	`span {line-height: 150%;}`

Work with Background and Color Properties

In HTML you are limited to setting background colors and images to the <body> and <table> elements. With CSS you can specify these background properties for almost all elements, which gives you a lot of flexibility in design and makes your job much easier when you want to create a "look" for your site. The properties for controlling background and color are listed in the following table:

To Do This	Use This Property	With These Values	Example
Set a foreground color (usually text color)	color	color (name, hex code, rgb)	`p {color: #ff0000;}`
Specify a background color	background-color	color (name, hex code, rgb) or transparent	`body {background-color: white;}`
Add a background image	background-image	image url	`div {background-image: url(pix.gif);}`
Tell a browser how to repeat an image	background-repeat	repeat (tile) | repeat-x (horizontally) | repeat-y (vertically) | no-repeat (display only one copy of image)	`body {background-image: url(pix.gif); background-repeat: repeat-y;}`
Turn image scrolling on and off	background-attachment	scroll | fixed	`body {background-image: url(pix.gif); background-repeat: repeat-y; background-attachment: fixed;}`
Position a background image	background-position	percentage or length (value a, value b) | or top/center/bottom and left/center/right	`body: background-image: url(pix.gif); background-position: top, left;}`

Understand and Use Box Properties

The box properties are organized in the following tables: Set Margins, Add Padding, Create Borders, and Specify Size and Float. If you think of every HTML element as being contained inside a rectangular "box," you'll have a pretty good idea of how to use these properties. If you learn to use these properties with class and ID selectors, you will have a virtually limitless array of possible styles at your disposal.

Set Margins

When you think of margins, remember to keep in mind the "box" concept. You are not setting margins for the entire page unless you are using "body" as a selector. Instead, you are setting the margins for that selector (element). Set your margins with the properties listed in the following table:

To Do This	Use This Property	With These Values	Example
Specify a top margin	margin-top	length, percentage, or auto	`blockquote {margin-top: 1em;}`
Specify a right margin	margin-right	length, percentage, or auto	`p {margin-right: .25in;}`
Specify the bottom margin	margin-bottom	length, percentage, or auto	`div {margin-bottom: .3cm;}`
Specify the left margin	margin-left	length, percentage, or auto	`span {margin-left: 5%;}`
Use shorthand to specify all four margins to the same size	margin: (value)	length, percentage, or auto	`h2 {margin: 10px;}(Sets a 10-pixel margin all the way around)`
Use shorthand to specify the top and bottom to one value, and the right and left to another	margin: (top/bottom value) (right/left value)	length, percentage, or auto	`h6 {margin: .25in .33in;}`
Use shorthand to set the top margin to one value, the right and left margins to another, and the bottom to a third value	margin: (top value) (right/left value) (bottom value)	length, percentage, or auto	`div {margin: 5% 7.5% 3%;}`
Use shorthand to set all four margins to different values	margin: (top) (right) (bottom) (left)	length, percentage, or auto	`p {margin: 2px .3em 4% .27cm;}`

Add Padding

What is the difference between specifying margins and adding padding? *Padding* represents the amount of space between the contents of the box and its edge. The idea is similar to cellspacing and cellpadding in XHTML tables. The following table lists the padding properties:

To Do This	Use This Property	With These Values	Example
Add top padding	padding-top	length, percentage, or auto	`span {padding-top: 15px;}`
Add padding to the right side	padding-right	length, percentage, or auto	`p {padding-right: 100mm;}`
Add bottom padding	padding-bottom	length, percentage, or auto	`blockquote {padding-bottom: .50in;}`
Add left-side padding	padding-left	length, percentage, or auto	`td {padding-left: 10%;}`

Create Borders

CSS allows you to surround any selector with a border. There is a variety of border styles to choose from (although some browsers will not display the full variety). The following table shows the properties necessary to specify borders:

To Do This	Use This Property	With These Values	Example
Specify a border style	border-style	inset | outset | ridge | solid | double | groove | dashed | dotted | none	`table {border-style: double;}`
Set border colors	border-color	color name, hex code, rgb values	`th {border-color: navy;}`
Set the width of the top border	border-top-width	thick | medium | thin or length	`td {border-top-width: 5px;}`
Set the width of the right border	border-right-width	thick | medium | thin or length	`tr {border-right-width: 1.2em;}`
Set the width of the bottom border	border-bottom-width	thick | medium | thin or length	`p {border-bottom-width: thin;}`
Set the width of the left border	border-left-width	thick | medium | thin or length	`p {border-left-width: medium;}`
Use shorthand to set the width of all borders with one declaration	border-width	thick | medium | thin or length (multiple values should be written the same as with margins)	`h6 {border-width: thin thick; }`
Define all values for a top border with one declaration	border-top	width | style | color	`span {border-top: thick dashed red;}`

To Do This	Use This Property	With These Values	Example
Define all values for a right border with one declaration	border-right	width | style | color	`div {border-right: thin groove yellow;}`
Define all values for a bottom border with one declaration	border-bottom	width | style | color	`h3 {border-bottom: medium ridge gold;}`
Define all values for a left border with one declaration	border-left	width | style | color	`td {border-left: 5px inset green;}`
Define identical values for all sides with one declaration	border	width | style | color	`p {border: thick outset navy;}`

Specify Image Size and Position

Although the properties in the following table can be used with text elements, more often than not they are used with images. These correspond closely to the XHTML attributes that often are applied to the <img /> element. The following table lists the properties related to position and size:

To Do This	Use This Property	With These Values	Example
Set an element's width (most often used with images)	width	length, percentage, or auto	`img.logo {width: 105px;}`
Set an element's height (most often used with images)	height	length or auto	`img.logo {height: 55px;}`
Position an element with text-wrapping	float	none | right | left	`image.logo {float: right;}`
Position an element without text-wrapping	clear	none | both | right | left	`image.logo {clear: left;}`

Understand Classification Properties

The properties in the following table can have some strange results if you aren't careful. For example, the display property can be used to prevent certain elements from displaying at all. However, most of these properties listed in the following table are used to tell the browser how to handle various elements, such as lists and list items:

To Do This	Use This Property	With These Values	Example
Control an element's display characteristics	display	none | list-item | block | inline	`img {display: none;}`
Control how a browser handles white space	white-space	normal | pre | nowrap	`p.code {display: pre;}`
Specify bullets and numbering for lists	list-style-type	disc | circle | square | decimal | lower-roman | upper-roman | lower-alpha | upper-alpha | none	`ol {list-style-type: lower-alpha;}`
Specify an image for list bullets	list-style-image	image url	`ul {list-style-image: url(bullet.gif);}`
Specify a position for image bullets	list-style-position	outside | inside	`ul {list-style-position: inside;}`
Use shorthand for specifying a list's characteristics	list-style	(type) (image) (position)	`ul.pix {list-style: disc url(px.gif) outside;}`

Understand Pseudo-classes and Pseudo-elements

Pseudo-classes and pseudo-elements represent ways of dealing with text that are not present in the actual structure of an HTML document. Pseudo-classes and pseudo-elements are limited in their scope but can be quite useful. The following table lists the pseudo-classes and pseudo-elements included in the CSS1 specification:

To Do This	Use This	Example
Set a link's characteristics	a:link	`a:link {color: red;}`
Set an active link's characteristics	a:active	`a:active {color: purple;}`

To Do This	Use This	Example
Set a visited link's characteristics	a:visited	`a:visited {color: green;}`
Cause a link to change color when a mouse passes over it	a:hover	`a:hover {color: magenta;}`
Modify the first letter of an element's text	(selector): first-letter	`p:first-letter {font-size: 150%; color: red;}`
Modify the first line of an element's text	(selector): first-line	`p:first-line {font-variant: italic; font-size: 1.2em; color: blue;}`

Although this CSS reference guide is by no means comprehensive, it gives you enough material to begin working with style sheets and using them on your site. After you have mastered CSS1, you might want to begin exploring the new features added in CSS2. Although many of these new features are not well supported yet, in time they will be. If you begin learning them now, you'll be ready to implement them in your Web sites when they are better supported.