APPENDIX B: CSS QUICK REFERENCE[1]

Style Rules

The example below illustrates a typical style rule. The `p` represents the HTML element `<p></p>`. The style rule contains instructions to use the font-family arial at 90% size. If the visitor's browser can't find the arialfont, it will use its default sans-serif font instead. A style rule consists of a **selector** and a set of one or more **declarations** surrounded by curly braces. In this example the `p` is a selector. A declaration is the combination of a **property** and its **values**. In this example, `font-family` and `font-size` are properties, and the font names and 90% are values. Use a colon between the property and its values. Use a semicolon at the end of each declaration.

```
p { font-family: arial, sans-serif; font-size: 90%; }
```

Selectors

Element selectors refer to HTML elements. They may be grouped as in the example below.

```
h1 { font-family: sans-serif; color: maroon; }
h1, h2, h3, h4, h5, h6 { font-family: sans-serif; color:
maroon; }
```

Contextual selectors indicate the style of an HTML element in a given context. The example below sets the color of bold text to green wherever the `<b></b>` tag pair is contained within a `<h1></h1>` tag pair.

```
h1 b { color: green; }
```

Class selectors change the style of HTML content identified as belonging to a given class using the `class` attribute. The example below shows the style rule and the HTML `class` attribute. Notice the selector begins with a period.

```
.huge { font-size: 300%; }
<p class="huge">This will be really big text.</p>
```

ID selectors change the style of HTML content identified using the `id` attribute. The example below shows the style rule and the HTML `id` attribute. Notice the selector begins with a hash mark.

---

[1] I adapted this appendix from my "Incredibly Abridged Guide to CSS," available at `http://www.stevenestrella.com/IWP`.

```
#tiny { font-size: 50%; }
<p id="tiny">This will be really small text.</p>
```

Pseudo-class selectors are available only for the `<a></a>` tag to reflect the different states of a link. Note the use of the colon within the selector.

```
a:link { color: blue; }
a:visited { color: black; }
a:active { color: green; }
a:hover { color: red; }
```

Pseudo-element selectors are available to identify subparts of an element. The commonly used pseudo-elements are first-letter (for drop caps, mostly) and first-line. The drop-cap effect is usually created using a class selector as seen below. This allows you to create a drop-cap effect on any paragraph just by changing its `class` attribute value.

```
p.dropcap:first-letter { font-size: 300%; float: left;
color: red; }
p:first-line { color: red; }
```

Values

You may choose values using Keywords, length values, percentage values, colors, and URLs.

Keywords include values such as small, large, medium, normal, dotted, and so on.

Length values are positive or negative numbers followed by a two-character abbreviation to indicate the unit of measurement. Absolute values can be stated in inches (`in`), centimeters (`cm`), points (`pt`; 1 inch is 72 points), or picas (`pc`; 1 pica is 12 points). You can state relative values in em-height (`em`; the height of the character box for a given font), X-height (`ex`; the height of the lowercase x in a given font), or pixels (`px`). In practice, `pt` is the most consistently implemented absolute value and `em` is the best choice for a relative value. X-height is reliable only on IE5Mac.

Percentage values are often used for font size and line height. Percentage values are relative to the default values of the visitor's browser.

Colors may be given in hexadecimal or RGB notations, or you can use selected color names.

The standard hex-pair notation gives red, green, and blue values for a color in three pairs of two-character hexadecimal numbers (#ffffff for white, #000000 for black, #ff0000 for red, and so on). The short-form hexadecimal notation uses only the first character of each pair and assumes the second character to be the same (#f0f would create the same purple color as #ff00ff).

RGB notation may use percentage values or decimal values (0 to 255) for red, green, and blue. Both notations below would produce a purple color by combining red and blue with no green.

```
rgb (100%,0%,100%)
rgb (255,0,255)
```

Keywords exist for the 16 colors aqua, black, blue, fuchsia, gray, green, lime, maroon, navy, olive, purple, red, silver, teal, white, and yellow.

URLs are often used to choose background images for a page. CSS defines URLs as relative to the page containing the style sheet.

```
url( niftybackground.jpg )
```

Style Sheet Placement

Inline styles may be added directly to a tag. The W3C discourages this usage, however, because it mixes structure with style in a fashion reminiscent of the old `<font>` tag.

```
<p style="color: green; font-size: 90%;">Small green
text here.</p>
```

Internal style sheets (also called embedded style sheets) can be added to the `<head></head>` section of a page.

```
<html><head><title>Always in style</title>
<style type="text/css">
body { font-family: serif; color: black; }
h1 { font-family: sans-serif; color: maroon; }
</style>
<body>
<h1>A heading in sans-serif and maroon</h1>
<p>All other text will be in a serif font and black</p>
</body>
</html>
```

External style sheets can be created as a set of rules in a separate text file. Use the `<link>` tag in the `<head></head>` section of the document to link to the external style sheet. Using external style sheets is the most efficient way to consistently style Web sites with many pages.

First create a plain text file called something like `mystyle.css` and add your style rules.

```
body { font-family: serif; color: black; }
h1 { font-family: sans-serif; color: maroon; }
```

Then link to the external style sheet using the `<link>` tag.

```
<html><head><title>Always in style</title>
<link rel="stylesheet" type="text/css"
href="mystyle.css">
<body>
<h1>A heading in sans-serif and maroon</h1>
<p>All other text will be in a serif font and black</p>
</body>
</html>
```

Conflicting Rules

A Web page may link to an external style sheet, use an internal style, and also use inline styles. When multiple style rules apply to the same content, the Web browser uses a set of rules to resolve the conflict based on a cascade order. The rules are complex but are based on easily understood principles of specificity and proximity. The more specifically a rule applies to an element, the more weight it is given in resolving any conflicts. An ID selector, for example, is more specific than a class selector, which in turn is more specific than an element selector. Proximity matters as well. An inline style will override an internal style, which in turn will override a rule found in an external style sheet. To force a given rule to take precedence you may use the `!important` value as part of the declaration.

```
h1 { color: maroon !important;}
```

Element Classification

CSS recognizes three types of elements. **Block-level elements** include paragraphs, headings, lists, tables, and divs. Block-level elements always begin on a new line. **Inline elements** include anchors (`<a></a>`), spans, form elements, and images. Inline elements exist in the natural flow of the document without forced line breaks. Inline elements may contain other inline elements but may not contain block-level elements. **List-item elements** are the `<li></li>` elements found in unordered and ordered lists. List-item elements have markers associated with them such as bullets, numbers, or letters.

Element Box

An invisible element box surrounds every element in CSS. Each element box has an outer edge, margin, border, padding, and inner edge that contain the width and height of the content.

Property Reference Tables

The property reference tables below show properties and values that work reliably in IE5Mac, NN6+, and IE5.5+ on Windows. These browsers will be referred to as **DHTML browsers** in this appendix.

Font Properties

Table B-1 lists the properties and values related to working with fonts.

Table B-1 Font Properties

Property	Values	Comments/Examples
`font-family`	Font family name (`arial`, `courier new`, and so on) Generic family name (`serif`, `sans-serif`, `monospace`, `cursive`, or `fantasy`)	Place quotes around family names with multiple words (for example, `"times new roman"`). The Web browser determines the default values for font properties. `p {font-family: arial;}`
`font-style`	`normal`, `italic`, or `oblique`	`p {font-style: italic;}`
`font-variant`	`normal` or `small-caps`	`p {font-variant: small-caps;}`
`font-weight`	`normal`, `bold`, `bolder`, or `lighter`	`p {font-weight: bold;}`
`font-size`	Absolute size (`14pt`, `1.2pc`) Relative size (`80%`, `1.2em`)	`p {font-size: 95%;}`
`font`	`font-style`, `font-variant`, `font-weight`, `font-size`, `line-height`, and `font-family`	This shorthand property specifies multiple font properties with one style rule.

When using the `font` shorthand property only `font-size` and `font-family` are required. The hard part is remembering the correct order of the properties required by CSS for this shortcut to work. Here are two examples.

```
p { font: italic small-caps bold 14pt/16pt arial,
sans-serif;}
p {font: 18pt serif; }
```

The first example specifies all font properties: `italic` is a font style, `small-caps` is a font variant, `14pt/16pt` indicates a 14-point font size with a 16-point line height, and `arial, sans-serif` is the font family. The second example specifies the minimum properties of size and family.

Browser-Safe Fonts

It is impossible to be sure exactly which fonts are available on your visitor's computer system. Given that most visitors use either Macintosh or Windows operating systems, however, you may wish to specify commonly available fonts in your style rules. Here are a few style rules to guide you. Each rule represents fonts usually available on both Macintosh and Windows systems. The order of the fonts is up to you. If you wish to give preference to a font available only on Macintosh or Windows, just add it as the first font in the list. The browser will use the second font in the list for those visitors using other platforms.

```
p { font-family: georgia, "times new roman", serif; }
p { font-family: verdana, arial, impact, sans-serif; }
p { font-family: "courier new", monaco,
"lucida console", monospace; }
```

Figure B-1 shows the fonts listed in the code above.

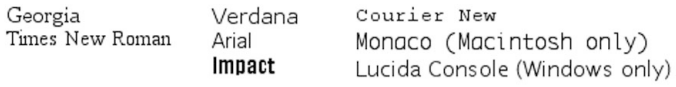

Figure B-1 Some Browser-Safe Fonts

You can reliably implement serif, sans-serif, and monospace fonts in style sheets. The remaining generic font families, cursive and fantasy, yield extremely unpredictable results and are not recommended at this time.

Color

The `color` property is commonly applied to fonts, but is also applied to other elements. Any of the 16 named color values may be used or any of millions of values may be given using hexadecimal numbers or RGB notation.

```
p { font-family: serif; color: red; }
p { font-family: serif; color: #ff0000; }
p { font-family: serif; color: rgb(255,0,0); }
```

Background Properties

Table B-2 lists the properties and values related to working with backgrounds.

Table B-2 Background Properties

Property	Values	Comments/Examples
background-color	Any valid color value	Backgrounds are transparent by default. `body {background-color: aqua;}`
background-image	Any valid absolute URL or a URL relative to the style sheet.	`body {background-image: url(images/niftybg.jpg);}`
background-repeat	`repeat` (default tiling) `repeat-x` (horizontal tiling) `repeat-y` (vertical tiling) `no-repeat` (no tiling)	This property is almost always used with a background image. The `repeat` value is the default for the usual tiling behavior of backgrounds. Use `no-repeat` to create interesting background effects together with a background color. `body { background-image: url(images/niftybg.jpg); background-color: aqua; background-repeat: no-repeat;}`
background-attachment	`scroll` (default) `fixed`	This property determines whether or not the background image scrolls with the page. `body { background-image: url(images/niftybg.jpg); background-attachment: fixed;}`

background-position	Percentage (`50% 50%`)
	
	Length (`2in 4in`)
	
	`top`, `center`, or `bottom`
	
	`left`, `center`, or `right`
		Set the initial horizontal and vertical position of a background image with this property. Use it with `background-repeat` to center a background image on a page.
		
		`body { background-image: url(images/niftybg.jpg); background-repeat: no-repeat; background-position: 50% 50%;}`

Table B-2 Background Properties *(continued)*

Property	Values	Comments/Examples
`background`	`background-color`, `background-image`, `background-repeat`, `background-attachment`, and `background-position`	This shorthand property specifies multiple background properties with one style rule. Separate the properties with spaces. `body { background: aqua url(images/niftybg.jpg) no-repeat fixed center top;}`

Text Properties

Table B-3 lists the properties and values related to working with text.

Table B-3 Text Properties

Property	Values	Comments/Examples
`letter-spacing`	Any valid length value	`p {letter-spacing: 0.2em;}`
`line-height`	Any valid number, length, or percentage value	Use of a simple number is recommended to make the line height relative to the font size. `p { line-height: 1.15; }`
`text-align`	`left`, `right`, `center`, or `justify`	`p { text-align: right;}`
`text-decoration`	`none`, `underline`, `overline`, `line-through`, or `blink`	This property is often used with the anchor tag to remove the underlining from links. Only `underline` and `none` work reliably on all DHTML browsers. `a {text-decoration: none;}`
`text-indent`	Any valid length or percentage value	Use this property with block-level elements only. It is often used with a class selector. `p.indented { text-indent: 4em; }`

Table B-3 Text Properties *(continued)*

Property	Values	Comments/Examples
`text-transform`	`capitalize` (This Text) `uppercase` (THIS TEXT) `lowercase` (this text)	`p { text-transform: uppercase;}`
`word-spacing`	Any valid length value	`p {word-spacing: 0.6em;}`

Box Properties

Table B-4 lists the properties and values related to working with boxes.

Table B-4 Box Properties

Property	Values	Comments/Examples
`margin-top`	Any valid length value Any valid percentage value `auto`	`p { margin-top: 5px; }`
`margin-right`	Any valid length value Any valid percentage value `auto`	`p { margin-right: 105%; }`
`margin-bottom`	Any valid length value Any valid percentage value `auto`	`p { margin-bottom: auto; }`
`margin-left`	Any valid length value Any valid percentage value `auto`	`p { margin-left: 2em; }`
`margin`	Any valid length value Any valid percentage value `auto`	This shorthand property sets all `margin` properties to the same value using one style rule. `p { margin: 12px; }`
`padding-top`	Any valid length value Any valid percentage value `0` (default)	`p { padding-top: 5px; }`
`padding-right`	Any valid length value Any valid percentage value `0` (default)	`p { padding-right: 105%; }`
`padding-bottom`	Any valid length value Any valid percentage value `0` (default)	`p { padding-bottom: 2em; }`

Table B-4 Box Properties *(continued)*

Property	Values	Comments/Examples
`padding-left`	Any valid length value Any valid percentage value `0` (default)	`p { padding-left: 129%; }`
`padding`	Any valid length value Any valid percentage value `0` (default)	This shorthand property sets all padding properties to the same value using one style rule. `p { padding: 12px; }`
`border-top-width`	`medium`, `thin`, `thick`, or any valid length value	`div { border-top-width: thin; }`
`border-right-width`	`medium`, `thin`, `thick`, or any valid length value	`div { border-right-width: thick; }`
`border-bottom-width`	`medium`, `thin`, `thick`, or any valid length value	`div { border-bottom-width: 5px; }`
`border-left-width`	`medium`, `thin`, `thick`, or any valid length value	`div { border-left-width: 2em; }`

`border-color`	Any valid color value	Use one color for all sides or list four colors for top, right, bottom, and left, respectively.
		
		`div { border-color: red; }`
		
		`div { border-color: red black yellow blue; }`

`border-style`	`none`, `dotted`, `dashed`, `solid`, `double`, `groove`, `ridge`, `inset`, or `outset`	This property is often used together with `border-color`. If a browser does not recognize a given value, `solid` is used instead. `div { border-color: red; border-style: dotted; }`
`border-top`	`border-top-width`, `border-style`, and `color`	This shorthand form sets the width, style, and color, respectively, of the top border using one style rule. `div { border-top: thin solid green; }`

Table B-4 Box Properties *(continued)*

Property	Values	Comments/Examples
`border-right`	`border-right-width`, `border-style`, and `color`	This shorthand form sets the width, style, and color, respectively, of the right border using one style rule. `div { border-right: 2px dashed red; }`
`border-bottom`	`border-bottom-width`, and `border-style`, `color`	This shorthand form sets the width, style, and color, respectively, of the bottom border using one style rule. `div { border-bottom: thick dotted black; }`
`border-left`	`border-left-width`, `border-style`, and `color`	This shorthand form sets the width, style, and color, respectively, of the left border using one style rule. `div { border-left: 2px solid blue; }`
`border`	`border-width`, `border-style`, and `color`	This shorthand property specifies all border properties using one style rule. `div { border: thin solid blue; }`
`width`	Any valid length value Any valid percentage value `auto` (default)	This property applies to block-level and replaced elements.* It also applies to CSS2 positioned elements with the `position` property set to `absolute`. `div { width: 600px; }`
`height`	Any valid length value Any valid percentage value `auto` (default)	This property applies to block-level and replaced elements.* It also applies to CSS2 positioned elements with the `position` property set to `absolute`. `div { height: 60px; }`

Table B-4 Box Properties *(continued)*

Property	Values	Comments/Examples
`float`	`none`, `left`, or `right`	This property is often applied to images to allow text to float around them. `img { float: left; }`
`clear`	`none`, `left`, `right`, or `both`	This property sets the side or sides of an element on which no image may appear. `br { clear: left; }`

* A replaced element is an element for which the CSS formatter knows only the intrinsic dimensions. In HTML, the IMG, INPUT, TEXTAREA, SELECT, and OBJECT elements can be replaced elements. For example, the content of the IMG element is often replaced by the image that the `src` attribute designates.

Classification Properties

Table B-5 lists the properties and values related to classification.

Table B-5 Classification Properties

Property	Values	Comments/Examples
`display`	`block`, `inline`, `list-item`, or `none`	Support for `block` and `none` is reliable. Support for `inline` and `list-item` is quirky in most browsers. `div {display: none;}`
`white-space`	`normal`, `pre`, or `nowrap`	Use `pre` to cause the browser to display multiple spaces and carriage returns. Use `nowrap` to prevent line breaks. `td { white-space: nowrap; }`

`list-style-type`	`disc`, `circle`, `square`, `decimal`, `lower-roman`, `upper-roman`, `lower-alpha`, `upper-alpha`, or `none`	`ul { list-style-type: circle;}`
		
		`ol { list-style-type: upper-roman;}`

`list-style-image`	`none` or any valid URL	`ul { list-style-image: url(mybullet.gif);}`

Table B-5 Classification Properties *(continued)*

Property	Values	Comments/Examples
`list-style-position`	`outside` (default) or `inside`	`ul { list-style-position: inside;}`
`list-style`	`list-style-type`, `list-style-position`, and `list-style-image`	This shorthand property specifies multiple list properties with one style rule. Separate the properties with spaces. `ul { list-style: square outer url(niftydot.jpg);}`

Positioning Properties (CSS2)

Table B-6 lists the properties and values related to positioning.

Table B-6 Positioning Properties

Property	Values	Comments/Examples
`position`	`static` (default), `absolute`, or `relative`	`div {position: relative;}`
`left`	Any valid length value Any valid percentage value `auto`	This property applies only to elements with the `position` property set to `absolute` or `relative`. `div { position: absolute; left: 100px; }`
`top`	Any valid length value Any valid percentage value `auto`	This property applies only to elements with the `position` property set to `absolute` or `relative`. `div { position: absolute; top: 10px; }`
`z-index`	Any positive or negative integer `auto`	Higher `z-index` values place the element closer to the viewer. `div { position: absolute; z-index: 4; }`

Table B-6 Positioning Properties *(continued)*

Property	Values	Comments/Examples
`clip`	`auto`, `rect`, or any valid shape value	This property applies to absolutely positioned elements. Today the only recognized shape is `rect`. The dimensions in parentheses represent the top, right, bottom, and left coordinates, respectively, of an imaginary rectangle through which we can view the content. `img { clip: rect(0px 300px 200px 30px);}`
`overflow`	`visible`, `hidden`, `scroll`, or `auto`	This property applies to absolutely positioned elements. `div { overflow: scroll;}`
`visibility`	`inherit`, `visible`, or `hidden`	`div { visibility: hidden; }`