

Appendix B: HTML Special Characters Quick Reference

Decimal Entity	Mnemonic Entity	Character
Τ	Τ	Greek capital letter tau
Υ	Υ	Greek capital letter upsilon
Φ	Φ	Greek capital letter phi
Χ	Χ	Greek capital letter chi
Ψ	Ψ	Greek capital letter psi
Ω	Ω	Greek capital letter omega
α	α	Greek small letter alpha
β	β	Greek small letter beta
γ	γ	Greek small letter gamma
δ	δ	Greek small letter delta
ε	ε	Greek small letter epsilon
ζ	ζ	Greek small letter zeta
η	η	Greek small letter eta
θ	θ	Greek small letter theta
ι	ι	Greek small letter iota
κ	κ	Greek small letter kappa
λ	λ	Greek small letter lambda
μ	μ	Greek small letter mu
ν	ν	Greek small letter nu
ξ	ξ	Greek small letter xi
ο	ο	Greek small letter omicron
π	π	Greek small letter pi
ρ	ρ	Greek small letter rho
ς	ς	Greek small letter final sigma
σ	σ	Greek small letter sigma
τ	τ	Greek small letter tau
υ	υ	Greek small letter upsilon
φ	φ	Greek small letter phi
χ	χ	Greek small letter chi
ψ	ψ	Greek small letter psi

continued

Appendix B: HTML Special Characters Quick Reference

TABLE B-8 (continued)

Decimal Entity	Mnemonic Entity	Character
ω	ω	Greek small letter omega
ϑ	ϑ	Greek small letter theta symbol
ϒ	ϒ	Greek upsilon with hook symbol
ϖ	ϖ	Greek pi symbol

Table B-9 lists a variety of mathematical symbols.

TABLE B-9

Mathematical Symbol Entities

Decimal Entity	Mnemonic Entity	Character
×	×	Multiplication sign
÷	&division;	Division sign
∀	∀	For all
∂	∂	Partial differential
∃	∃	There exists
∅	∅	Empty set = null set = diameter
∇	∇	Nabla = backward difference
∈	∈	Element of
∉	∉	Not an element of
∋	∋	Contains as member
∏	∏	n-ary product = product sign
∑	∑	n-ary summation
−	−	Minus sign
∗	∗	Asterisk operator
√	√	Square root = radical sign
∝	∝	Proportional to
∞	∞	Infinity

Appendix B: HTML Special Characters Quick Reference

Decimal Entity	Mnemonic Entity	Character
∠	∠	Angle
∧	∧	Logical and = wedge
∨	∨	Logical or = vee
∩	∩	Intersection = cap
∪	∪	Union = cup
∫	∫	Integral
∴	∴	Therefore
∼	∼	Tilde operator = varies with = similar to
≅	≅	Approximately equal to
≈	≈	Almost equal to = asymptotic to
≠	≠	Not equal to
≡	≡	Identical to
≤	≤	Less than or equal to
≥	≥	Greater than or equal to
⊂	⊂	Subset of
⊃	⊃	Superset of
⊄	⊅	Not a subset of
⊆	⊆	Subset of or equal to
⊇	⊇	Superset of or equal to
⊕	⊕	Circled plus = direct sum
⊗	⊗	Circled times = vector product
⊥	⊥	Up tack = orthogonal to = perpendicular
⋅	⋅	Dot operator
⌈	⌈	Left ceiling
⌉	⌉	Right ceiling
⌊	⌊	Left floor
⌋	⌋	Right floor
〈	⟨	Left-pointing angle bracket
〉	⟩	Right-pointing angle bracket

Appendix B: HTML Special Characters Quick Reference

Table B-10 lists other miscellaneous entities.

TABLE B-10

Miscellaneous Entities

Decimal Entity	Mnemonic Entity	Character
¡	¡	Inverted exclamation mark
¦	¦	Broken bar = broken vertical bar
§	§	Section sign
¨	¨	Diaeresis = spacing diaeresis
ª	ª	Feminine ordinal indicator
«	«	Left-pointing double angle quotation mark = left pointing guillemet
¬	¬	Not sign
­	­	Soft hyphen = discretionary hyphen
¯	¯	Macron = spacing macron = overline = APL overbar
°	°	Degree sign
±	±	Plus-minus sign = plus-or-minus sign
²	²	Superscript two = superscript digit two = squared
³	³	Superscript three = superscript digit three = cubed
´	´	Acute accent = spacing acute
µ	µ	Micro sign
¶	¶	Pilcrow sign = paragraph sign
·	·	Middle dot = Georgian comma = Greek middle dot
¸	¸	Cedilla = spacing cedilla
¹	¹	Superscript one = superscript digit one
º	º	Masculine ordinal indicator
»	»	Right-pointing double angle quotation mark = right pointing guillemet
¼	¼	Vulgar fraction one quarter = fraction one quarter
½	½	Vulgar fraction one half = fraction one half
¾	¾	Vulgar fraction three quarters = fraction three quarters
¿	¿	Inverted question mark = turned question mark
Œ	Œ	Latin capital ligature OE
œ	œ	Latin small ligature oe

Appendix B: HTML Special Characters Quick Reference

Decimal Entity	Mnemonic Entity	Character
Š	Š	Latin capital letter S with caron
š	š	Latin small letter s with caron
Ÿ	Ÿ	Latin capital letter Y with diaeresis
ˆ	ˆ	Modifier letter circumflex accent
˜	˜	Small tilde
 	 	Thin space
‌	‌	Zero width non-joiner
‍	‍	Zero width joiner
‎	‎	Left-to-right mark
‏	‏	Right-to-left mark
‚	‚	Single low-9 quotation mark
„	„	Double low-9 quotation mark
†	†	Dagger
‡	‡	Double dagger
‰	‰	Per mille sign
‹	‹	Single left-pointing angle quotation mark
›	›	Single right-pointing angle quotation mark

CSS 2.1 Properties Quick Reference

This appendix follows CSS 2.1, which is a specification intended to represent the most commonly supported properties in modern browsers.

Note that aural stylesheet properties are not covered in this appendix because adequate coverage of their use and capabilities goes well beyond a quick reference. For information on aural properties and their use, visit The Alliance for Technology Access website at www.ataccess.org.

Unfortunately, solid support for many of the CSS properties is spotty, so testing on a wide variety of Web browsers is recommended. In addition, several websites provide extensive information on CSS support across the popular browsers.

Cross-Ref

For an overview of CSS selectors, see Appendix D. ■

Property Listings

In each table that follows, words under Supported values that are capitalized are placeholders for either a set of possible values or values drawn from a related property. Examples of placeholders include the following:

- **Length** — Number followed by a unit of measurement, such as “px” for pixel
- **Percentage** — Number followed by a percent sign
- **Integer** — Whole number

IN THIS APPENDIX

Property Listings

Property List: Quick Reference

Background

List

Generated content

Font and text

Text direction

Block

Positioning

Borders

Table

Printing

Miscellaneous

Appendix C: CSS 2.1 Properties Quick Reference

Inherited refers to whether a given property will be drawn from the element's parents if it is not explicitly provided.

Property List: Quick Reference

The following list provides an overview of the full list of CSS properties. Use this list as a reference to the assorted attributes listed within the appendix.

Background

- background-image
- background-repeat
- background-attachment
- background-position
- background-color
- background

List

- list-style-type
- list-style-position
- list-style-image
- list-style

Generated content

- content
- quotes
- counter-increment
- counter-reset

Font and text

- text-align
- text-decoration
- text-indent
- text-transform
- color
- font-family
- font-size
- font-style
- font-variant
- font-weight
- font

letter-spacing

word-spacing

white-space

Text direction

unicode-bidi

direction

Block

margin-left, margin-right, margin-top, margin-bottom

margin

padding-left, padding-right, padding-top, padding-bottom

padding

clip

overflow

height, width

max-height, max-width

min-height, min-width

line-height

vertical-align

Positioning

visibility

display

position

float

top, bottom, left, right

z-index

clear

Borders

border-color, border-top-color, border-bottom-color, border-left-color, border-right-color

border-style, border-top-style, border-bottom-style, border-left-style, border-right-style

border-width, border-top-width, border-bottom-width, border-left-width, border-right-width

border

outline-color

outline-style

outline-width

outline

Appendix C: CSS 2.1 Properties Quick Reference

Table

- table-layout
- border-collapse
- border-spacing
- empty-cells
- caption-side

Printing

- page-break-after, page-break-before
- page-break-inside
- orphans
- widows

Miscellaneous

- cursor

Background

Listings of the background properties follow.

background-image

Place an image behind an element (typically the body of a document).

```
table { background-image: url("/images/draft.gif"); }
```

Supported values	url(), none, inherit
Default value	none
Inherited	No
Applies to	All

background-repeat

Define the background image behavior if it fails to fill its element.

```
table { background-repeat: none; }
```

Supported values	repeat, repeat-x, repeat-y, no-repeat, inherit
Default value	repeat
Inherited	No
Applies to	All

background-attachment

Specify whether the background image scrolls with the enclosing element.

```
table { background-attachment: fixed; }
```

Supported values	scroll, fixed, inherit
Default value	scroll
Inherited	No
Applies to	All
Note(s)	Browsers not required to support “fixed”

background-position

Declare the initial position of a background image.

```
table { background-position 25% 25%; }
```

Supported values	Percentage, Length, top, center, bottom, left, right, inherit
Default value	0 0
Inherited	No
Applies to	All
Note(s)	If two values are supplied, the first is a horizontal position and the second vertical. If one numeric value is supplied, it is treated as a horizontal position, and vertical will be 50 percent.

background-color

Define the background color for an element.

```
body { background-color: black; }
```

Supported values	Color, transparent, inherit
Default value	transparent
Inherited	No
Applies to	All

Appendix C: CSS 2.1 Properties Quick Reference

background

Consolidate background properties.

```
table { background: url("/images/draft.gif") none fixed 25% 25%; }
```

Supported values	Color, Image, Repeat, Attachment, Position, inherit
Default value	transparent, none, repeat, scroll, 0 0
Inherited	No
Applies to	All

List

These properties apply to the rendering of lists. See also `counter-increment` and `counter-reset` in the section “Generated Content” later in this appendix.

list-style-type

Select the bullet markers for a list.

```
ul.nobullet { list-style-type: none; }
```

Supported values	disc, circle, square, decimal, decimal-leading-zero, lower-roman, upper-roman, lower-greek, lower-latin, upper-latin, armenian, georgian, lower-alpha, upper-alpha, none, inherit
Default value	disc
Inherited	Yes
Applies to	li

list-style-position

Indicate whether the list markers should be treated as internal to the box enclosing each list item.

```
ul.paragraphs { list-style-position: inside; }
```

Supported values	inside, outside, inherit
Default value	outside
Inherited	Yes
Applies to	li

list-style-image

Refer to an image to be used for bullet markers.

```
ul { list-style-image: url("/images/daggers.gif"); }
```

Supported values	url(), none, inherit
Default value	none
Inherited	Yes
Applies to	li

list-style

Consolidate list-style properties.

```
ul { list-style: circle outside url("/images/daggers.gif"); }
```

Supported values	Type Position Image, inherit
Default value	disc outside none
Inherited	Yes
Applies to	li
Note(s)	If both a type and an image are supplied, the list-style type will be used if the image cannot be retrieved.

Generated content

CSS provides for the insertion of new text in certain locations via the `content` property.

The other properties in this section affect the text inserted by the `content` property by modifying the open/close quotes or by impacting a named counter value. Many of these properties and methods are not widely supported.

content

Text to be displayed by autogeneration properties.

```
.quote:before { content: open-quote; }  
.quote:after { content: close-quote; }
```

Supported values	normal, String, url(), counter(), counters(), attr(), open-quote, close-quote, no-open-quote, no-close-quote, inherit
Default value	Normal
Inherited	No
Applies to	:before, :after pseudo-elements

Appendix C: CSS 2.1 Properties Quick Reference

quotes

Define quotation marks for use with `q` elements and content properties.

```
body { quotes: "\00AB" "\00BB"; }
```

Supported values	String, none, inherit
Default value	Browser-defined
Inherited	Yes
Applies to	All

counter-increment

Indicate that the named counter should be incremented by one or the numeric value provided.

```
div.section { counter-increment: sectionheading; }
```

Supported values	Identifier Integer, none, inherit
Default value	none
Inherited	No
Applies to	All

counter-reset

Indicate that the named counter should be set back to zero or the numeric value provided.

```
div.section { counter-reset: sectionsubheading; }
```

Supported values	Identifier Integer, none, inherit
Default value	none
Inherited	No
Applies to	All

Font and text

These properties are used to specify the way in which text is rendered.

text-align

Specify the text alignment within the block.

```
pre.poem { text-align: center; }
```

Appendix C: CSS 2.1 Properties Quick Reference

Supported values	center, left, right, justify, inherit
Default value	left (but see Notes in this table)
Inherited	Yes
Applies to	Block elements, td, th, and form input fields
Note(s)	The default value is “correct” if <code>direction: rtl</code> is set.

text-decoration

Augment the text with underlining or similar properties.

```
p.annoying { text-decoration: line-through blink; }
```

Supported values	none, underline, overline, line-through, blink, inherit
Default value	none
Inherited	No
Applies to	All
Note(s)	Several decorations may be listed with whitespace separation.

text-indent

Specify the indentation for the first line in a block.

```
p { text-indent: 1em; }
```

Supported values	Length, Percentage, inherit
Default value	0
Inherited	Yes
Applies to	Block elements, td, th, and form input fields.

text-transform

Convert text to uppercase or lowercase.

```
span.customername { text-transform: uppercase; }
```

Appendix C: CSS 2.1 Properties Quick Reference

Supported values	capitalize, uppercase, lowercase, none, inherit
Default value	none
Inherited	Yes
Applies to	All

color

Define text color.

```
div.hardtoread { color: yellow; }
```

Supported values	Color, inherit
Default value	Browser-defined
Inherited	Yes
Applies to	All
Note(s)	Any borders in this scope will default to this color.

font-family

Define the desired typeface.

```
body { font-family: Garamond, serif; }
```

Supported values	Family (one or more comma-separated values), inherit
Default value	Browser-defined
Inherited	Yes
Applies to	All
Note(s)	Use quotes around font family names that include spaces. Be sure to provide generic families as alternatives should the browser not be able to locate the font you prefer.

font-size

Specify the type size.

```
caption { font-size: x-large; }
```


Appendix C: CSS 2.1 Properties Quick Reference

Supported values	Length, Percentage, xx-large, x-large, large, medium, small, x-small, xx-small, larger, smaller, inherit
Default value	medium
Inherited	Yes
Applies to	All

font-style

Render the enclosed text as italic, oblique, or normal.

```
.booktitle { font-style: italic; }
```

Supported values	normal, italic, oblique, inherit
Default value	normal
Inherited	Yes
Applies to	All

font-variant

Render the enclosed text as small capitals or normal.

```
span.manufacturername { font-variant: small-caps; }
```

Supported values	normal, small-caps, inherit
Default value	normal
Inherited	Yes
Applies to	All

font-weight

Specify the “boldness” of text.

```
caption { font-weight: 900; }
```

Appendix C: CSS 2.1 Properties Quick Reference

Supported values	normal, bolder, bold, lighter, 100, 200, 300, 400, 500, 600, 700, 800, 900
Default value	normal
Inherited	Yes
Applies to	All
Note(s)	normal is equivalent to 400; bold to 700.

font

Consolidate font properties or specify system fonts.

```
h6 { font: menu; }
```

Supported values	caption, icon, menu, message-box, small-caption, status-bar, inherit (and see Note(s))
Default value	See font-style, font-variant, font-weight, font-size
Inherited	Yes
Applies to	All
Note(s)	The value can be one of the preceding or a combination of the other font properties with line-height thrown in to confuse things.

letter-spacing

Add to the spacing between letters.

```
blockquote { letter-spacing: 0.1em; }
```

Supported values	Length, normal, inherit
Default value	normal
Inherited	Yes
Applies to	All
Note(s)	The value may be negative.

word-spacing

Add to the spacing between words.

```
h2 { word-spacing: 1em; }
```

Supported values	Length, normal, inherit
Default value	normal
Inherited	Yes
Applies to	All

white-space

Specify the handling of whitespace, including line wrapping.

```
blockquote.poem { white-space: pre; }
```

Supported values	normal, pre, nowrap, pre-wrap, pre-line, inherit
Default value	normal
Inherited	Yes
Applies to	All

Text direction

These properties are required to deal with the problems arising from the fact that some languages read right to left while others read left to right.

unicode-bidi

This, combined with the `direction` property, handles the directionality of text for a document. This property is useful only when two languages of different directionality are present.

```
span.arabic { unicode-bidi: embed; direction: rtl; }  
span.english { unicode-bidi: embed; direction: ltr; }
```

Supported values	normal, embed, bidi-override, inherit
Default value	normal
Inherited	No
Applies to	All

Appendix C: CSS 2.1 Properties Quick Reference

direction

Define the direction for the enclosed text:

```
p.english { direction: ltr; }
```

Supported values	ltr, rtl, inherit
Default value	ltr
Inherited	Yes
Applies to	All except for inline elements with <code>unicode-bidi: normal</code>

Block

The key differentiator between padding and margin: The padding is inside any border around a block, and the margin is outside that border.

margin-left, margin-right, margin-top, margin-bottom

Define the size of the margin on a given side of a block.

```
blockquote { margin-left: 10%; }
```

Supported values	Length, Percentage, auto
Default value	0
Inherited	No
Applies to	All except for table components (e.g., td, tfoot, tr)

margin

Consolidate margin widths.

```
p { margin: 1em 0 1em 0; }
```

Supported values	Length, Percentage, auto (up to four values defining up to the four sides)
Default value	0 0 0 0
Inherited	No
Applies to	All except for table components (that is, td, tfoot, tr)
Note(s)	Order of values: top, right, bottom, left.

padding-left, padding-right, padding-top, padding-bottom

Define the size of the padding on a given side of a block.

```
div.withborder { padding-top: 2%; }
```

Supported values	Length, Percentage, inherit
Default value	0
Inherited	No
Applies to	All excluding table components (but including td, th)

padding

Consolidate padding widths.

```
div.withborder { padding: 2% 0 0 0; }
```

Supported values	Length, Percentage, inherit (up to four values)
Default value	0 0 0 0
Inherited	No
Applies to	All excluding table components (but including td and th)
Note(s)	Order of values: top, right, bottom, left.

clip

Define a boundary for an element outside of which any presentation (text, border) should be clipped.

```
blockquote { clip: rect(5px, 20px, 20px, 5px); overflow: scroll; }
```

Supported values	Shape, auto, inherit
Default value	auto
Inherited	No
Applies to	Absolutely positioned elements
Note(s)	If overflow is visible, this has no effect. The only recognized shape is rect().

Appendix C: CSS 2.1 Properties Quick Reference

overflow

Specify what happens when a block's content is larger than the clipping area.

```
blockquote { clip: rect(5px, 20px, 20px, 5px); overflow: scroll; }
```

Supported values	visible, hidden, scroll, auto, inherit
Default value	visible
Inherited	No
Applies to	Block elements, img, object, td, th

height, width

Specify the height or width of an element.

```
img.logo { height: 5cm; width: 5cm; }
```

Supported values	Length, Percentage, auto, inherit
Default value	auto
Inherited	No
Applies to	Block elements, img, object, and form input fields

max-height, max-width

Constrain element size.

```
table { max-width: 50%; }
```

Supported values	Length, Percentage, none, inherit
Default value	none
Inherited	No
Applies to	Block elements, img, object, and form input fields

min-height, min-width

Define a minimum element size.

```
textarea { min-width: 25%; }
```

Supported values	Length, Percentage, inherit
Default value	0
Inherited	No
Applies to	Block elements, <code>img</code> , <code>object</code> , and form input fields

line-height

Define line height. For block elements, this is the minimal line height; for inline, it is the specific height.

```
p { line-height: 150%; }
```

Supported values	Number, Length, Percentage, normal, inherit
Default value	normal
Inherited	Yes
Applies to	All
Note(s)	Unless an absolute measurement such as “cm” is used, this will be relative to the font size.

vertical-align

Define the vertical-alignment characteristics of this element relative to its line box (when top or bottom are specified) or its parent.

```
span.superscript { vertical-align: super; }
```

Supported values	Length, Percentage, baseline, sub, super, top, text-top, middle, bottom, text-bottom, inherit
Default value	baseline
Inherited	No
Applies to	Inline, <code>td</code> , <code>th</code>

Positioning

The preceding block properties specify the internal characteristics of the boxes that are used to lay out a page. The properties in this section can be used to describe the desired positions of those boxes.

Appendix C: CSS 2.1 Properties Quick Reference

visibility

Specify whether an element should be visible.

```
li.answer { visibility: hidden; }
```

Supported values	visible, hidden, collapse, inherit
Default value	visible
Inherited	Yes
Applies to	All
Note(s)	Unless <code>display: none</code> is set, the element will still occupy space, even if hidden.

display

Specify how an element should be presented.

```
.invisible { display: none; }
```

Supported values	none, inline, block, list-item, run-in, inline-block, table, inline-table, table-row-group, table-header-group, table-footer-group, table-row, table-column-group, table-column, table-cell, table-caption, inherit
Default value	inline
Inherited	No
Applies to	All
Note(s)	Other than removing objects from the document flow by setting <code>display</code> to <code>none</code> , this property is most valuable for defining the presentation of XML documents with no inherent style.

position

Specify the algorithm to be used for placing this element's containing box on the page.

```
div#menu { position: absolute; top: 3.8cm; left: 0; }
```


Supported values	static, relative, absolute, fixed, inherit
Default value	static
Inherited	No
Applies to	All

float

For elements that are not absolutely positioned, define their relationship to elements surrounding them.

```
div#logo { float: left; }
```

Supported values	left, right, none, inherit
Default value	none
Inherited	No
Applies to	All elements without display: none

top, bottom, left, right

For absolutely positioned elements, define the distance to the enclosing box's edges.

```
div#menu { position: absolute; top: 3.8cm; left: 0; }
```

Supported values	Length, Percentage, auto, inherit
Default value	auto
Inherited	No
Applies to	Positioned elements

z-index

Define stacking order for overlapping elements.

```
div#logo { z-index: 99; }
```

Appendix C: CSS 2.1 Properties Quick Reference

Supported values	Integer, auto, inherit
Default value	auto
Inherited	No
Applies to	Positioned elements
Note(s)	The higher the number, the higher on the stack.

clear

Specify which sides of an element's box may not be adjacent to a floating element.

```
h1 { clear: both; }
```

Supported values	none, left, right, both, inherit
Default value	none
Inherited	No
Applies to	Block elements
Note(s)	This element will be shifted to be below any floater.

Borders

These properties specify borders and outlines for the boxes used to lay out the page. Note that outlines are not widely supported.

border-color, border-top-color, border-bottom-color, border-left-color, border-right-color

Specify border colors.

```
div#logo { border-color: green; }
```

Supported values	Color, transparent, inherit
Default value	Element color property value
Inherited	No
Applies to	All

border-style, border-top-style, border-bottom-style, border-left-style, border-right-style

Specify the border design.

```
div#logo { border-style: groove; }
```

Supported values	none, hidden, dotted, dashed, solid, double, groove, ridge, inset, outset, inherit
Default value	none
Inherited	No
Applies to	All

border-width, border-top-width, border-bottom-width, border-left-width, border-right-width

Specify the border size.

```
div#logo { border-width: thin; }
```

Supported values	Length, thin, medium, thick, inherit
Default value	medium
Inherited	No
Applies to	All

border

Consolidate border properties.

```
div#logo { border: green groove thin; }
```

Supported values	Color Style Width, inherit
Default value	color, none, medium
Inherited	No
Applies to	All

Appendix C: CSS 2.1 Properties Quick Reference

outline-color

Specify outline color.

```
span.acronym { outline-color: blue; }
```

Supported values	Color, invert, inherit
Default value	invert
Inherited	No
Applies to	All

outline-style

Specify the outline style.

```
span.acronym { outline-style: dotted; }
```

Supported values	none, dotted, dashed, solid, double, groove, ridge, inset, outset, inherit
Default value	none
Inherited	No
Applies to	All

outline-width

Specify the outline width.

```
span.acronym { outline-width: thin; }
```

Supported values	Length, thin, medium, thick, inherit
Default value	medium
Inherited	No
Applies to	All

outline

Consolidate outline properties.

```
span.acronym { outline: blue dotted thin; }
```

Supported values	Color Style Width, inherit
Default value	invert, none, medium
Inherited	No
Applies to	All

Table

These properties specify how tables are rendered, primarily table borders.

table-layout

Specify a table layout algorithm. If `auto`, the table's contents will be scanned before generation to calculate the proper width of each column.

```
table.huge { table-layout: fixed; }
```

Supported values	auto, fixed, inherit
Default value	auto
Inherited	No
Applies to	table

border-collapse

Specify whether adjacent table cell borders should be consolidated.

```
table { border-collapse: collapse; }
```

Supported values	collapse, separate, inherit
Default value	separate
Inherited	Yes
Applies to	table

border-spacing

Define the space between internal table borders.

```
table { border-spacing: 2pt 4pt; }
```

Appendix C: CSS 2.1 Properties Quick Reference

Supported values	Length (1 or 2 values), <code>inherit</code>
Default value	0
Inherited	Yes
Applies to	<code>table</code>
Note(s)	If two values, the first is horizontal, the second vertical; otherwise, the value is applied to both dimensions.

empty-cells

Specify whether empty cells should be rendered with background and border.

```
table { empty-cells: hide; }
```

Supported values	<code>show</code> , <code>hide</code> , <code>inherit</code>
Default value	<code>show</code>
Inherited	Yes
Applies to	<code>td</code> , <code>th</code>

caption-side

Specify whether a caption is placed above or below its table.

```
table.figure { caption-side: bottom; }
```

Supported values	<code>top</code> , <code>bottom</code> , <code>inherit</code>
Default value	<code>top</code>
Inherited	Yes
Applies to	<code>caption</code>

Printing

These properties provide instructions to the user agent about how the page should be handled when printed. These can be used to help prevent inappropriate page breaks and otherwise format a printed page.

page-break-after, page-break-before

Specify whether a printed page break should occur before or after this block element.

```
h1 { page-break-after: avoid; }
```

Supported values	auto, always, avoid, left, right, inherit
Default value	auto
Inherited	No
Applies to	Block elements

page-break-inside

Specify a preference regarding page breaks internal to a block element.

```
table { page-break-inside: avoid; }
```

Supported values	avoid, auto, inherit
Default value	auto
Inherited	Yes
Applies to	Block elements

orphans

Define the minimum number of lines in a paragraph that must be left at the bottom of a page. Any fewer and the entire paragraph will wrap to the following page.

```
body { orphans: 3; }
```

Supported values	Integer, inherit
Default value	2
Inherited	Yes
Applies to	Block elements

widows

Define the minimum number of lines in a paragraph that must be available for the top of a page. Any fewer and the entire paragraph will wrap to that page.

```
body { widows: 4; }
```

Appendix C: CSS 2.1 Properties Quick Reference

Supported values	Integer, <code>inherit</code>
Default value	2
Inherited	Yes
Applies to	Block elements

Miscellaneous

The `cursor` property specifies what the user agent cursor should be when it is over the element to which the property is applied.

cursor

Define the type of cursor to be used when the mouse is over this element.

```
body.annoyuser { cursor: wait; }
```

Supported values	<code>auto</code> , <code>crosshair</code> , <code>default</code> , <code>pointer</code> , <code>move</code> , <code>nw-resize</code> , <code>n-resize</code> , <code>ne-resize</code> , <code>e-resize</code> , <code>se-resize</code> , <code>s-resize</code> , <code>sw-resize</code> , <code>w-resize</code> , <code>text</code> , <code>wait</code> , <code>help</code> , <code>progress</code> , <code>inherit</code> , <code>url()</code>
Default value	<code>auto</code>
Inherited	Yes (but see Note(s))
Applies to	All
Note(s)	The specification states that this is inherited, but browsers do not consistently do so. Use the <code>inherit</code> value if you want the parent's cursor to carry over to elements such as hyperlinks that would ordinarily have their own cursor type.

CSS 2.1 Selectors Quick Reference

CSS selectors are specific patterns used to match elements that will have the corresponding properties applied to them. CSS has many different patterns to match many different aspects of elements — their name/type, class, ID, place in the document hierarchy, and more.

In addition to using a single pattern to match elements, you can also combine patterns to create more specific matches. For example, the following selector matches all `h1` elements:

```
h1 { properties }
```

If you need more specificity, you can add a class selector as in the following example, which matches all `h2` elements with a class of `section`:

```
h2.section { properties }
```

You can take the selector even one step further by adding a descendant selector, as in the following example, which matches all `h2` elements with a class of `section` that are also descendants of `h1` elements:

```
h1 h2.section { properties }
```

The following sections provide a quick reference into the various CSS selector patterns.

Basic Element Selectors

The basic element selectors are used to match specific elements by name (e.g., `p`, `h1`, and so on).

IN THIS APPENDIX

Basic Element Selectors

Descendant Selectors

Child Selectors

Adjacent Sibling Selectors

Class Selectors

ID Selectors

Attribute Selectors

Appendix D: CSS 2.1 Selectors Quick Reference

Syntax:

```
E { properties }
```

Matches all E elements.

Syntax:

```
* { properties }
```

Matches all elements.

Note

The universal selector (*) guarantees only a universal (all element) match if it is the sole criteria in the selector. If additional conditions are added, they too must be met to match elements. For example, the following selector will match elements with a class of book only:

```
*.book { properties }
```

However, because of the use of the universal selector, *all* elements with a class of book will be matched. ■

Syntax:

```
E, F, G { properties }
```

Matches all E, F, and G elements.

Note

The comma separator can be used to specify a variety of selector patterns for the same selector definition, not just element name selectors. ■

Descendant Selectors

Descendant selectors are used to match elements that are descendants of other, specific elements.

Syntax:

```
E F { properties }
```

Matches all F elements that are descendants of E elements.

Child Selectors

Child selectors are used to match elements that are children (direct descendants) of other, specific elements.

Syntax:

```
E > F | properties { }
```

Matches all *F* elements that are children of *E* elements.

Note

Child selectors are very much like descendant selectors in scope in that they select elements that are descendants of a specific ancestor element. The distinction, however, is that children must be *direct descendants* of the specific ancestor element. That is, there can be no elements between the ancestor (parent) and descendant (child). ■

Adjacent Sibling Selectors

Adjacent sibling selectors are used to match the second of two elements that share the same parent.

Syntax:

```
E + F | properties { }
```

Matches all *F* elements that have the same direct sibling relationship (share the same parent) as *E* elements. Note that the *E* element must immediately precede the *F* element in the document tree.

Class Selectors

Class selectors are used to match elements of a particular class (elements that have a particular value for their `class` attribute).

Syntax:

```
E.c { properties }
```

Matches all *E* elements that have a class of *c*.

Tip

You can also use the attribute selector method to select elements that have the class attribute set to a specific value or set to any value. See the “Attribute Selectors” section later in this appendix for more information. ■

ID Selectors

ID selectors are used to match elements of a particular ID (elements that have a particular value for their `ID` attribute).

Appendix D: CSS 2.1 Selectors Quick Reference

Syntax:

```
E#i { properties }
```

Matches all E elements that have an ID of i.

Attribute Selectors

Attribute selectors are used to match elements that have a particular value for a particular attribute. The attribute selector has three different formats.

Syntax:

```
E[a] { properties }
```

Matches all E elements that have an attribute named a, no matter what the value.

```
E[a="v"]
```

Matches all E elements that have an attribute named a, with a value of v.

```
E[a~="v"]
```

Matches all E elements that have an attribute named a, with a space-separated list of values, one of which is v.

Tip

This selector method can also be used to match elements with a specific class using the format

```
E[class~="c"]
```

where c is the class value to match.

```
E[a|="v"]
```

Matches all E elements that have an attribute named a, with a hyphen-separated list of values, the first of which is v. ■

Pseudo-Elements and Pseudo-Classes Quick Reference

CSS selectors are designed to match HTML elements that are in a static state. Occasionally, however, you will want to match pieces of a document that cannot be clearly delimited by HTML entities, or match elements that are in a particular phase of a dynamic state. For these purposes, pseudo-elements and pseudo-classes exist.

Pseudo-elements provide the means to match certain parts of a document that aren't delimited by standard elements — the first line or first letter of an element's content, for example.

Pseudo-classes provide the means to match elements that are in a certain state — being the first child of a parent element, having the mouse hovering over the element, and so on.

Both pseudo-elements and pseudo-classes have the same format: a colon followed by a keyword that is appended to the end of a selector. For example, the following selector will select the first line of all `h1` elements:

```
h1:first-line { properties }
```

The following sections provide a quick reference for pseudo-elements and pseudo-classes.

Note

The pseudo-elements and pseudo-classes covered in this appendix are implemented in almost all modern browsers. Exceptions are the *before* and *after* pseudo-elements, which are not yet supported in Internet Explorer (as of version 7.0), and the *lang* pseudo-class, which is supported only in Internet Explorer for the Mac. The World Wide Web Consortium (W3C) has suggested many more pseudo-constructs for CSS level 3, which, as of this writing, is still in draft form. Pseudo-constructs are a powerful feature of CSS but should be tested on your target platform(s) before being implemented on a large scale. ■

IN THIS APPENDIX

Pseudo-Elements

Pseudo-Classes

Pseudo-Elements

Syntax:

```
E:first-line { properties }
```

Matches the first line of all E elements.

Syntax:

```
E:first-letter { properties }
```

Matches the first letter of all E elements.

Syntax:

```
E:before { properties } E:after { properties }
```

Matches the space immediately before (:before) and after (:after) all E elements.

Pseudo-Classes

Syntax:

```
E:first-child { properties }
```

Matches all E elements that are a first child of their parent element.

Syntax:

```
E:link { properties } E:visited { properties }
```

Matches all E elements that represent a link to another resource (:link) that has not been visited, or a link to another resource that has been previously visited (:visited).

Syntax:

```
E:active { properties } E:hover { properties } E:focus { properties }
```

Matches all E elements in an active state (:active) that are being hovered over by the mouse (:hover), or have the current focus (:focus).

Syntax:

```
E:lang(c)
```

Matches all E elements that are in language "c".

Index

A

- `<a>`. See anchor tags
- `<abbr>`. See abbreviation tag
- abbr (table cell attribute), 114
- abbreviation tag (`<abbr>`), 67–68, 619
- above (frame attribute value), 110
- absolute links, 90–92
- absolute positioning, 506–507
- absolute size, 441
- absolute size keywords (font size value metric), 441
- accented special characters, 237–239, 670–672
- accept attribute, 164
- accept-charset, 164
- accesskey attribute, 94–95, 174–175
- `<acronym>`, 68, 620
- ACSS (Aural CSS), 573
- action attribute, 162, 163, 174
- :active, 413, 551, 600
- active (link status mode), 96–97
- ActiveBorder (system color keyword), 584
- ActiveCaption (system color keyword), 584
- ActiveX controls, 223, 443
- acute accent, 243, 256, 676
- addCoupon, 179
- `<address>`, 337, 338, 340, 620–621
- adjacent sibling selectors, 280, 407, 410–412, 591, 707
- Adobe Dreamweaver, 314–315, 366, 590–591
- Adobe Fireworks, 196, 200, 215, 318
- Adobe Flash, 227, 229, 230, 299, 318–319, 352
- Adobe Freehand, 318
- Adobe HomeSite, 311–312
- Adobe Illustrator, 318
- Adobe Photoshop, 11, 196, 197, 198, 318
- Adobe's TechNote site, 220
- :after, 417–418, 419, 527, 530–531, 709
- align attribute
 - images and, 202–203
 - inline frame tag, 156
 - table cell, 114
 - table row tag, 112
- alignment
 - captions, 468–470
 - horizontal text, 445–448
 - images, 201–204
 - tables, 103–107
 - text, 445–450
 - valign attribute
 - table cell, 114
 - table row tag, 112
 - vertical text, 448–450
 - vertical-align property, 448–450, 451, 461, 695
- all (CSS media type), 554
- all (rules attribute value), 111
- Alliance for Technology Access website, 679
- almost equal to, 242, 675
- alpha (Greek letter), 672, 673

Index

- alphabets, in Unicode, 250–254
- alt attribute, 37, 200, 204, 205, 211, 356
- Altova, 347
- always (page-break-before/after property value), 561
- American Standard Code for Information Interchange (ASCII), 231, 235, 323, 324, 381
- ampersand (&), 32, 232, 233, 255, 667
- ancestors/descendants, 410
- anchor pseudo-classes, 551–552
 - :active, 413, 551, 600
 - :hover, 413, 551
 - :link, 413, 451
 - :visited, 413, 551
- anchor styles, 413–414
- anchor tags (<a>), 35–37, 96, 618–619
 - bookmarks and, 96
 - creating, 96
 - pseudo-classes for, 413–414
 - reference, 618–619
- angle (mathematical symbol entity), 242, 675
- angle brackets, 7, 13, 14, 31, 32, 137. *See also* brackets
 - left-pointing, 243, 675
 - right-pointing, 243, 675
- angles (aural metric), 429
- animated images, 199–200, 214–216
 - event triggers and, 265
 - GIF and, 215
- animation/video formats, 216–219
- Apache rewrite module, 355
- APIs (application program interfaces), 273, 343
- APL overbar, 243, 676
- apostrophe special characters, 669
- appendChild(), 277
- Apple Safari, 4, 368, 573
- application program interfaces (APIs), 273, 343
- approximately equal to, 242, 675
- AppWorkSpace (system color keyword), 584
- Arabic numbers, 72, 73
- <area>, 621
- Armenian value, 75, 250, 474, 684
- arrow special characters, 236, 669
- ascension (font characteristic), 438
- ASCII (American Standard Code for Information Interchange), 231, 235, 323, 324, 381
- ASP, 262
- ASP.NET, 262
- asterisk operator, 242, 674
- ASX markup language, 219, 223
- asymptotic to, 242, 675
- @ sign, entity equivalent and, 380
- @import rule, 434, 556
- @media rule, 554, 555, 569
- @page rule, 557–560, 564, 569
- attribute selectors, 707, 708
- attributes. *See also* event attributes; tag attributes; *specific attributes*
 - HTML5, 300–305
 - JavaScript DOM property binding, 276
 - XML, 333–334
- aural (CSS media type), 554
- Aural CSS (ACSS), 573
- aural style sheets
 - Alliance for Technology Access website and, 679
 - angles and, 423, 429–430
 - frequencies and, 423, 429–430
 - time values and, 423, 429–430
- author styles, 433–434
- auto (overflow property value), 516

auto (page-break-before/after property value), 561

auto value (cursor property), 582

auto-completion

- tag, 17
- text, 590

autogenerated text, 460. *See also* generated content

automatic numbering (for elements), 532–536

automation, form, 289–295, 382–387

auto-numbering system, chapter/section, 533–536

avoid (page-break-before/after property value), 561

axis (table cell attribute), 114

azimuth, 429, 430

B

. *See* bold tag

Babel Fish, 249

Background (system color keyword), 584

background colors

- CSS and, 492–495
- documents, 46–47
- tables, 119–120

background images, 47–48

- CSS and, 496–502
- positioning, 501

background properties (CSS 2.1), 680, 682–684

- background, 192, 502, 684
- background-attachment, 498–500, 683
- background-color, 47, 119–120, 191–193, 493–494, 683
- background-image, 47, 496–498, 682
- background-position, 501, 683
- background-repeat, 498–500, 682
- shortcut, 502

backward difference (nabla), 241, 674

<base> tag, 43, 622–623

<basefont> tag, 62, 303

baseline (font characteristic), 438

baseline (vertical-align property value), 448

baseline alignment, 112

Basic Latin, 249–250

<bdo>, 623

BECSS (Behavioral Extensions to CSS), 573

:before, 417–418, 419, 527, 530–531, 709

Beginning XML, Third Edition (WROX), 329

Behavioral Extensions to CSS (BECSS), 573

below (frame attribute value), 110

Benefits of Membership caption, 114–117

beta (Greek letter), 672, 673

bicolor attributes, 46, 111, 119, 185, 186, 191, 222, 304

<bgsound> tag, 228–229

<big>, 29, 30, 623–624

bigText.style.fontSize, 540

BitStream, 443

_blank value, 92, 153

Blast Radius, 347

block elements, 21–31

block properties (CSS 2.1), 681, 692–695

<blockquote> tags, 24, 25, 57–58, 624. *See also* <q>

blocks and navigational menus (table formatting), 139–141

<body> tags, 19–20, 624–625

bold tag (), 65–66, 621–622. *See also*

bookmarks, anchor tags for, 96

border (frame attribute value), 110

border attribute

image tag, 206

<table> tag, 9, 108, 110, 379, 409, 462

border styles, 484–485

borders

collapsing, 464–466, 522, 523

colors, 485–486

conflicts, 484

element, 483–487

on empty cells, 465–467

image, 206–208

multiple color, 576–577

table, 108–110

borders properties (CSS 2.1), 461–462, 681, 698–699

border, 206, 418, 486–487, 699

border-bottom, 192, 207, 418

border-bottom-color, 192, 207, 419, 485, 698

border-bottom-style, 207, 419, 484, 699

border-bottom-width, 207, 419, 483, 699

border-collapse, 418, 464–466, 487, 701

border-color, 192, 207, 418, 485, 698

border-left, 192, 207, 418

border-left-color, 192, 207, 419, 485, 698

border-left-style, 207, 419, 484, 699

border-left-width, 207, 419, 483, 699

border-right, 192, 206, 418

border-right-color, 192, 207, 419, 485, 698

border-right-style, 207, 419, 484, 699

border-right-width, 207, 419, 483, 699

border-spacing, 418, 461, 462, 463, 487, 701–702

border-style, 207, 419, 484, 699

border-top, 192, 206, 418

border-top-color, 192, 207, 418, 485, 698

border-top-style, 207, 419, 484, 699

border-top-width, 207, 419, 483, 699

border-width, 207, 419, 483, 699

outline, 700–701

outline-color, 700

outline-style, 700

outline-width, 700

shortcuts, 486–487

shorthand expressions and, 418–420

border-side, 207, 420

border-<side>-color property, 574

bottom (CSS property), 508, 697

bottom (vertical-align property value), 448

box (frame attribute value), 110

box formatting model (CSS), 479–490. *See also*

borders; margins; padding

borders and, 483–487

margins and, 487–489

padding and, 482–483

page box formatting model v., 556–558

box model/vertical module (CSS3), 572

boxes, rounded. *See* rounded corners

 (line break tag), 51–52, 625–626

braces { }, 406, 421, 589, 591

brackets [], 14, 169, 409. *See also* angle

brackets

Braille (CSS media type), 554

brass ring. *See* rounded corners

broken bar/broken vertical bar, 243, 255, 676

browser detection (browser sniffing), 288–289

browsers. *See also* Firefox; Internet Explorer

Check Browser Compatibility feature, 315, 316

cross-browser compatibility issues (DHTML), 288–289

CSS3 properties and, 573–574

Gecko, 539, 561
 as graphical FTP client, 326
 hiding scripts from, 269–270
 HTML and, 4
 JavaScript event handlers for, 287
 Konqueror, 368
 nongraphical, 204–205
 Opera, 4, 368, 560, 561, 563
 @page rule and, 557
 Safari, 4, 368, 573
 bulleted lists. *See* unordered lists
 <button>, 626–627
 button field, 171, 172, 179
 ButtonFace, 584
 ButtonHighlight, 584
 buttons
 custom, 171–172
 expanding, 597–600
 radio, 165–166
 Reset, 174
 Submit, 172, 174
 ButtonShadow, 584
 ButtonText, 584

C

C++, 343
 calling external scripts, 264
 cap (intersection), 242, 675
 capital A special characters, 256
 capital C, cedilla, 257
 capital E special characters, 257
 capital Eth, Icelandic, 257
 capital I special characters, 257
 capital N, tilde, 257
 capital O special characters, 257
 capital THORN, Icelandic, 257
 capital U special characters, 257
 capitalization, 457–458. *See also* text-transform property
 caption (user interface font keyword), 587
 captions (<caption>), 114–117, 627
 aligning/positioning, 468–470
 reference for, 627
 caption-side property, 468–469, 702
 CaptionText, 584
 cascade, 433–435
 cascade/inheritance/value assignment module (CSS3), 572
 Cascading Style Sheets. *See* CSS
 cascading styles, 402–404
 CDATA sections, 14, 334, 341, 342
 cedilla, 244, 256, 676
 cell padding, 107–108
 cell spacing, 107–108
 cellpadding attribute, 107, 461
 cells. *See* table cells
 cellspacing attribute, 107, 461, 463
 cent symbol, 235, 255, 668
 center justification, 447
 centimeters (CSS property values measure), 425
 CGI (Common Gateway Interface), 262
 CGI Resource Index, 183
 changing document's nodes, 278–280
 chapter/section auto-numbering system, 533–536
 char
 table cell attribute, 114
 table row tag attribute, 112
 character encodings, 231–232. *See also* internationalization; special characters
 character entities. *See* special characters
 character formatting, 61–70

- charoff, 112, 114
- charset, 98
- chdirscroll(), 393
- check boxes
 - DHTML and, 289–295
 - forms and, 166
- Check Browser Compatibility feature, 315, 316
- checked attribute, 166
- chi (Greek letter), 673
- child selectors, 410–412, 706–707
- childNodes, 276, 280
- children/parents, 411
- circle polygons, 209, 210
- circled plus = direct sum, 242, 675
- circled times = vector product, 242, 675
- `<cite>`, 627
- CJK Compatibility, 253
- CJK Compatibility Forms, 254
- CJK Compatibility Ideographs, 254
- CJK Letters and Months, enclosed, 253
- CJK Radicals Supplement, 253
- CJK Strokes, 253
- CJK Symbols and Punctuation, 253
- CJK Unified Ideographs, 253
- CJK Unified Ideographs Extension A, 253
- cjk-ideographic, 75, 474
- Clark, James, 347
- class attribute (classes), 12–13, 16, 68, 69, 413,
414, 415, 665, 707
 - IDs *v.*, 12
 - style attribute *v.*, 69
- class selectors, 415, 561, 705, 707
- classid attribute, 220, 224
- clear property, 453–454, 512, 698
- clickable regions, 209–211
- client-side image maps, 208
- client-side scripting. *See* scripts
- clip, 693
- cloneNode(), 277
- closing item tags (``), 81
- closing tags
 - column tags and, 126
 - form tags and, 161
 - image tag and, 200
 - script hiding tags and, 269
 - XHTML and, 617
- `<code>`, 628
- code pages, 249
- codebase attribute, 220, 224
- CoffeeCup FTP client, 326
- `<col>`, 125–126, 628–629
- ColdFusion, 313. *See also* NetObjects Fusion
- `<colgroup>`, 125–126, 629
- collapsible menus, 292–295
- collapsing borders, 464–466, 522, 523
- color attributes, 10, 11, 61, 63, 186, 192, 414
- color codes, 665
- color keywords, 486, 492, 584–586
- color property, 57, 64, 191–193, 491–492,
688
- color/gamma/color profiles module (CSS3), 572
- colors, 185–193. *See also* background colors
 - basics, 185–186
 - border, 485–486
 - CSS property values and, 428–429
 - 8-bit palette, 194
 - element, 491–495
 - evolution of (on Web), 187–191
 - extended named color palette, 189–191
 - foreground, 491–492
 - hexadecimal format, 10–11, 186, 187, 188,
189, 190, 191, 486

- link status, 96–98
- RGB decimal format, 10–11, 186, 486
- 16 colors palette, 185, 188
- specifying, 191–193
- 216 colors palette, 187, 188
- user interface, 583–586
- values, 10–11
- Web-safe palette of, 187–188
- cols (rules attribute value), 111
- colspan (table cell attribute), 114, 120, 379
- columns. *See* table columns
- columns module (CSS3), 573
- comments, 13–14, 38
 - nesting, 13
 - table formatting and, 133
 - uses for, 38
 - XML and, 334
- Common Gateway Interface. *See* CGI
- Compact HTML, 350
- compression, of images, 193–194
- CompuServe, 194
- conflict resolution, CSS, 435
- contact element, 344
- contains as member (mathematical symbol entity), 241, 674
- content property, 525–527, 685
 - counters and, 76–77
- coords attribute, 211
- copyright issues, images and, 196–197
- copyright symbol, 234–235, 255, 668
- Core DOM, 281–282
- corners, rounded, 575–577, 605–608
- corporate website, tables and, 131–132
- counter object, 532
- counter-increment, 76, 532–533, 686
- counter-reset, 76, 532, 533, 686
- counters, 76–77, 532–536
- CPAN, 347
- createElement(), 277
- createTextNode(), 277
- crop marks, 560
- cross marks, 560
- cross-browser compatibility issues (DHTML), 288–289
- crosshair value (cursor property), 582
- CSS (Cascading Style Sheets). *See also* borders
 - properties; style definitions
 - Aural, 573
 - background colors and, 492–495
 - background images and, 496–502
 - background properties. *See* background properties
 - basics, 397–404
 - box formatting model, 479–490, 556–558
 - cascade and, 433–435
 - conflict resolution, 435
 - content property, 525–527
 - DHTML with, 539–552
 - editing, Firefox add-ons for, 592–593
 - element positioning, 503–523
 - fonts and, 437–444
 - future of. *See* CSS3
 - generated content. *See* generated content
 - /HTML rift, 62, 142, 305
 - inheritance and, 431–433
 - layouts, 127, 503–523
 - levels of, 400, 571
 - list properties. *See* list properties
 - lists, 472–478
 - manipulation, 545–552
 - media types, 553–570
 - metrics, 423–430

CSS (Cascading Style Sheets). *See also* borders
 properties; style definitions (*continued*)
 positioning properties. *See* positioning
 properties
 precedence rules, 403–404
 properties
 accessed with JavaScript, 539–545
 list, 679–704
 table attributes and, 460–467
 property values, 421–430
 pseudo-elements, 527–531
 rounded corners, 575–577, 605–608
 selectors. *See* selectors
 specificity and, 435–436
 table layout v. CSS layout, 127
 testing, 589–591
 text formatting, 445–470
 text hiding/showing and, 545–547
 tips/tricks, 595–615
 validating, 591–592
 validation tool (W3C), 367
CSS Juice, 607
CSS level 3. *See* CSS3
CSS Validator, 592
CSS1, 400
CSS2, 400
CSS3 (CSS level 3), 400, 571–577
 modules, 572–573
 properties, usage of, 573–574
 pseudo constructs for, 709
 rounded corners and, 575–577
 selectors, 574–575
cubed, 243, 676
cup (union), 242, 675
currency special characters, 235, 668
currency symbol, general, 235, 255

currentStyle, 541, 542, 543
cursive fonts, 438, 439
cursor property, 581–583, 704
custom list numbering, 536
CuteFTP, 326

D

dagger, 244, 677
<dd> (definition description tag), 24, 25, 71,
 629–630
decimal color values, 10–11, 186, 486
decimal value, 75, 473, 684
decimal-leading-zero, 75, 473, 684
default path, 43–44
default value (cursor property), 582
definition description tag. *See* <dd>
definition lists (<dl>), 24, 71, 81–83,
 631–632
degree sign, 243, 256, 676
degrees (aural metric), 429
. *See* delete tag
delete tag (), 31, 68, 630
delta (Greek letter), 672, 673
deprecated attributes (HTML5), 304–305
deprecated elements (HTML5), 303–304
descendant selectors, 407, 410–412, 705, 706,
 707
descendants/ancestors, 410
descension (font characteristic), 438
descriptors, 88
<dfn>, 630–631
DHTML (dynamic HTML), 271–295
 checkboxes and, 289–295
 collapsible menus and, 292–295
 cross-browser compatibility issues, 288–289
 with CSS, 539–552

- DOM and, 272–280
- examples, 289–295
- mechanics of, 272
- need for, 271–272
- rollovers and, 291–292
- diaeresis, 243, 255, 258, 676
- diameter (empty set = null set = diameter), 241, 674
- dimensions, image, 193, 379
- dir attribute, 15, 18, 665
- direct sum = circled plus, 242, 675
- direction property, 692
- disabled attribute, 175, 176, 177
- discretionary hyphen, 243, 676
- display property, 293, 472, 523, 545, 546, 548, 549, 603, 696
- dithering, 194
- `<div>`, `</div>`. *See* division tags
- division sign, 241, 258, 674
- division tags (`<div>`, `</div>`), 27–28, 29, 52–56, 70, 631
 - code example, 53–56
 - paragraph tags and, 53
 - rounded corners and, 605–606
 - section elements (HTML5) v., 297–298
 - `` tag v., 68
- `<dl>`. *See* definition lists
- `<!DOCTYPE>` tag, 17–18
- document object, 283–284
- Document Object Model. *See* DOM
- document title, 41, 42
- Document Type Definitions. *See* DTDs
- documents. *See* HTML documents
- document's nodes, changing, 278–280
- dollar symbol, 235
- DOM (JavaScript Document Object Model), 280–286
 - Core, 281–282
 - Internet Explorer and, 273, 539
 - method bindings, 277–278
 - property bindings, 276–277
- DOM (WC3 Document Object Model), 272–280
 - basis of, 273–275
 - document (example), 273–276
 - history of, 273
 - online information, 272
- `doScroll()`, 393
- dot operator, 242, 675
- double dagger, 245, 677
- double low-9 quotation mark, 244, 677
- double quote mark, 233, 667
- double-side printing, 564
- down arrow, 236, 669
- down arrow with corner leftwards, 236, 669
- down double arrow, 669
- downwards double arrow, 236
- Dreamweaver (Adobe), 314–315, 366, 590–591
- drop shadow effect, 133, 134, 271, 459, 539, 576, 609
- `<dt>`, 71, 632
- DTDs (Document Type Definitions), 17–18
 - code validation and, 365–366
 - frameset, 145, 152, 155
 - XML and, 332, 337–342
 - attributes in, 341–342
 - elements in, 338–340
- dynamic HTML. *See* DHTML
- dynamic outlines. *See* outlines

E

E ~ F, 575

E:checked, 575

ECMAScript, 280–282

EditCSS, 592

editing

 CSS, Firefox add-ons and, 592–593

 graphics, 317–318

 GUI HTML, 125

 HTML-specific editors, 311–312

 image editors, 216

 Jase, 363

 raster-based editors, 318

 smart text editors, 310–311

 text-oriented editors, 309–312

 WYSIWYG HTML editors, 312–317

 XML, 346–347

editorial insertions/deletions. *See* delete tag;

 insert tag

E:empty, 575

E:first-of-type, 575

E[foo =“bar”], 575

E[foo\$=“bar”], 575

E[foo*=“bar”], 575

EGA (Enhanced Graphics Adapter), 185

8-bit palette, 194

E:last-child, 575

E:last-of-type, 575

electronic robots (spiders), 87

element of (mathematical symbol entity), 241,
 674

element positioning (CSS), 503–523

 absolute positioning model, 506–507

 fixed positioning model, 507–508

 relative positioning model, 504–505

 specifying, 508–511

 stacking elements in layers, 517–522

 static positioning model, 503–504

element selectors, 407, 705–706

elements. *See also* pseudo-elements

 automatic numbering, 532–536

 borders, 483–487

 colors, 491–495

 deprecated (HTML5), 303–304

 floating, to left/right, 511–514

 flowing, 608–611

 head, 41–48

 HTML (reference), 617–665

 in HTML documents, 17–40

 HTML5, 300–305

 listing of, 618–663

 margins, 487–489

 maximum/minimum sizes for, 515

 organizational, 32–35

 overflow property and, 501, 515–517, 694

 padding, 482–483

 stacking, 517–522

 visibility property and, 522–523, 696

 XHTML Basic, 352–353

 XML, 332–333, 338–340

. *See* emphasis tags

Em dash, 233–234, 668

em metric, 426, 427

Em space, 233–234, 668

Emacs, 310, 346, 347

e-mail addresses, protection of, 379–381

<embed> tag, 220, 222, 223, 228, 229, 230

embedding fonts, 442–444

embedding multimedia, 219–228

 Windows Media Player, 223–226

 YouTube videos, 226–228

embossed (CSS media type), 554

emphasis tags (), 31, 62–64, 66, 632–633
 italic tags v., 66
 list of, 64
 reference for, 632–633

empty cells, 113, 682, 702
 borders on, 465–467
 nonbreaking space and, 113

empty set, 241, 674

empty-cells property, 113, 465–467, 702

En dash, 233–234, 668

En space, 233–234, 668

encoding e-mail addresses, 380–381

enctype, 164

English pound, 235, 255, 668

Enhanced Graphics Adapter (EGA), 185

E:not(s), 575

E:nth-child(n), 575

E:nth-last-child(n), 575

E:nth-last-of-type(n), 575

E:nth-of-type(n), 575

entities. *See also* special characters
 special characters as, 232–233
 XML, 335, 342

entity equivalent, @ sign and, 380

E:only-child, 575

E:only-of-type, 575

epsilon (Greek letter), 672, 673

e-resize (cursor property), 582

Ericsson Mobility World Developer Program, 357

E:root, 575

essential special characters, 233, 667

eta (Greek letter), 672, 673

E:target, 575

euro, 235, 668

event attributes, 180, 663–664

event handlers, 286–287

events
 forms and, 179–182
 triggering scripts with, 264–269

evolution of colors (on Web), 187–191

expandable title bars, 374–376

expanding buttons, 597–600

expat (XML parser), 347

expires meta tag, 44

extended named color palette, 189–191

Extensible Markup Language. *See* XML

Extensible Stylesheet Language Transformations.
See XSLT

external scripts, calling, 264

F

fantasy fonts, 439

feminine ordinal indicator, 243, 256, 676

field labels, 164

<fieldset> tag, 178, 633

fieldsets, 177–179

file fields, 172–173

File Transfer Protocol. *See* FTP

FileZilla, 325

findNode(), 278

Firebug, 316–317, 593

Firefox
 add-ons
 for CSS editing, 592–593
 WYSIWYG HTML editing and, 316–317
 getStyleVal() and, 544
 multiple-color borders and, 576
 rounded corners and, 576

FireFTP, 326

- Fireworks (Adobe), 196, 200, 215, 318
- :first, 564
- firstChild, 276
- :first-child, 414
- :first-letter, 416–417, 527, 528–530
- :first-line, 415–416, 527, 528
- fixed positioning, 507–508
- Flash (Adobe), 227, 229, 230, 299, 318–319, 352
- float property, 451–454, 511–514, 697
- floating. *See also* flowing
 - elements (to left/right), 511–514
 - images, 451–453, 511–514
 - page layouts, 132–138
- flowing
 - elements, 608–611
 - text, 611–615
- :focus, 413
- font property (CSS 2.1), 64, 690
- font size value metrics, 441
- tags, 30, 61–62
- font-family, 64, 688
- fonts (CSS), 437–444. *See also* specific fonts
 - characteristics, 437–438
 - cursive, 438, 439
 - defined, 437
 - embedding, 442–444
 - fantasy, 439
 - font family types, 438–440
 - glyphs and, 437–438
 - monospace, 58, 61, 66, 455, 649, 662
 - OpenType standard, 443
 - rights for use, 443
 - Sans-serif, 438, 439, 440
 - Serif, 438, 439, 440
 - sizing, 440–441
 - styling, 441–442
 - TrueDoc standard, 443
 - user interface, 587
- fonts module (CSS3), 572
- font-size, 64, 440, 688–689
- font-size-adjust, 440, 441
- font-stretch, 441–442
- font-style, 64, 69, 441–442, 689
- font/text properties (CSS 2.1), 64, 680–681, 686–691
- font-variant, 64, 441–442, 689
- font-weight, 64, 69, 441–442, 689–690
- footers (table footers), 117–119
- for all, 241, 674
- foreground colors, 491–492
- form data verification, event triggers and, 265
- form handlers, 159, 182–183
- form input validation, 384–387
- form objects, 284, 382–384
- <form> tag, 162–164
 - attributes, 163–164
 - reference, 633–634
- formatting
 - box formatting model (CSS) and, 479–490, 556–558
 - characters, 61–70
 - image formats, 192–195
 - inline formatting elements. *See* inline formatting elements
 - paragraphs. *See* paragraph tags
 - with tables, 127–142
 - text (CSS), 445–470
- formmail.cgi, 182, 183
- forms (HTML forms), 34–35, 159–183
 - automation, 289–295, 382–387
 - basic (code example), 159–162

- buttons. *See* buttons
 - check boxes, 166
 - events with, 179–182
 - field labels, 164
 - fieldsets, 177–179
 - file fields, 172–173
 - hidden fields, 170–171
 - images on, 172
 - inserting, 162–164
 - keyboard shortcuts and, 174–175
 - large text input, 169–170
 - legends, 177–179
 - list boxes, 167–169
 - password input boxes, 165
 - preventing changes in, 175–177
 - radio buttons, 165–166
 - Reset button, 174
 - script services, 183
 - Submit button, 172, 174
 - tab order and, 174–175
 - table layout and, 127–131
 - text input boxes, 164–165
 - ForumNokia, 357
 - Fotolia, 197
 - fraction one-half, 244, 256, 676
 - fraction one-quarter, 244, 256, 676
 - fraction three-quarters, 244, 256, 676
 - frame attribute values, 110
 - frame attributes, 110, 145, 148
 - frame borders, 150, 151
 - frame tag (<frame>), 148–150
 - frame tag attributes, 148–149
 - frameborder attribute, 148, 149, 156
 - frame-name value, 92, 153
 - frames, 143–158
 - inline, 155–158
 - overview, 143–144
 - resizing, 150–151
 - targeting, 151–154
 - frameset DTDs, 145, 152, 155
 - frameset tag (<frameset>), 146–147
 - framesets, 144–148
 - creating, 144–146
 - nested, 154–155
 - free-form polygons, 209, 210
 - Freehand (Adobe), 318
 - frequencies (aural metric), 429
 - FTP (File Transfer Protocol), 14, 321–327
 - FTP clients, 322–326
 - FTP commands, 324–325
 - FTP Voyager, 326
 - fully justified text, 447
 - FuturePlayer, 218
- ## G
- gamma (Greek letter), 672, 673
 - gamma/color profiles/color module (CSS3), 572
 - Gecko browsers, 539, 561
 - general currency symbol, 235, 255
 - generated content (CSS 2.1), 680, 685–686
 - content property, 525–527, 685
 - counter-increment, 686
 - counter-reset, 686
 - pseudo-elements and, 525–537
 - quotes, 531–532, 686
 - generated content/markers module (CSS3), 573
 - Georgian comma, 244, 676
 - Georgian value, 75, 251, 252

GET (HTTP GET), 163, 173
getAttribute(), 277
getElementById(), 277, 280, 287–288
getElementByTagName(), 277
getPropertyValue(), 542
getScrollXY(), 393
getStyleVal(), 543, 544, 545
Getty Images, 197
GIFs (Graphics Interchange Format), 37,
 194–195
 animated images and, 215
 compression and, 194–195
 interlaced, 197–198
The GIMP, 318
global attributes (HTML5), 303
glyphs, 437–438
Gnome, 326
grads (aural metric), 429
graphical FTP clients, 323–326
graphics editors, 317–318
Graphics Interchange Format. *See* GIFs
graphics/text, in nonstandard format (table
 formatting), 134–139
GrayText, 584
greater than (>), 7, 233, 255, 667
greater than or equal to, 242, 675
Greek middle dot, 244, 676
Greek special characters, 239–241, 672–674
grid.jpg, 47, 48
grouping columns, 125–126
groups (rules attribute value), 111
GUI HTML editors. *See* HTML editors
guidelines/standards, W3C and, 7
guillemet
 left-pointing, 243, 256, 676
 right-pointing, 244, 256, 676

H

<h1> through <h6>. *See* headings
handheld (CSS media type), 554
Handset Detection, 355
hanging indents, 595–597
hasChildNodes(), 277
head elements, 41–48
 <head> tags, 18–19, 635
headers (HTTP 1.1), 43
headers (table cell attribute), 114
headers (table headers), 117–119
headings (<h1> through <h6>), 22–24,
 634–635
Hebrew value, 75, 250, 474
height attribute
 image tags and, 205, 206
 inline frame tag, 156
height property (CSS), 514–515, 694
help value (cursor property), 582
hexadecimal color format, 10–11, 186, 187,
 188, 189, 190, 191, 486
hidden (overflow property value), 516
hidden fields, 170–171
hiding scripts, from older browsers, 269–270
hiding/showing text, 545–547
hierarchy, HTML document, 410–412
Highlight (system color keyword), 584
highlighting, 4–6
HighlightText, 584
hiragana, 75, 253, 474
hiragana-iroha, 75, 474
history object, 285
HomeSite (Adobe), 311–312
horizontal collapsible menus, 292–295
horizontal rule tag (<hr>), 56–57, 636
horizontal text alignment, 445–448

- `:hover`, 413, 551
- `hover` (CSS anchor pseudo-class), 291
- `hover` (link status mode), 96–97
- `<hr>`. *See* horizontal rule tag
- `href` (hypertext reference attribute), 36, 37
- `hreflang`, 98
- `hsides` (frame attribute value), 110
- HTML (Hypertext Markup Language). *See also* DHTML; HTML5; mobile HTML
 - angle brackets and, 7
 - browsers and, 4
 - /CSS rift, 62, 142, 305
 - defined, 4
 - elements (reference), 617–665
 - future of. *See* HTML5
 - Hypertext and, 4
 - markup languages and, 6–8
 - purpose of, 6
 - send only, 159
 - styles and, 398–400
 - tips/tricks, 371–393
 - validation tool (W3C), 366
 - values/units, 9–16
 - Web documents and, 4
- HTML documents (Web documents)
 - background color, 46–47
 - background images, 47–48
 - default path, 43–44
 - elements, 17–40
 - hierarchy, 410–412
 - HTML and, 4
 - organizational elements of, 32–35
 - printing setup for, 556–564
 - profiles, 45–46
 - publishing, 321–327
 - redirecting, 44
 - refreshing, 43–44
 - screen media (online), 565–568
 - script sections, 45
 - scripts added to, 263–264. *See also* scripts
 - structure/sections, 18–20
 - testing, 368–369
- HTML editors
 - HTML-specific editors, 311–312
 - spanning columns/rows and, 125
 - WYSIWYG, 312–317
- HTML forms. *See* forms
- HTML lists. *See* lists
- HTML tables. *See* tables
- `<html>` tag, 18, 636–637
- HTML Tidy tool, 362–365
- HTML5, 297–305
 - deprecated attributes, 304–305
 - deprecated elements, 303–304
 - elements/attributes, 300–305
 - multimedia and, 298–300
 - publishing/layout features in, 297–298
 - section elements, 297–298
 - W3C and, 297
 - XHTML and, 301
- HTMLTrim, 362
- HTTP (Hypertext Transfer Protocol), 14
- HTTP 1.1 headers, 43
- HTTP GET, 163, 173
- HTTP POST, 163, 173, 290, 367, 385
- HTTP-EQUIV parameter, 42, 44, 232, 263, 644
- Hypertext, 4
- Hypertext Markup Language. *See* HTML
- hypertext reference attribute. *See* href
- Hypertext Transfer Protocol. *See* HTTP

I

- `<i>`. See *italic tag*
- icon (user interface font keyword), 587
- ID selectors, 707–708
- identical to, 242, 675
- identifiers, for tags, 12–13
- IDs (id attribute), 12, 665
 - classes v., 12
 - `<form>` tag, 163, 164
 - `<iframe>` tag, 155–158, 302
- Illustrator (Adobe), 318
- image compression, 193–194
- image editors, 216
- image formats, 192–195
- image maps, 37, 208–212
- image tags (``), 37, 637–638
 - border attribute, 206
 - closing tag and, 200
 - width attribute, 205
- images, 37
 - alignment, 201–204
 - animated, 199–200, 214–216, 265
 - background, 47–48
 - CSS and, 496–502
 - positioning, 501
 - borders around, 206–208
 - compression of, 193–194
 - copyright issues and, 196–197
 - dimensions, 193, 379
 - floating, 451–453, 511–514
 - on forms, 172
 - inserting, 200–201
 - licensed (online), 197
 - as list markers, 476–477
 - preloading, 371–373
 - repeating, 498–501
 - scrolling, 498–501
 - sizing, 205–206
 - transparency and, 194, 198–199
 - zooming, 548–549
- ``. See *image tags*
- important rule, 403, 404, 434, 435
- @import rule, 434, 556
- InactiveBorder, 584
- InactiveCaption, 584
- InactiveCaptionText, 584
- inches (CSS property values measure), 425
- indenting text, 450–451
- index document, 88
- infinity (mathematical symbol entity), 242, 674
- InfoBackground, 584
- InfoText, 584
- inherit (page-break-before/after property value), 561
- inherit keyword, 424
- inheritance, 431–433
- inheritance/value assignment/cascade module (CSS3), 572
- initscroll(), 392–393
- inline formatting elements, 28–31
 - `<big>`, 29, 30, 623–624
 - bold tag, 65–66, 621–622
 - emphasis tags, 31, 62–64, 66, 632–633
 - italic tag, 65–66, 637
 - `<small>`, 29, 30, 652–653
 - `` tag and, 68–69
 - ``, 29, 31, 653–654
 - teletype tag, 29, 30, 66, 554, 662
- inline frame tag attributes, 156–157
- inline frames, 155–158
- `<input>`, 638–639
- `<ins>`. See *insert tag*

- insert tag (<ins>), 31, 68, 639
- insertBefore(), 277
- inserting
 - forms, 162–164
 - images, 200–201
- instructions. *See* markup instructions
- integral, 242, 675
- interlaced GIFs, 197–198
- internationalization, 247–248, 623. *See also*
 - localization
 - attributes, 665
 - character encoding and, 231–232
 - localization v., 247–248
- Internet Explorer
 - access keys and, 95
 - :after and, 709
 - @page rule and, 557
 - :before and, 709
 - blurs and, 539
 - border-spacing property and, 464
 - caption-side property and, 469
 - cross-browser compatibility issues and, 288
 - document title in, 41, 42
 - DOMs and, 273, 539
 - drop shadows and, 539
 - empty-cells property and, 466
 - expanding buttons and, 600
 - file field in, 173
 - frameborder setting and, 149
 - getStyleVal() and, 544
 - horizontal rule in, 57
 - HTML and, 4
 - image alignment and, 201, 202
 - :lang and, 709
 - link titles and, 93, 94
 - OpenType and, 443
 - page-break properties and, 560, 561
 - positioning support and, 508
 - <q> and, 650
 - RSS document and, 345, 346
 - testing HTML with, 368
 - user styles and, 434
 - Windows Media Player and, 223
- Internet service providers (ISPs), 262, 320
- intersection, 242, 675
- inverted exclamation mark, 243, 255, 676
- inverted question mark, 244, 256, 676
- iota (Greek letter), 672, 673
- ISO-8859-1, 42, 254–258, 259, 260
- ISPs (Internet service providers), 262, 320
- iStockphoto, 197
- italic tag (<i>), 65–66, 637. *See also* emphasis tags

J

- James Clark's XML parser, 347
- Japanese yen, 235, 255, 668
- Jase, 363
- Java, 262, 343, 344, 362
- JavaScript, 38, 39. *See also* scripts
 - CSS properties accessed with, 539–545
 - ECMAScript, 280–282
 - event handlers, 286–287
 - form automation with, 382–387
 - form input validation with, 384–387
 - rollovers and, 291–292
 - scripting and, 181, 261
 - scrollbar control and, 387–393
 - targeting frames with, 151
 - text hiding/showing and, 545–547
 - window.open(), 93, 281
- JavaScript Bible (Wiley, 2009), 295, 387

JavaScript DOM, 280–286

 method bindings, 277–278

 property bindings, 276–277

Jigsaw tool, 591–592, 593

Joint Photographic Experts Group. *See* JPEGs

JPEGs (Joint Photographic Experts Group), 37,
 195

 compression and, 195

 progressive, 197–198

JScript, 261

K

K Desktop Environment, 326

kappa (Greek letter), 672, 673

katakana, 75, 253, 474

katakana-iroha, 75, 474

<kbd>, 640

keyboard shortcuts

 for forms, 174–175

 for links, 94–95

keywords

 absolute size, 441

 color, 486, 492, 584–586

 inherit, 424

 property values, 424

 relative size, 441

 user interface font, 587

Konqueror, 368

L

label attribute (<option> tag), 167

<label> tag, 164, 640–641

lambda (Greek letter), 672, 673

:lang, 414, 417, 709

lang attribute, 15, 18, 665

language codes, 15

large text input, 169–170

lastChild, 276

Latin capital letter S with caron, 244, 677

Latin capital letter Y with diaeresis, 244, 677

Latin capital ligature OE, 244, 676

Latin encodings, 249–254

Latin Extended Additional, 260

Latin Extended-A, 259

Latin Extended-B, 260

Latin letters (accented special characters),
 237–239, 670–672

Latin small letter S with caron, 244, 677

Latin small ligature oe, 244, 676

Latin-1 Supplement, 254–255

layout features, HTML5 and, 297–298

layouts (CSS), 503–523

:left, 564

left (CSS property), 508, 697

left (page-break-before/after property value),
 561

left arrow, 236, 669

left ceiling, 242, 675

left double arrow, 669

left floor, 243, 675

left justification, 447

left right arrow, 236, 669

left right double arrow, 236, 669

left/opening double-quote, 236, 669

left/opening single-quote, 236, 669

left-pointing angle bracket, 243, 675

left-pointing double angle quotation mark (left
 pointing guillemet), 243, 256, 676

left-to-right mark, 244, 677

leftwards double arrow, 236

<legend> tag, 178, 179, 641

- legends, 177–179
 - length (vertical-align property value), 448
 - less than (<), 7, 233, 255, 667
 - less than or equal to, 242, 675
 - letter-spacing property, 455–457, 690
 - lhs (frame attribute value), 110
 - (closing item tags), 81
 - (list item tag), 24, 72, 641–642
 - licensed images, 197
 - line box model module (CSS3), 572
 - line break tag (
), 51–52, 625–626
 - line spacing, 442
 - line-height, 442, 695
 - :link, 413, 551
 - Link (link status mode), 96–97
 - <link> tag, 99, 642
 - links, 35–37, 87–100
 - absolute, 90–92
 - colors, 96–98
 - components, 87–89
 - destination details, 98–99
 - keyboard shortcuts, 94–95
 - relative, 90–92
 - status modes, 96–97
 - tab order, 95
 - target attribute, 92–93
 - targeting links to frames, 151–154
 - titles, 93–94, 665
 - URIs and, 14–15
 - to Web pages, 89–90
 - Linux graphical FTP clients, 326
 - list boxes, 167–169
 - list item tag. *See*
 - list markers
 - images as, 476–477
 - positioning of, 475–476
 - list numbering, custom, 536
 - list properties (CSS 2.1), 680, 684–685
 - list-style, 75, 473, 476, 685
 - list-style-image, 79–80, 476–477, 685
 - list-style-position, 75–76, 78, 475–476, 684
 - list-style-type, 73–75, 78, 473–475, 684
 - lists (HTML lists), 24–26, 71–85
 - CSS and, 472–478
 - definition, 24, 71, 81–83, 631–632
 - display format, 72
 - display options, 72
 - nested, 83–84
 - ordered (numbered) lists, 24, 25, 26, 72–77, 471–472, 646
 - overview of, 471–472
 - unordered (bulleted), 24, 77–81, 471–472
 - localization, 247–260
 - internationalization v., 247–248
 - software, 249
 - location object, 284–285
 - logical and, 242, 675
 - logical or, 242, 675
 - longdesc attribute, 148, 149, 156
 - lower-greek, 75, 473, 684
 - lower-latin, 75, 473, 684
 - lower-roman, 75, 473, 684
 - LZW compression, 194
- ## M
- macron, 243, 256, 676
 - manipulating
 - CSS, 545–552
 - form objects, 382–384
 - map, United States, 208, 212
 - <map> tags, 208, 211, 212, 643

- margin properties, 559–560
 - margin, 487–488, 692
 - margin-bottom, 487–488, 692
 - margin-left, 487–488, 595, 597, 692
 - margin-right, 487–488, 692
- marginheight attribute, 148, 149, 156
- margins (element margins), 487–489
- margin-top, 692
- marginwidth attribute, 148, 149, 157
- markers. *See* list markers
- markers/generated content module (CSS3), 573
- marks property, 560
- markup instructions
 - highlighting and, 5–6
 - requirements for, 6–7
- markup languages, 6–8. *See also* XML
 - ASX, 219, 223
 - defined, 7
 - HTML and, 6–8
 - WML, 350, 351, 355
- masculine ordinal indicator, 244, 256, 676
- matching (with selectors), 407–412
 - elements
 - by attributes, 409
 - child/descendant/adjacent sibling, 410–412
 - by class, 408
 - by identifier, 409
 - by type, 407
 - universal selector and, 407–408
- math module (CSS3), 573
- mathematical special characters, 241–243, 674–675
- Max Design site, 471
- max-height, 515, 694
- maximum/minimum sizes, for elements, 515
- max-width, 515, 694
- mean line (font characteristic), 438
- media queries module (CSS3), 573
- media types (CSS), 553–570
- @media rule, 554, 555, 569
- Menu (system color keyword), 584
- menu (user interface font keyword), 587
- menu buttons, with rollovers, 549–552
- menus, tabbed, 603–605
- MenuText, 584
- message-box (user interface font keyword), 587
- META tag, 263
- <meta> tags, 41–43, 643–644
- metrics (property values), 423–430
- micro sign, 243, 676
- middle (vertical-align property value), 448
- middle dot, 244, 256, 676
- millimeters (CSS property values measure), 425
- MIME type, 45, 98, 216, 217, 221, 222, 401
- min-height, 515, 694–695
- minimum/maximum sizes, for elements, 515
- minus sign, 242, 674
- min-width, 515, 694–695
- miscellaneous special characters, 676–677
- mobile HTML, 349–357
 - development tools, 356–357
 - history, 350–352
 - tips, 349
- Mobile Profile standard (XHTML), 351
- “Mobile Web Best Practices” document, 353
- modifier letter circumflex accent, 244, 677
- Modularization, XHTML, 352
- modules, XHTML Basic, 352–353

monospace (typewriter) fonts, 58, 61, 66, 439, 455, 649, 662. *See also* teletype tag

MOTODEV, the Motorola Developer Network, 357

mouse

- event triggers and mouse navigation, 265–269
- scrollbar control (exercise), 387–393

move value (cursor property), 582

-moz prefix, 573

-moz-border-*<side>*-colors property, 574

MSHTML component, 557

mu (Greek letter), 672, 673

multimedia, 213–230. *See also* Adobe Flash

- defined, 213
- document creation, 564–570
- embedding, 219–228
- file creation, 229
- HTML5 and, 298–300

multiple attribute, 169

multiple color borders, 576–577

multiplication sign, 241, 257, 674

N

nabla, 241, 674

name attribute

- anchor tags and, 96, 619
- (un)check all check box and, 383
- form elements and, 284
- <form>* tag, 163, 164
- frame and, 148, 151
- object element and, 221
- radio button and, 165

namespaces, XML, 336

n-ary product = product sign, 241, 674

n-ary summation, 241, 674

navigational menus and blocks (table formatting), 139–141

ne-resize (cursor property), 582

nested CDATA tags, 14

nested comments, 13

nested framesets, 154–155

nested lists, 83–84

nested *<script>* tags, 263

nested tables, 103, 130

NetObjects Fusion, 312–314

new windows, opening, 93

newChild, 277

newspaper columns, 377–379

nextSibling, 276, 280

nodeName, 276

nodeType, 276

nodeValue, 277

nonbreaking space, 113, 233, 255, 373, 667

none value, 111, 293, 475

nongraphical browsers, text for, 204–205

non-parsed data (XML), 334–335

noresize attribute, 150

<noscript>, 644

not a subset of, 242, 675

not an element of (mathematical symbol entity), 241, 674

not equal to, 242, 675

not sign, 243, 256, 676

Notepad (Windows), 309–310, 346

nowrap attribute, 114, 305, 454, 455, 691

n-resize, 582

nu (Greek letter), 672, 673

null set, 241, 674

numbered lists. *See* ordered lists

nw-resize, 582

O

<object> tag

attributes, 221

 <embed> tag and, 222–223

 embedding multimedia and, 219–228

 reference for, 645–646

. *See* ordered lists

omega (Greek letter), 673, 674

omicron (Greek letter), 672, 673

onAbort, 180, 282, 287

onblur, 177, 180, 265, 287, 664

onchange, 39, 177, 180, 265, 287, 291, 385,
 664

onclick, 39, 172, 174, 177, 179, 180–182,
 265, 266, 267, 268, 282, 285, 286,
 291, 383, 540, 664

ondblclick, 180, 265, 664

one-half (fraction), 244, 256, 676

one-quarter (fraction), 244, 256, 676

onError, 180, 282, 287

onfocus, 180, 265, 287, 664

onkeydown, 180, 265, 664

onkeypress, 180, 265, 664

onkeyup, 180, 265, 284, 664

online (screen media) document, 565–568

onLoad, 180, 282, 287, 372, 391, 392

onload, 19, 39, 265, 282, 625, 664

onmousedown, 180, 265, 664

onmousemove, 180, 265, 664

onmouseout, 180, 265, 266, 267, 287, 295,
 391, 393, 549, 664

onmouseover, 180, 265, 266, 267, 282, 287,
 291, 295, 391, 549, 664

onmouseup, 180, 265, 664

onreset, 180, 265, 282, 664

onResize, 180

onselect, 180, 265, 287, 664

onsubmit, 180, 265, 282, 287, 385, 664

onunload, 19, 39, 180, 265, 287, 625, 664

open(), 93, 281–283

Open Mobile Alliance, 351

opening new windows (debate), 93

OpenType standard, 443

Opera, 4, 368, 560, 561, 563

<optgroup>, 167, 168, 646–647

option groups, 168

<option> tag, 167–168, 647

 attributes, 167

 reference for, 647

ordered (numbered) lists (), 24, 25, 26,
 72–77, 471–472, 646

 default, 72–73

 start value of, 76–77

organizational elements (of HTML documents),
 32–35. *See also* forms; tables

orphans, 562–564, 703

orthogonal to, 242, 675

outline properties

 outline, 192, 489–490, 700–701

 outline-color, 192, 489–490, 700

 outline-style, 489–490, 700

 outline-width, 489–490, 700

outlines (dynamic outlines), 489–490

overflow property, 501, 515–517, 694

overline, 243, 676

ownerDocument, 277

<oxygen>, 347

P

<p>, </p>. *See* paragraph tags

padding

 cell, 107–108