| **Selector** | **Description** |
| --- | --- |
| \* | Selects all elements. |
| E | Selects an element of type E |
| E:not(s1, s2) | Selects an E element that does not match either s1 or s2. |
| E:matches(s1, s2) | Selects an E element that matches s1 and/or s2. |
| E:has(rs1, rs2) | Selects an E element, if either of the relative selectors rs1or rs2, when evaluated with E as the :scope elements, match an element. |
| E.classname | Selects an E element belonging to the class named classname. |
| E#myid | Selects an E element with an ID of myid. |
| E[foo] | Selects an E element with a foo attribute. |
| E[foo="bar"] | Selects an E element whose foo attribute value is exactly equal to bar. |
| E[foo="bar" i] | Selects an E element whose foo attribute value is exactly equal to bar, regardless of its case. Basically, using i allows you to specify "case-sensitive" when specifying the value.  So for example, Text, text, and TEXT will all be selected if i is specified. |
| E[foo~="bar"] | Selects an E element whose foo attribute value is a list of whitespace-separated values, one of which is exactly equal to bar. |
| E[foo^="bar"] | Selects an E element whose foo attribute value begins exactly with the string bar. |
| E[foo$="bar"] | Selects an E element whose foo attribute value ends exactly with the string bar. |
| E[foo\*="bar"] | Selects an E element whose foo attribute value contains the substring bar. |
| E[foo|="fruit"] | Selects an E element whose foo attribute value is a hyphen-separated list of values beginning with en. |
| E:dir(ltr) | Selects an element of type E in with left-to-right directionality (the document language specifies how directionality is determined). |
| E:lang(zh, "\*-hant") | Selects an element of type E tagged as being either in Chinese (any dialect or writing system) or othewise written with traditional Chinese characters. Simply replace zh with the applicable language code, and \*-hant with the applicable character code. |
| E:any-link | Selects an E element being the source anchor of a hyperlink.  This is the equivalent of using both the :link and :visited pseudo-classes. |
| E:link | Selects an E element being the source anchor of a hyperlink of which the target is not yet visited. |
| E:visited | Selects an E element being the source anchor of a hyperlink of which the target is already visited. |
| E:target | Selects an E element being the target of the referring URL. |
| E:scope | Selects an E element being a designated reference element. |
| E:current | Selects an E element that is currently presented in a time-dimensional canvas.  For example, if the contents are being read out while being displayed on a screen, this selector could be used to style the current item as it is being read out. |
| E:current(s) | Selects an E element that is the deepest :currentelement that matches selector s. |
| E:past | Selects an E element that is in the past in a time-dimensional canvas.  For example, if the contents are being read out while being displayed on a screen, this selector could be used to style the items that have already been read out. |
| E:future | Select an E element that is in the future in a time-dimensional canvas.  For example, if the contents are being read out while being displayed on a screen, this selector could be used to style the items that have yet to be read out. |
| E:active | Selects an E element that is in an activated state. |
| E:hover | Selects an E element that is under the cursor, or that has a descendant under the cursor. |
| E:focus | Selects an E element that has user input focus. |
| E:drop | Selects an E element that can possibly receive a drop.  For example, in HTML you can specify a drop target with the dropzone attribute. This selector can be used style those elements while the user is dragging another element to be dropped. |
| E:drop(active) | Selects an E element that is the current drop target for the item being dragged.  In other words, if the user were to release the drag, the element would be dropped on to this drop target. |
| E:drop(valid) | Selects an E element that could receive the item currently being dragged. It only matches if the drop target is valid for the object currently being dragged.  For example, the HTML dropzone attribute can be used to specify which types of files can be dropped onto an element. |
| E:drop(invalid) | Selects an E element that cannot receive the item currently being dragged, but could receive some other item. |
| E:enabled | Selects a user interface element E that is enabled. |
| E:disabled | Selects a user interface element E that is disabled. |
| E:read-write | Selects a user interface element E that is user alterable.  For example, an HTML [input](https://www.quackit.com/html/tags/html_input_tag.cfm) element that is not disabled and not readonly is :read-write. Also, any HTML element with contenteditable attribute set to the true state is also :read-write. |
| E:read-only | Selects a user interface element E that is not user alterable.  For example, an HTML [input](https://www.quackit.com/html/tags/html_input_tag.cfm) element that is disabled or readonly is :read-only. |
| E:placeholder-shown | Selects an input control currently showing placeholder text.  This allows you to style the placeholder text. For example, change its color, etc. |
| E:default | Selects a user interface element E that is the default item in a group of related choices.  For example, it could select the initially selected [option](https://www.quackit.com/html/tags/html_option_tag.cfm) element in a [select](https://www.quackit.com/html/tags/html_select_tag.cfm) element, or the default button in the form, etc. |
| E:checked | Selects a user interface element E that is checked/selected (for example, a radio-button or checkbox).  Can be used to style the element's label. |
| E:indeterminate | Selects a user interface element E that is in an indeterminate state (neither checked nor unchecked).  Can be used to style the element's label. |
| E:valid | Selects a user-input element E that meets its data validity semantics. |
| E:invalid | Selects a user-input element E that doesn't meet its data validity semantics. |
| E:in-range | Selects a user-input element E whose value is in-range. |
| E:out-of-range | Selects a user-input element E whose value is out-of-range. |
| E:required | Selects a user-input element E that requires input.  For example, a form element with the requiredattribute. |
| E:optional | Selects a user-input element E that does not require input.  For example, a form element without the requiredattribute. |
| E:user-error | Selects a user-altered user-input element E with incorrect input (invalid, out-of-range, omitted-but-required). |
| E:root | Selects an E element, root of the document.  In most cases, when using HTML this will match the [html](https://www.quackit.com/html/tags/html_html_tag.cfm) element, however, this may not always be the case if using another language such as SVG or [XML](https://www.quackit.com/xml/tutorial/). |
| E:empty | Selects an E element that has no children (not even text nodes). |
| E:blank | Selects an E element that has no content except maybe white space.  Similar to :empty, however, :blank will also select elements that contain white space (which :empty will not). |
| E:nth-child(n [ofS]?) | Selects an E element, the n-th child of its parent matching S. |
| E:nth-last-child(n[of S]?) | Selects an E element, the n-th child of its parent matching S, counting from the last one. |
| E:first-child | Selects an E element, first child of its parent. |
| E:last-child | Selects an E element, last child of its parent. |
| E:only-child | Selects an E element, only child of its parent. |
| E:nth-of-type(n) | Selects an E element, the n-th sibling of its type. |
| E:nth-last-of-type(n) | Selects an E element, the n-th sibling of its type, counting from the last one. |
| E:first-of-type | Selects an E element, first sibling of its type. |
| E:last-of-type | Selects an E element, last sibling of its type. |
| E:only-of-type | Selects an E element, only sibling of its type. |
| E F | Selects an F element descendant of an E element. |
| E >> F | Selects an F element descendant of an E element.  Note that this selector does the same thing as the previous selector (E F). This selector was introduced in CSS Selectors Level 4 to give the descendant combinator a visible, non-whitespace form and to bridge the gap between the child combinator (>) and the shadow-piercing descendant combinator (>>>) |
| E > F | Selects an F element child of an E element. |
| E + F | Selects an F element immediately preceded by an E element. |
| E ~ F | Selects an F element preceded by an E element. |
| F || E | Selects an E element that represents a cell in a grid/table belonging to a column represented by an element F. |
| E:nth-column(n) | Selects an E element that represents a cell belonging to the nth column in a grid/table. |
| E:nth-last-column(n) | Selects an E element that represents a cell belonging to the nth column in a grid/table, counting from the last one. |
| :playing | Selects an element representing an audio, video, or similar resource that is capable of being "played" or "paused", when that element is "playing". |
| :paused | Selects an element representing an audio, video, or similar resource that is capable of being "played" or "paused", when that element is "paused". This includes both an explicit "paused" state, and other non-playing states like "loaded, hasn't been activated yet", etc. |

**Background Properties**

|  |  |  |  |
| --- | --- | --- | --- |
| **Property** | **Description** | **Possible Values** | **Examples** |
| background-attachment | Declares the attachment of a background image (to scroll with the page content or be in a fixed position). | fixed scroll | div { background-attachment:fixed; }  div { background-attachment:scroll; } |
| background-color | Declares the background color. | Valid color names, RGB values, hexidecimal notation. | div { background-color:green; }  div { color:#00FF00; } |
| background-image | Declares the background image of an element. | URL values. | div { background-image:url(images/img.jpg); }  body { background-image:url(img.jpg); } |
| background-position | Declares the position of a background image. | Lengths or percentages for the x and y positions, or one of the predefined values:  top left top center top right center left center center center right bottom left bottom center bottom right | div { background-position:10px 50px; }  div { background-position:bottom right; } |
| background-repeat | Declares how and/or if a background image repeats. | repeat repeat-x repeat-y no-repeat | div { background-repeat:repeat-x; }  div { background-repeat:no-repeat; } |
| background | Used as a shorthand property to set all the background properties at once. | Separate values by a space in the following order (those that are not defined will use inherited or default initial values):  background-color background-image background-repeat  background-attachment  background-position | div { background:green url(image.jpg) no-repeat fixed center center; }  div { background:url(image.jpg) fixed; } |

**Border Properties**

|  |  |  |  |
| --- | --- | --- | --- |
| **Property** | **Description** | **Possible Values** | **Examples** |
| border-top-color | Declares the color of the top border. | Valid color names, RGB values, hexidecimal notation, or the predefined value transparent. | div { border-top-color:green; }  div { border-top-color:#00FF00; } |
| border-top-style | Declares the style of the top border. | none hidden dotted dashed solid double groove ridge inset outset | div { border-top-style:solid; }  div { border-top-style:inset; } |
| border-top-width | Declares the width of the top border. | Lengths or the following predefined values:  thin medium thick | div { border-top-width:2px; }  div { border-top-width:thin; } |
| border-top | Used as a shorthand property to set all the border-top properties at once. | Separate values by a space in the following order (those that are not defined will use inherited or default initial values):  border-top-width border-top-style border-top-color | div { border-top:2px solid green; }  div { border-top:thick double #00FF00; } |
| border-right-color | Declares the color of the right border. | Valid color names, RGB values, hexidecimal notation, or the predefined value transparent. | div { border-right-color:green; }  div { border-right-color:#00FF00; } |
| border-right-style | Declares the style of the right border. | none hidden dotted dashed solid double groove ridge inset outset | div { border-right-style:solid; }  div { border-right-style:inset; } |
| border-right-width | Declares the width of the right border. | Lengths or the following predefined values:  thin medium thick | div { border-right-width:2px; }  div { border-right-width:thin; } |
| border-right | Used as a shorthand property to set all the border-right properties at once. | Separate values by a space in the following order (those that are not defined will use inherited or default initial values):  border-right-width border-right-style border-right-color | div { border-right:2px solid green; }  div { border-right:thick double #00FF00; } |
| border-bottom-color | Declares the color of the bottom border. | Valid color names, RGB values, hexidecimal notation, or the predefined value transparent. | div { border-bottom-color:green; }  div { border-bottom-color:#00FF00; } |
| border-bottom-style | Declares the style of the bottom border. | none hidden dotted dashed solid double groove ridge inset outset | div { border-bottom-style:solid; }  div { border-bottom-style:inset; } |
| border-bottom-width | Declares the width of the bottom border. | Lengths or the following predefined values:  thin medium thick | div { border-bottom-width:2px; }  div { border-bottom-width:thin; } |
| border-bottom | Used as a shorthand property to set all the border-bottom properties at once. | Separate values by a space in the following order (those that are not defined will use inherited or default initial values):  border-bottom-width border-bottom-style border-bottom-color | div { border-bottom:2px solid green; }  div { border-bottom:thick double #00FF00; } |
| border-left-color | Declares the color of the left border. | Valid color names, RGB values, hexidecimal notation, or the predefined value transparent. | div { border-left-color:green; }  div { border-left-color:#00FF00; } |
| border-left-style | Declares the style of the left border. | none hidden dotted dashed solid double groove ridge inset outset | div { border-left-style:solid; }  div { border-left-style:inset; } |
| border-left-width | Declares the width of the left border. | Lengths or the following predefined values:  thin medium thick | div { border-left-width:2px; }  div { border-left-width:thin; } |
| border-left | Used as a shorthand property to set all the border-left properties at once. | Separate values by a space in the following order (those that are not defined will use inherited or default initial values):  border-left-width border-left-style border-left-color | div { border-left:2px solid green; }  div { border-left:thick double #00FF00; } |
| border-color | Declares the border color of all four borders at once. | Valid color names, RGB values, hexidecimal notation, or the predefined value transparent.  Separate the color for each border by a space, declaring the colors for the borders in the following order:  border-top-color border-right-color border-bottom-color border-left-color  Undeclared values work as further shorthand notation. If only one color value is declared, all four borders will use that color. If two colors are declared, the top and bottom borders will use the first color while the right and left borders will use the second color. If three colors are declared, the top border will use the first color, the right and left borders will use the second color, and the bottom border will use the third color. | div { border-color:green red blue olive; }  div { border-color:green; }  div { border-color:green red; }  div { border-color:green red blue; } |
| border-style | Declares the border style of all four borders at once. | none hidden dotted dashed solid double groove ridge inset outset  Undeclared values work as further shorthand notation. If only one style value is declared, all four borders will use that style. If two styles are declared, the top and bottom borders will use the first style while the right and left borders will use the second style. If three styles are declared, the top border will use the first style, the right and left borders will use the second style, and the bottom border will use the third style. | div { border-style:solid dotted dashed double; }  div { border-style:solid; }  div { border-style:solid dotted; }  div { border-style:solid dotted dashed; } |
| border-width | Declares the width of all four borders at once. | Lengths or the following predefined values:  thin medium thick  Undeclared values work as further shorthand notation. If only one width value is declared, all four borders will use that width. If two widths are declared, the top and bottom borders will use the first width while the right and left borders will use the second width. If three widths are declared, the top border will use the first width, the right and left borders will use the second width, and the bottom border will use the third width. | div { border-width:1px 3px 5px 2px; }  div { border-width:thin; }  div { border-width:2px 4px; }  div { border-width:2px 4px 5px; } |
| border | Used as a shorthand to declare the border properties when all four borders will have the same appearance. | Separate values by a space in the following order (those that are not defined will use inherited or default initial values):  border-width border-style border-color | div { border:1px double green; }  div { border:thin solid #00FF00; } |

**Classification and Positioning Properties**

|  |  |  |  |
| --- | --- | --- | --- |
| **Property** | **Description** | **Possible Values** | **Examples** |
| clear | Declares the side(s) of an element where no previous floating elements are allowed to be adjacent. | left right both none | div { clear:right; }  div { clear:both; } |
| cursor | Declares the type of cursor to be displayed. | URL values, and the following prefefined values:  auto crosshair default pointer move e-resize ne-resize nw-resize n-resize se-resize sw-resize s-resize w-resize text wait help | div { cursor:crosshair; }  div { cusrsor:url(image.csr); }  div { cusrsor:url(image.csr), pointer; } |
| display | Declares if/how the element displays. | none inline block list-item run-in compact marker table inline-table table-row-group table-header-group table-footer-group table-row table-column-group table-column table-cell table-caption | div { display:none; }  div { display:inline; }  div { display:marker; } |
| float | Declares whether a box should float to the left or right of other content, or whether it should not be floated at all. | left right none | div { float:left; }  div { float:right; } |
| visibility | Declares the visibility of boxes generated by an element. | visible hidden collapse | div { visibility:visible; }  div { visibility:hidden; } |
| Top | Declares the distance that the top content edge of the element is offset below the top edge of its containing block. The position property of the element must also be set to a value other than static. | Lengths, percentages, and the predefined value auto. | div { top:15px; }  div { top:2%; } |
| Right | Declares the distance that the right content edge of the element is offset to the left of the right edge of its containing block. The position property of the element must also be set to a value other than static. | Lengths, percentages, and the predefined value auto. | div { right:15px; }  div { right:2%; } |
| bottom | Declares the distance that the bottom content edge of the element is offset above the bottom edge of its containing block. The position property of the element must also be set to a value other than static. | Lengths, percentages, and the predefined value auto. | div { bottom:15px; }  div { bottom:2%; } |
| Left | Declares the distance that the left content edge of the element is offset to the right of the left edge of its containing block. The position property of the element must also be set to a value other than static. | Lengths, percentages, and the predefined value auto. | div { left:15px; }  div { left:2%; } |
| position | Declares the type of positioning of an element. | static relative absolute fixed | div { position:absolute; }  div { position:relative; } |
| Clip | Declares the shape of a clipped region when the value of the overflow property is set to a value other than visible. | Shapes, or the predefined value auto.  In CSS 2, the only valid shape is a rectangle, using the following format to specify the offset lengths from each side of the box:  rect(top, right, bottom, left) | div { clip:auto; }  div { clip:rect(2px, 4px, 7px, 5px); } |
| overflow | Declares how content that overflows the element's box is handled. | visible hidden scroll auto | div { overflow:hidden; }  div { overflow:scroll; } |
| vertical-align | Declares the vertical alignment of an inline-level element or a table cell. | Lengths, percentages, and the following predefined values:  baseline sub super top text-top middle bottom text-bottom | span { vertical-align:middle; }  td { vertical-align:top; } |
| z-index | Declares the stack order of the element. | Integer values and the predefined value auto. | div { z-index:2; }  div { z-index:auto; } |

**Dimension Properties**

|  |  |  |  |
| --- | --- | --- | --- |
| **Property** | **Description** | **Possible Values** | **Examples** |
| height | Declares the height of the element. | Lengths, percentages, and the predefined value auto. | div { height:200px; }  div { height:50%; } |
| max-height | Declares the maximum height of the element. | Lengths, percentages, and the predefined value auto. | div { max-height:200px; }  div { max-height:50%; } |
| min-height | Declares the minimum height of the element. | Lengths, percentages, and the predefined value auto. | div { min-height:200px; }  div { min-height:50%; } |
| width | Declares the width of the element. | Lengths, percentages, and the predefined value auto. | div { width:500px; }  div { width:75%; } |
| max-width | Declares the maximum width of the element. | Lengths, percentages, and the predefined value auto. | div { max-width:500px; }  div { max-width:75%; } |
| min-width | Declares the minimum width of the element. | Lengths, percentages, and the predefined value auto. | div { min-width:500px; }  div { min-width:75%; } |

**Font Properties**

|  |  |  |  |
| --- | --- | --- | --- |
| **Property** | **Description** | **Possible Values** | **Examples** |
| font-family | Declares the name of the font to be used. Previously set in HTML via the *face* attribute in a <font> tag. | Valid font family names or generic family names, i.e. *Arial*, *Verdana*, *sans-serif*, *"Times New Roman"*, *Times*, *serif*, etc.  Font family names can be separated by a comma in the same declaration to allow additional and/or generic family names to be used if the preferred font is unable to be displayed. | div { font-family:Arial; }  div { font-family:Arial, Helvetica, sans-serif; } |
| font-size | Declares the size of the font. Previously set in HTML via the *size* attribute in a <font> tag. | Lengths (number and unit type— i.e. *1em*, *12pt*, *10px*, *80%*) or one of the following predefined values:  xx-small x-small small medium large x-large xx-large smaller larger | div { font-size:70%; }  div { font-size:0.85em; }  div { font-size: medium; } |
| font-size-adjust | **Limited browser support: Was part of CSS 2, but not in CSS 2.1. This property may return in CSS 3.**  Declares the *aspect value* (font size divided by x-height). | Numeric value | div { font-size-adjust:0.54; }  div { font-size-adjust:0.46; } |
| font-stretch | **Limited browser support: Was part of CSS 2, but not in CSS 2.1. This property may return in CSS 3.**  Declares the stretch of the font face. | normal wider narrower ultra-condensed extra-condensed condensed semi-condensed semi-expanded expanded extra-expanded ultra-expanded | div { font-stretch:narrower; }  div { font-stretch:ultra-expanded; } |
| font-style | Declares the font style. | normal italic oblique | div { font-style:italic; }  div { font-style:oblique; } |
| font-variant | Declares the font variant. | normal small-caps | div { font-variant:normal; }  div { font-variant:small-caps; } |
| font-weight | Declares the font weight (lightness or boldness) | normal bold bolder lighter 100 200 300 400 500 600 700 800 900 | div { font-weight:bolder; }  div { font-weight:200; } |
| font | Used as a shorthand property to declare all of the font properties at once (except font-size-adjust and font-stretch). | Separate values by a space in the following order (those that are not defined will use inherited or default initial values):  font-style font-variant font-weight font-size line-height font-family | div { font:italic small-caps bold 1em 1.2em Arial }  div { font:bold 0.8em Verdana } |

**Generated Content Properties**

|  |  |  |  |
| --- | --- | --- | --- |
| **Property** | **Description** | **Possible Values** | **Examples** |
| content | Generates content in the document in conjunction with the :before and :after pseudo-elements. | String values, URL values, and predefined value formats:  counter(name) counter(name, list-style-type) counters(name, string) counters(name, string, list-style-type) attr(X) open-quote close-quote no-open-quote no-close-quote | div:before { content:"some text"; }  div:after { content:url(page2.html); } |
| counter-increment | Declares the counter increment for each instance of a selector. | Integers and the predefined value none. |  |
| counter-reset | Declares the value the counter is set to on each instance of a selector. | Integers and the predefined value none. |  |
| quotes | Declares the type of quotation marks to use for quotations and embedded quotations. | String values and the predefined value none. |  |

**List Properties**

|  |  |  |  |
| --- | --- | --- | --- |
| **Property** | **Description** | **Possible Values** | **Examples** |
| list-style-type | Declares the type of list marker used. | disc circle square decimal decimal-leading-zero lower-roman upper-roman lower-alpha upper-alpha lower-greek lower-latin upper-latin hebrew armenian georgian cjk-ideographic hiragana katakana hiragana-iroha katakana-iroha | ol { list-style-type:upper-roman; }  ul { list-style-type:square; } |
| list-style-position | Declares the position of the list marker. | inside outside | ol { list-style-position:inside; }  ul { list-style-position:outside; } |
| list-style-image | Declares an image to be used as the list marker. | URL values. | ul { list-style-image:url(image.jpg); } |
| list-style | Shorthand property to declare three list properties at once. | Separate values by a space in the following order (those that are not defined will use inherited or default initial values):  list-style-type list-style-position list-style-image | ul { list-style:disc inside url(image.gif); }  ol { list-style:upper-roman outside; } |
| marker-offset | Declares the marker offset for elements with a value of marker set for the display property. | Lengths and the predefined value auto. | li:before { display:marker; marker-offset:5px; } |

**Margin Properties**

|  |  |  |  |
| --- | --- | --- | --- |
| **Property** | **Description** | **Possible Values** | **Examples** |
| margin-top | Declares the top margin for the element. | Lengths, percentages, and the predefined value auto. | div { margin-top:5px; }  div { margin-top:15%; } |
| margin-right | Declares the right margin for the element. | Lengths, percentages, and the predefined value auto. | div { margin-right:5px; }  div { margin-right:15%; } |
| margin-bottom | Declares the bottom margin for the element. | Lengths, percentages, and the predefined value auto. | div { margin-bottom:5px; }  div { margin-bottom:15%; } |
| margin-left | Declares the left margin for the element. | Lengths, percentages, and the predefined value auto. | div { margin-left:5px; }  div { margin-left:15%; } |
| margin | Shorthand property used to declare all the margin properties at once. | Separate values by a space in the following order (those that are not defined will use inherited or default initial values):  margin-top margin-right margin-bottom margin-left  Undeclared values work as further shorthand notation. If only one length value is declared, all four margins will use that length. If two lengths are declared, the top and bottom margins will use the first length while the right and left margins will use the second length. If three lengths are declared, the top margin will use the first length, the right and left margins will use the second length, and the bottom margin will use the third length. | div { margin:5px 12px 4px 7px; }  div { margin:5px; }  div { margin:5px 10px; }  div { margin:5px 7px 4px; } |

**Outline Properties**

|  |  |  |  |
| --- | --- | --- | --- |
| **Property** | **Description** | **Possible Values** | **Examples** |
| outline-color | Declares the outline color. | Valid color names, RGB values, hexidecimal notation. | div { outline-color:green; }  div { outline-color:#00FF00; } |
| outline-style | Declares the style of the outline. | none dotted dashed solid double groove ridge inset outset | div { outline-style:solid; }  div { outline-style:inset; } |
| outline-width | Declares the width of the outline. | Lengths or the following predefined values:  thin medium thick | div { outline-width:2px; }  div { outline-width:thin; } |
| outline | Used as a shorthand property to set all the background properties at once. | Separate values by a space in the following order (those that are not defined will use inherited or default initial values):  outline-color outline-style outline-width | div { outline:green solid 2px; }  div { outline:#00FF00 double thick; } |

**Padding Properties**

|  |  |  |  |
| --- | --- | --- | --- |
| **Property** | **Description** | **Possible Values** | **Examples** |
| padding-top | Declares the top padding for the element. | Lengths, percentages, and the predefined value auto. | div { padding-top:5px; }  div { padding-top:15%; } |
| padding-right | Declares the right padding for the element. | Lengths, percentages, and the predefined value auto. | div { padding-right:5px; }  div { padding-right:15%; } |
| padding-bottom | Declares the bottom padding for the element. | Lengths, percentages, and the predefined value auto. | div { padding-bottom:5px; }  div { padding-bottom:15%; } |
| padding-left | Declares the left padding for the element. | Lengths, percentages, and the predefined value auto. | div { padding-left:5px; }  div { padding-left:15%; } |
| Padding | Shorthand property used to declare all the margin properties at once. | Separate values by a space in the following order (those that are not defined will use inherited or default initial values):  padding-top padding-right padding-bottom padding-left  Undeclared values work as further shorthand notation. If only one length value is declared, all four sides will use that length. If two lengths are declared, the top and bottom sides will use the first length while the right and left sides will use the second length. If three lengths are declared, the top side will use the first length, the right and left sides will use the second length, and the bottom side will use the third length. | div { padding:5px 12px 4px 7px; }  div { padding:5px; }  div { padding:5px 10px; }  div { padding:5px 7px 4px; } |

**Page Properties**

|  |  |  |  |
| --- | --- | --- | --- |
| **Property** | **Description** | **Possible Values** | **Examples** |
| marks | Declares the type of marks to display outside the page box. | crop cross | @page { marks:crop; } |
| orphans | Declares the minimum number of lines of a paragraph that must be left at the bottom of a page. | Integers | @page { orphans:2; } |
| page | Declares the type of page where an element should be displayed. | Indentifiers |  |
| page-break-after | Declares a page break. | auto always avoid left right |  |
| page-break-before | Declares a page break. | auto always avoid left right |  |
| page-break-inside | Declares a page break. | auto avoid |  |
| size | Declares the size and orientation of a page box. | Lengths, and the following predefined values:  auto landscape potrait |  |
| widows | Declares the minimum number of lines of a paragraph that must be left at the top of a page. | Integers | @page { widows:2; } |

**Table Properties**

|  |  |  |  |
| --- | --- | --- | --- |
| **Property** | **Description** | **Possible Values** | **Examples** |
| border-collapse | Declares the way borders are displayed. | collapse separate | table { border-collapse:collapse; }  table { border-collapse:separate; } |
| border-spacing | Declares the distance separating borders (if border-collapse is separate). | Lengths for the horizontal and vertical spacing, separated by a space.  If one length is value is declared, that length is used for both the horizontal and vertical spacing. If two lengths are declared, the first one is used for horizontal spacing and the second one is used for vertical spacing. | table { border-spacing:5px; }  table { border-spacing:5px 10px; } |
| caption-side | Declares where the table caption is displayed in relation to the table. | top bottom left right | caption { caption-side:top; }  caption { caption-side:right; } |
| empty-cells | Declares the way empty cells are displayed (if border-collapse is separate). | show hide | table { empty-cells:show; }  table { empty-cells:hide; } |
| table-layout | Declares the type of table layout. | auto fixed | table { table-layout:auto; }  table { table-layout:fixed; } |

**Text Properties**

|  |  |  |  |
| --- | --- | --- | --- |
| **Property** | **Description** | **Possible Values** | **Examples** |
| color | Declares the color of the text. | Valid color names, RGB values, hexidecimal notation.  The predefined color names are:  aqua black blue  fuchsia gray green lime maroon navy olive purple red silver teal white yellow | div { color:green; }  div { color:rgb(0,255,0); }  div { color:#00FF00; } |
| direction | Declares the reading direction of the text. | ltr rtl  ltr = left-to-right rtl = right-to-left | div { direction:ltr; }  div { direction:rtl; } |
| line-height | Declares the distance between lines. | Numbers, percentages, lengths, and the predefined value of normal. | div { line-height:normal; }  div { line-height:2em; }  div { line-height:125%; } |
| letter-spacing | Declares the amount of space between text characters. | A length (in addition to the default space) or the predefined value of normal. | div { letter-spacing:normal; }  div { letter-spacing:5px; }  div { letter-spacing:-1px; } |
| text-align | Declares the horizontal alignment of inline content. | left right center justify  If used on a set of table cells, this property can be given a string value to which the text of each row of the column will be aligned. | div { text-align:center; }  div { text-align:right; }  td { text-align:"."; } |
| text-decoration | Declares the text decoration. | none underline overline line-through blink | div { text-decoration:none; }  div { text-decoration:underline; } |
| text-indent | Declares the indentation of the first line of text. | Lengths and percentages. | div { text-indent:12px; }  div { text-indent:2%; } |
| text-shadow | Declares shadow effects on the text. | A list containg a color followed by numeric values (separated by spaces) that specify:   1. The color for the shadow effect 2. Horizontal distance to the right of the text 3. Vertical distance below the text 4. Blur radius | div { text-shadow:green 2px 2px 7px; }  div { text-shadow:olive -3px -4px 5px; } |
| text-transform | Declares the capitalization effects on the letters in the text. | none capitalize uppercase lowercase | div { text-transform:uppercase; }  div { text-transform:lowercase; } |
| unicode-bidi | Declares values relating to bidirectional text. May be used in conjunction with the the *direction* property. | normal embed bidi-override | div { unicode-bidi:embed; }  div { unicode-bidi:bidi-override; } |
| white-space | Declares how white space is handled in an element. | normal pre nowrap | div { white-space:pre; }  div { white-space:nowrap; } |
| word-spacing | Declares the space between words in the text. | A length (in addition to the default space) or the predefined value of normal. | div { word-spacing:normal; }  div { word-spacing:1.5em; } |

**Other Properties**

|  |  |  |  |
| --- | --- | --- | --- |
| **Property** | **Description** | **Possible Values** | **Examples** |
| azimuth | Declares the angle that sound travels to the listener. | Angle values in degrees (deg), or one of the following predefined values:  left-side far-left  left  center-left  center  center-right  right  far-right  right-side  behind  leftwards  rightwards | div { azimuth:90deg; }  div { azimuth:behind; } |
| cue-after | Declares an audio cue to play after an element. | URL values and the predefined value none. | div { cue-after:url(sound.wav); }  div { cue-after:none; } |
| cue-before | Declares an audio cue to play before an element. | URL values and the predefined value none. | div { cue-before:url(sound.wav); }  div { cue-before:none; } |
| cue | Shorthand proerty to set both cue values at once. | URL values and the predefined value none. Separate the values by a space in the following order:  cue-before cue-after  If only one cue value is declared, it is used for both before and after. | div { cue:url(sound.wav) url(sound2.wav); }  div { cue:url(sound.wav); } |
| elevation | Declares the elevation of a sound. | Angle values in degrees (deg), or one of the following predefined values:  below level above higher lower | div { elevation:30deg; }  div { elevation:higher; } |
| pause-after | Declares the amount of time to pause after an element. | Time in milliseconds (ms) or percentages. | div { pause-after:100ms; }  div { pause-after:20%; } |
| pause-before | Declares the amount of time to pause before an element. | Time in milliseconds (ms) or percentages. | div { pause-before:100ms; }  div { pause-before:20%; } |
| pause | Shorthand proerty to set both pause values at once. | Separate the values by a space in the following order:  pause-before pause-after  If only one pause value is declared, it is used for both before and after. | div { pause:200ms 100ms; }  div { pause:100ms; } |
| pitch | Declares the average speaking pitch of a voice. | Frequencies in hertz (Hz) or the following predefined values:  x-low low medium high x-high | div { pitch:120Hz; }  div { pitch:high; } |
| pitch-range | Declares a change in the pitch range of a voice. | Number values between 0 and 100 (lower values indicate a flat voice while higher values indicate an animated voice). | div { pitch-range:50; }  div { pitch-range:99; } |
| play-during | Declares a background sound to be played while the current element is spoken. | URL value, followed by one or more of the following keywords, separated by spaces:  mix repeat  Alternatley, one of the following keywords:  auto none | div { play-during:url(music.wav); }  div { play-during:url(music.wav) repeat; }  div { play-during:none; } |
| richness | Declares the richness of the voice in spoken text. | Numeric values between 0 and 100 (lower values have less richness and higher values have more richness). | div { richness:50; }  div { richness:0; } |
| speak | Declares if/how text is spoken. | normal none spell-out | div { speak:none; }  div { speak:spell-out; } |
| speak-header | Declares how often table header cells are spoken. | once always | th { speak-header:once; }  th { speak-header:always; } |
| speak-numeral | Declares how numerals are spoken. | digits continuous | div { speak-numeral:digits; }  div { speak-numeral:continuous; } |
| speak-punctuation | Declares how punctuation is spoken. | code none | div { speak-punctuation:code; }  div { speak-punctuation:none; } |
| speech-rate | Declares the speech rate of spoken text. | A number indicating the number of words per minute, or one of the following predefined values:  x-slow slow medium fast x-fast faster slower | div { speech-rate:50; }  div { speech-rate:medium; } |
| stress | Declares the stress of the voice on spoken text. | Numeric values between 0 and 100 (lower values have less stress and higher values have more stress). | div { stress:50; }  div { stress:0; } |
| voice-family | Declares the voice family of spoken text. | Generic or specific voice family names. |  |
| volume | Declares the median volume. | Numbers between 0 and 100, percentages, or one of the following predefined values:  silent x-soft soft medium loud x-loud | div { volume:50; }  div { volume:silent; } |