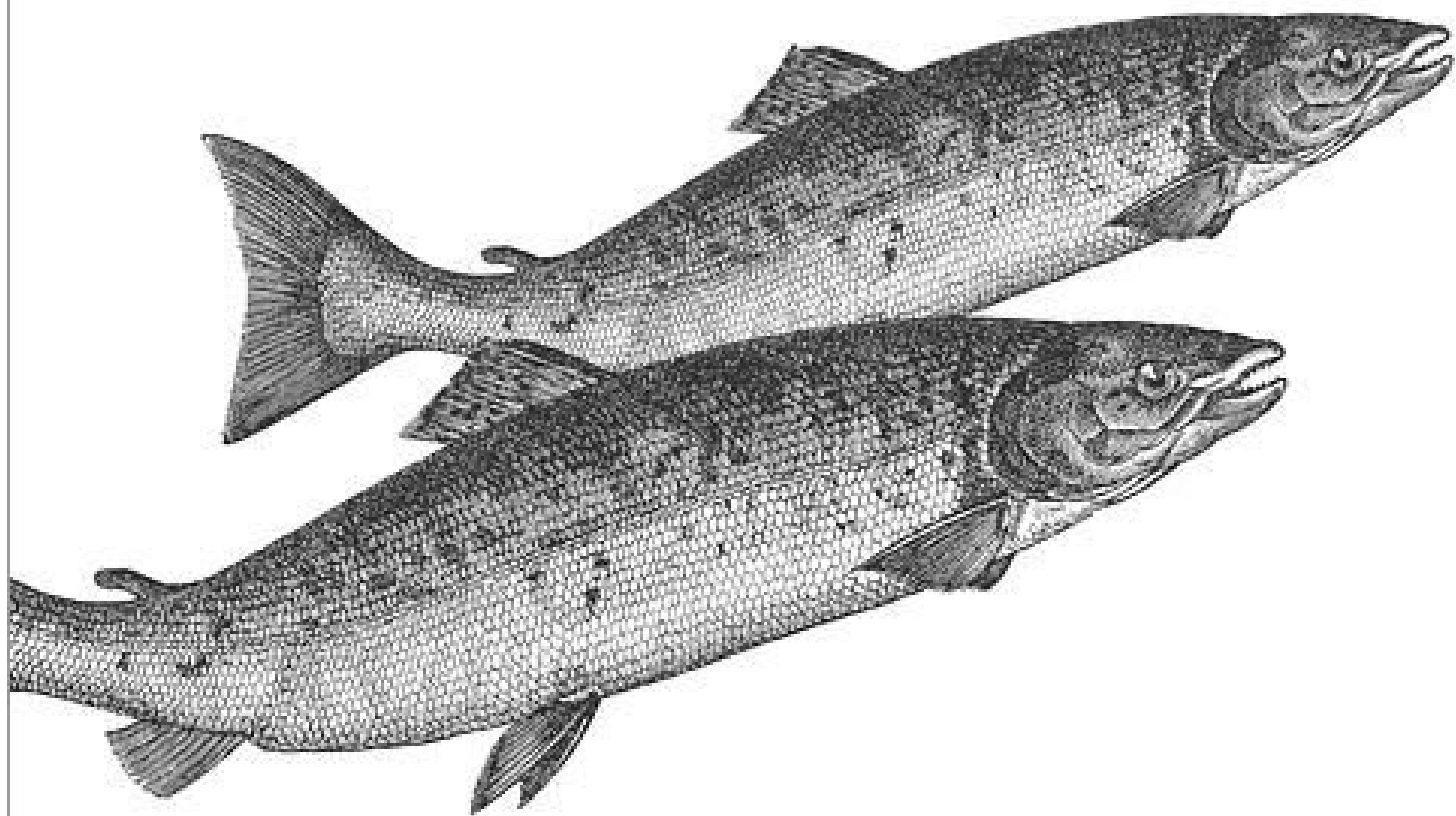


*Visual Presentation for the Web*

**3rd Edition**

# CSS

*The Definitive Guide*



**O'REILLY®**

*Eric Meyer*

<b>Appendix A. Property Reference.....</b>	<b>1</b>
Section A.1. Visual Media.....	1
Section A.2. Tables.....	30
Section A.3. Paged Media.....	32
Section A.4. Dropped from CSS2.1.....	34
Section A.5. Visual Styles.....	34
Section A.6. Paged Media.....	35
Section A.7. Aural Styles.....	36

---

# Property Reference

## Visual Media

---

### background

This is a shorthand method to express all of the individual background properties within a single declaration. Use of this property is generally encouraged over the individual properties, as it has a slightly better support profile in older browsers and doesn't take as long to type.

**Values:** [ <background-color> || <background-image> || <background-repeat> || <background-attachment> || <background-position> ] | inherit

**Initial value:** Refer to individual properties

**Applies to:** All elements

**Inherited:** No

**Percentages:** Values are allowed for <background-position>

**Computed value:** See individual properties

---

### background-attachment

This property defines whether the background image scrolls along with the element when the document is scrolled. This can be used to create “aligned” backgrounds; see Chapter 9 for more details.

**Values:** scroll | fixed | inherit

**Initial value:** scroll

**Applies to:** All elements

**Inherited:** No

**Computed value:** As specified

---

## background-color

This sets a solid color for the background of the element. This color fills the content, padding, and border areas of the element, extending to the outer edge of the element's border. Borders that have transparent sections, such as dashed borders, will show the background color through the transparent sections.

**Values:** <color> | transparent | inherit

**Initial value:** transparent

**Applies to:** All elements

**Inherited:** No

**Computed value:** As specified

---

## background-image

This places an image in the background of the element. Depending on the value of background-repeat, the image may tile infinitely, along one axis, or not at all. The initial background image (the origin image) is placed according to the value of background-position.

**Values:** <uri> | none | inherit

**Initial value:** none

**Applies to:** All elements

**Inherited:** No

**Computed value:** Absolute URI

---

## background-position

This property sets the position of the background's origin image (as defined by background-image); this is the point from which any background repetition will occur.

**Values:** [[<percentage> | <length> | left | center | right] [<percentage> | <length> | top | center | bottom]?] | [[left | center | right] || [top | center | bottom]] | inherit

**Initial value:** 0% 0%

**Applies to:** Block-level and replaced elements

**Inherited:** No

**Percentages:** Refer to the corresponding point on both the element and the origin image

**Computed value:** The absolute length offsets, if <length> is specified; otherwise, percentage values

---

## background-repeat

This defines the tiling pattern for the background image. Note that the axis-related repeat values actually cause repetition in *both* directions along the relevant axis. The repetition begins from the origin image, which is defined as the value of background-image and is placed according to the value of background-position.

**Values:** repeat | repeat-x | repeat-y | no-repeat | inherit

**Initial value:** repeat

**Applies to:** All elements

**Inherited:** No

**Computed value:** As specified

---

## border

This is a shorthand property that defines the width, color, and style of an element's border. Note that while none of the values are actually required, omitting a border style will result in no border being applied because the default border style is none.

**Values:** [ <border-width> || <border-style> || <border-color> ] | inherit

**Initial value:** Refer to individual properties

**Applies to:** All elements

**Inherited:** No

**Computed value:** As specified

---

## border-bottom

This shorthand property defines the width, color, and style of the bottom border of an element. As with border, omission of a border style will result in no border appearing.

**Values:** [ <border-width> || <border-style> || <border-color> ] | inherit

**Initial value:** Not defined for shorthand properties

**Applies to:** All elements

**Inherited:** No

**Computed value:** See individual properties (border-width, etc.)

---

## border-bottom-color

This property sets the color for the visible portions of the bottom border of an element. Only a solid color can be defined, and the border's style must be a value other than none or hidden for any border to appear.

<b>Values:</b>	<color>   transparent   inherit
<b>Initial value:</b>	The value of color for the element
<b>Applies to:</b>	All elements
<b>Inherited:</b>	No
<b>Computed value:</b>	If no value is specified, use the computed value of the property color for the same element; otherwise, as specified

---

## border-bottom-style

This defines the style for the bottom border of an element. The value must be something other than none for any border to appear. In CSS1, HTML user agents were only required to support solid and none.

<b>Values:</b>	none   hidden   dotted   dashed   solid   double   groove   ridge   inset   outset   inherit
<b>Initial value:</b>	none
<b>Applies to:</b>	All elements
<b>Inherited:</b>	No
<b>Computed value:</b>	As specified

---

## border-bottom-width

This sets the width for the bottom border of an element, which will take effect only if the border's style is something other than none. If the border style is none, the border width is effectively reset to 0. Negative length values are not permitted.

<b>Values:</b>	thin   medium   thick   <length>   inherit
<b>Initial value:</b>	medium
<b>Applies to:</b>	All elements
<b>Inherited:</b>	No
<b>Computed value:</b>	Absolute length; 0 if the style of the border is none or hidden

---

## border-color

This shorthand property sets the color for the visible portions of the overall border of an element or sets a different color for each of the four sides. Remember that a border's style must be something other than none or hidden for any border to appear.

<b>Values:</b>	[ <color>   transparent ]{1,4}   inherit
<b>Initial value:</b>	Not defined for shorthand properties

**Applies to:** All elements  
**Inherited:** No  
**Computed value:** See individual properties (border-top-color, etc.)

---

## border-left

This shorthand property defines the width, color, and style of the left border of an element. As with border, omission of a border style will result in no border appearing.

**Values:** [ <border-width> || <border-style> || <border-color> ] | inherit  
**Initial value:** Not defined for shorthand properties  
**Applies to:** All elements  
**Inherited:** No  
**Computed value:** See individual properties (border-width, etc.)

---

## border-left-color

This property sets the color for the visible portions of the left border of an element. Only a solid color can be defined, and the border's style must be something other than none or hidden for any border to appear.

**Values:** <color> | transparent | inherit  
**Initial value:** The value of color for the element  
**Applies to:** All elements  
**Inherited:** No  
**Computed value:** If no value is specified, use the computed value of the property color for the same element; otherwise, as specified

---

## border-left-style

This defines the style for the left border of an element. The value must be something other than none for any border to appear. In CSS1, HTML user agents were only required to support solid and none.

**Values:** none | hidden | dotted | dashed | solid | double | groove | ridge | inset | outset | inherit  
**Initial value:** none  
**Applies to:** All elements  
**Inherited:** No  
**Computed value:** As specified

---

## border-left-width

This sets the width for the left border of an element, which will take effect only if the border's style is something other than none. If the border style is none, the border width is effectively reset to 0. Negative length values are not permitted.

**Values:** thin | medium | thick | <length> | inherit

**Initial value:** medium

**Applies to:** All elements

**Inherited:** No

**Computed value:** Absolute length; 0 if the style of the border is none or hidden

---

## border-right

This shorthand property defines the width, color, and style of the right border of an element. As with border, omission of a border style will result in no border appearing.

**Values:** [ <border-width> || <border-style> || <border-color> ] | inherit

**Initial value:** Not defined for shorthand properties

**Applies to:** All elements

**Inherited:** No

**Computed value:** See individual properties (border-width, etc.)

---

## border-right-color

This property sets the color for the visible portions of the right border of an element. Only a solid color can be defined, and the border's style must be something other than none or hidden for any border to appear.

**Values:** <color> | transparent | inherit

**Initial value:** The value of color for the element

**Applies to:** All elements

**Inherited:** No

**Computed value:** If no value is specified, use the computed value of the property color for the same element; otherwise, as specified



---

## border-right-style

This defines the style for the right border of an element. The value must be something other than none for any border to appear. In CSS1, HTML user agents were only required to support solid and none.

**Values:** none | hidden | dotted | dashed | solid | double | groove | ridge | inset | outset | inherit

**Initial value:** none

**Applies to:** All elements

**Inherited:** No

**Computed value:** As specified

---

## border-right-width

This sets the width for the right border of an element, which will only take effect if the border's style is something other than none. If the border style is none, the border width is effectively reset to 0. Negative length values are not permitted.

**Values:** thin | medium | thick | <length> | inherit

**Initial value:** medium

**Applies to:** All elements

**Inherited:** No

**Computed value:** Absolute length; 0 if the style of the border is none or hidden

---

## border-style

This shorthand property can be used to set the styles for the overall border of an element, or for each side individually. The value of any border must be something other than none for the border to appear.

**Values:** [ none | hidden | dotted | dashed | solid | double | groove | ridge | inset | outset ]{1,4} | inherit

**Initial value:** Not defined for shorthand properties

**Applies to:** All elements

**Inherited:** No

**Computed value:** See individual properties (border-top-style, etc.)

**Note:** In CSS1, HTML user agents were only required to support solid and none; the rest of the values (except for hidden) may be interpreted as solid

---

## border-top

This shorthand property defines the width, color, and style of the top border of an element. As with `border`, omission of a border style will result in no border appearing.

**Values:** [ <border-width> || <border-style> || <border-color> ] | inherit

**Initial value:** Not defined for shorthand properties

**Applies to:** All elements

**Inherited:** No

**Computed value:** See individual properties (`border-width`, etc.)

---

## border-top-color

This property sets the color for the visible portions of the top border of an element. Only a solid color can be defined, and the border's style must be something other than `none` or `hidden` for any border to appear.

**Values:** <color> | transparent | inherit

**Initial value:** The value of `color` for the element

**Applies to:** All elements

**Inherited:** No

**Computed value:** If no value is specified, use the computed value of the property `color` for the same element; otherwise, as specified

---

## border-top-style

This defines the style for the top border of an element. The value must be something other than `none` for any border to appear. In CSS1, HTML user agents were only required to support `solid` and `none`.

**Values:** none | hidden | dotted | dashed | solid | double | groove | ridge | inset | outset | inherit

**Initial value:** none

**Applies to:** All elements

**Inherited:** No

**Computed value:** As specified

---

## border-top-width

This sets the width for the top border of an element, which will only take effect if the border's style is something other than none. If the style is none, the width is effectively reset to 0. Negative length values are not permitted.

**Values:** thin | medium | thick | <length> | inherit

**Initial value:** medium

**Applies to:** All elements

**Inherited:** No

**Computed value:** Absolute length; 0 if the style of the border is none or hidden

---

## border-width

This shorthand property can be used to set the width for the overall border of an element or for each side individually. The width will take effect for a given border only if the border's style is something other than none. If the border style is none, the border width is effectively reset to 0. Negative length values are not permitted.

**Values:** [ thin | medium | thick | <length> ]{1,4} | inherit

**Initial value:** Not defined for shorthand properties

**Applies to:** All elements

**Inherited:** No

**Computed value:** See individual properties (border-top-style, etc.)

---

## bottom

This property defines the offset between the bottom outer margin edge of a positioned element and the bottom edge of its containing block.

**Values:** <length> | <percentage> | auto | inherit

**Initial value:** auto

**Applies to:** Positioned elements (that is, elements for which the value of position is something other than static)

**Inherited:** No

**Percentages:** Refer to the height of the containing block

**Computed value:** For relatively positioned elements, see Note; for static elements, auto; for length values, the corresponding absolute length; for percentage values, the specified value; otherwise, auto

**Note:** For relatively positioned elements, if both `bottom` and `top` are auto, their computed values are both 0; if one of them is auto, it becomes the negative of the other; if neither is auto, `bottom` will become the negative of the value of `top`

---

## clear

This defines the sides of an element on which no floating elements may appear. In CSS1 and CSS2, this is accomplished by automatically increasing the top margin of the cleared element. In CSS2.1, clearance space is added above the element's top margin, but the margin itself is not altered. In either case, the end result is that the element's top outer border edge is just below the bottom outer margin edge of a floated element on the declared side.

**Values:** left | right | both | none | inherit

**Initial value:** none

**Applies to:** Block-level elements

**Inherited:** No

**Computed value:** As specified

---

## clip

This is used to define a clipping rectangle inside of which the content of an absolutely positioned element is visible. Content outside this clipping area is treated according to the value of `overflow`. The clipping area can be smaller or larger than the content area of the element.

**Values:** `rect(top, right, bottom, left)` | auto | inherit

**Initial value:** auto

**Applies to:** Absolutely positioned elements (in CSS2, `clip` applied to block-level and replaced elements)

**Inherited:** No

**Computed value:** For a rectangle, a set of four computed lengths representing the edges of the clipping rectangle; otherwise, as specified

---

## color

This property sets the foreground color of an element, which in HTML rendering means the text of an element; raster images are not affected by color. This is also the color applied to any borders of the element, unless overridden by `border-color` or one of the other border color properties (`border-top-color`, etc.).

**Values:** `<color>` | `inherit`

**Initial value:** User agent-specific

**Applies to:** All elements

**Inherited:** Yes

**Computed value:** As specified

---

## content

This is the property used to define the generated content placed before or after an element. By default, this is likely to be inline content, but the type of box the content creates can be controlled using the property `display`.

**Values:** `normal` | [`<string>` | `<uri>` | `<counter>` | `attr(<identifier>)` | `open-quote` | `close-quote` | `no-open-quote` | `no-close-quote` ]+ | `inherit`

**Initial value:** `normal`

**Applies to:** `:before` and `:after` pseudo-elements

**Inherited:** No

**Computed value:** For `<uri>` values, an absolute URI; for attribute references, the resulting string; otherwise, as specified

---

## counter-increment

With this property, counters can be incremented (or decremented) by any value, positive or negative. If no `<integer>` is supplied, it defaults to 1.

**Values:** [`<identifier> <integer>?` ]+ | `none` | `inherit`

**Initial value:** User agent-dependent

**Applies to:** All elements

**Inherited:** No

**Computed value:** As specified

---

## counter-reset

With this property, counters can be set or reset to any value, positive or negative. If no `<integer>` is supplied, it defaults to 0.

**Values:** [`<identifier> <integer>? ]+ | none | inherit`

**Initial value:** User agent-dependent

**Applies to:** All elements

**Inherited:** No

**Computed value:** As specified

---

## cursor

This defines the cursor shape to be used when a mouse pointer is placed within the boundary of an element (although CSS2.1 does not define which edge creates this boundary).

**Values:** `[[<uri>,* [ auto | default | pointer | crosshair | move | e-resize | ne-resize | nw-resize | n-resize | se-resize | sw-resize | s-resize | w-resize | text | wait | help | progress ]] | inherit`

**Initial value:** auto

**Applies to:** All elements

**Inherited:** Yes

**Computed value:** For `<uri>` values, an absolute URI; otherwise, as specified

---

## direction

This property specifies the base writing direction of blocks and the direction of embeddings and overrides for the Unicode bidirectional algorithm. User agents that do not support bidirectional text are permitted to ignore this property.

**Values:** `ltr | rtl | inherit`

**Initial value:** ltr

**Applies to:** All elements

**Inherited:** Yes

**Computed value:** As specified

---

## display

This is used to define the kind of display box an element generates during layout. Gratuitous use of display with a document type such as HTML can be dangerous, as it upsets

---

the display hierarchy already defined in HTML. In the case of XML, which has no such built-in hierarchy, display is indispensable.

<b>Values:</b>	none   inline   block   inline-block   list-item   run-in   table   inline-table   table-row-group   table-header-group   table-footer-group   table-row   table-column-group   table-column   table-cell   table-caption   inherit
<b>Initial value:</b>	inline
<b>Applies to:</b>	All elements
<b>Inherited:</b>	No
<b>Computed value:</b>	Varies for floated, positioned, and root elements (see CSS2.1, section 9.7); otherwise, as specified
<b>Note:</b>	The values compact and marker appeared in CSS2 but were dropped from CSS2.1 due to a lack of widespread support

---

## float

This defines the direction in which an element is floated. It has traditionally been applied to images to let text flow around them, but in CSS, any element may be floated. A floated element will generate a block-level box no matter what kind of element it may be. Floated nonreplaced elements should be given an explicit width; otherwise, they may tend to become as narrow as possible.

<b>Values:</b>	left   right   none   inherit
<b>Initial value:</b>	none
<b>Applies to:</b>	All elements
<b>Inherited:</b>	No
<b>Computed value:</b>	As specified

---

## font

This is a shorthand property used to set two or more aspects of an element's font all at once. It can also be used to set the element's font to match an aspect of the user's computing environment using keywords such as icon. Note that if these keywords are not used, the minimum font value must include the font size and family.

<b>Values:</b>	[[ <font-style>    <font-variant>    <font-weight> ]? <font-size> [ / <line-height> ]? <font-family>]   caption   icon   menu   message-box   small-caption   status-bar   inherit
<b>Initial value:</b>	Refer to individual properties
<b>Applies to:</b>	All elements

<b>Inherited:</b>	Yes
<b>Percentages:</b>	Calculated with respect to the parent element for <font-size> and with respect to the element's <font-size> for <line-height>
<b>Computed value:</b>	See individual properties (font-style, etc.)

---

## font-family

This defines a font family to be used in the display of an element's text. Note that use of a specific font family (e.g., Geneva) is wholly dependent on that family being available on a user's machine; no font downloading is implied by this property. Therefore, the use of generic family names as a fallback is strongly encouraged.

<b>Values:</b>	[[<family-name>   <generic-family>],]* [<family-name>   <generic-family>]   inherit
<b>Initial value:</b>	User agent-specific
<b>Applies to:</b>	All elements
<b>Inherited:</b>	Yes
<b>Computed value:</b>	As specified

---

## font-size

This sets the size of the font for an element. Note that it actually sets the height of the character boxes in the font; the actual character glyphs may be taller or shorter than these boxes (usually shorter). Each keyword must be larger than the next smallest keyword and smaller than the next biggest keyword. Negative length and percentage values are not permitted.

<b>Values:</b>	xx-small   x-small   small   medium   large   x-large   xx-large   smaller   larger   <length>   <percentage>   inherit
<b>Initial value:</b>	medium
<b>Applies to:</b>	All elements
<b>Inherited:</b>	Yes
<b>Percentages:</b>	Calculated with respect to the parent element's font size
<b>Computed value:</b>	An absolute length



---

## font-style

This sets the font to use an italic, oblique, or normal font face. Italic text is generally defined as a separate face within the font family. It is theoretically possible for a user agent to compute a slanted font face from the normal face.

**Values:** italic | oblique | normal | inherit

**Initial value:** normal

**Applies to:** All elements

**Inherited:** Yes

**Computed value:** As specified

---

## font-variant

This property is basically used to define small-caps text. It is theoretically possible for a user agent to compute a small-caps font face from the normal face.

**Values:** small-caps | normal | inherit

**Initial value:** normal

**Applies to:** All elements

**Inherited:** Yes

**Computed value:** As specified

---

## font-weight

This property sets the font weight used in rendering an element's text. The numeric value 400 is equivalent to the keyword `normal`, and 700 is equivalent to `bold`. Each numeric value must be at least as light as the next lowest number and at least as heavy as the next highest number.

**Values:** normal | bold | bolder | lighter | 100 | 200 | 300 | 400 | 500 | 600 | 700 | 800 | 900 | inherit

**Initial value:** normal

**Applies to:** All elements

**Inherited:** Yes

**Computed value:** One of the numeric values (100, etc.), or one of the numeric values plus one of the relative values (`bolder` or `lighter`)

---

## height

This defines the height of an element's content area, outside of which padding, borders, and margins are added. This property is ignored for inline nonreplaced elements. Negative length and percentage values are not permitted.

<b>Values:</b>	<length>   <percentage>   auto   inherit
<b>Initial value:</b>	auto
<b>Applies to:</b>	Block-level and replaced elements
<b>Inherited:</b>	No
<b>Percentages:</b>	Calculated with respect to the height of the containing block
<b>Computed value:</b>	For auto and percentage values, as specified; otherwise, an absolute length, unless the property does not apply to the element (then auto)

---

## left

This property defines the offset between the left outer margin edge of a positioned element and the left edge of its containing block.

<b>Values:</b>	<length>   <percentage>   auto   inherit
<b>Initial value:</b>	auto
<b>Applies to:</b>	Positioned elements (that is, elements for which the value of position is something other than static)
<b>Inherited:</b>	No
<b>Percentages:</b>	Refer to the width of the containing block
<b>Computed value:</b>	For relatively positioned elements, see Note; for static elements, auto; for length values, the corresponding absolute length; for percentage values, the specified value; otherwise, auto
<b>Note:</b>	For relatively positioned elements, the computed value of left always equals right

---

## letter-spacing

This defines the amount of whitespace to be inserted between the character boxes of text. Since character glyphs are typically narrower than their character boxes, length values create a modifier to the usual spacing between letters. Thus, *normal* is synonymous with 0. Negative length values are permitted and will cause letters to bunch closer together.

<b>Values:</b>	<length>   normal   inherit
<b>Initial value:</b>	normal

**Applies to:** All elements  
**Inherited:** Yes  
**Computed value:** For length values, the absolute length; otherwise, normal

---

## line-height

This property influences the layout of line boxes. When applied to a block-level element, it defines the minimum distance between baselines within that element, but not the maximum. The difference between the computed values of line-height and font-size (called “leading” in CSS) is split in half and added to the top and bottom of each piece of content in a line of text. The shortest box that can enclose all of those pieces of content is the line box. A raw number value assigns a scaling factor, which is inherited instead of a computed value. Negative values are not permitted.

**Values:** <length> | <percentage> | <number> | normal | inherit  
**Initial value:** normal  
**Applies to:** All elements (but see text regarding replaced and block-level elements)  
**Inherited:** Yes  
**Percentages:** Relative to the font size of the element  
**Computed value:** For length and percentage values, the absolute value; otherwise, as specified

---

## list-style

This is a shorthand property that condenses all the other list-style properties. Because it applies to any element that has a display of list-item, it will apply only to li elements in ordinary HTML and XHTML, although it can be applied to any element and inherited by list-item elements.

**Values:** [ <list-style-type> || <list-style-image> || <list-style-position> ] | inherit  
**Initial value:** Refer to individual properties  
**Applies to:** Elements whose display value is list-item  
**Inherited:** Yes  
**Computed value:** See individual properties

---

## list-style-image

This specifies an image to be used as the marker on an ordered or unordered list item. The placement of the image with respect to the content of the list item can be broadly controlled using list-style-position.

**Values:** <uri> | none | inherit  
**Initial value:** none  
**Applies to:** Elements whose display value is list-item  
**Inherited:** Yes  
**Computed value:** For <uri> values, the absolute URI; otherwise, none

---

## list-style-position

This property is used to declare the position of the list marker with respect to the content of the list item. Outside markers are placed some distance from the border edge of the list item, but the distance is not defined in CSS. Inside markers are treated as though they were an inline element inserted at the beginning of the list item's content.

**Values:** inside | outside | inherit  
**Initial value:** outside  
**Applies to:** Elements whose display value is list-item  
**Inherited:** Yes  
**Computed value:** As specified

---

## list-style-type

This is used to declare the type of marker system to be used in the presentation of a list.

**CSS2.1 values:** disc | circle | square | decimal | decimal-leading-zero | lower-roman | upper-roman | lower-greek | lower-latin | upper-latin | armenian | georgian | none | inherit  
**CSS2 values:** disc | circle | square | decimal | decimal-leading-zero | upper-alpha | lower-alpha | upper-roman | lower-roman | lower-greek | hebrew | armenian | georgian | cjk-ideographic | hiragana | katakana | hiragana-iroha | none | inherit  
**Initial value:** disc  
**Applies to:** Elements whose display value is list-item  
**Inherited:** Yes  
**Computed value:** As specified

---

---

## margin

This shorthand property sets the width of the overall margin for an element or sets the widths of each individual side margin. Vertically adjacent margins of block-level elements are collapsed, whereas inline elements effectively do not take top and bottom margins. The left and right margins of inline elements do not collapse, nor do margins on floated elements. Negative margin values are permitted, but use them with caution.

**Values:** [ <length> | <percentage> | auto ]{1,4} | inherit

**Initial value:** Not defined

**Applies to:** All elements

**Inherited:** No

**Percentages:** Refer to the width of the containing block

**Computed value:** See individual properties

---

## margin-bottom

This sets the width of the bottom margin for an element. Negative values are permitted, but use them with caution.

**Values:** <length> | <percentage> | auto | inherit

**Initial value:** 0

**Applies to:** All elements

**Inherited:** No

**Percentages:** Refer to the width of the containing block

**Computed value:** For percentages, as specified; for length values, the absolute length

---

## margin-left

This sets the width of the left margin for an element. Negative values are permitted, but use them with caution.

**Values:** <length> | <percentage> | auto | inherit

**Initial value:** 0

**Applies to:** All elements

**Inherited:** No

**Percentages:** Refer to the width of the containing block

**Computed value:** For percentages, as specified; for length values, the absolute length

---

## margin-right

This sets the width of the right margin for an element. Negative values are permitted, but use them with caution.

**Values:** <length> | <percentage> | auto | inherit

**Initial value:** 0

**Applies to:** All elements

**Inherited:** No

**Percentages:** Refer to the width of the containing block

**Computed value:** For percentages, as specified; for length values, the absolute length

---

## margin-top

This sets the width of the top margin for an element. Negative values are permitted, but use them with caution.

**Values:** <length> | <percentage> | auto | inherit

**Initial value:** 0

**Applies to:** All elements

**Inherited:** No

**Percentages:** Refer to the width of the containing block

**Computed value:** For percentages, as specified; for length values, the absolute length

---

## max-height

The value of this property sets a maximum constraint on the height of the element. Thus, the element can be shorter than the specified value, but not taller. Negative values are not permitted.

**Values:** <length> | <percentage> | none | inherit

**Initial value:** none

**Applies to:** All elements except inline nonreplaced elements and table elements

**Inherited:** No

**Percentages:** Refer to the height of the containing block

**Computed value:** For percentages, as specified; for length values, the absolute length; otherwise, none

---

## max-width

The value of this property sets a maximum constraint on the width of the element. Thus, the element can be narrower than the specified value, but not wider. Negative values are not permitted.

<b>Values:</b>	<length>   <percentage>   none   inherit
<b>Initial value:</b>	none
<b>Applies to:</b>	All elements except inline nonreplaced elements and table elements
<b>Inherited:</b>	No
<b>Percentages:</b>	Refer to the width of the containing block
<b>Computed value:</b>	For percentages, as specified; for length values, the absolute length; otherwise, none

---

## min-height

The value of this property sets a minimum constraint on the height of the element. Thus, the element can be taller than the specified value, but not shorter. Negative values are not permitted.

<b>Values:</b>	<length>   <percentage>   inherit
<b>Initial value:</b>	0
<b>Applies to:</b>	All elements except inline nonreplaced elements and table elements
<b>Inherited:</b>	No
<b>Percentages:</b>	Refer to the height of the containing block
<b>Computed value:</b>	For percentages, as specified; for length values, the absolute length

---

## min-width

The value of this property sets a minimum constraint on the width of the element. Thus, the element can be wider than the specified value, but not narrower. Negative values are not permitted.

<b>Values:</b>	<length>   <percentage>   inherit
<b>Initial value:</b>	0
<b>Applies to:</b>	All elements except inline nonreplaced elements and table elements
<b>Inherited:</b>	No
<b>Percentages:</b>	refer to the width of the containing block
<b>Computed value:</b>	For percentages, as specified; for length values, the absolute length; otherwise, none

---

## outline

This shorthand property is used to set the overall outline for an element. Outlines can be of irregular shape, and they do not change or otherwise affect the placement of elements.

**Values:** [ <outline-color> || <outline-style> || <outline-width> ] | inherit

**Initial value:** Not defined for shorthand properties

**Applies to:** All elements

**Inherited:** No

**Computed value:** See individual properties (outline-color, etc.)

---

## outline-color

This property sets the color for the visible portions of the overall outline of an element. Remember that an outline's style must be something other than none for any border to appear.

**Values:** <color> | invert | inherit

**Initial value:** invert (or user agent-specific; see text)

**Applies to:** All elements

**Inherited:** No

**Computed value:** As specified

---

## outline-style

This property is used to set the style for the overall border of an element. The style must be something other than none for any outline to appear.

**Values:** none | dotted | dashed | solid | double | groove | ridge | inset | outset | inherit

**Initial value:** none

**Applies to:** All elements

**Inherited:** No

**Computed value:** As specified

---

## outline-width

This property sets the width for the overall outline of an element. The width will take effect only for a given outline if the outline's style is something other than none. If the style is none, the width is effectively reset to 0. Negative length values are not permitted.

---



**Values:** thin | medium | thick | <length> | inherit  
**Initial value:** medium  
**Applies to:** All elements  
**Inherited:** No  
**Computed value:** Absolute length; 0 if the style of the border is none or hidden

---

## overflow

This defines what happens to content that overflows the content area of an element. For the value scroll, user agents are supposed to provide a scrolling mechanism whether or not it is actually needed; thus, for example, scrollbars would appear even if all content can fit within the element box.

**Values:** visible | hidden | scroll | auto | inherit  
**Initial value:** visible  
**Applies to:** Block-level and replaced elements  
**Inherited:** No  
**Computed value:** As specified

---

## padding

This shorthand property sets the width of the overall padding for an element or sets the widths of each individual side padding. Padding set on inline nonreplaced elements does not affect line-height calculations; therefore, such an element with both padding and a background may visibly extend into other lines and potentially overlap other content. The background of the element will extend throughout the padding. Negative padding values are not permitted.

**Values:** [ <length> | <percentage> ]{1,4} | inherit  
**Initial value:** Not defined for shorthand elements  
**Applies to:** All elements  
**Inherited:** No  
**Percentages:** Refer to the width of the containing block  
**Computed value:** See individual properties (padding-top, etc.)  
**Note:** Padding can never be negative

---

---

## padding-bottom

This property sets the width of the bottom padding for an element. Bottom padding set on inline nonreplaced elements does not affect line-height calculations; therefore, such an element with both bottom padding and a background may visibly extend into other lines and potentially overlap other content. Negative padding values are not permitted.

**Values:** <length> | <percentage> | inherit

**Initial value:** 0

**Applies to:** All elements

**Inherited:** No

**Percentages:** Refer to the width of the containing block

**Computed value:** For percentage values, as specified; for length values, the absolute length

**Note:** Padding can never be negative

---

## padding-left

This property sets the width of the left padding for an element. Left padding set for an inline nonreplaced element will appear only on the left edge of the first inline box generated by the element. Negative padding values are not permitted.

**Values:** <length> | <percentage> | inherit

**Initial value:** 0

**Applies to:** All elements

**Inherited:** No

**Percentages:** Refer to the width of the containing block

**Computed value:** For percentage values, as specified; for length values, the absolute length

**Note:** Padding can never be negative

---

## padding-right

This property sets the width of the right padding for an element. Right padding set for an inline nonreplaced element will appear only on the right edge of the last inline box generated by the element. Negative padding values are not permitted.

**Values:** <length> | <percentage> | inherit

**Initial value:** 0

**Applies to:** All elements

**Inherited:** No

**Percentages:** Refer to the width of the containing block

**Computed value:** For percentage values, as specified; for length values, the absolute length

**Note:** Padding can never be negative

---

## padding-top

This property sets the width of the top padding for an element. Top padding set on inline nonreplaced elements does not affect line-height calculations; therefore, such an element with both top padding and a background may visibly extend into other lines and potentially overlap other content. Negative padding values are not permitted.

**Values:** <length> | <percentage> | inherit

**Initial value:** 0

**Applies to:** All elements

**Inherited:** No

**Percentages:** Refer to the width of the containing block

**Computed value:** For percentage values, as specified; for length values, the absolute length

**Note:** Padding can never be negative

---

## position

This defines the positioning scheme used to lay out an element. Any element may be positioned, although elements positioned with `absolute` or `fixed` will generate a block-level box no matter what kind of element they are. An element that is relatively positioned is offset from its default placement in the normal flow.

**Values:** `static` | `relative` | `absolute` | `fixed` | `inherit`

**Initial value:** `static`

**Applies to:** All elements

**Inherited:** No

**Computed value:** As specified

---

## quotes

This property is used to determine the quotation pattern used with quotes and nested quotes. The actual quote marks are inserted via the property content.

**Values:** [`<string>` `<string>`]+ | `none` | `inherit`

**Initial value:** User agent-dependent

**Applies to:** All elements

**Inherited:** Yes

**Computed value:** As specified

---

## right

This property defines the offset between the right outer margin edge of a positioned element and the right edge of its containing block.

**Values:** <length> | <percentage> | auto | inherit

**Initial value:** auto

**Applies to:** Positioned elements (that is, elements for which the value of position is something other than static)

**Inherited:** No

**Percentages:** Refer to the width of the containing block

**Computed value:** For relatively positioned elements, see Note; for static elements, auto; for length values, the corresponding absolute length; for percentage values, the specified value; otherwise, auto

**Note:** For relatively positioned elements, the computed value of left always equals right

---

## text-align

This property sets the horizontal alignment of text within a block-level element by defining the point with which line boxes are aligned. The value justify is supported by allowing user agents to programmatically adjust the letter and word spacing of the line's content; results may vary by user agent.

**CSS2.1 values:** left | center | right | justify | inherit

**CSS2 values:** left | center | right | justify | <string> | inherit

**Initial value:** User agent-specific; may also depend on writing direction

**Applies to:** Block-level elements

**Inherited:** Yes

**Computed value:** As specified

**Note:** CSS2 included a <string> value that was dropped from CSS2.1 due to a lack of widespread support

---

## text-decoration

This property allows certain text effects such as underlining. These decorations will “span” descendant elements that do not have decorations of their own. User agents are not required to support blink.

**Values:** none | [ underline || overline || line-through || blink ] | inherit  
**Initial value:** none  
**Applies to:** All elements  
**Inherited:** No  
**Computed value:** As specified

---

## text-indent

Used to define the indentation of the first line of content in a block-level element. This is most often used to create a “tab” effect. Negative values are permitted and cause “outdent” (or “hanging indent”) effects.

**Values:** <length> | <percentage> | inherit  
**Initial value:** 0  
**Applies to:** Block-level elements  
**Inherited:** Yes  
**Percentages:** Refer to the width of the containing block  
**Computed value:** For percentage values, as specified; for length values, the absolute length

---

## text-transform

This property changes the case of letters in an element, regardless of the case of the text in the document source. The determination of which letters are to be capitalized by the value capitalize is not precisely defined, as it depends on user agents knowing how to recognize a “word.”

**Values:** uppercase | lowercase | capitalize | none | inherit  
**Initial value:** none  
**Applies to:** All elements  
**Inherited:** Yes  
**Computed value:** As specified

---

## top

This property defines the offset between the top outer margin edge of a positioned element and the top edge of its containing block.

<b>Values:</b>	<length>   <percentage>   auto   inherit
<b>Initial value:</b>	auto
<b>Applies to:</b>	Positioned elements (that is, elements for which the value of position is something other than static)
<b>Inherited:</b>	No
<b>Percentages:</b>	Refer to the height of the containing block
<b>Computed value:</b>	For relatively positioned elements, see Note; for static elements, auto; for length values, the corresponding absolute length; for percentage values, the specified value; otherwise, auto
<b>Note:</b>	For relatively positioned elements, if both top and bottom are auto, their computed values are both 0; if one of them is auto, it becomes the negative of the other; if neither is auto, bottom will become the negative of the value of top

---

## unicode-bidi

This allows the author to generate levels of embedding within the Unicode embedding algorithm. User agents that do not support bidirectional text are permitted to ignore this property.

<b>Values:</b>	normal   embed   bidi-override   inherit
<b>Initial value:</b>	normal
<b>Applies to:</b>	All elements
<b>Inherited:</b>	No
<b>Computed value:</b>	As specified

---

## vertical-align

This defines the vertical alignment of an inline element's baseline with respect to the baseline of the line in which it resides. Negative length and percentage values are permitted, and they lower the element instead of raising it. In table cells, this property sets the alignment of the content of the cell within the cell box.

<b>Values:</b>	baseline   sub   super   top   text-top   middle   bottom   text-bottom   <percentage>   <length>   inherit
<b>Initial value:</b>	baseline

<b>Applies to:</b>	Inline elements and table cells
<b>Inherited:</b>	No
<b>Percentages:</b>	Refer to the value of line-height for the element
<b>Computed value:</b>	For percentage and length values, the absolute length; otherwise, as specified
<b>Note:</b>	When applied to table cells, only the values baseline, top, middle, and bottom are recognized

---

## visibility

This specifies whether the element box generated by an element is rendered. This means the element could occupy the space it would ordinarily, but be completely invisible. The value collapse is used in tables to remove columns or rows from the table's layout.

<b>Values:</b>	visible   hidden   collapse   inherit
<b>Initial value:</b>	visible
<b>Applies to:</b>	All elements
<b>Inherited:</b>	Yes
<b>Computed value:</b>	As specified

---

## white-space

This declares how whitespace within an element is handled during layout. The values pre-wrap and pre-line were added in CSS2.1.

<b>Values:</b>	normal   nowrap   pre   pre-wrap   pre-line   inherit
<b>Initial value:</b>	normal
<b>Applies to:</b>	All elements (CSS2.1); block-level elements (CSS1 and CSS2)
<b>Inherited:</b>	No
<b>Computed value:</b>	As specified

---

## width

This defines the width of an element's content area, outside of which padding, borders, and margins are added. This property is ignored for inline nonreplaced elements. Negative length and percentage values are not permitted.

<b>Values:</b>	<length>   <percentage>   auto   inherit
<b>Initial value:</b>	auto

<b>Applies to:</b>	Block-level and replaced elements
<b>Inherited:</b>	No
<b>Percentages:</b>	Refer to the width of the containing block
<b>Computed value:</b>	For auto and percentage values, as specified; otherwise, an absolute length, unless the property does not apply to the element (then auto)

---

## word-spacing

This defines the amount of whitespace to be inserted between words in an element. For the purposes of this property, a “word” is defined as a string of characters surrounded by whitespace. Length values create a modifier to the usual spacing between words; thus, `normal` is synonymous with 0. Negative length values are permitted and will cause words to bunch closer together.

<b>Values:</b>	<length>   <code>normal</code>   <code>inherit</code>
<b>Initial value:</b>	<code>normal</code>
<b>Applies to:</b>	All elements
<b>Inherited:</b>	Yes
<b>Computed value:</b>	For <code>normal</code> , the absolute length 0; otherwise, the absolute length

---

## z-index

This property sets the placement of a positioned element along the *z*-axis, which is defined as the axis that extends perpendicular to the display area. Positive numbers are closer to the user, and negative numbers are further away.

<b>Values:</b>	<integer>   <code>auto</code>   <code>inherit</code>
<b>Initial value:</b>	<code>auto</code>
<b>Applies to:</b>	Positioned elements
<b>Inherited:</b>	No
<b>Computed value:</b>	As specified

# Tables

---

## border-collapse

This property defines the layout model used in laying out the borders in a table—i.e., those applied to cells, rows, and so forth. Although the property applies only to tables, it is inherited by all the elements within the table.



<b>Values:</b>	collapse   separate   inherit
<b>Initial value:</b>	separate
<b>Applies to:</b>	Elements with the display value table or inline-table
<b>Inherited:</b>	Yes
<b>Computed value:</b>	As specified
<b>Note:</b>	In CSS2, the default value was collapse

---

## border-spacing

This specifies the distance between cell borders in the separated borders model. The first of the two length values is the horizontal separation, and the second is the vertical. This property is ignored unless border-collapse is set to separate. Although the property only applies to tables, it is inherited by all of the elements within the table.

<b>Values:</b>	<length> <length>?   inherit
<b>Initial value:</b>	0
<b>Applies to:</b>	Elements with the display value table or inline-table
<b>Inherited:</b>	Yes
<b>Computed value:</b>	Two absolute lengths
<b>Note:</b>	This property is ignored unless border-collapse value is separate

---

## caption-side

This specifies the placement of a table caption with respect to the table box. The caption is rendered as though it were a block-level element placed just before (or after) the table.

<b>Values:</b>	top   bottom
<b>Initial value:</b>	top
<b>Applies to:</b>	Elements with the display value table-caption
<b>Inherited:</b>	Yes
<b>Computed value:</b>	As specified
<b>Note:</b>	The values left and right appeared in CSS2 but were dropped from CSS2.1 due to a lack of widespread support

---

## empty-cells

This defines the presentation of table cells that contain no content. If shown, the cell's borders and background are drawn. This property is ignored unless border-collapse is set to separate.

<b>Values:</b>	show   hide   inherit
<b>Initial value:</b>	show
<b>Applies to:</b>	Elements with the display value table-cell
<b>Inherited:</b>	Yes
<b>Computed value:</b>	As specified
<b>Note:</b>	This property is ignored unless border-collapse value is separate

---

## table-layout

This property specifies which layout algorithm is used to lay out a table. The fixed layout algorithm is faster but less flexible, while the automatic algorithm is slower but more reflective of traditional HTML tables.

<b>Values:</b>	auto   fixed   inherit
<b>Initial value:</b>	auto
<b>Applies to:</b>	Elements with the display value table or inline-table
<b>Inherited:</b>	Yes
<b>Computed value:</b>	As specified

## Paged Media

---

### orphans

This specifies the minimum number of text lines within the element that can be left at the bottom of a page. This can affect the placement of page breaks within the element.

<b>Values:</b>	<integer>   inherit
<b>Initial value:</b>	2
<b>Applies to:</b>	Block-level elements
<b>Inherited:</b>	Yes
<b>Computed value:</b>	As specified

---

### page-break-after

This declares whether page breaks should be placed after an element. While it is possible to force breaks with always, it is not possible to guarantee prevention; the best an author can do is ask the user agent to avoid inserting a page break if possible.

<b>Values:</b>	auto   always   avoid   left   right   inherit
<b>Initial value:</b>	auto
<b>Applies to:</b>	Nonfloated block-level elements with a position value of relative or static
<b>Inherited:</b>	No
<b>Computed value:</b>	As specified

---

## page-break-before

Declares whether page breaks should be placed before an element. While it is possible to force breaks with `always`, it is not possible to guarantee prevention; the best an author can do is ask the user agent to avoid inserting a page break if possible.

<b>Values:</b>	auto   always   avoid   left   right   inherit
<b>Initial value:</b>	auto
<b>Applies to:</b>	Nonfloated block-level elements with a position value of relative or static
<b>Inherited:</b>	No
<b>Computed value:</b>	As specified

---

## page-break-inside

This declares whether page breaks should be placed inside an element. Because an element might be taller than a page box, it is not possible to guarantee prevention; the best an author can do is ask the user agent to avoid inserting a page break if possible.

<b>Values:</b>	auto   avoid   inherit
<b>Initial value:</b>	auto
<b>Applies to:</b>	Nonfloated block-level elements with a position value of relative or static
<b>Inherited:</b>	Yes
<b>Computed value:</b>	As specified

---

## widows

This specifies the minimum number of text lines within the element that can be left at the top of a page. This can affect the placement of page breaks within the element.

<b>Values:</b>	<integer>   inherit
<b>Initial value:</b>	2

**Applies to:** Block-level elements

**Inherited:** Yes

**Computed value:** As specified

## Dropped from CSS2.1

The following properties appeared in CSS2 but were dropped from CSS2.1 due to a lack of widespread support. They do not have computed value information since computed values were first explicitly defined in CSS2.1.

## Visual Styles

---

### font-size-adjust

The aim of this property is to allow authors to trigger font scaling such that substitute fonts will not look too different from the font the author intended, even if the substitute font has a different x-height. Note that this property does not appear in CSS2.1.

**Values:** <number> | none | inherit

**Initial value:** none

**Applies to:** All elements

**Inherited:** Yes

---

### font-stretch

With this property, the character glyphs in a given font can be made wider or narrower, ideally by selected condensed or expanded faces from the font's family. Note that this property does not appear in CSS2.1.

**Values:** normal | wider | narrower | ultra-condensed | extra-condensed | condensed | semi-condensed | semi-expanded | expanded | extra-expanded | ultra-expanded | inherit

**Initial value:** normal

**Applies to:** All elements

**Inherited:** Yes

---

## marker-offset

This property specifies the distance between the nearest border edge of a marker box and its associated element box.

**Values:** <length> | auto | inherit

**Initial value:** auto

**Applies to:** Elements with a display value of marker

**Inherited:** No

**Note:** This property is obsolete as of CSS2.1 and will likely not reappear in CSS3, and the same is true for the display value of marker; as of this writing, it appears that other mechanisms will be used to achieve these effects

---

## text-shadow

This permits the assignment of one or more “shadows” to the text in an element. The first two length values in a shadow definition set horizontal and vertical offsets, respectively, from the element’s text. The third length defines a blurring radius. Note that this property does not appear in CSS2.1.

**Values:** none | [<color> || <length> <length> <length>? ,]\* [<color> || <length> <length> <length>? ] | inherit

**Initial value:** none

**Applies to:** All elements

**Inherited:** No

## Paged Media

---

### marks

This property defines whether “cross marks” (otherwise known as register marks or registration marks) should be placed outside the content area but within the printable area of the canvas. The exact placement and rendering of the marks is not defined. Note that this value does not appear in CSS2.1.

**Values:** [ crop || cross ] | none | inherit

**Initial value:** none

**Applies to:** Page context

**Inherited:** N/A

---

## page

This property, in conjunction with `size`, specifies a particular page type to be used in the printing of an element. Note that this property does not appear in CSS2.1.

**Values:** <identifier> | inherit

**Initial value:** auto

**Applies to:** Block-level elements

**Inherited:** Yes

---

## size

With this property, authors can declare the size and orientation of the page box used in the printing of an element. It can be used in conjunction with `page`, although this is not always necessary. Note that this property does not appear in CSS2.1.

**Values:** <length>{1,2} | auto | portrait | landscape | inherit

**Initial value:** auto

**Applies to:** The page area

**Inherited:** No

# Aural Styles

---

## azimuth

This property sets the angle along the horizontal plane (otherwise known as the horizon) from which a sound should seem to emanate. This is used in conjunction with `elevation` to place the sound at a point on a hypothetical sphere with the user at its center.

**Values:** <angle> | [[ left-side | far-left | left | center-left | center | center-right | right | far-right | right-side ] || behind ] | leftwards | rightwards | inherit

**Initial value:** center

**Applies to:** All elements

**Inherited:** Yes

**Computed value:** Normalized angle

---

## cue

This is a shorthand property that allows an author to define cues that precede and follow the audio rendering of an element's content. A “cue” is something like an auditory icon.

**Values:** [ <cue-before> || <cue-after> ] | inherit  
**Initial value:** none  
**Applies to:** All elements  
**Inherited:** No  
**Computed value:** See individual properties (cue-before, etc.)

---

## cue-after

This property allows an author to define a cue that follows the audio rendering of an element's content.

**Values:** <uri> | none | inherit  
**Initial value:** none  
**Applies to:** All elements  
**Inherited:** No  
**Computed value:** For <uri> values, the absolute URI; otherwise, none

---

## cue-before

This property allows an author to define a cue that precedes the audio rendering of an element's content.

**Values:** <uri> | none | inherit  
**Initial value:** none  
**Applies to:** All elements  
**Inherited:** No  
**Computed value:** For <uri> values, the absolute URI; otherwise, none

---

## elevation

This property sets the angle above or below the horizontal plane (otherwise known as the horizon) from which a sound should seem to emanate. This is used in conjunction with azimuth to place the sound at a point on a hypothetical sphere with the user at its center.

**Values:** <angle> | below | level | above | higher | lower | inherit  
**Initial value:** level  
**Applies to:** All elements  
**Inherited:** Yes  
**Computed value:** Normalized angle

---

---

## pause

This is a shorthand property that allows an author to define pauses that precede and follow the audio rendering of an element’s content. A “pause” is an interval in which no content is audibly rendered, although background sounds may still be audible.

**Values:** `[[<time> | <percentage>]{1,2} ] | inherit`

**Initial value:** `0`

**Applies to:** All elements

**Inherited:** No

**Computed value:** See individual properties (pause-before, etc.)

---

## pause-after

This property allows an author to define the length of a pause that follows the audio rendering of an element’s content.

**Values:** `<time> | <percentage> | inherit`

**Initial value:** `0`

**Applies to:** All elements

**Inherited:** No

**Computed value:** The absolute time value

---

## pause-before

This property allows an author to define the length of a pause that precedes the audio rendering of an element’s content.

**Values:** `<time> | <percentage> | inherit`

**Initial values:** `0`

**Applies to:** All elements

**Inherited:** No

**Computed value:** The absolute time value

---

## pitch

Specifies the average pitch (frequency) of the speaking voice used to audibly render the element’s content. The average pitch of a voice will depend greatly on the voice family.



**Values:** <frequency> | x-low | low | medium | high | x-high | inherit  
**Initial value:** medium  
**Applies to:** All elements  
**Inherited:** Yes  
**Computed value:** The absolute frequency value

---

## pitch-range

This property specifies the variation in average pitch used by the speaking voice, while audibly rendering the element’s content. The higher the variation, the more “animated” the voice will sound.

**Values:** <number> | inherit  
**Initial value:** 50  
**Applies to:** All elements  
**Inherited:** Yes  
**Computed value:** As specified

---

## play-during

This provides a sound to be played “in the background” while the element’s contents are audibly rendered. The sound can be mixed with other background sounds (set using play-during on an ancestor element), or it can replace other sounds for the duration of the element’s audio rendering.

**Values:** <uri> | [mix || repeat]? | auto | none | inherit  
**Initial value:** auto  
**Applies to:** All elements  
**Inherited:** No  
**Computed value:** For <uri> values, the absolute URI; otherwise, as specified

---

## richness

This property sets the “brightness” of the speaking voice used when audibly rendering the element’s content. The brighter the voice, the more it will “carry.”

**Values:** <number> | inherit  
**Initial value:** 50

---

**Applies to:** All elements

**Inherited:** yes

**Computed value:** As specified

---

## speak

This determines how an element's contents will be audibly rendered, or indeed if they will be rendered at all. The value `spell-out` is typically used for acronyms and abbreviations, such as W3C or CSS. If the value is `none`, the element is skipped (it takes no time to be audibly rendered).

**Values:** `normal` | `none` | `spell-out` | `inherit`

**Initial value:** `normal`

**Applies to:** All elements

**Inherited:** Yes

**Computed value:** As specified

---

## speak-header

This specifies whether the content of table headers is spoken before every cell associated with those headers, or only when the header associated with a cell is different from the header associated with the previously rendered cell.

**Values:** `once` | `always` | `inherit`

**Initial value:** `once`

**Applies to:** Elements containing table header information

**Inherited:** Yes

**Computed value:** As specified

---

## speak-numeral

This property determines how numbers are spoken during audible rendering.

**Values:** `digits` | `continuous` | `inherit`

**Initial value:** `continuous`

**Applies to:** All elements

**Inherited:** Yes  
**Computed value:** As specified

---

## speak-punctuation

This property determines how punctuation is spoken during audible rendering. The value code causes punctuation symbols to be rendered literally.

**Values:** code | none | inherit  
**Initial value:** none  
**Applies to:** All elements  
**Inherited:** Yes  
**Computed value:** As specified

---

## speech-rate

This sets the average rate at which words are spoken when an element's content is audibly rendered.

**Values:** <number> | x-slow | slow | medium | fast | x-fast | faster | slower | inherit  
**Initial value:** medium  
**Applies to:** All elements  
**Inherited:** Yes  
**Computed value:** An absolute number

---

## stress

This property affects the height of peaks in the intonation of a speaking voice. These peaks are in turn generated by stress marks within a language.

**Values:** <number> | inherit  
**Initial value:** 50  
**Applies to:** All elements  
**Inherited:** Yes  
**Computed value:** As specified

---

## voice-family

This property is used to define a list of voice families that can be used in the audio rendering of an element's content, and is comparable to font-family. The permitted generic voices are male, female, and child.

**Values:** [[<specific-voice> | <generic-voice> ],]\* [<specific-voice> | <generic-voice> ] | inherit

**Initial value:** User agent-dependent

**Applies to:** All elements

**Inherited:** Yes

**Computed value:** As specified

---

## volume

This sets the median volume level for the waveform of the audibly rendered content. Thus, a waveform with large peaks and valleys may go well above or below the volume level set with this property. Note that 0 is not the same as silent.

**Values:** <number> | <percentage> | silent | x-soft | soft | medium | loud | x-loud | inherit

**Initial value:** medium

**Applies to:** All elements

**Inherited:** Yes

**Computed value:** An absolute number