|  |  |  |  |
| --- | --- | --- | --- |
| @font-face | font-family | src |  font- {stretch, style, weight} | unicode-range | | Use non-default fonts |
| @keyframes name | from | to | *%* | | Define animation |
| @media [not|only *mediatype*] and (*expressions*) | mediatype = all, print, screen, speech  expr = orientation, resolution, viewport/device width/height | | Check media properties  , = logical OR between = XOR |
| @charset | *charset* | | Define stylesheet character encoding  Must be the first rule |
| @font-feature-values *name* | { <declaration-list> } | | Defines common name in font-variant-alternate for OpenType |
| @import | url() *media-queries*  “*file*” *media-queries* | | Import Stylesheet |
| attr() | *Attribute* | | Return *attribute* value |
| calc() | *expression* | | Calculate *expression* |
| counter() | *name*, [*counterstyle*] | | Returns *name* value as a string |
| var() | --*name*, [*fallback* *val*] | | Use a variable’s value |
| min() | *value,* … | | Selects minimum value |
| max() | *value*, … | | Selects maximum value |
| cubic-bezier() | *x1*, *y1*, *x2*, *y2* | | Define a cubic-bezier curve |
| rgb() | n, n, n | | Define color using RGB model |
| rgba() | n, n, n, 0-1 | | Define color using RGBA model |
| hsl() | *degree*, *n*%, *n*% | | Define color with HSL model |
| hsla() | *degree*, *n*%, *n*%, 0-1 | | Define color with HSLA model |
| conic-gradient()  repeating- | [from *angle* deg]  [at *position* %], *color* [*degree*], … | | Transition around center point |
| linear-gradient() | [to *direction*], *color* [*stop*%], … | | Transition linearly |
| repeating- | [*angle*] OR  [to *direction*], *color* [*stop*%]*, …* | | Repeat linear gradients |
| radial-gradient()  repeating- | [*shape*, *size*] [at *position*],  *start-color* [*%*], … *stop-color* [*%*] | | Transition from center  shape=ellipse, circle; size=  farthest-corner, closest-side, closest-corner, farthest-side |
| \* | Select all elements | | | |
| .*class* | Elements with *class* | | | |
| .*class1*.*class2* | Elements with *class1* and *class2* | | | |
| .*class1* .*class2* | *class2* element inside *class1* element | | | |
| #*id* | Element with *id* | | | |
| *elem* | HTML element | | | |
| *elem*.*class* | Element with *class* | | | |
| *elem1*, *elem2* | Element that is *elem1* or *elem2* | | | |
| *elem1* *elem2* | *elem2* inside *elem1* | | | |
| *elem1* > *elem2* | *elem2* with *elem1* as direct parent | | | |
| *elem1* + *elem2* | First *elem2* after *elem1* | | | |
| *elem1* ~ *elem2* | All *elem2* after elem1 | | | |
| [attr] | HTML elements with attribute defined | | | |
| [attr=*value*] | Attribute equals *value* | | | |
| [attr|=*value*] | Attribute equals or begins with *value* | | | |
| [attr^=*value*] | Attribute begins with *value* | | | |
| [attr$=*value*] | Attribute ends with *value* | | | |
| [attr\*=*value*] | Attribute contains substring *value* | | | |
| [attr~=*value*] | Attribute contains word *value* | | | |
| :link | Unvisited link | | | |
| :active | Active link | | | |
| :visited | Visited link | | | |
| :hover | Hovered element (must be after :link and :visited) | | | |
| :target | Targeted link with # | | | |
| :checked | Checked <*radio* *checkbox*> <option> | | | |
| :default | Default <button> <*radio checkbox*> <option> | | | |
| :indeterminate | Indeterminate <*radio checkbox*> <progress> | | | |
| :in-range | <input> within min and max attributes | | | |
| :out-of-range | <input> outside min and max attributes | | | |
| :focus | Focused form element | | | |
| :required | Required form element | | | |
| :optional | Optional form element | | | |
| :valid | Valid form elements | | | |
| :invalid | Invalid form element | | | |
| :enabled | Enabled form element | | | |
| :disabled | Disabled form element | | | |
| :read-only | Read-only form element | | | |
| :read-write | Read-write, enabled form element | | | |
| :empty | No children or content | | | |
| :only-child | Only child of a parent | | | |
| :first-child | First child of a parent | | | |
| :nth-child(*n*) | nth child of a parent | | | |
| :nth-last-child(*n*) | nth to last child of a parent | | | |
| :last-child | Last child of parent | | | |
| :only-of-type | Only element of a parent | | | |
| :first-of-type | First element of a parent | | | |
| :nth-of-type(*n*) | nth element of a parent | | | |
| :nth-last-of-type(*n*) | nth to last element of a parent | | | |
| :last-of-type | Last element of a parent | | | |
| :fullscreen | Full-screened element | | | |
| :lang(*language*) | Element where lang=“*language*” | | | |
| :not(*selector*) | Element != *selector* | | | |
| :root | Document’s root element (<html>) | | | |
| ::after | Insert after element’s contents | | | |
| ::before | Insert before element’s contents | | | |
| ::first-letter | First letter of content | | | |
| ::first-line | First line of content | | | |
| ::marker | List item marker | | | |
| ::placeholder | Form input’s placeholder | | | |
| ::selection | Selected element | | | |
| /\* \*/ | | CSS Comment | |
| !important | | Override all previous stylings | |
| display | | Element Display | |
| inline | block | contents | flex | grid | inline-{block, flex, grid, table} | list-item | run-in | none | table | table-{caption, {column, row, header, footer}-group, row, cell, column, row} | | | |
| visibility | | Element Visibility | |
| visible | hidden | collapse | | | |
| opacity | | Element Opacity | |
| *number* | | | |
| filter | | Element Filter (space-separated) | |
| none | blur(px) | brightness(%) | contrast(%) | drop-shadow(*h-shadow v-shadow blur spread color*) | grayscale(%) | hue-rotate(deg) | invert(%) | opacity(%) | saturate(%) | sepia(%) | url() | | | |
| Rotate | | Element Rotation | |
| [x, y, z] {deg, rad, turn} | *vector* *angle* (*vector* = *n n n*) | | | |
| Scale | | Element Scaling  \*up to 3 values | |
| *n* | *%* | | | |
| translate | | Element Position Translation  \*up to 3 values (x, y, z)  \*for z-axis, need perspective: *length* | |
| *length* | *%* | | | |
| position | | Element Position | |
| static | relative | fixed | absolute | sticky | | | |
| top  right  bottom  left | | Element Placement  \*relative: move top edge around normal position  \*absolute/fixed: set top edge around nearest ancestor  \*sticky: relative when in viewport, fixed when outside | |
| auto | {*length*, *%*} | | | |
| inset | | Distance between self and parent | |
| auto | top right bottom left | | | |
| -block | | Inset in block direction | |
| auto | -start -end | | | |
| -start | | Block Start | |
| auto | *length* | *%* | | | |
| -end | | Block End | |
| auto | *length* | *%* | | | |
| -inline | | Inset in inline direction | |
| auto | -start -end | | | |
| -start | | Inline Start | |
| auto | *length* | *%* | | | |
| -end | | Inline End | |
| auto | *length* | *%* | | | |
| z-index | | Element Stack Position (needs position or display: flex) | |
| auto | *number* | | | |
| clip-path | | Element Content Clip | |
| none| *clip-source* | *basic-shape* | margin-box | border-box | padding-box | content-box |  fill-box | stroke-box | view-box | | | |
| overflow | | Element Content Overflow (needs height)  \*specify axis: -x, -y | |
| visible | hidden | scroll | auto | | | |
| -wrap | | Element Content Break Around Words | |
| normal | anywhere | break-word | | | |
| -anchor | | Scroll Anchoring Toggle | |
| auto | none | | | |
| float | | Element Floating Position (no effect: display: absolute) | |
| none | left | right | | | |
| clear | | Element Flow After Float | |
| none | left | right | both | | | |
| box-sizing | | Element Sizing Calculation Algorithm | |
| content-box | border-box | | | |
| box-shadow | | Element Shadow | |
| none | *h-offset* *v-offset* [*blur* *spread* *color*] | | | |
| box-reflect | | Element Reflection | |
| none | below | above | left | right [offset] [gradient] | | | |
| box-decoration | | Element Box Behavior when Fragmented | |
| slice | clone | | | |
| flex | | Flexbox Item Flexible Length | |
| -grow -shrink -basis | auto (1 1 auto) | none (0 0 auto) | | | |
| -grow | | Item Growth Relative to Other Items | |
| *number* | | | |
| -shrink | | Item Shrink Relative to Other Items | |
| *number* | | | |
| -basis | | Item Initial Length | |
| auto | *number* | | | |
| -flow | | Item Content Flow | |
| flex-direction flex-wrap | | | |
| -direction | | Item Content Direction | |
| row | row-reverse | column | column-reverse | | | |
| -wrap | | Item Content Break | |
| nowrap | wrap | wrap-reverse | | | |
| order | | Item Content Order | |
| *number* | | | |
| justify-content | | Inline Horizontal Alignment | |
| flex-start | flex-end | center | space-between | space-around | space-evenly | | | |
| align-items | | Inline Vertical Alignment | |
| normal | stretch | center | flex-start | flex-end | start | end | baseline | | | |
| align-content | | Flex Lines Alignment | |
| stretch | center | flex-start | flex-end | space-between | space-around | space-evenly | | | |
| align-self | | Block Direction Alignment (overrides align-items) | |
| auto | stretch | center | flex-start | flex-end | baseline | | | |
| grid | | Element Grid Display | |
| none | -template-rows / -template-columns | -template-areas |  -template-rows / [-auto-flow] -auto-columns |  [-auto-flow] -auto-rows / -template-columns | | | |
| -template | | Grid Template | |
| none | -template-rows / -template-columns | -template-areas | | | |
| -template-rows | | Grid Template for Rows | |
| none | auto | min-content | max-content | *length* | | | |
| -template-columns | | Grid Template for Columns | |
| none | auto | min-content | max-content | *length* | | | |
| -template-areas | | Grid Template Using Named Areas | |
| none | ‘*itemname*’ \*. refers to unnamed areas | | | |
| -auto-flow | | Auto-placed Item Insertion | |
| row | column | dense | row dense | column dense | | | |
| -auto-rows | | Default Size of Grid Rows | |
| auto | min-content | max-content | | | |
| -auto-columns | | Default Size of Grid Columns | |
| auto | *length* | *%* | min-content | max-content | minmax(min.max) | fit-content() | | | |
| -area | | Item Size and Location in Grid | |
| -row-start / -column-start / -row-end / column-end | *itemname* | | | |
| -row-start | | Row Grid Item Begins | |
| auto | row-line | | | |
| -column-start | | Column Grid Item Begins | |
| auto | span *n* | column-line | | | |
| -row-end | | Row Grid Item Ends | |
| auto | span *n* | row-line | | | |
| -column-end | | Column Item Ends | |
| auto | span *n* | column-line | | | |
| -row | | Item Size and Row Location | |
| -row-start / -row-end | | | |
| -column | | Item Size and Column Location | |
| -column-start / -column-end | | | |
| Justify-items | | Grid Item Inline Alignment from Grid Container | |
| legacy | normal | stretch | | | | |
| Justify-content | | Grid Item Inline Alignment with Unused Grid Space | |
| flex-start | flex-end | center | space-between | space-around | space-evenly | | | |
| Justify-self | | Grid Item Inline Alignment withing Grid Cell | |
| auto | normal | stretch | start | left | center | end | right | safe | unsafe | *baseline-alignment* | | | |
| Place-content | | Content Placement in Grid/flexbox | |
| Align-content justify-content | | | |
| Place-items | | Item Placement in Grid/Flexbox | |
| align-items justify-content | | | |
| Place-self | | Unused Space Alignment in Grid/Flexbox | |
| Align-self justify-self | | | |
| background | | Element Background | |
| -color -image -position/-size -repeat -origin -clip -attachment | | | |
| -color | | Color | |
| *color* | transparent | | | |
| -image | | Image | |
| url(“*url*, …”) | none | gradient() | | | |
| -position | | Image Start Position | |
| {top, center, bottom} {left, center, right} | {x*%*, *number*} {y*%*, *number*} | | | |
| -x | | Image X-Coord Start | |
| top-bottom left-right | x%|unit | {left, center, right} *offset* | | | |
| -y | | Image Y-Coord Start | |
| top-bottom left-right | y%|unit | {top, center, bottom} *offset* | | | |
| -size | | Image Size | |
| auto | cover | contain | *length* | *percent* | | | |
| -repeat | | Image Repeat | |
| repeat-x | repeat-y | no-repeat | space | round | | | |
| -origin | | Image Origin Position | |
| border-box | padding-box | content-box | | | |
| -clip | | Extend within Element | |
| border-box | padding-box | content-box | | | |
| -attachment | | Image Follows Scroll | |
| scroll | fixed | local | | | |
| -blend-mode | | Layer Blend | |
| normal | multiply | screen | overlay | darken | lighten | color-dodge | saturation | color | luminosity | | | |
| Isolation | | Control if new stacking content must be created | |
| auto |isolate | | | |
| Mix-blend-mode | | Element Content Blend with Direct Parent Background | |
| normal | multiply | screen | overlay | darken | lighten | color-dodge | color-burn | difference | exclusion | hue | saturation | color | luminosity | | | |
| outline | | Element Background Border | |
| -width -style -color | | | |
| -width | | Thickness | |
| *length* | thin | medium | thick | | | |
| -style | | Pattern | |
| none | hidden | dotted | dashed | solid | double | groove | ridge | inset | outset | | | |
| -color | | Color | |
| *color* | invert | | | |
| -offset | | Space Between Outline and Border | |
| *length* | | | |
| margin | | Element Outer Space (collapse = higher vertical)  \*specify edge: -top, -right, -bottom, -left | |
| *length* | *%* | auto | | | |
| -block | | Space in Block Direction  \*specify direction: -start, -end | |
| *length* | *%* | auto | | | |
| -inline | | Space in Inline Direction  \*specify direction: -start, -end | |
| *length* | *%* | auto | | | |
| border | | Element Border  \*specify side: -top, -right, -bottom, -left  \*specify group: -block, -inline | |
| -width -style -color | | | |
| -width | | Thickness | |
| *length* | thin | medium | thick | | | |
| -style | | Pattern | |
| none | hidden | dotted | dashed | solid | double | groove | ridge | inset | outset | | | |
| -color | | Color | |
| *color* | transparent | | | |
| -radius | | Corner Radius | |
| {*length*, *%*}| *%* / *length* | *%* | | | |
| -block | | Border in Block Direction (top to bottom)  \*specify position: -start, -end | |
| width style color | | | |
| -inline | | Border in Inline Direction (left to right)  \*specify position: -start, -end | |
| width style color | | | |
| -image | | Image | |
| source slice width outset repeat | | | |
| -source | | Image Source | |
| none | url() | | | |
| -slice | | Image Slice (four corners, four edges, middle) | |
| *number* | *%* | fill | | | |
| -width | | Border Width to Fill | |
| *number* | *%* | auto | | | |
| -outset | | Image Extension from Border | |
| *length* | *number* | | | |
| -repeat | | Image Repeat | |
| stretch | repeat | round | space | | | |
| padding | | Element Inner Space  \*specify edge: -top, -right, -bottom, -left | |
| *length* | *%* | | | |
| -block | | Space in Block Direction  \*specify direction: -start, -end | |
| *length* | *%* | auto | | | |
| -inline | | Space in Inline Direction  \*specify direction: -start, -end | |
| *length* | *%* | auto | | | |
| height | | Content Height  \*specify bound: min-, max- | |
| *length* | *%* | auto | none (for max) | | | |
| width | | Content Width  \*specify bound: min-, max- | |
| *length* | *%* | auto | none (for max) | | | |
| aspect-ratio | | Width/Height Ratio  \*if set with width, height will follow | |
| *n* / *n* | | | |
| block-size | | Element Size in Block Direction  \*specify min-, max- | |
| auto | *length* | *%* | | | |
| inline-size | | Element Size in Inline Direction  \*specify min-, max- | |
| auto | *length* | *%* | | | |
| color | | Text Color | |
| *color* | | | |
| vertical-align | | Text Vertical Alignment | |
| *length* | *%* | baseline | sub | super | top | text-top | middle | bottom | text-bottom | | | |
| text-align | | Text Horizontal alignment | |
| left | center | right | justify | | | |
| -last | | Text Horizontal Alignment of Last Line | |
| left | center | right | justify | auto | start | end | | | |
| direction | | Text Direction | |
| ltr | rtl | | | |
| unicode-bidi | | Override Text to support multiple languages in same document | |
| normal | embed | bidi-override | isolate | isolate-override | plaintext | | | |
| Writing-mode | | Lines laid out horizontally or vertically | |
| horizontal-tb | vertical-rl | vertical-lr | | | |
| text-decoration | | Text Line Decoration | |
| -line -color -style -thickness | | | |
| -line | | Decoration Placement | |
| none | underline | overline | line-through | | | |
| -color | | Decoration Color | |
| *color* | | | |
| -style | | Decoration Pattern | |
| solid | double | dotted | dashed | wavy | | | |
| -thickness | | Decoration Height | |
| auto | from-font | *length* | *%* | | | |
| text-transform | | Text Capitalization | |
| none | capitalization | uppercase | lowercase | | | |
| text-indent | | Text First Line Indentation | |
| *length* | *%* | | | |
| text-shadow | | Text Shadow (comma-separated) | |
| *h-shadow v-shadow blur-radius* color | none | | | |
| Text-justify | | Justification Method | |
| auto | inter-word | inter-character | none | | | |
| Text-orientation | | Character Orientation (writing-mode: vertical) | |
| mixed | upright | sideways | sideways-right | use-glyph-orientation | | | |
| Text-overflow | | Overflow Signal (white-space: nowrap; overflow: hidden) | |
| clip | ellipsis | *string* | | | |
| text-underline-position | | Underline position of text-decoration  \*up to 2 values | |
| auto | under | left | right | | | |
| Text-emphasis | | Text Emphasis | |
| -color -style | | | |
| -color | | Emphasis Color | |
| *color* | | | |
| -style | | Emphasis Style | |
| filled | open | dot | circle | double-circle | triangle | sesame | *string* | none | | | |
| text-combine-upright | | Multiple Character Combination into Single Space Control | |
| none | all | digits [4] | | | |
| line-height | | Text Line Space | |
| *number* | *length* | *%* | normal | | | |
| white-space | | Text Space at Line End | |
| normal | nowrap | pre | pre-line | pre-wrap | | | |
| letter-spacing | | Text Space Between Letters | |
| *length* | normal | | | |
| word-spacing | | Text Space Between Words | |
| *length* | normal | | | |
| Line-break | | Line Break Definition | |
| auto | loose | normal | strict | anywhere | | | |
| Word-break | | Word Break End-of-Line Toggle | |
| normal | break-all | keep-all | break-word | | | |
| Word-wrap | | Allow Word Fragmentation | |
| normal | break-word | | | |
| User-select | | Text Selection Toggle | |
| auto | none | text | all | | | |
| font | | Element Font | |
| -style -variant -weight -size -family| caption | icon | menu | message-box | small-caption | status-bar | | | |
| -style | | Font Angle | |
| normal | italic | oblique | | | |
| -variant | | Font Variation | |
| normal | small-caps | | | |
| -weight | | Font Thickness | |
| normal | bold | bolder | lighter | *x*00 | | | |
| -size | | Font Size  \*set % in body, then em for rest | |
| xx-small | x-small | small | smaller | medium | larger | large | x-large |  xx-large | *length* | *%* | | | |
| -family | | Font Family (comma-separated)  \*Sans-Serif (Arial, Verdana, Tahoma, Trebuchet MS)  \*Serif (Times New Roman, Georgia, Garamond)  \*Monospace (Courier New), Cursive (Brush Script MT) | |
| *family-name* | *generic-family* | | | |
| -feature-settings | | OpenType Font Advanced Typographic Feature Control | |
| normal | *feature-value* (format: string [1|0|on|off]) | | | |
| -kerning | | Letter Spacing | |
| auto | normal | none | | | |
| -language-override | | Override Glyphs defined in HTML lang attribute | |
| normal | *STR* | | | |
| -size-adjust | | Preserve font-size for fallback fonts | |
| none | *number* | | | |
| -stretch | | Face Type | |
| normal | ultra-condensed | extra-condensed | condensed | semi-condensed | semi-expanded | expanded | extra-expanded | ultra-expanded | | | |
| -synthesis | | Logographic Typeface Feature Toggle | |
| -weight -style -small-caps | | | |
| -weight | | Synthesis Weight | |
| auto | none | | | |
| -style | | Synthesis Style | |
| auto | none | | | |
| -small-caps | | Synthesis Small Caps | |
| auto | none | | | |
| -variant-alternates | | Alternate Glyphs Control Defined in @font-feature-values  \*space-delimited | |
| normal | historical-forms | stylistic() | styleset() | character-variant | swash() | ornaments() | annotation() | | | |
| -variant-caps | | Alt Glyphs Capital Letters | |
| normal | small-caps | all-small-caps | petite-caps | all-petite-caps | unicase | titling-caps | | | |
| -variant-east-asian | | Alt glyphs for East-Asian Scripts  \*space-delimited | |
| normal | ruby | jis78 | jis83 | jis90 | jis04 | simplified | traditional | full-width | proportional-width | | | |
| -variant-ligatures | | Ligature and Contextual Form Control | |
| normal | none | common-ligatures | no-common-ligatures | discretionary-ligatures | no-discretionary-ligatures | historical-ligatures | no-historical-ligaturtes | contextual | no-contextual | | | | |
| -numeric | | Alt glyphs for numbers, fractions, and ordinal markers  \*space delimited | |
| normal | ordinal | slashed-zero | lining-nums | oldstyle-nums | proportional-nums | tabular-nums | diagonal-fractions | stacked-fractions | | | |
| -position | | Alt glyphs smaller size position control | |
| normal | sub | super | | | |
| -settings | | Alt glyph low-level control | |
| normal | ‘*STRG*’ *number* | | | |
| content | | Generate content with ::before and ::after | |
| normal | none | counter(*id*) | attr(*attr*) | url(*url*) | *string* | {no-}{close, open}-quote | | | |
| resize | | Element Resize Toggle | |
| none | horizontal | vertical | both | | | |
| list-style | | Element List | |
| -type -position -image | | | |
| -type | | Bullet Type | |
| disc | circle | square | decimal -leading-zero | lower- upper-alpha greek latin roman | hiragana -iroha | katakana -iroha | armenian | cjk-ideographic | georgian | hebrew | none | | | |
| -position | | Bullet Position | |
| inside | outside | | | |
| -image | | Bullet Image | |
| url() | none | | | |
| Hanging-punctuation | | Punctuation can be placed outside linebox | |
| none | first | last | allow-end | force-end | | | |
| Hyphens | | Split-Word Layout in Paragraphs | |
| manual | auto | none | | | |
| Quotes | | Quotation Type | |
| none | “str” x4 | | | |
| counter-reset | | Generate Counter | |
| none | *id* [*number*] | | | |
| counter-increment | | Increment Counter | |
| none | *id* [*number*] | | | |
| gap | | Grid/Flexbox Gap | |
| row-gap column-gap | | | |
| Row-gap | | Gap between grid rows | |
| normal | *length* | | | |
| columns | | Element Columns | |
| auto | column-width column-count | | | |
| column-width | | Column Minimum Width | |
| auto | *length* | | | |
| column-count | | Number of Columns | |
| auto | *n* | | | |
| column-span | | Number of Columns to Span | |
| none | all | | | |
| column-fill | | Column Content Balance | |
| balance | auto | | | |
| column-gap | | Gap Between Columns | |
| normal | *length* | | | |
| column-rule | | Line Between Columns | |
| -width -style -color | | | |
| -width | | Rule Thickness | |
| medium | thin | thick | *length* | | | |
| -style | | Rule Pattern | |
| none | hidden | dotted | dashed | solid | double | groove | ridge | inset | outset | | | |
| -color | | Rule Color | |
| *color* | | | |
| border-collapse | | Table Border Collapse | |
| separate | collapse | | | |
| border-spacing | | Table Border Margin (no effect if border-collapse: collapse) | |
| *h-length v-length* | | | |
| table-layout | | Table Layout Algorithm | |
| auto | fixed | | | |
| caption-side | | Table Caption Placement | |
| top | bottom | | | |
| empty-cells | | Empty Cell Display (no effect if border-collapse: collapse) | |
| show | hide | | | |
| transform | | Element Transformation | |
| none | translate(x, y) | rotate(deg) | scale(x, y) | skew(x, y) | matrix(n, n, n, n, n, n) | translate3d(x, y, z) | scale3d(x, y, z) | rotate3d(x, y, z, deg) | perspective | matrix3d(n x16) | | | |
| -origin | | Transformed Element Position | |
| {left, right, center, *length*, *%*} {top, center, bottom, *length*, *%*} {*length*} | | | |
| -style | | Nested Element Rendering | |
| flat | preserve-3d | | | |
| perspective | | Child Element’s 3D Distance Perspective | |
| none | *length* | | | |
| -origin | | User 3D Viewing Position | |
| {left, center, right, *length*, *%*} (top, center, bottom, *length*, *%*) | | | |
| backface-visibility | | Back Face Visibility | |
| visible | hidden | | | |
| transition | | Element Transition Effect | |
| -property -duration -timing-function -delay | | | |
| -property | | Transitioned Property | |
| all | none | *property* | | | |
| -duration | | Transition Length | |
| time{ms, s} | | | |
| -timing-function | | Transition Speed Curve | |
| ease | ease {in, out, in-out} | step-{start, end} | steps(n, start|end) | linear  cubic-bezier(n, n, n, n) | | | |
| -delay | | Transition Start Delay | |
| *time*{ms, s} | | | |
| animation | | Element Animation | |
| -name -duration -timing-function -delay -iteration-count -direction -fill-mode -play-state | | | |
| -name | | Animation Definition Name | |
| none | *keyframename* | | | |
| -duration | | Animation Length | |
| *time*{ms, s} | | | |
| -timing-function | | Animation Speed Curve | |
| ease | ease {in, out, in-out} | step-{start, end} | steps(n, start|end) | linear  cubic-bezier(n, n, n, n) | | | |
| -delay | | Animation Start Delay | |
| *time*{ms, s} | | | |
| -iteration-count | | Animation Cycles | |
| *n* | infinite | | | |
| -direction | | Animation Playback Direction | |
| normal | reverse | alternate | alternate-reverse | | | |
| -fill-mode | | Animation Idle Style | |
| none | forwards | backwards | both | | | |
| -play-state | | Animation Playback State | |
| running | paused | | | |
| offset | | Animation Offset | |
| auto | -anchor -distance -path -rotate | | | |
| -anchor | | Element Fixed Point that follows Path | |
| auto | {left, center, right} {top, center, bottom} | *xpos ypos* | *x% y%* | | | |
| -distance | | Starting Position on Path | |
| 0 | auto | *length* | *%* | | | |
| -path | | Path to Follow | |
| none | path() | | | |
| -rotate | | Element Rotation on Path | |
| auto | *angle* | auto *angle* | reverse | | | |
| Image-rendering | | Algorithm for Image Scaling | |
| auto | smooth | high-quality | crisp-edges | pixelated | | | |
| Paint-order | | SVG Painting Order  \*one, two, or three values | |
| normal | *values* (stroke, fill) | | | |
| object-fit | | Resize Media in Container (<img> and <video>) | |
| fill | contain | cover | none | scale-down | | | |
| object-position | | Resized Media Position in Container (1st = x-axis, 2nd = y-axis) | |
| left | center | right | top | bottom | length | % | | | |
| mask | | Element Image Mask | |
| -image -mode -repeat -position -clip -origin -size -composite | | | |
| -image | | Image Mask Layer | |
| none | image | url | | | |
| -mode | | Image Mask Type | |
| match-source | luminance | alpha | | | |
| -repeat | | Image Mask Repetition | |
| repeat | repeat-x | repeat-y | space | round | no-repeat | | | |
| -position | | Image Mask Position | |
| {left, center, right} {top, center, bottom} | % | n | | | |
| -clip | | Mask Area  \*Comma-delimited | |
| border-box | content-box | padding-box | fill-box | stroke-box | view-box | no-clip | | | |
| -origin | | Image Mask Area Position | |
| border-box | content-box | padding-box | margin-box | fill-box | stroke-box | view-box | | | |
| -size | | Image Mask Size | |
| auto | *size* | contain | cover | | | |
| -composite | | Mask Compositing Operation | |
| add | subtract | intersect | exclude | | | |
| backdrop-filter | | Filters Transparent Background | |
| none | blur() | brightness() | contrast() | drop-shadow() | grayscale() | hue-rotate() | invert() | opacity() | sepia() | saturate() | url() | | | |
| accent-color | | Input Accent Color | |
| auto | *color* | | | |
| caret-color | | Input Caret Color | |
| auto | *color* | | | |
| cursor | | Cursor Over Element | |
| auto | default | alias | all-scroll | cell | col-resize | context-menu | copy | crosshair | e-resize | ew-resize | grab | grabbing | help | move | n-resize | ne-resize | news-resize | ns-resize | nw-resize | nwse-resize | no-drop | none | not-allowed | pointer | progress | row-resize | s-resize | se-resize | sw-resize | text | vertical-text | w-resize | wait | zoom-in | zoom-out | url() | | | |
| Pointer-events | | Pointer Interaction Toggle | |
| auto | none | | | |
| Tab-size | | Tab Character Width | |
| *number* | | | |
| all | | Property Reset except unicode-bidi and direction | |
| initial | inherit | unset | | | |
| break-before | | Page-, Column-, Region-Break Occur Before Element | |
| auto | all | always | avoid | avoid-column | avoid-page | avoid-region | column | left | page | recto | region | right | verso | | | |
| break-inside | | Page-, Column-, Region-Break Occur Inside Element | |
| auto | all | always | avoid | avoid-column | avoid-page | avoid-region | column | left | page | recto | region | right | verso | | | |
| break-after | | Page-, Column-, Region-Break Occur After Element | |
| auto | all | always | avoid | avoid-column | avoid-page | avoid-region | column | left | page | recto | region | right | verso | | | |
| Page-break-before | | Page breaks Before Element  \*cant use with empty <div> or position: absolute | |
| auto | always | avoid | left | right | | | |
| Page-break-inside | | Page breaks Inside Element | |
| auto | avoid | | | |
| Page-break-after | | Page breaks After Element | |
| auto | always | avoid | left | right | | | |
| Orphans | | Minimum number of lines left at the bottom of a page/column | |
| *integer* | | | |
| Widows | | Minimum number of lines at top of page/column | |
| *integer* | | | |
| Overscroll-behavior | | Scroll Chaining or Overscroll Affordance Toggle  \*specify -x, -y (shorthand)  \*specify -block, -inline | |
| auto | contain | none | | | |
| Scroll-behavior | | Scroll Animation | |
| auto | smooth | | | |
| Scroll-margin | | Distance Between Snap Position and Container  \*specify -top, -right, -bottom, -left  \*specify -block, -inline; optionally -start, -end | |
| 0 | *length* | | | |
| Scroll-padding | | Distance Between Snap Position and Child Elements  \*specify -top, -right, -bottom, -left  \*specify -block, -inline; optionally -start, -end | |
| 0 | *length* | *%* | | | |
| Scroll-snap-type | | Scroll Snap Behavior | |
| none | x | y | block | inline | both | mandatory | proximity | | | |
| Scroll-snap-align | | Element Snap Position when Scrolling Stops for Children  \*up to 2 values (block inline) | |
| none | start | end | center | | | |
| Scroll-snap-stop | | Scroll Past or Onto Child | |
| normal | always | | | |

RWD

User box-sizing: border-box

12 col = 8.33% in a page

Breakpoints: 600-768-992-1200px

Images and videos: width:100% height:auto or max/min-width

SASS = Syntactically Awesome Stylesheet

CSS Preprocessor based on Ruby that needs to be transpiled

// = inline comment

$var: value = variable declaration -> strings, numbers, colors, Booleans, lists, nulls

!global = overrides variable scope

Nest properties or same-prefix properties like font: {} as shorthand

@import *filename* directive rids of extra HTTP request for CSS import

\_filename = partials, where Sass will not transpile (import without \_)

@mixin *name* {} reusable CSS code throughout website using @include *mixin*

- and \_ are the same

Can take arguments with ($arg, …)

can include default ($arg: val) and call with @include mixin($arg: newVal)

best with vendor prefixes: -ms- and -webkit-

@extend *selector* to share properties from a selector to another

String Functions

quote(str) – adds quotes to strings

unquote(str) – removes quotes from strings

str-index(str, substr) – returns index of first occurrence of substr within str

str-insert(str, ins, idx) – return str with ins inserted at idx

str-length(str) – returns length of str

str-slice(str, start, end) – extract characters from string at indexes

to-lower-case(str) – convert to lower case

to-upper-case(str) – convert to upper case

unique-id() – return unique randomly generated string

Numeric Functions

abs(n) – returns absolute value

round(n) – round to nearest integer

ceil(n) – ceiling function

floor(n) – floor function

comparable(n1, n2) – checks if numbers are comparable

max(n, …) – return highest number

min(n, …) – returns lowest number

percentage(n) – converts to percentage

random() – random number between 0 to 1

random(n) – rand between 1 and n

List Functions – Lists are immutable, starts at 1

append(*list*, *val*, [*separator*]) – appends val to lis; separator: auto, comma, space

join(list1, list2, [separator, bracketed]) – appends list2 into list1; bracketed: auto, true, false

index(list, val) – return idx of val

is-bracketed(list) – check if list has square brackets

list-separator(list) – return separator used as string

length(list) – list length

nth(list, n) – returns nth element in list

set-nth(list, n, value) – set nth element as value

zip(lists) – combines lists into single multidimensional list

Map Functions – Maps are immutable

map-get(map, key) – return val of key in map

map-has-key(map, key) – checks if map has key

map-keys(map) – returns all keys

map-merge(map1, map2) – appends map2 to map1

map-remove(map, keys, …) – removes keys from map

map-values(map) – returns all values in map

Selector Functions

is-superselector(super, sub) – check if super matches all elements sub matches

selector-append(selector, …) – append selectors to first selector

selector-extend(selector, extendee, extender)

selector-nest(selectors) – returns selector with nested list of last selector in other selectors

selector-parse(selector) – returns list of strings contained in selector with same format as parent

selector-replace(selector, original, replacement) – return new selector

selector-unify(selector1, selector2) – return selector with matching elements from both

simple-selector(selectors) – return list of individual selectors

Introspection Functions

call(function, arguments) – calls a functions and returns result

content-exists() – check if mixin was passed a @content block

feature-exists(feature) – checks if feature is supported by current Sass implementation

function-exists(function) – check if function exists

get-function(function, css:false) – returns specified function. If css:true, return plain CSS func

global-variable-exists(var) – check if variable exists

inspect(val) – return value as string

mixin-exists(mixin) – check if mixin exists

type-of(val) – returns type of val: number, string, color, list, map, bool, null, function, arglist

unit(n) – return unit of n

unitless(n) – check if n has unit associated

variable-exist(var) – check if var exists

Color Functions

rgb(…), rgba(…), hsl(…), hsla(…), grayscale(color), complement(color), invert(color, [weight])

red, green, blue, hue, saturation, lightness, alpha, opacity (color)

mix(c1, c2, weight), adjust-hue(c, degrees), rgba(color, alpha)

adjust-color, change-color (color, {red, green, blue, hue, saturation, lightness, alpha})

lighten, darken, saturate, desaturate, opacity, fade-in, transparentize, fade-out (color, amount)