

# CSS Cheat Sheet



# Selectors

<code>*{}</code>	Universal Selector
<code>#id {}</code>	ID Selector
<code>.class {}</code>	Class Selector
<code>h1, h2{}</code>	Type Selector
<code>h1+p {}</code>	Adjacent Sibling Selector
<code>ul &gt; li{}</code>	Child Selector
<code>h1 ~ p {}</code>	General Sibling Selector
<code>p a{}</code>	Descendant Selector
<code>div[att="val"]{}</code>	Attribute Selector

# Units

<code>%</code>	Percentage
<code>cm</code>	Centimeter
<code>in</code>	Inch
<code>mm</code>	Millimeter
<code>pc</code>	Pica (1 pica = 12 points)
<code>pt</code>	Point (1 point = 1/72 inch)
<code>px</code>	Pixel (1 pixel = 1/96 inch)
<code>ch</code>	Width of the "0" glyph in the font size
<code>em</code>	<code>1em</code> = Current font size
<code>ex</code>	X-height of the element's font
<code>gd</code>	Grid defined by 'layout-grid'
<code>rem</code>	Font size of the root element
<code>vh</code>	Viewport's height
<code>vw</code>	Viewport's width
<code>vm</code>	Smaller of viewport's height or width

# Pseudo Selectors

:active	Activated element
:focus	Focused element
:hover	Hovered element
:link	Unvisited link
:disabled	Disabled element
:enabled	Enabled element
:checked	Checked element
:nth-child(n)	N-th sibling
:nth-last-child(n)	N-th sibling from the end
:first-child	First sibling
:last-child	Last sibling
:only-child	Only child
:nth-of-type(n)	N-th sibling of its type
:nth-last-of-type(n)	N-th sibling of its type from end
:last-of-type	Last sibling of its type
:first-of-type	First sibling of its type
:only-of-type	Only child of its type
:empty	Element with no children
:root	Root element
:not(x)	Element not matching 'x'
:target	Target element specified by a URI
::first-letter	Style for the first letter of text
::first-line	Style for the first line of text
::before	Insert content before an element
::after	Insert content after an element

## List Styling

```
// List Style  
list-style-type: disc | circle | square | none;  
  
// List Position  
list-style-type: inside | outside;  
  
// List Image  
list-style-img: url()
```

## Position

```
// Position  
position: static | relative | absolute | fixed | sticky;  
  
// Position Element  
top | right | bottom | left  
  
// Float Element  
float: left | right | none  
  
// Z-index  
z-index: 3 | auto | none  
  
// Clear Floating  
clear: none | left | right | both
```

## Background

```
background-color: #FFF2EB  
  
background-image: url()  
  
background-repeat: repeat-x | repeat-y | repeat | space | round  
| no-repeat  
  
background-attachment: scroll | fixed | local | initial | inherit  
  
background-position: top | right | bottom | left | center
```

## Font Properties

```
font-style: normal | italic | oblique  
  
font-variant: normal | small-caps  
  
font-size: 13px | 0.8rem | 80%  
  
font-weight: normal | bold | bolder | lighter | 100-900  
  
letter-spacing: normal | 4px  
  
line-height: normal | 3rem | 34%  
  
font-family: 'Open sans', sans-serif
```

## Text Properties

```
text-align: left | right | center | justify  
text-transform : capitalise | lowercase | uppercase  
text-indent: 23px  
vertical-align: baseline | 10px | sub | super | top | text-top  
| middle | bottom | text-bottom | initial  
text-align-last: auto | left | right | center | justify  
| start | end | initial | inherit  
text-decoration: none | underline | overline | line-through  
text-justify: auto | inter-word | inter-character | none  
| initial | inherit  
text-overflow: clip | ellipsis | string | initial | inherit  
text-shadow: h-shadow v-shadow | blur-radius color | none  
| initial | inherit
```

## Transition

```
transition-timing-function: ease | linear | ease-in | ease-out  
| ease-in-out | cubic-Bezier(num, num, num, num);  
  
transition-property: none | all  
  
transition-delay: time;  
  
transition-duration: time;  
  
transition: transition-property transition-duration  
transition-timing-function transition-delay;
```

## Animation

```
animation-timing-function: ease | linear | ease-in | ease-out | ease-in-out  
| cubic-Bezier (number, number, number, number)  
  
animation-name: none | IDENT;  
  
animation-duration : time;  
  
animation-delay: time;  
  
animation-iteration-count : inherit | number;  
  
animation-direction: normal | alternate;  
  
rotation: angle rotation-point position;  
  
animation-play-state: running | paused;  
  
// shorthand  
animations: animation-name animation-duration animation-timing-function  
           animation-delay animation-iteration-count animation-direction
```

## Transform

```
// 2D Transform
transform: translate(x, y):           Translate (move) element
transform: rotate(angle):             Rotate element around a specified angle
transform: scale(x, y):               Scale element
transform: skew(x-ang,y-ang):         Skew (slant) element
transform: skewX(angle):              Skew (slant) element along the X-axis
transform: skewY(angle):              Skew (slant) element along the Y-axis

// 3D Transform
transform: translate3d(x, y, z):     Translate (move) element
transform: rotateX(angle):            Rotate element around the X-axis
transform: rotateY(angle):            Rotate element around the Y-axis
transform: rotateZ(angle):            Rotate element around the Z-axis
transform: scale3d(x, y, z):          Scale element
transform: perspective(value):       Set the perspective view
```