

# THE COMPLETE CSS CHEAT SHEET

## TABLE OF CONTENT

TOPIC	PAGE
• Font, Text, Background, Color	2
• The Box Model	3
• Column, animation, Transition	4
• Pseudo-classes and element, units of measurement	5
• Position, Outline, List, Transform	6
• Semantic tags, Tables, Lists & Images	7
• CSS Grid, Flexbox	8

Powered by **TRENIDE TECH**

Definition Reference: w3schools

# FONT | TEXT | BACKGROUND | COLOR

## FONT

The CSS font properties define the font family, boldness, size, and the style of a text

### font-family

**Generic Family - Serif, Sans-serif, Monospace**

-Font families: Times New Roman, Verdana, Courier New

### font-style

**Mostly used to specify italic text.**

-Values: normal, italic, oblique

### font-stretch

**Allows you to make text narrower or wider**

-Values: normal, condensed, expand, ultra-condensed, ultra-expanded, semi-condensed, extra-condensed

### font-size

**Default: 16px.**

- values: px, %, em, rem, vw, small, large, etc.

### font-weight

**Sets how thick chars in text shld b displayed**

-Values: normal, bold, bolder, lighter, 100 - 900

### font-variant

**For displaying caps letters with a smaller text size**

-Values: normal, small-caps, inherit

## TEXT

### text-decoration

**Specifies the decoration added to text**

- Values: underline, overline, line-through

### direction

**Set text direction from left to right or vice versa**

- Values: rtl, ltr, inherit, initial

### text-transform

**Controls the capitalization of text**

- Values: capitalize, uppercase, lowercase, none

### text-indent

**Specifies indentation of the 1st line in a text-block**

- Values: px, em, %, etc

### text-align

**Set the horizontal alignment of texts in a div**

- Values: right, left, justify, center

### text-overflow

**Specifies how overflowed text shld b displayed**

- Values: clip, ellipsis, stringz

### text-shadow

**eg. text-shadow: 2px 2px 8px #FF0000**

- Value order: horizontal, vertical, blur-radius, color

### letter-spacing

**increase/decreases space between chars in a text**

- Value: px, em, etc

## BACKGROUND

**Short hand eg - background: lightblue url("img\_tree.gif") no-repeat fixed center;**

-Order: background-color, background-image, background-repeat, background-attachment, background-position

### background-color

**Specifies the background color to be used**

- Value: color (name, hex, rgba, hsl)

### background-position

**Specifies the position of the background images**

- Values: center(commonly used), left center, etc

### background-attachment

**Specifies mov't of element relative to container**

- Values: scroll, fixed, local

### background-origin

**Specifies the positioning area of the bg images**

- Values: border-box-padding-box, content-box

### text-align

**Specifies 1 or more bg images to be used**

- Values: URLs, seperated with a comma

### background-size

**Specifies the size of the background images**

- Values: cover(common), auto, contain

### background-repeat

**Sets if/how a bg image will be repeated**

- Values: repeat, no-repeat, repeat-x, repeat-y, space

### background-clip

**Specifies the painting area of the bg images**

- Values: border-box-padding-box, content-box

## COLOR

**Common Hex:** White - #ffffff, Black - #000000, Red - #ff0000, Green - #00ff00, Blue - #0000ff....[See the pattern?](#)

**Common rgb:** White - rgb(255,255,255), Black - rgb(0, 0, 0), Red - rgb(255, 0, 0), Green - ? [Same Pattern](#) - ff = 255

# CSS BOX MODEL

It is essentially a box that wraps around every HTML element. It consists of: margins, borders, padding, and the actual content

- Unless Specified, Value is always a length(px, em, rem, etc) or %

## CONTENT

The content of the box, where text and images appear.

### height & Width

Specifies the height & width of content

### max-height & max-width

Cannot be exceeded by height or width value

### float

Positions elements to far left or right

- Values: left, right. Make sure the elements are the only 2 in their div. Float one left & the other right

### display

Specifies if/how an element is displayed

- Values: none, block, inline, inline-block

## BORDER

A border that goes around the padding and content, not visible until given a width

- Value: border-width, border-style, border-color eg. border: 3px, solid, red; ( in that order  
- **BORDER STYLES**: solid, dotted, dashed, double. 3D versions - groove, ridge, inset, outset. none, hidden  
Values can also be properties on their own. eg. border-style: 3px; border-top-color: 8px; border-right-style: solid;

### border-top

Specifies only the top border

- Same values and same arrangement as border

### border-right

Specifies only the right border

- Same values and same arrangement as border

### border-bottom

Specifies only the bottom border

- Same values and same arrangement as border

### border-left

Specifies only the left border

- Same values and same arrangement as border

## MARGIN

Used to create space around elements, outside of any defined borders

- Value: margin-top, margin-right, margin-bottom, margin-left - eg. margin: 5px, 10px, 5px, 10px;

### margin-top

Specifies only the top margin

### margin-right

Specifies only the right margin

### margin-bottom

Specifies only the bottom margin

### margin-left

Specifies only the left margin

### margin-shorthand

When only 2 vals are specified. eg. margin: 5px 10px  
1st val - margin-top & bottom. 2nd val - margin-right & left

### margin - shortest hand

When only 1 value is specified. eg. margin: 5px  
The value represent all margins

## PADDING

Used to generate space around an element's content, inside of any defined borders

- Value: padding-top, padding-right, padding-bottom, padding-left - eg. padding: 5px, 10px, 5px, 10px;

### padding-top

Specifies only the top padding

### padding-right

Specifies only the right padding

### padding-bottom

Specifies only the bottom padding

### padding-left

Specifies only the left padding

### padding - shorthand

When only 2 vals are specified. eg padding: 5px 10px  
1st val - padding-top & bottom. 2nd val - padding-right & left

### padding - shortest hand

When only 1 value is specified. eg. padding: 5px  
The value represent all paddings

# COLUMN | ANIMATION | TRANSITION

## COLUMN

### column-count

**Spec no of columns an element shld b divided into**  
-Values: auto(default), numberm(of columns)

### column-gap

**Specifies the gap between the columns**  
-Values: length(px, em,rem), normal

### column-rule-color

**Specifies the color of the rule between columns**  
-Values: color

### column-rule-width

**Specifies the width of the rule b/n columns**  
-Values: length(px, em,rem), medium, thin, thick

### column-width

**Specifies the column width**  
-Values: auto, length(px, em, rem)

### column-fill

**Specifies how to fill columns, balanced or not**  
- values: balance(default) , auto

### column-rule

**Sets width, style, & color of rule b/n columns**  
-Values: column-rule-width, column-rule-style

### column-rule-style

**Specifies the style of the rule between columns**  
-Values: none, hidden, dotted, dashed, solid, etc

### column-span

**Spec how many colmns a elemnt shld span across**  
-Values: column-rule-width, column-rule-style

### columns

**Shorthand for column-width & column-count**  
-Values: column-width column-count , auto

## ANIMATION

**An animation lets an element gradually change from one style to another.**

- The animation is declared with **@Keyframes**, and the following properties are used to modify it's behavior

### animation

**A shorthand 4 setting all animation properties**  
-Values: animation-name, -duration, -timing-function, -delay, -iteration-count, -direction

### animation-duration

**Spec time animation shld take to complete 1 cycle**  
- Value: time. eg, 4s

### animation-delay

**Specifies a delay for the start of an animation**  
- Value: time. eg, 4s

### animation-direction

**Specifies the direction of play of the animation**  
- Values: normal, reverse, alternate, alternate-reverse

### animation-name

**Specifies the name of the @keyframes animation**  
-Values: text

### animation-timing-function

**Specifies the speed curve of the animation**  
-Values: ease, linear, ease-in, ease-out, ease-in-out

### animation-iteration-count

**Specif no of times an animation shld b played**  
- Value: number

### animation-play-state

**Spec whether animation is running or paused**  
- Values: paused, running

## TRANSITIONS

**Allows you to change property values smoothly, over a given duration.**

- To add transition, at least 2 things must bet specified, the transition-property and transition-duration.

### transition

**A shorthand 4 setting all transition properties**  
-Val: transition-property, -duration, -timing-function, -delay

### transition-duration

**Spec time(millisecond) it takes to complete transition**  
- Value: time. eg 4

### transition-timing function

**Specifies the speed curve of the transition effect**  
- Values: ease, linear, ease-in, ease-out, ease-in-out

### transition-property

**Spec name of CSS propaty the transition effect is for**  
- Value: CSS Property. eg. width, color

### transition-delay

**Spec a delay (in sec) for the transition effect**  
- Value: time eg. 1

### Transition vs Animation

**Transition is a form of animation**

! Except it works on only a start & end value, for more flexibility, use animation

# PSEUDO PROPERTIES | UNITS OF MEASUREMENTS

## PSEUDO-ELEMENTS

### ::first-letter & ::first-line

Adds special styling to 1st letter & line of a text

### ::before & ::after

Adds some content before & after an element

## PSEUDO-CLASSES

### :link & :visited

Styles an unvisited link & a visited link

### :enabled & :disabled

Styles an enabled element & a disabled element

### :first-child, :last-child & :only-child

Styles 1st child, last child & only child of an element

### :first-of-type, :last-of-type & :only-of-type

An element that is 1st, last or only type of its siblings

### :root

Styles an unvisited link & a visited link

### :active

Styles an activated element

### :focus & :hover

An element when with focus & when u mouser over it

### :checked & :selection

A checked form element and a highlighted element

### :nth-child(n) & :nth-last-child(n)

The nth child & the nth child counting from the end

### :nth-of-type(n) & :nth-last-of-type(n)

The nth sibling or nth counting from last of its siblings

### :empty

An element when with focus & when u mouser over it

### :target

Styles the current active target element

## ABSOLUTE UNITS

**Absolute measurement units do not change with changing browser window or device screen size**

- Not good for responsiveness. use only when you know what you are doing

### Centimeter - cm

### Millimeter - mm

### Inch - in

### Point - pt

## RELATIVE UNITS

**Relative measurement units change with changing screen size, pixels, width, etc**

- Very good for responsivity. Preferred over absolute units

### Percentage - %

Styles an unvisited link & a visited link

### em

Relative to font size of the current element

### vh & vw

Relative to the viewport height(vh) or width(vw)

### Pixel - px

An element when with focus & when u mouser over it

### Root em - rem

Relative to font size of root element(mostly <html> )

### vm

Relative to vh or vw, whichever is smaller

## ANGLES

### Degrees - deg

### Radians - rad

### Grads - grad

### Turns - turn

## FREQUENCY

### Hertz - Hz

### Kilo-Hertz - kHz

## TIME

### Seconds - s

### Milli-seconds - ms

# POSITION | OUTLINE | LIST | TRANSFORM

## POSITION

**Specifies the type of positioning method used for an element**

Use properties - top, bottom, right, & left to specify the position after specifying the type of positioning

### absolute

Relative to it's first positioned ancestor.

### relative

Relative to the elements original position

### fixed

Relative to browser window. Fixed to specified location

### sticky

Relative to a scroll position. first relative then become fixed

## OUTLINE

**A line that is drawn around elements, outside the borders, to make the element "stand out"**

Shorthand - outline: outline-width outline-style outline-color. eg. outline: 2px solid black

### outline-width

Values: length(px, em,etc), medium, thin, thick

### outline-color

Value: color

### outline-style

Values: none, hidden, dotted, dashed. solid, double ect

### outline-offset

Adds space b/n an outline and the edge or border, value: px

## LIST-STYLE

**Specifies the styling applied to a list of items, <li>**

Shorthand - list-style: list-style-type list-style-position list-style-image. eg. list-style: square inside url("img.jpg")

### list-style-type

Values: disc(default), square, circle, decimal, armanian, etc

### list-style-image

Value: image URL. eg. url(my-img.jpg)

### list-style-position

Values: outside(default), inside

### Note

Any missing list-style value will be replaced by the default

## TRANSFORM

**Applies a 2D or 3D transformation to an element. allows you to translate, scale, rotate, skew**

Values can be an angle(deg), length(px,em,etc), percentage, decimal(unitless)

### translate(x,y)

Defines a 2D translation

### translateX(x),translateY(y),translateZ(z)

Defines a translation, using value for the X, Y or Z-axis

### scale3d(x,y,z)

Defines a 3D scale transformation

### rotate(angle)

Defines a 2D rotation along an angle(parameter)

### rotateX(angle),rotateY(angle),rotateZ(angle)

Defines a 3D rotation along the X, Y or Z-axis

### skewX(x-angle), skewY(y-angle)

Defines a 2D skew transformation along the X or Y-axis

### translate3d(x,y,z)

Defines a 3D translation

### scale(x,y)

Defines a 2D scale transformation

### scaleX(x),scaleY(y),scaleZ(z)

Def a scale trans4mation by givin a val 4 X,Y or Z-axis

### rotate3d(x,y,z,angle)

Defines a 3D rotation

### skew(x-angle, y-angle)

Defines a 2D skew transformation along the X & Y-axis

### perspective(n)

Defines a perspective view for a 3D transformed element

## TRANSFORM Properties

### transform-origin

allows changing the position of transformed elements

### transform-style

Specifies how nested elements are rendered in 3D space

# SPEECH | CSS SELECTORS

## SPEECH

**Essentially converts your document into speech and feeds it into screen reader for the user**  
- It is best for people with impaired vision or people with reading problems.

### cue

**Sets the cue properties in one declaration**  
-Values: cue-before, cue-after

### elevation

**Sets where the sound should come from**  
-Values: angle,below,level,above,higher,lower

### pause-before & pause-after

**A pause b4 & after speaking an element's content**  
-Values: time, %

### pitch

**Specifies the speaking voice**  
-Values: frequency, x-low, low, medium, high, x-high

### play-during

**A sound to play while speaking an elemnt's content**  
-Values: auto, none, url, mix repeat

### speak

**Specifies whether content will render aurally**  
-Values: normal, nonne, spell-out

### speak-numeral

**Specifies how to speak numbers**  
-Values: digits, continuous

### speech-rate

**Specifies the speed of the speaking**  
-Values: number, slow, medium, fast, slower, etc

### cue-before & cue-after

**Sound to play b4 & after speaking elemnt's contnt**  
-Values: none, url

### pause

**Sets the pause properties in one declaration**  
-Values: pause-before, pause-after

### voice-family

**Specifies the voice family of the speaking**  
-Values: male, female, child, other specifics

### pitch-range

**Specifies the variation in the speaking voice.**  
-Value: number

### richness

**Specifies the richness of the speaking voice.**  
-Values: number

### speak-header

**Specifies how to handle table headers**  
-Values: always, once

### volume

**Specifies the volume of the speaking**  
-Values: number, %, silent, soft, medium, loud

### stress

**Specifies the "stress" in the speaking voice**  
-Values: number

## CSS SELECTORS

**Used to "find" (or select) the HTML elements you want to style.**

categories of selectors: simple(name,id,class), combinator, Pseudo-classes, pseudo-elements, attribute

### Universal - \*

**Selects all elements**  
- eg. \* { color: red; }

### Class - .

**Select single or multiple elements using their class**  
-eg. .button { color: #ffffff }

### Attribute - []

**Select an elemnt that has a particular attribute**  
-eg. a[target="\_blank"] { color: #00ff00; }

### Pseudo-element - ::

**To style specified part of an element**  
-eg. p::first-line { color: ff0000 }

### Child - >

**To style a direct child of an element**  
-eg. .caption > p { color: ff0000 }

### Name

**Select using element name**  
-eg. p { font-size: 18px; }

### id - #

**Select a sinlge element using its unique id**  
-eg #submit-button { disabled: true }

### Pseudo-class - :

**To Style an element in a particular state.**  
-eg a:hover { background-color: 0000ff; }

### Multiple Elements

**to give multiple elements the same styling**  
-eg. h1, p, .button { font-size: 17px; }

### Descendant

**To style a descendant, could be child or lower**  
-eg. .caption p { color: ff0000 }



# CSS GRID | FLEXBOX

## GRID

**Offers a grid-based layout system, with rows and columns, making it easier to design web pages**

- grid: grid-template-rows -template-columns -template-areas -auto-rows -auto-columns -auto-flow

### grid-template-columns

**Specif no(& widths) of columns in a grid layout**

-Values: none, auto, max-content, min-content, length

### grid-auto-flow

**Controls how auto-placed items get inserted in grid**

-Values: row, column, dense, row dense

### grid-auto-rows

**Sets a size 4 the rows in a grid container**

-Values: auto, max-content, min-content, length

### grid-row

**Shorthand property for the grid-row properties**

- grid-row: grid-row-start grid-row-end

### grid-template-rows

**Specifies no(& heights) of the rows in a grid layout.**

-Values: none, auto, max-content, min-content, length

### grid-template

**Shorthand property 4 the grid template properties**

-grid-template: grid-template-rows, -template-columns

### grid-auto-columns

**Sets a size 4 the columns in a grid container**

-Values: auto, max-content, min-content, length

### grid-column

**Shorthand property for the grid-column properties**

- grid-column: grid-column-start grid-column-end

## GRID-AREA

**Specifies a grid item's size and location in a grid layout. Shorthand property for:**

- grid-area: grid-row-start, grid-column-start, grid-row-end, grid-column-end

### grid-row-start

**Specif on which row to start displaying the item**

-Values: auto(default), row-line

### grid-row-end

**Specif on which row-line to stop displaying the item**

-Values: auto(default), span n, column-line

### grid-column-start

**Specif on which column to start displaying the item**

-Values: auto(default), span n, column-line

### grid-column-end

**Specif on which column-line to stop displaying item**

-Values: auto(default), span n, column-line

## GRID-GAP

**Defines the size of the gap between the rows and columns in a grid layout**

- grid-gap: grid-row-gap, grid-column-gap.

### grid-row-gap

**Sets the size of the gap b/n rows in a grid layout**

Value: length( px, etc), %. 0 is default

### grid-column-gap

**Sets the size of the gap b/n columns in a grid layout**

Value: length( px, etc), %. 0 is default

## CSS FLEXBOX

**The Flexible Box Layout Module, makes it easier to design flexible responsive layout structure**

-Flex container properties: flex-direction, flex-wrap, flex-flow, justify-content, -align-items, align-content

### display

**Specifies the type of box used for an HTML element**

Set value to "flex" to use flexbox

### flex-direction

**Specif direction of the items inside a flex container**

Values: row, row-reverse, column, column-reverse

### flex-flow

**Shorthand property 4 flex-direction and flex-wrap**

- flex-flow: flex-direction flex-wrap

### align-items

**Vertically aligns the flex items**

Values: stretch, center, flex-start, flex-end, baseline

### flex-wrap

**Specifies whether the flex items should wrap or not**

- Values: nowrap, wrap, wrap-reverse

### justify-content

**Horizontally aligns the flex items**

Values: flex-start, flex-end, center, space-between, space-around

### align-content

**Modifies behavior of the flex-wrap property**

Values: stretch, center, flex-start, flex-end, space-between, etc

### order

**Specif order of a flexible item relative to the rest**

- Value: number. default = 0.