

# CSS CHEATSHEET

## → Font

There are many properties related to the font, such as the face, weight, style, etc. These properties allow you to change the style or complete look of your text.

### 1. Font - Family

→ font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;

### 2. Font - Style

→ font-style: italic;

### 3. Font - Variant

→ font-variant: small-caps;

### 4. Font - Weight

→ font-weight: bold;

### 5. Font - Size

→ font-size: larger;

### 6. Font

→ font: style variant weight size family;

## → Text

Text properties allow one to manipulate alignment, spacing, decoration, indentation, etc., in the document.

### 1. Text - Align

→ text-align: justify;

### 2. Letter - Spacing

→ letter-spacing: -15em;

### 3. Text - Decoration

→ text-decoration: underline;

### 4. Word - Spacing

→ word-spacing: 0.25em;

### 5. Text - Transform

→ text-transform: uppercase;

### 6. Text - Indent

→ text-indent: 0.5em;

7. Line-Height: line-height: normal;

→ **Background**  
As the name suggests, these properties are related to background, i.e., you can change the color, image, position, size, etc., of the document.

1. **Background - Image**

→ `background-image: url("path");`

3. **Background - Size**

→ `background-size: cover;`

5. **Background - Attachment**

→ `background-attachment: scroll;`

7. **Background**

→ `background: color image repeat attachment position`

2. **Background - Position**

→ `background-position: right top;`

4. **Background - Repeat**

→ `background-repeat: no-repeat;`

6. **Background - Color**

→ `background-color: cyan;`

→ **Border**

Border properties are used to change the style, radius, color, etc., of buttons or other items of the document.

1. **Border - Width**

→ `border-width: 5px;`

3. **Border - Color**

→ `border-color: aqua;`

5. **Border**

→ `border: width style color;`

2. **Border - Style**

→ `border-style: solid;`

4. **Border - Radius**

→ `border-radius: 15px;`

→ **Box Model**

In laymen's terms, the CSS model box is a container that wraps around every HTML element. It consists of margins, borders, paddings, and the actual



1. Float  
→ float: none;
2. Clear  
→ clear: none;
3. Display  
→ display: block;
4. Height  
→ height: fit-content;
5. Width  
→ width: auto;
6. Margin  
→ margin: top right bottom left;
7. Padding  
→ padding: top right bottom left;
8. Overflow  
→ overflow: hidden;
9. Visibility  
→ visibility: visible;

## → Colors

With the help of the color property, one can give color to text, shape, or any other object.

1. Color  
→ color: pink;
2. Opacity  
→ opacity: 4;

## → Template Layout

Specifies the visual look of the content inside a template.

1. Box - Align  
→ box-align: start;
2. Box - Direction  
→ box-direction: normal;
3. Box - Flex  
→ box-flex: normal;
4. Box - Flex - Group  
→ box-flex-group: 2;
5. Box - Orient  
→ box-orient: inline;
6. Box - Pack  
→ box-pack: justify;
7. Box - Sizing  
→ box-sizing: margin-box;
8. max-width  
→ max-width: 800px;
9. min-width  
→ min-width: 500px;
10. max-height  
→ max-height: 100px;
11. min-height  
→ min-height: 80px;



## → Table

Table properties are used to give style to the tables in the document. You can change many things like border spacing, table layout, option, etc.

### 1. Border - Collapse

→ border-collapse: separate;

### 3. Border - Spacing

→ border-spacing: 2px;

### 5. Caption - Side

→ caption-side: bottom;

### 7. Column - Gap

→ column-gap: 5px;

### 9. Column - rule - width

→ column-rule-width: medium;

### 11. Column - width

→ column-width: 10px;

### 2. Empty - Cells

→ empty-cells: show;

### 4. Table - Layout

→ table-layout: auto;

### 6. Column - Count

→ column-count: 10;

### 8. Column - rule - style

→ column-rule-style: dotted;

### 10. Column - rule - color

→ column-rule-color: black;

### 12. Column - span

→ column-span: all;

## → List & Markers

List and marker properties are used to customize lists in the document.

### 1. List - style - type

→ list-style-type: square

### 3. List - style - image

→ list-style-image: url(image.gif);

### 2. List - style - position

→ list-style-position: 2.5px;

### 4. Marker - offset

→ marker-offset: auto;

## → Animations

CSS animations allow one to animate transitions on other media files on the web pages.



1. Animation-name
- animation-name: myanimation;
2. Animation-duration
- animation-duration: 5s;
3. Animation-timing-function
- animation-timing-function: ease;
4. Animation-delay
- animation-delay: 2ms;
5. Animation-iteration-count
- animation-iteration-count: 1;
6. Animation-direction
- animation-direction: normal;
7. Animation-play-state
- animation-play-state: running;
8. Animation-fill-mode
- animation-fill-mode: both;

### → Transitions

Transitions let you define the transition between two states of an element.

1. Transition-property
- transition-property: none;
2. Transition-duration
- transition-duration: 1s;
3. Transition-timing-function
- transition-timing-function: ease-in-out;
4. Transition-delay
- transition-delay: 20ms;

### → CSS Flexbox

Flexbox is a layout of CSS that lets you format HTML easily. Flexbox makes it simple to align items vertically and horizontally using rows and columns. Items will "flex" to different sizes to fill the spaces. And overall, it makes the responsive design more manageable.

### \* Parent Properties (flex container)

1. display
- display: flex;
2. Flex-Direction
- flex-direction: row / row-reverse / column / column-reverse;
3. flex-wrap
- flex-wrap: nowrap / wrap / wrap-reverse;
4. Flex-flow
- flex-flow: column wrap;



5. justify-content: flex-start | flex-end | center | space-between | space-around | space-evenly | start | end | left | right | safe | unsafe;
6. align-items: stretch | flex-start | flex-end | center | baseline | first baseline | last baseline | start | end | self-start | self-end | safe | unsafe;
7. align-content: flex-start | flex-end | center | space-between | space-around | space-evenly | start | end | stretch | baseline | first baseline | last baseline | safe | unsafe;

### \* Child Properties (flex items)

1. order: 3; /\* default is 0 \*/
2. flex-grow: 5; /\* default is 0 \*/
3. flex-shrink: 2; /\* default is 1 \*/
4. flex-basis: auto;
5. flex-shorthand: none | [<'flex-grow'> <'flex-shrink'> | <'flex-basis'>];
6. align-self: auto | flex-start | flex-end | center | baseline | stretch;

### → Grid

Grid layout is a 2-Dimensional grid system to CSS that creates complex responsive web-designs, layouts more easily and consistently across browsers.

### \* Parent Properties (Grid Container)

1. display: grid | inline-grid
2. grid-template-columns: 12px 13px 13px;
3. grid-template-rows: 5px auto 12px;
4. grid-template: none | <grid-template-row> / <grid-template-column>;
5. column-gap: 5px;
7. grid-column-gap: 10px;
6. row-gap: 5px;
8. grid-row-gap: 10px;
9. gap: <grid-row-gap> <column-gap>;
10. grid-gap: <grid-row-gap> <grid-column-gap>;
11. justify-items & align-items: start | end | center | stretch;
12. place-items: center;



13. justify-content & align-content: start | end | center | stretch | space-around | space-between | space-evenly;
14. place-content: <align-content> / <justify-content>;
15. grid-auto-columns & grid-auto-rows: <track size>;
16. grid-auto-flow: column | column dense | row | row dense;

#### \* Child Properties (Grid items)

1. grid-column-start: <number> | span <number> | <name> | span <name> | auto;
2. grid-row-start: <number> | span <number> | <name> | span <name> | auto;
3. grid-column-end & grid-row-end: <number> | span <number> | <name> | span <name> | auto;
4. grid-column & grid-row: <start-line> / <end-line> / span <value>;
5. grid-area: <name> | <row-start> / <column-start> / <row-end> / <column-end>;
6. justify-self: start | end | center | stretch;
7. align-self: start | end | center | stretch;
8. place-self: center;