

CSS Cheatsheet

Cascading Style Sheets

- You can input `<style>` tags in HTML, or compose a separate style sheet referenced in the HTML.
- The idea is that you select an HTML element as target in the DOM, then apply attributes to it.
- DOM: Document Object Model
- Case sensitive
- It is best practice to end inline styles with a semi colon ;
-
- Creating variables in the `<style>` `:root {--varname: pink}` allows for global availability for any selector & heir.
- Declaration order determines conflicts, so the 'lowest' or last `<style>` declaration takes priority. This can be creatively used to produce fallbacks for older browsers, by listing the fallback first.
- `!important` in the stylesheet overrides all. `.blue-text {color: blue !important;}`
- Order of declaration priority follows the following rule: *!important ; style= ; id= ; stylesheet*
- Define mobile and tablet differences using: `@media (max-width: 350px) {:root {varname: value;}}`

Allows control of:

1. Color
2. Fonts
3. Positioning
4. Spacing
5. Sizing
6. Decorations
7. Transitions

`<!-- -->` At first, comment out code when you need be rid of it.

HTML

Rules

```
<style> h2 {color:blue;} </style>
```

ID

```
<style>
```

```
#cat-photo-form {background-colour: green;}
```

```
<form id="cat-photo-form"> ID has a primacy over classes, allowing for customisation.
```

IDs are ALWAYS hashtags.

Class

```
<style>
```

```
Body {background-colour: black; colour: green; font-family: monospace;}
```

```
.links
```

```
Class {position: fixed/relative/absolute; top/right/bottom/left: 10px;} moves w/ a push, not pull
```

Class {float: left/right; width: 20%;}

Class {z-index: 2} *to create illustrative layers*

Class {height: 100px; width: 100px; margin: auto;} *margin: auto centres the element*

```
Class:hover {transform:
skewX(1deg)/skewY(1deg)
scale(1.1);
rotate(-45deg)
}
```

.blue-text {color:blue;} </style> <h2 class="blue-text">
Class attributes lose the full stop & shift from <style> to <element class="class1 class2 class 3">

H1 {font-size: 30px;}

H2 {font-family: "sans serif", back up font;} *(Back up when using external styles)*

H3 {font-weight: 500}

H4 {line-height: 25px}

Class {background-color: hsl(180, 80%, 25%); *hue, saturation and lightness*

background: linear-gradient(35deg, #ccffff, #ffcccc); *creates an appealing gradient of colour*
Background: repeating-linear-gradient(90deg, yellow 0px, blue 40px, green 40px, Red 90px);

```
.thin-red-border {border-colour: blue; border-width: 5px; border-style: solid; border-radius: 50%}
.silver-background {background-colour: silver;}
```

```
.red-box {background-colour: red; colour: #fff; padding: 40px 20px 20px 40px; margin: 40px 20px
20px 40px; border-radius: 60px;}
```

```
[attr='value'] e.g. [type='checkbox'] {margin: 10px 0px 15px 0px;}
```

```
.penguin {--penguin-skin: green;} .penguin-hands {var(--penguin-skin, fallback);}
```

</style>

Inline

```
<h2 style="color: red;">
```

External Styles

```
<link href="https://fonts.googleapis.com/css?family=Lobster" rel="stylesheet" type="text/css">
```

Visual Design

Box-shadow: offset-x, offset-y, blur-radius, spread-radius, color *in that order. Commas can be sequentially used to apply multiple shadows.*

Animation

```
button:hover {
  animation-name: background-color;
  animation-duration: 500ms;
  animation-fill-mode: forwards;
  Animation-iteration-count: infinite/1/2/3
  Animation-timing-function: ease-out/ease-in/linear
}
```

```
@keyframes background-color {
  100% {
    background-color: #4791d0;
  }
}
```

animation-timing-function: cubic-bezier(0.25, 0.25, 0.75, 0.75;) *custom animation timing*

```
@keyframes bounce{
  0% {
    top: 0px;
  }
  50% {
    top: 249px;
    width: 130px;
    height: 70px;
  }
  100% {
    top: 0px;
  }
}
```

Opacity:0.1
Top/left: 25px

Misc

Text-align: centre

— — — IMAGE SCALES TO SCREEN — — —

```
img {
  max-width: 100%;
  height: auto;
}
```

vw (viewport width): 10vw would be 10% of the viewport's width.

vh (viewport height): 3vh would be 3% of the viewport's height.

vmin (viewport minimum): 70vmin would be 70% of the viewport's smaller dimension (height or width).

vmax (viewport maximum): 100vmax would be 100% of the viewport's bigger dimension (height or width).