
CSS Cheat Sheet

1. Font-Face

The `@font-face` rule lets you define custom fonts for your webpage.

Syntax

```
@font-face {  
  font-family: "CustomFont";  
  src: url("customfont.woff2") format("woff2"),  
       url("customfont.woff") format("woff");  
}
```

Example

```
@font-face {  
  font-family: "Roboto";  
  src: url("Roboto-Regular.woff2") format("woff2");  
}  
  
body {  
  font-family: "Roboto", Arial, sans-serif;  
}
```

Key Points

- Use `.woff2` for modern browsers and `.woff` as a fallback.
- Always include a generic fallback font in your `font-family`.

2. Display: Grid

CSS Grid is a powerful layout system for creating rows and columns.

Key Properties

Property	Description	Example
<code>display: grid;</code>	Turns a container into a grid layout.	<code>display: grid;</code>
<code>grid-template-rows</code>	Defines rows in the grid.	<code>grid-template-rows: 100px 200px auto;</code>
<code>grid-template-columns</code>	Defines columns in the grid.	<code>grid-template-columns: 1fr 2fr;</code>
<code>gap</code>	Adds space between rows and columns.	<code>gap: 10px;</code>
<code>place-items</code>	Centers content within the grid items.	<code>place-items: center;</code>

Example

```
.container {
  display: grid;
  grid-template-columns: 1fr 2fr;
  grid-template-rows: auto 100px;
  gap: 20px;
}

.item1 {
  grid-column: 1 / span 2; /* Spans two columns */
}
```

Visualization

```
|---Grid Template---|
| Item 1 spans 2 columns → Item 2 |
| Item 3 | Item 4 |
```

3. Animations

CSS animations bring motion and life to your webpage.

Key Properties

Property	Description	Example
<code>@keyframes</code>	Defines the animation's steps.	<code>@keyframes slide { from { ... } }</code>
<code>animation-name</code>	Links an element to a <code>@keyframes</code> .	<code>animation-name: slide;</code>
<code>animation-duration</code>	Sets the animation's length.	<code>animation-duration: 2s;</code>
<code>animation-timing-function</code>	Controls the pace of the animation.	<code>animation-timing-function: ease-in;</code>

Example

```
@keyframes fade {
  from {
    opacity: 0;
  }
  to {
    opacity: 1;
  }
}

.box {
  animation-name: fade;
  animation-duration: 3s;
  animation-timing-function: ease-in-out;
}
```

4. Transitions

Transitions allow for smooth changes between property values.

Syntax

```
transition: property duration timing-function delay;
```

Example

```
button {  
  background-color: blue;  
  transition: background-color 0.5s ease;  
}  
  
button:hover {  
  background-color: lightblue;  
}
```

Key Properties

Property	Description	Example
<code>transition-property</code>	Specifies which property to animate.	<code>transition-property: color;</code>
<code>transition-duration</code>	Defines how long the transition lasts.	<code>transition-duration: 1s;</code>
<code>transition-timing-function</code>	Specifies the speed curve.	<code>transition-timing-function: ease-out;</code>

5. Float

Float is used to position elements to the left or right, often for layouts.

Key Properties

Property	Description	Example
<code>float</code>	Moves an element to the left or right.	<code>float: left;</code>
<code>clear</code>	Prevents elements from wrapping around floated elements.	<code>clear: both;</code>

Example

```
img {  
  float: left;  
  margin-right: 10px;  
}  
  
p {  
  clear: both;  
}
```

Key Points

- Floats are commonly used for simple layouts (e.g., text wrapping around images).
 - Use **clear** to avoid overlap.
-