CSS

cheat-sheet

🌈 CSS Cheatsheet (with Short Description & Analogies)

1. Basic Syntax

```
selector { property: value; }
```

📝 **Meaning:** Says *which element* to style and *how*.
🎯 **Analogy:** Like telling a painter — “Paint the wall (selector) red (property:value)”.

---

2. Selectors

Used to **target** elements.
🎯 **Analogy:** Like calling someone by name, nickname, or ID.

Selector	Example	Meaning
Universal	`*`	All elements
Element	`p`	All `<p>` tags
Class	`.box`	Elements with class "box"
ID	`#main`	Element with id "main"
Descendant	`div p`	`<p>` inside `<div>`
Child	`div > p`	Direct child only
Pseudo-class	`a:hover`	On hover
Attribute	`[type="text"]`	Input type text

---

3. Text & Font

Controls how text looks.
🎯 **Analogy:** Like choosing handwriting style, size, and color.

```
color: red;
font-size: 18px;
font-family: Arial;
```

```
text-align: center;
text-transform: uppercase;
```

---

4. Box Model

Every element = a **box** (Content + Padding + Border + Margin).
🎯 **Analogy:** Like a photo (content) inside a frame (border) placed in a wall space (margin).

```
div {
  margin:10px; padding:10px;
  border:2px solid black;
}
```

---

5. Background

Sets image or color behind an element.
🎯 **Analogy:** Like wallpaper behind your photo.

```
background-color: lightblue;
background-image: url('bg.jpg');
background-size: cover;
```

---

6. Position

Moves elements around.
🎯 **Analogy:** Like fixing your photo — stuck, floating, or scroll-sticky.

```
position: relative | absolute | fixed | sticky;
top: 10px; left: 20px;
```

---

7. Display & Layout

Defines **how elements appear** on screen.
🎯 **Analogy:** Like choosing how books are arranged — stacked (block) or side by side (inline).

```
display: block | inline | flex | grid;
```

---

8. Flexbox

Used to align and distribute items in a line or column.
🎯 **Analogy:** Like organizing photos evenly in one row.

```
display: flex;
justify-content: center;
align-items: center;
gap: 10px;
```

---

9. Grid

Creates a **2D layout** (rows + columns).
🎯 **Analogy:** Like placing items on a chessboard.

```
display: grid;
grid-template-columns: repeat(3, 1fr);
gap: 10px;
```

---

10. Transition & Animation

Adds **motion** or **smooth changes**.
🎯 **Analogy:** Like slowly fading lights instead of instant on/off.

```
transition: all 0.3s ease;
@keyframes move {
  from {left:0;}
  to {left:100px;}
}
```

---

11. Colors

Defines colors using **names, RGB, Hex, RGBA**.
🎯 **Analogy:** Like choosing shades from a color palette.

```
color: red;
color: #ff0000;
color: rgba(255,0,0,0.5);
```

---

12. Media Query

Used for **responsive design** (mobile, tablet, PC).
🎯 **Analogy:** Like clothes that fit all sizes — adjusts layout for screen width.

```
@media (max-width:600px) {
  body { background-color: lightgray; }
```