@CODE.CLASH

CSS
Cheat Sheet

Selectors

```
*{}                     Universal Selector
#id {}                  ID Selector
.class {}               Class Selector
h1, h2{}                Type Selector
h1+p {}                 Adjacent Sibling Selector
ul > li{}               Child Selector
h1 ~ p {}               General Sibling Selector
p a{}                   Descendant Selector
div[att="val"]{}        Attribute Selector
```

Units

```
%       Percentage
cm      Centimeter
in      Inch
mm      Millimeter
pc      Pica (1 pica = 12 points)
pt      Point (1 point = 1/72 inch)
px      Pixel (1 pixel = 1/96 inch)
ch      Width of the "0" glyph in the font size
em      1em = Current font size
ex      X-height of the element's font
gd      Grid defined by 'layout-grid'
rem     Font size of the root element
vh      Viewport's height
vw      Viewport's width
vm      Smaller of viewport's height or width
```

Pseudo Selectors

`:active`	Activated element
`:focus`	Focused element
`:hover`	Hovered element
`:link`	Unvisited link
`:disabled`	Disabled element
`:enabled`	Enabled element
`:checked`	Checked element
`:nth-child(n)`	N-th sibling
`:nth-last-child(n)`	N-th sibling from the end
`:first-child`	First sibling
`:last-child`	Last sibling
`:only-child`	Only child
`:nth-of-type(n)`	N-th sibling of its type
`:nth-last-of-type(n)`	N-th sibling of its type from end
`:last-of-type`	Last sibling of its type
`:first-of-type`	First sibling of its type
`:only-of-type`	Only child of its type
`:empty`	Element with no children
`:root`	Root element
`:not(x)`	Element not matching `'x'`
`:target`	Target element specified by a URI
`::first-letter`	Style for the first letter of text
`::first-line`	Style for the first line of text
`::before`	Insert content before an element
`::after`	Insert content after an element

Background

```
background-color: #FFF2EB

background-image: url()

background-repeat: repeat-x | repeat-y | repeat | space | round
                   | no-repeat

background-attachment: scroll | fixed | local | intial | inherit

background-position: top | right | bottom | left | center
```

Font Properties

```
font-style: normal | italic | oblique

font-variant: normal | small-caps

font-size: 13px | 0.8rem | 80%

font-weight: normal | bold | bolder | lighter | 100-900

letter-spacing: normal | 4px

line-height: normal | 3rem | 34%

font-family: 'Open sans', sans-serif
```

```
text-align: left | right | center | justify

text-transform : capitalise | lowercase | uppercase

text-indent: 23px

vertical-align: baseline | 10px | sub | super | top | text-top
              | middle | bottom | text-bottom | initial

text-align-last: auto | left | right | center | justify
               | start | end | initial | inherit

text-decoration: none | underline | overline | lint-through

text-justify: auto | inter-word | inter-character | none
            | initial | inherit

text-overflow: clip | ellipsis | string | initial | inherit

text-shadow: h-shadow v-shadow | blur-radius color | none
           | initial| inherit
```

```
transition-timing-function: ease | linear | ease-in | ease-out
                          | ease-in-out | cubic-Bezier(num,num,num,num);

transition-property: none | all

transitions-delay: time;

transitions-duration: time;

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

Animation

```
animation-timing-function: ease | linear | ease-in | ease-out | ease-in-out
                         | cubic-Bezier (number, number, number, number)

animation-name: none | IDENT;

animation-duration : time;

animation-delay: time;

animation-iteration-count : inherit | number;

animation-direction: normal | alternate;

rotation: angle rotation-point position;

animation-play-state: running | paused;

// shorthand
animations: animation-name animation-duration animation-timing-function
            animation-delay animation-iteration-count animation-direction
```

List Styling

```
// List Style
list-style-type: disc | circle | square | none;

// List Position
list-style-type: inside | outside;

// List Image
list-style-img: url()
```

Position

```
// Position
position: static | relative | absolute | fixed | sticky;

// Position Element
top | right | bottom | left

// Float Element
float: left | right | none

// Z-index
z-index: 3 | auto | none

// Clear Floating
clear: none | left | right | both
```

```
// 2D Transform
transform: translate(x, y):     Translate (move) element
transform: rotate(angle):       Rotate element around a specified angle
transform: scale(x, y):         Scale element
transform: skew(x-ang,y-ang):   Skew (slant) element
transform: skewX(angle):        Skew (slant) element along the X-axis
transform: skewY(angle):        Skew (slant) element along the Y-axis


// 3D Transform
transform: translate3d(x, y, z): Translate (move) element
transform: rotateX(angle):       Rotate element around the X-axis
transform: rotateY(angle):       Rotate element around the Y-axis
transform: rotateZ(angle):       Rotate element around the Z-axis
transform: scale3d(x, y, z):     Scale element
transform: perspective(value):   Set the perspective view
```

WEB DEBVELOEPR

@COMPUTER__PROGRAMMER

CSS

Beginner To Advanced

Day 1: Introduction to CSS

- CSS: Cascading Style Sheets (CSS) is a stylesheet language used to describe the presentation of a document written in HTML or XML.
- Selectors: CSS selectors are patterns used to select the element(s) you want to style.
- Properties: CSS properties are attributes that define the appearance and behavior of elements on a webpage.

Day 2: Basic Selectors and Properties

- Element Selector: Selects elements based on their tag name. Example: **p { color: blue; }**
- ID Selector: Selects an element based on its unique ID attribute. Example: **#header { background-color: pink; }**
- Class Selector: Selects elements with a specific class attribute. Example: **.highlight { font-weight: bold; }**
- Universal Selector: Selects all elements on the page. Example: *** { margin: 0; padding: 0; }**
- Properties: Common properties like **color**, **background-color**, **font-size**, **font-family**, **margin**, **padding**, etc.

Day 3: Box Model and Layout

- Box Model: Describes how elements on a webpage are structured as rectangular boxes with content, padding, border, and margin.
- Margin: Space outside the border of an element.
- Padding: Space between the content and the border of an element.
- Border: Border around the content of an element.
- Width & Height: Specifies the width and height of an element's content area.
- Display Property: Defines how an element is displayed, like **block, inline, inline-block, flex, grid,** etc.
- Positioning: CSS properties like **position, top, right, bottom,** and **left** for controlling the position of elements.

Day 5: Responsive Design and Media Queries

- Responsive Design: Design approach aimed at crafting sites to provide an optimal viewing experience across a wide range of devices.

- Media Queries: Conditional statements that apply styles based on device characteristics like **screen width, height, orientation,** etc.

- Viewport Meta Tag: **<meta name="viewport" content="width=device-width, initial-scale=1.0">** adjusts the viewport to the device's width.

- Breakpoints: Specific points where the layout of a webpage will respond to suit different screen sizes.

- Mobile-first Design: Designing for mobile devices first, then scaling up for larger screens using media queries.

- Flexbox and Grid Layout: CSS layout models that facilitate the creation of flexible and responsive layouts.

Day 4: Advanced Selectors

- Attribute Selectors: Select elements based on their attribute values. Example: **input[type="text"] { background-color: yellow; }**

- Pseudo-classes: Select elements based on their state or position. Example: **a:hover { color: red; }**

- Pseudo-elements: Style specific parts of an element. Example: **p::first-line { font-weight: bold; }**

- Child Selectors: Select elements that are direct children of another element. Example: **ul > li { list-style-type: none; }**

- Adjacent Sibling Selectors: Select an element that is directly preceded by another element. Example: **h2 + p { font-style: italic; }**

- General Sibling Selectors: Select elements that are siblings of another element. Example: **h2 ~ p { color: green; }**

Day 6: CSS Preprocessors

- CSS Preprocessors: Tools that extend the functionality of CSS, allowing for variables, mixins, nested rules, inheritance, etc.

- Sass (Syntactically Awesome Stylesheets): A popular CSS preprocessor that adds features like variables, nesting, and mixins.

- Less: Another CSS preprocessor that also introduces variables, nesting, and other enhancements.

- Stylus: Yet another CSS preprocessor with a unique syntax and feature set.

Day 7: Advanced Layout Techniques

- Grid Layout: CSS Grid Layout provides a two-dimensional grid-based layout system, making it easier to design complex layouts.
- Flexbox: CSS Flexible Box Layout (Flexbox) provides a one-dimensional layout model, making it easier to distribute space among items in a container.
- Multi-column Layout: CSS Multi-column Layout provides a way to flow content into multiple columns.
- Floats & Clearing: Techniques used for creating layouts where elements float around each other.
- Positioning: Advanced positioning techniques like **position: absolute**, **position: relative**, and **position: fixed**
- CSS Frameworks: Utilizing CSS frameworks like Bootstrap or Foundation for responsive, pre-designed layout components.


WEB DEVELOPER
@COMPUTER__PROGRAMMER

CSS

CheatSheet

Beginner:

- **Selectors:**
  - Target elements by tag name (e.g., **h1 { ... }** for all <h1> elements)
  - Style elements with a unique **ID** using # (e.g., **#main-heading { ... }**)
  - Apply styles to elements with a **class** using . (e.g., **.important { ... }**)

Basic Properties:

- **font-family:** Change the font (e.g., **font-family: Arial;**)
- **font-size:** Control text size (e.g., **font-size: 16px;**)
- **color:** Set text color (e.g., **color: #FF0000;**)
- **background-color:** Set background color (e.g., **background-color: #f0f0f0;**)

- **Text Properties:**
    - **text-align** (center, left, right),
    - **text-decoration** (underline),
    - **text-transform** (uppercase, lowercase)
- **Box Model:** Basic understanding of **padding, margin,** and **border** for element spacing.

Selectors (Go Beyond the Basics):

- **Universal Selector:** Target all elements with * (use cautiously for broad styling).
- **Group Selectors:** Apply styles to a comma-separated list of selectors (e.g., **h1, h2, h3 { ... }**).
- **Pseudo-elements:** Style specific parts of an element (e.g., **::first-letter** for the first letter of an element).
- **Attribute Selectors:** Target elements based on their attributes (e.g., **a[href^="https"]** for links starting with "**https**").

Intermediate:

- **Selectors:**
  - Descendant selectors (e.g., **div p { ... }**)
  - Child selectors (e.g., **ul > li { ... }**)
  - Pseudo-classes (e.g., **:hover** for hover effects)
- **Properties:**
  - **border** properties (style, width, color)
  - **display** property (control how elements are displayed)
  - Positioning: **position: absolute;** for precise placement

Properties (Extend Your Toolkit):

- **Font Properties:** Explore **font-weight** (boldness), **font-style** (italic), **letter-spacing, line-height**.
- **Background Properties:** Utilize **background-image** for gradients with **linear-gradient** or **radial-gradient** functions, and **background-position** to control image placement.
- **Text Properties:** Master **text-shadow** for adding shadows to text, and **text-overflow** to handle overflowing text with ellipsis (**...**).
- **List Properties:** Style unordered lists with **list-style-type** (disc, circle, square) and **list-style-position** (inside, outside).
- **Flexbox:** Control how elements flex within a container using properties like **flex-direction** (row, column), **justify-content** (horizontal alignment), and **align-items** (vertical alignment).


Advanced:

- **Media Queries:** Responsive design for different screen sizes
- **CSS Grid:** Advanced layout system
- **CSS Animations and Transitions:** Dynamic effects
- **CSS Preprocessors (Sass/LESS):** Efficient and maintainable code

If You Want Post On Other Topic Please Comment

WEB DEVELOPER

@COMPUTER__PROGRAMMER

CSS

CHEATSHEET 2024

1. CSS SYNTAX

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

2. SELECTORS

```
/* Element Selector: */

p { color: blue; }


/*Class Selector: */

.className { color: red; }


/*ID Selector: */

#idName { color: green; }
```

3. TEXT STYLING

```
color: blue;
font-size: 16px;
font-family: Arial, sans-serif;
text-align: center;
font-weight: bold;
```

4. BOX MODEL

```
padding: 20px;
margin: 15px;
border: 2px solid black;
width: 100px;
height: 100px;
```

5. BACKGROUNDS

```
background-color: #f0f0f0;
background-image: url('image.jpg');
background-repeat: no-repeat;
background-size: cover;
```

6. DISPLAY PROPERTIES

```
display: block;
display: inline;
display: inline-block;
display: flex;
```

7. POSITIONING

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

8. FLEXBOX

```
display: flex;
justify-content: center;
align-items: center;
flex-direction: row;
flex-wrap: wrap;
```

9. GRID

```
display: grid;
grid-template-columns: 1fr 1fr 1fr;
grid-gap: 10px;
grid-area: header;
```

10. CSS UNITS

```
width: 100px;
height: 50vh;
padding: 2em;
margin: 10%;
```

11. MEDIA QUERIES

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

15. PSEUDO-CLASSES

```
a:hover { color: red; }
input:focus { border: 2px solid blue; }
```

16. PSEUDO-ELEMENTS

```
p::before { content: "Prefix"; }
p::after { content: "Suffix"; }
```

17. Z-INDEX

```
position: relative;
z-index: 10;
```

18. OPACITY

```
opacity: 0.5;
```

12. TRANSITIONS

```
transition: background-color 0.3s ease-in-out;
```

13. TRANSFORMS

```
transform: rotate(45deg);
transform: scale(1.5);
```

14. ANIMATIONS

```
@keyframes slide {
  from { left: 0px; }
  to { left: 100px; }
}

div {
  animation: slide 2s infinite;
}
```