# Introduction:

Name: Mohammad Sameer T A

Domain: FrontEnd

complete css notes

css:

**css selectors**  
**1. Basic Selectors**

* \* → Selects **all elements**
* element → Selects all elements of that type  
  Example: p {} → all <p>
* .class → Selects elements with that class  
  Example: .btn {}
* #id → Selects element with that ID  
  Example: #header {}
* element.class → Selects element with specific class  
  Example: p.intro {}

## **Grouping & Universal**

* div, p, span {} → Selects multiple elements
* \* {} → All elements

**Combinators**

* A B (Descendant) → Selects B inside A  
  Example: div p {}
* A > B (Child) → Direct child  
  Example: ul > li {}
* A + B (Adjacent sibling) → Next sibling  
  Example: h1 + p {}
* A ~ B (General sibling) → All siblings  
  Example: h1 ~ p {}

**Pseudo-classes**

* :hover → On mouse hover
* :active → When clicked
* :focus → When focused (form fields)
* :visited → For visited links
* :first-child → First child of parent
* :last-child → Last child
* :nth-child(n) → nth child
* :nth-child(odd/even) → Odd/even child
* :nth-of-type(n) → nth of same type
* :not(selector) → Excludes elements

**Pseudo-elements**

* ::before → Insert content before element
* ::after → Insert content after element
* ::first-letter → First letter styling
* ::first-line → First line styling
* ::selection → Highlighted text styling
* ::placeholder → Input placeholder

**Special**

* :root → Root element (HTML)
* :checked → Radio/checkbox checked
* :disabled → Disabled form field
* :enabled → Enabled field
* :empty → Empty element
* :only-child → Only child element
* :only-of-type → Only element of that type

CSS Box Model:

This way you can remember it as **outside → inside**:  
**Margin → Border → Padding → Content**

selector {

width: 200px; /\* Content width \*/

height: 100px; /\* Content height \*/

padding: 20px; /\* Space inside border \*/

border: 5px solid red; /\* Line around padding + content \*/

margin: 30px; /\* Space outside the element \*/

}

CSS Font, Text & Color Properties:

Font Properties:

selector {

font-family: "Arial", sans-serif; /\* Font type \*/

font-size: 16px; /\* Font size \*/

font-weight: bold; /\* normal | bold | 100–900 \*/

font-style: italic; /\* normal | italic | oblique \*/

font-variant: small-caps; /\* normal | small-caps \*/

}

Text Properties:

selector {

color: blue; /\* Text color \*/

text-align: center; /\* left | right | center | justify \*/

text-decoration: underline; /\* none | underline | overline | line-through \*/

text-transform: uppercase; /\* none | uppercase | lowercase | capitalize \*/

text-indent: 40px; /\* Indent first line \*/

letter-spacing: 2px; /\* Space between letters \*/

word-spacing: 5px; /\* Space between words \*/

line-height: 1.5; /\* Space between lines \*/

white-space: nowrap; /\* Handling of whitespace \*/

text-shadow: 2px 2px 5px gray; /\* x-offset y-offset blur color \*/

}

Color Properties:

selector {

color: red; /\* Named color \*/

color: #ff0000; /\* Hex \*/

color: rgb(255, 0, 0); /\* RGB \*/

color: rgba(255, 0, 0, 0.5); /\* RGB with transparency \*/

color: hsl(0, 100%, 50%); /\* HSL \*/

color: hsla(0, 100%, 50%, 0.5); /\* HSL with transparency \*/

background-color: yellow; /\* Background fill \*/

}

CSS Sizing Units:

**Absolute Units (Fixed size)**

* px → Pixels (most common)
* pt → Points (1/72 of an inch, used in print)
* pc → Picas (1pc = 12pt)
* cm → Centimeters
* mm → Millimeters
* in → Inches

**Relative Units (Scales with parent or viewport)**

* % → Relative to parent element size
* em → Relative to element’s font size
* rem → Relative to root (<html>) font size
* ex → Relative to height of lowercase "x"
* ch → Relative to width of "0" (zero character)

**Viewport Units (Responsive sizing)**

* vw → 1% of viewport width
* vh → 1% of viewport height
* vmin → 1% of the smaller (width or height)
* vmax → 1% of the larger (width or height)

**CSS Functions for Sizing**

* min() → Takes the smaller of values
* max() → Takes the larger of values
* clamp(min, preferred, max) → Sets flexible range

CSS Display Properties:

**Common Display Values**

* block → Takes full width, starts on a new line.  
  (e.g., <div>, <p>, <h1> by default)
* inline → Fits only its content, does not start a new line.  
  (e.g., <span>, <a>, <strong> by default)
* inline-block → Behaves like inline (in a line), but allows setting width/height/padding/margin.
* none → Hides the element completely (removed from flow).

p { display: block; }

span { display: inline; }

button { display: inline-block; }

.hidden { display: none; }

**2. Table Display**

* table → Acts like a <table> element
* table-row → Acts like <tr>
* table-cell → Acts like <td>

div.table { display: table; }

div.row { display: table-row; }

div.cell { display: table-cell; }

**Flexbox Display**

* flex → Creates a **flex container** (one-dimensional: row/column)
* inline-flex → Same as flex, but inline container

.container {

display: flex;

justify-content: space-between;

align-items: center;

}

**Grid Display**

* grid → Creates a **grid container** (two-dimensional layout)
* inline-grid → Grid but inline

.container {

display: grid;

grid-template-columns: 1fr 1fr 1fr;

}