⑧

CSS

what is css?

-) Cascading Style sheets

-) Describes how HTML elements are to be displayed on screen

-) It can control layout of multiple pages all at ones

-) External style sheet are stored in css files

Why use CSS?

used to define styles of your web pg, including the design, layout & variations in display for different devices & screen sizes.

CSS Syntax?

selector { Declaration block ; }

h1 { color : blue ; }

property ↓   value ↓

How to add styling in HTML?

* Inline CSS

* Internal CSS (style Tag)

* External CSS

Selectors in CSS

A css selector selects the HTML element(s) you want

to style.

1) Simple selectors (Element selector, class selector, ID selector)
2) Psuedo-class selectors
3) ~~Multiple Selector~~ Combinator selectors
4) Psuedo-elements selectors   5) Attribute selectors

Element Selector

Selects ~~ele~~ HTML elements based on the element name

En   p {
       text-align: center;
       colon: red;
    }

Class Selector

selects HTML elements with a speific class attribute

To select elements with a speific class write a period (.) character, followed by class name

En   .center {
         colon: red;
      }

ID selector

Used to select unique element.

En   #para1 {
         colon: red;
      }

Psuedo class selectors:

Selects elements based on certain states.

-) hover
-) active
-) etc

Syntax :   selector : pseudo - class {
                   property : value;
              }

Combinator selectors

Selects elements based on specific relationship between them.

-) Descendant (space)
-) child (>)
-) Next sibling (+)
-) Subsequent - sibling (~)

Pseudo - elements selectors

select and style a part of an element.

Syntax: selector :: pseudo- element {
                  property : value ;
             }

Attribute selectors

select elements based on an attribute or attribute value.

Universal selectors (*)

Multiple (Grouping ~~sted~~ selector) & h1, h2, p {

color: red;

}

Nested selector

Selector specificity

If there are 2 or more CSS rules that point to the same element, the selector with highest specificity value will "win", and its style decoration will be applied to that HTML element.

Specificity Hierarchy

Every CSS selector has its place in the specificity hierarchy

4 categories which define the specifity level of a selector:

* Inline styles (Highest)

* IDs (less than inline)

* classes, pseudo-classes, attribute selectors (Lower)

* elements & pseudo-elements (lowest)

!important Rule

-) !important rule is used to add more importance to a property / value then normal

-) It will override all previous rules

-) Do not use unless you absolutely have to

Ex

```
p {
    color : red !important;
}
```

Cascading in CSS

Cascading ensures that styles from different sources (inline, internal and external) are applied correctly, with specific rules taking precedence over general ones.
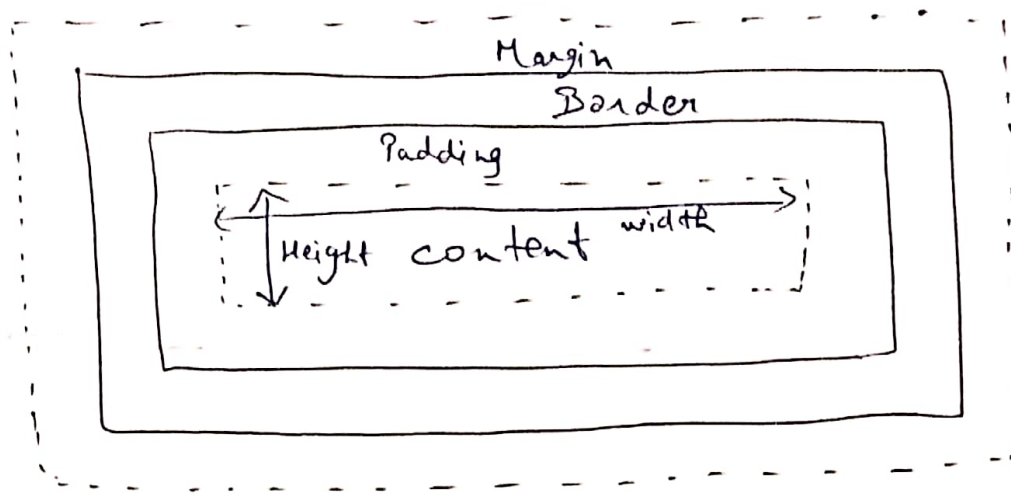
Box Model in CSS

Basic building block of CSS

According to the Box model concept every element on a page is a rectangular box & may have width, height, padding, borders & margins.

Box Model

Diagram showing the Box Model with nested boxes labeled Margin, Border, Padding, and content with Height and width arrows.

* **Content:** The content of the box, where text & images appear

* **Padding:** clears an area around the content. The padding is transparent.

* **Border:** A border that goes around padding & content

* **Margin:** clears an area outside the border. The margin is transparent.

Colors in CSS

Colors in CSS can be specified by following methods:

→ Hexadecimal colors

→ RGB colors

→ Predefined / Cross browser color names

→ RGBA colors

→ HSL colors

→ HSLA colors

1) Hexadecimal Colors (HEX)

A hexadecimal color is specified with : #RRGGBB

RR ⇒ Red , GG ⇒ Green , BB ⇒ Blue

Values must be btw 00 to FF
                        ↓   ↓
                     Lowest   Highest
                     value   value

Black = #000000

White = # ffffff

2) RGB Colors

Red , Green , Blue

rgb(red, green, blue)

Each parameter defines the intensity of color & can be an integer b/w 0 to 255 on a percentage value from 0% to 100%.

eg rgb(255, 99, 71)

3) Predefined / Cross-browser color names

140 color names are predefined in HTML & CSS color specification.

4) RGBA

Extension of RGB with A as an alpha channel - which specifies the opacity for a color.

Alpha parameter is number between 0.0 (fully transparent) and 1.0 (not transparent at all)

rgba (255, 99, 71, 0.8)

5) HSL colors

hsl (hue, saturation, lightness)

Hue is degree on color wheel from 0 to 360

0 =) Red , 120 =) Green , 240 =) Blue

Saturation is a percentage value 0% means a shade of grey , 100% means full color

Lightness is also a percentage 0% is black , 50% is neither light on dark , 100% is white

6) HSLA colors

Extension of HSL with A as alpha channel which specifies opacity for a color.
Alpha between 0.0 to 1.0

Ex hsla(0, 100%, 50%, 1)

Units in CSS

1) Absolute Unit

fixed"

cm

mm

in

pn

qt

pc

2) Relative Unit

Length relative to another length property

i) Relative Unit to font size

(Parent or self) em :- Relative to font size of element

(noot) rem :- Relative to font-size of root element

ii) Relative Unit to parent element

% =) Relative to parent.

iii) ~~Rela~~ Relative Unit to Viewport

* vw =) ~~100~~ 1% ~~*~~ of width of viewport
* vh =) 1% of height of viewport

Fonts

Sets all font properties in one declaration.

* Font family

  (Specifies the font family for text)

* Font weight (weight of font)

* Font style specifies font style for text.

* External Fonts use google fonts

@ ~~font-face~~ {

~~font-family : myfont;~~

~~src : url (~~

CSS Background

Add background effects for elements

CSS Borders

Allows to specify the style, width & color of an element's border.

Display Property

specifies how an element is shown on web page. ~~to~~ It can change the default display behavior of HTML elements (i.e Inline or Block)

display: inline-block

behavior :- inline

style :- block

when elements need to flow inline but retain block level styling

CSS Position

Determines how an element is positioned in the document.

i) **Static** Default for all elements
elements are placed in normal doc flow
(one after another)
Top, right, bottom, left does not work

ii) **Relative** Positioned relative to its normal position.
Moves when you use top, right, bottom, left, but
its space in layout remains reserved

iii) **Absolute** Positioned relative to normal ancestor. If
none exists, it is positioned relative to <html>
(the document)
Removed from normal flow, doesn't effect other
elements

iv) **fixed** Positioned relative to viewport, so it doesn't
move when scrolling
Great for sticky headers or navigation bars

v) sticky

Min of relative a fixed

Acts like relative unitil you scrobl to certain point, then it becomes fixed

Z-index property

-) Controls the stacking order of elements, determing which elements appears in front of on behind others useful when elements overlap.

-) Higher z-index values appear in front of lower ones

-) only works on elements with a positioning content.

Overflow Property

Specifies what should happen if content overflow's an element's box

This works only with block elements with specified height.

overflow: visible | hidden | clip | scroll | auto;

Opacity

Speifies the transpareny/opacity of an element

the Lower the value, the more transparent

Range :- 0.0 to 1.0