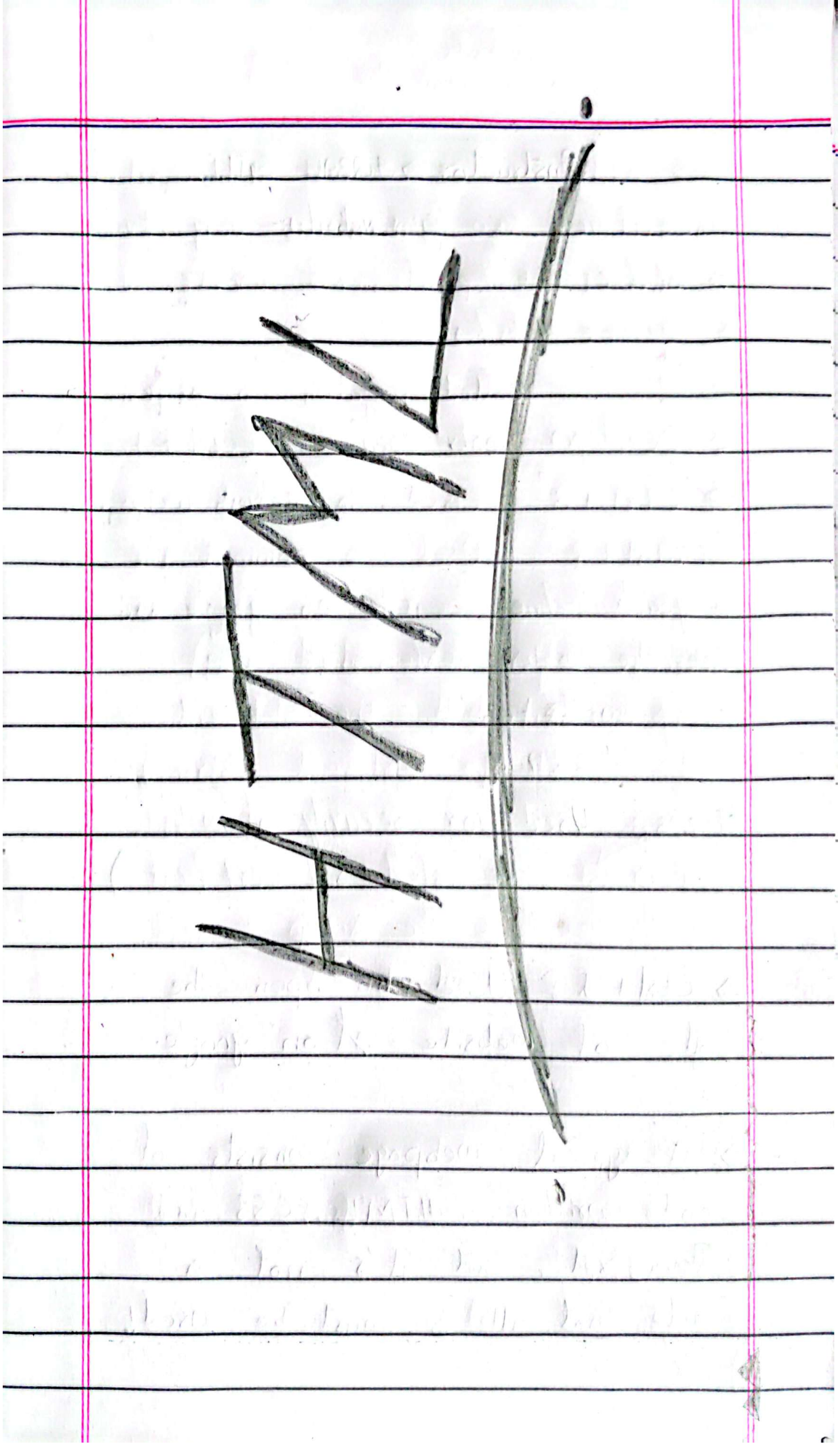


Instructor ⇒ IQBAL GILL

There are two types websites:-

⇒ Static Websites

⇒ Dynamic Websites

⇒ Developer tools open (ctrl+shift+i)

Tool:-

Wayback Machine: Search on google and see the older screen shot of every website.

Https: Used for security of website
(For 's' we install SSL certificate)

⇒ Ctrl+U: Used to open the Code of website on google.

⇒ A typical webpage consists of Code written in HTML, CSS and Javascript, but it's not a rule that all 3 must be used.

⇒ Current version of HTML is '5'

⇒ Current version of Css is '3'

⇒ Current version of react is '21'

⇒ HTML ⟶ Noun

⇒ CS ⟶ Adjectives.

⇒ Javascript ⟶ verb.

Randev :-

Used this term to see the output on search engine.

⇒ Aguler Brackets '< >'

* There are two types of tags:

i. Paired Tag.

ii. self closing tag (without closing tag)

⇒ Ctrl+Z : Used to adjust the lorem paragraph according to screen.

⇒ Alt+ click : Used to multiple cursor

⇒ ctrl+D : Also used to multiple cursors

Website To learn the Coding concept.

=> 'Mdn' is Just like 'w3school' But it is 'mdn' is very advance and Used profesional words and terms.

=> '!' called negation in programing.

=> "Ctrl + R" Used to Refresh webpage

=> "Ctrl + ?" Used to comment line

Strong tag: (Paired tag)
Used to bold the Statement.
e.g Statement

em tag:
Used to change the design to statement in italian format.

e.g.

Lists:-
There are two types of list
i) Unordered list
ii) Ordered list.
iii) Nested list

Unordered list:

e.g.
shirt
glasses

ordered list:

e.g
html 5
CSS3
Javascript

Nested list:-

e.g
Mall

This Sharp in programming.

Attributes :-

Used to extend the functionality of tag function.

e.g google

Attributes.

This is anchor tag used to add links.

=>
 tag Used to break the statement and move to next line.

img tag : (self closing tag).

Used to set image.

e.g <img src="link of img" alt="link">

* sub (subscript)

=> Sub is an element H_2O

=> This is used to write the Notation in html.

* Sup (superscript)

=> Sup is an element (used in 'P' tag)

=> This is used to write the power 2^5

photo, videos, audio

* Here some websites used to download without copyright
  - Unsplash .com
  - Freepik .com
  - Pixaway .com
  - mixkit .com

Forms:-

⇒ Group form Control
(what inputs are part of the form?)

⇒ send the data somewhere
(where does the form data go when submitted)
↗ where the form data send.

```
<form action="/signup" method="get">
</form>
```

Text input:-

⇒ The input element is extremely versatile. e.g set type="text" to create a standard text. It has no closing tag.

```
<input type="text">
```

Button :-

<button type= "submit"> submit </button>

Name :-

<input type = "text" name = "city"

↓

The name that the data will be sent to the server under.

Label :-

<label for = "username"> Enter your username </label>

<input type= "text" id= "username">

=> Match a label element to a form control using the for attribute. It must match the id attribute on input.

Required:-

<input type="password" required />

=> Use the required attribute to mark an input as required!

Range:-

=> Use type="range" to create a range input (slider)

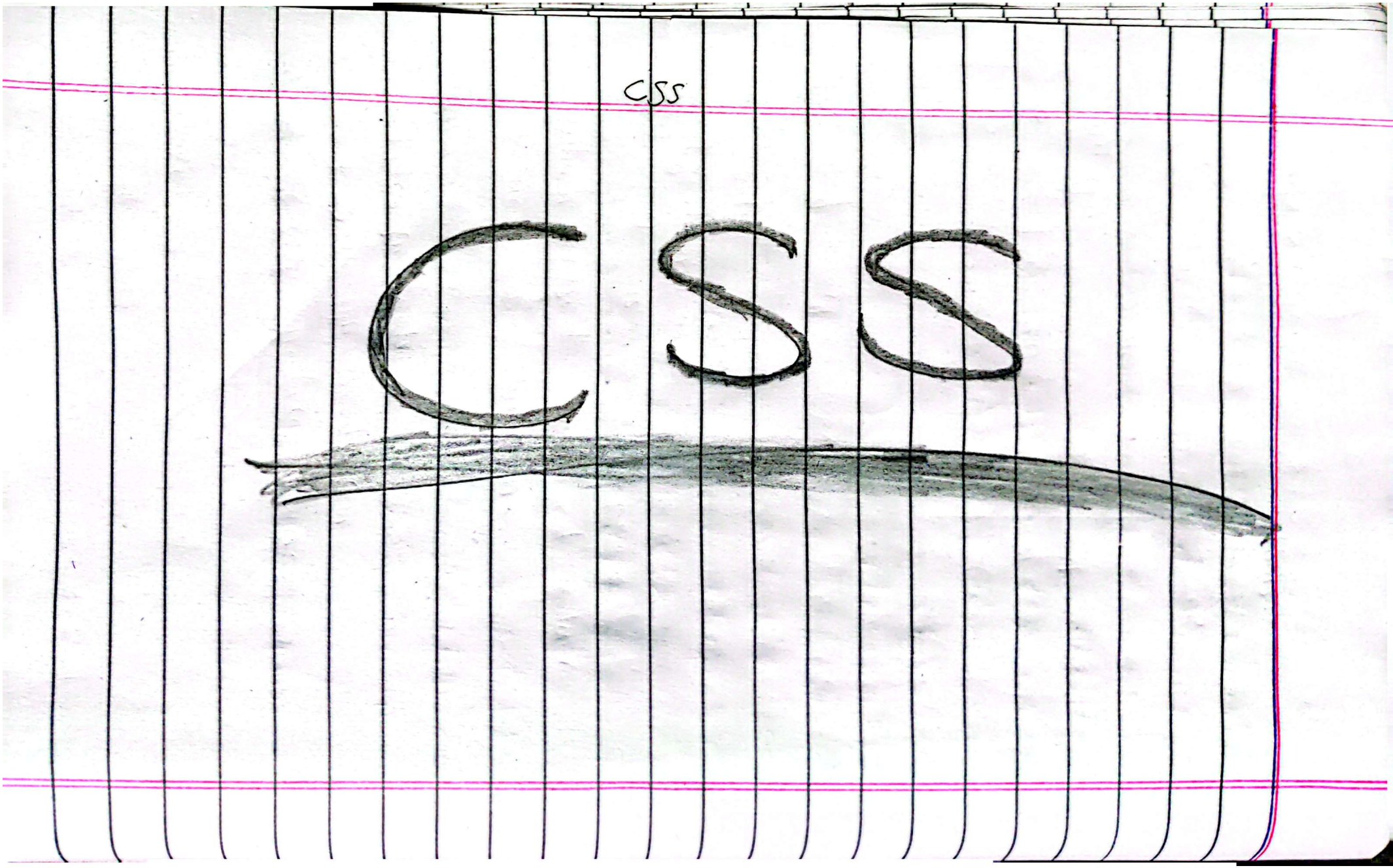
<input type="range" min="0" max="10" />

~~to create a~~

Checkbox:-

=> Use type="checkbox" to create a checkbox element.

<input type="checkbox" name="subscribe" />



(geeksforgeeks) website for more info

★ CSS

★ Choose hexadecimal clow on:-

https://colorshunt.co

⇒ Ctrl + space → (show properties & values)

Google fonts:-
⇒ Use google fonts to use of different style of fonts.

★ Responsiveness :-

⇒ To make a website Responsive following term are used:

⇒ Box Model (imp)
⇒ Grid System
⇒ flex box
⇒ Position property
⇒ Media Quries.

CSS

* CSS (Cascading style sheets)
  - ⇒ Changes text color, size and fonts
  - ⇒ Adjust layout, positioning and spacing
  - ⇒ Adds background, borders and animations etc.

* CSS can be apply to an html documents in three different ways:

1. Inline CSS (Directly inside Html elements)
   - ⇒ Used for quick styling of a single element
   - ⇒ Not recomented in large projects.

eg:

```
<p style="color:red; font-size: 20px">
    This is red text. </p>
```

2. Internal CSS (inside "style" tag in the html "head")
   - ⇒ Best for single page styling.
   - ⇒ Keeps CSS sperate from Html content.

syntax:

```
<head>
    :
    <title> Internal CSS </title>
    <style>
        body {
            background-color: lightgray;
        }
        h1 {
            color: blue;
            text-align: center;
        }
    </style>
</head>
```

Main

3. External CSS (using an external .css file)

⇒ Best for larg wesites.

⇒ Keeps HTml and CSS seperate for better organization.

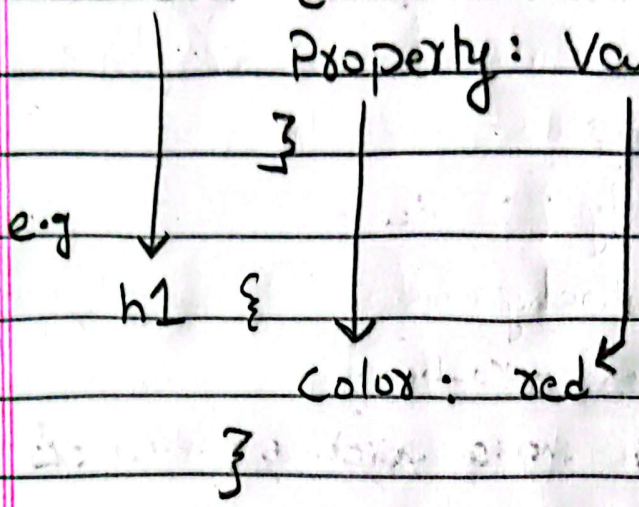
Imp steps:

⇒ Create a seperat CSS file e.g style.css.

⇒ link it in the html file (index.html)

<link rel="stylesheet" href="style.css">

Basic syntax for CSS:-



★ Selectors in CSS:

1: CSS element Selector:

⇒ The element selector selects Html elements based on element name.

E.g: Here <P> is an element.

```
P {
    text-align: center;
}
```

2. CSS id selector:

⇒ The id selector use the id attribute of an html element.

⇒ To select an element

with a specific id, write a '#' character, followed by the id of the element.

E.g:

```
#para1 {
        color: red;
       }
```

3: CSS class selector:

=> The CSS class selector selects html elements with a specific class attributes.

=> To select elements with a specific class, write period (.) character and then write class name.

e.g:

```
.hightlig {
           font-weight: bold;
          }
```

4. CSS Universal selector:

=> The Universal selector (*) selectects all html elements on the page.

Example:
The CSS rule below will effects every html element on the page:

```
* {
    text-align: Center;
    Color: blue;
  }
```

5: CSS Grouping selectors: (List selector)
=> The grouping selector selects all the html elements with the same style definitions.
Example:

```
h1, h2, P {
    text-align: Center;
    Color: red;
  }
```

6. Descendant selector:
=> select all <a>'s that are nested inside
eg `li a { color: red; }`
7. Attribute Selector:-
=> select all input elements where the type is text
eg `input [type = "text"] { width: 3 or Px;`
`color: red;`
`}`

✱ CSS Colors:

=> Color are specified using Predefined color names, or RGB, HEX, HSL, RGBA, HSLA values.

1: RGB Colors: (Red, Green, blue)

=> In CSS, a color can be specified as an RGB value, using formula:

rgb(red, green, blue)

=> Each color Range (0 — 255)

=> e.g

rgb(255, 0, 0) ⟶ Red
rgb(0, 0, 0) ⟶ Black
rgb(255, 255, 255) ⟶ white.

2: CSS HEX Colors:

=> A hexadecimal color is specified with # RRGGBB where 'RR(Red), GG(Green) BB(Blue) hexdecimal integers specify the component of the color.

=> Where RR (red), gg (Green) and BB (Blue) are hexadecimal values Between (00 - ff) (same as decimal (0-255)

Formate

ff6347

3: HSL Colors: (hue, Saturation, lightness)

=> Hue is a degree on the color wheel from 0 to 360. where: (0 → Red, 120 is green, 240 is blue)

=> Saturation is percentage value 0% means a shade of gray and 100% is the full color.

=> lightness is also percentage 0% is black, 100% is whight 50% light or dirk.

Format:

hsl(0, 100%, 50%)

4: RGBA (Red, Green, Blue, Alpha)

=> RGB values Range (0 — 255)

=> Alpha controls transperency 0 = Full transperent, 1 = Fully dark

=> Alpha range (0, 0.1 — 0.9, 1)

Format:

rgba (255, 0, 0, 0.5)

5: HSLA (Hue, Saturation, lightness, Alpha)

=> HUE Range (0° — 360°)

=> Saturation range (0% — 100%)

=> lightness range (0% — 100%)

=> Alpha range (0, 0.1 — 0.9, 1)

Format:

hsla (200, 100%, 50%, 0.5)

In Code:

```
p {
    color: hsla (200, 100%, 50%, 0.5)
}
```

↓

Blue with 50% transparency

* CSS Background properties:-

1: Background-color:-

=> Used to add background effects. and specifiece the background color.

Format:

```
body {
    background-color: lightblue;
}
```

2. Background-image:-

=> Used to specifies the background image.

Format:

```
body {
    Background-image: url("etc.jpg");
}
```

3. background-position:-

=> Used to set the position of Background image.

Format:-

```
background position: x-axis y-axis;
```

- Positioning with keywords:

Format:

```
div {
    background-position: center center;
}
```

Keyword position	Description
left top	Aligns to top-left
right top	Aligns to top-right
left bottom	Aligns to bottom-left
right bottom	Aligns to bottom-right
center center	Centers the image

- Positing with pixel:

```
div {
    background-Position: 50px 100px;
}
```

- positing with percentage:

```
div {
    background-position: 30% 70%;
}
```

4. background-repeat:-

=> It control background image repeat and in which direction.

e.g background-repeat: value;

Value	Description
no-repeat	No repetition (only one image)
repeat-x	repeat only horizontally
repeat-y	repeat only vertically

✱ Margin property:-

⇒ Used to create space around elements, outside of any defined border.

⇒ Margin properties are as follow:

- margin-top (margin-top: 100px)
- margin-bottom (margin-bottom: 100px)
- margin-right (margin-right:: 150 px)
- Yeargin-left (margin-left: 80px)

✱ Padding Property:-

⇒ Used to generate space around an element content, inside of any defined border.

⇒ Margin properties are as follow:

- Padding-top (Padding-top: 50px)
- Padding-right (Padding-right: 30px)
- Padding-bottom (Padding-bottom: 50px)
- Padding-left (Padding-left: 80px)

✱ Shorthand property:

⇒ Used both for Margin & Padding

⇒ It is specify all the values in one property.

e.g:

top right bottom left

margin / Padding: 25px, 50px, 75px, 100px;

★ hight and width property:-

⇒ hight and width property used to set the hight and width of an element.

Format:

```
div {
    hight: 200px ;
    width: 50px ;
}
```

} also used in percent %.

★ Border Properties: (important ones)

⇒ Used to apply the border on content.

Syntax: selector{
border: width style colors }

⇒ common border properties:

Property + value	Discription
border-left: 2px;	sets the thickness of border left
border-style: dashed;	Defines the type.
border-color: blue;	sets the border color
border-radius: 10px;	rounds the border.
border-bottom: 2px	Sets the thickness of border bottom
border-width: 2px	Apply on all four sides

⇒ Also apply short hand Property.

border-width: 2px 4px 6px 8px

To Right Button left

* CSS Text properties:-

⇒ text-align
⇒ font-weight
⇒ text-decoration
⇒ line-height
⇒ letter-spacing

- text-align:

⇒ Use to set the horizontal alignment of text.

Syntax:

```
text-align: value;
```

⇒ Values of text-align:

Values	Description
text-align: left	Aligns text to left (Default)
text-align: right	Aligns text to right
text-align: center	Aligns text to center
textalign: Justify	Align text both left or right edge.

- font-weight:

⇒ Used to control boldness of text.

Syntax:

```
font-weight : value;
```

⇒ Values of font-weight;

Values	Description
font-weight: normal;	Default text (400)
font-weight: bold	Makes text bold (700)
font-weight: bolder	Increase baldness compared to the parent element
font-weight: lighter	Decrease boldness compared to parent element

⇒ Values assign by numeric:

Range of baldness text:

100 ——→ 900

e.g. font-weight: 900;

font-weight: 700;

- text-decoration :-

⇒ Used to add or (remove) visual effects like Unedline, overline, line-through and blinking on text.

Syntax: text-decoration: Value;

Values	Description
text-decoration: none	Remove any text decoration
text-decoration: underline	Adds an underline below the text
text-decoration: overline	Add a line above the text
text-decoration: line-through;	Add a strikethrough (cross out line)
text-decoration: blink;	Makes the text blink

• line-height :-

=> Used to control vetical spacing between lines of text.

Syntax:

```
line-height: value;
```

=> Values of line-height;

1. normal:

=> Default line height.

e.g `p{line-height: normal;}`

2. number:

=> Multiplier of font size.

e.g
```
p{
    font-size: 16px;
    line-height: 1.5;
}
```

Its mean 16px X 1.5 = 24px (spacing)

3. length:

=> Fixed hights using units (px, em, rem)

e.g `p{ line-height: 30px}`

4. Percentage:

=> Percentage of font size.

e.g `p{ line-height: 180 %;}`

• letter-spacing :-

⇒ Used to control the space between characters in text.

Syntax:

letter-spacing : value;

⇒ for increase spacing : (+ive, -ive values)

e.g h1 { letter-spacing : 3Px; }

⇒ For decrease spacing : (+ive, -ive values)

e.g h1 { letter-spacing : -1Px; }

★ font-size :-

⇒ Used to control the text appearence (big or small)

Syntax:

```
p {
    font-size: 16 px; (standard size)
    font-size: 52px; (large the text)
}
```

★ Font-family:-

=> Used to define the typeface (text style) for text.

Syntax:-

font-family: "font Name" ;

eg. Calibri

=> We use google fonts for typeface.

=> almost 1800 fonts available on google fonts.

★ Pseudo Classes:-

=> Keyword added to a selector that specifies a special state of the selected element.

=> Used to style element based on their state or position without modifying the HTML structure.

=> Pseudo Classes coritten after the selector with a colon(:) :

Syntax:

```
Selector : pseudo-class {
        Property : value;
    }
```

=> Common Pesudo Classes:-

1. :active
2. :checked
3. :focus
4. :first-child
5. :hover
6. :not ()
7. :nth-child()
8. :nth-of-type()

1. :active :-

⇒ Used to apply styles to an element when it is being clicked or activated by the user.

e.g. button:active {
background-color: red;
color: white; }

2. :checked :-

⇒ Used to styles a checkbox or radio button.

e.g. input:checked {
outline: 2px solid red;
}

3. :first-child :-

⇒ Used to styles the first child element inside a parent.

e.g. p:first-child {
color: black; }

4. :focus :-

⇒ Used to styles an element when it gain focus.

e.g. input:focus {
border: 2px solid black;
outline: none; }

5. :Hover:-

⇒ Applies styles when the user hovers over an element.

e.g. button: hover {
background-color: black;
color: white; }

6. :not( ):-

⇒ styles all elements except the specified ones.

e.g p:not(.special) {
color: black; }

7. :nth-child( );

⇒ Styles a element based on its position.

e.g li:nth-child(2) {
color: green;
}

8. :nth-of-type( );

⇒ Used to select elements based on their type (tag name) and position within their parent and apply style to element of the same type.

e.g p:nth-of-type(2) {
color: green; }

✱ Pseudo Elements:-

⇒ Keyword added to a selector that lets you style a particular part of selected elements.

1. ::after
2. ::before
3. ::first-letter
4. ::first-line
5. ::selection

⇒ Used to styles specific parts of an element without modifying the HTML structure.

⇒ It allows you to apply styles to parts of an element.

Syntax:

```
Selector::Pseudo-element {
        Property: value;
    }
```

1. ::after:-

⇒ Insert content after an element.

e.g

```
h1::after { Content: "★";   (emoji)
            Color: gold;
          }
```

2. :: befor:-

⇒ Insert content befor element.

e.g. h1:: before {

content: " 😊 "

color: orange; }

3. :: first-letter :-

⇒ Styles only the first letter of a block of text.

e.g. P:: first-letter {

font-size: 2em;

color: red }

4. :: first-line :-

⇒ Styles only the first line of a block of text.

e.g. P:: first-line {

font-weight: bold;

color: blue; }

5. ::Selection :-

⇒ Changes the style of highlighted (selected) text.

e.g P::selection {

background-color: Yellow;

Color: black;

}

★ Inline Elements :-

=> Inline Elements fit in alongside other elements.

e.g My name

★ Block Elements :-

=> Block level element take up a whole "block" of space.

e.g my name

* Display Property:-

1 Inline:- (stay on same line)

=> width & height are ignored: Margin & padding push elements away horizontally but not vertically.

e.g span {
display : inline;
background: yellow; }

2 Block:- (take the full width of page)

=> Block elements break the flow of a document. width, height, margin & padding are respected.

e.g div { display : block;
background blue; }

3. Inline-block: (allow setting width, height etc)

=> Behaved like an inline element except width, height, Margin & Padding are respected.

e.g button {
display : inline-block;
width : 100 px;
height : 50 px;
background : green;
}

★ Specificity:-

⇒ Determines which styles take priority when multiple rules target the same element.

⇒ Specificity Calculation:

Most specific

1. Inline Styles
2. ID Selectors
3. Class, Attributes, Pseudo-Class
4. Type, Pseudo-elements

least Specific

⇒ Important keyword:-

It decline the specificity rules and follow the important keyword.

e.g.

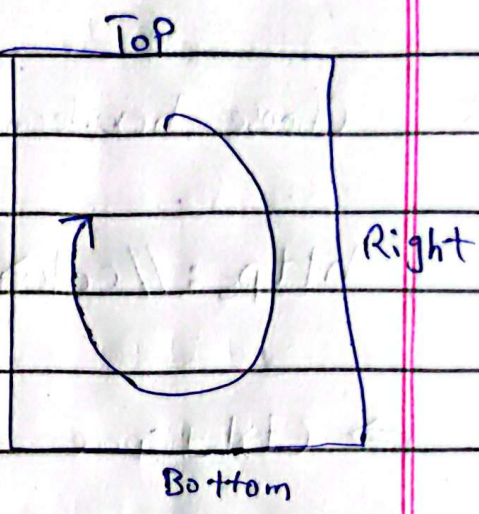
```
.h1 {
    color: blue !important;
}
```

(generally not a good idea)

(Library of loaders, buttons, check...

imp ⇒ Uiverse.io → Used for loaders buttons.

Properties apply:-



⇒ Margin
⇒ Pedding
⇒ Border.

Top Right Bottom left.
e.g margin : 20px 5px 15px 10px.

Tool:-

Eyedropper :-

It is a Chrome extention which used to find colour values on screen.

design-seeds:-

⇒ design-seeds.com.
⇒ Provide nature clours.

devdocs:-

⇒ devdocs.io
⇒ Used to prepare documentation.

* emojipedia.org:-
=> emojipedia.org.
=> Provide emojies for websites.

* Flaticon:-
=> flaticon.com.
=> Provide icons like insta, facebook etc.

* Caniuse:-
=> caniuse.com.

* Htmlwasher:-
=> htmlwasher.com.
=> It will minify the code.

* W3 validator:-

* Enjoy CSS:-
=> enjoycss.com
=> Easily adjust shadows, gradients borders, transitions, and animation

* CSS3buttongenerator.com :-

⇒ It will generate buttons.

* themeforest.net :-

⇒ This website have websites.

⇒ These website are Paid.

Bootstrap

Bootstrap.

* Important topic in Bootstrap:-

=> Bootstrap 5

1. Containers
2. Grid + Columns
3. Typography
4. Images + Tables
5. Form Sections
6. Components
7. Background + Borders + Section
8. Display + position.

* Important website of bootstrap for Gk:

=> getbootstrap.com.

* Tool:-

Animation:-

=> animation.css (animation.style).

-> Used for add animations in website.

Start bootstrap.com.

* Bootstrap :- (Latest version 5)

=> It is a popular CSS framework used to create responsive and mobile-first web design.

=> It provides pre-built CSS and Javascript file.

* Bootstrap Cdn: (content delivery network)

=> CDN is the easiest way to add bootstrap to your project without downloading any file.

=> Bootstrap provides CSS and JS file through a CDN.

=> Get files links on getbootstrap.com website.

e.g <head>

:

<title> bootstrap </title>

(CSS link Paste here)

(/head)

<body>

:

(JS link Paste here)

</body>