Hyper Text Markup Language.

`<!Doctype.html>` → specifies this is html 5 doc.

`<html>` → root of an HTML page

`<title>` sunny `</title>` ⇒ title of web page

`<body>`

`</body>`

`</html>`

---

comments in HTML

`<!-- comment -->`

case sensitive

`<H1>`, `<h1>` ⇒ Same

Heading Tag

`<H1>` सबसे बड़ा heading

`<H2>`

`<H3>`

`<H4>`

`<H5>`

`<H6>` → सबसे छोटा Heading

`<P>` --- `</P>` ⇒ Paragraph

Line Break tag `<Br>`
act as empty tag. no need of closing Tag.

Horizontal Line

`<hr>` ___

Preserve formatting
जैसे Text लिखा है वैसे उसी format में लिखाएगा

`<pre>` This is
       sunny
`</pre>`

1989 - Tim Berner Lee.
invent — WWW

1991 _____ HTML

1993 - Dave Roggett drafted
HTML+

---

non breaking space

`<P>` An example "12
       Angry   Ma </P>

→ Output ⇒ An example "12 Angry.

HTML link

`<a href = " specify link here ">`
       click to go to link `</a>`

Image.

`<img src = " Location of image"
       width ="200px" height = "200px"
       alt = " There must be some issue with
       the image."

जब image load नहीं हो पाया या रहा तब ये text लिखा हुआ दिखाएगा

`<blockquate> </blockquate>`
great people के quote लिखने के लिए Text को centre में ले आता है But छोटे quates पर उतना effective नहीं है।

`<q> -- <q>` ⇒ " " sign लगाने के लिए

`<abbr title = " Hyper Text Markup
       Language "> HTML </abbr>`

`<address> ... <address>`
address को italic कर देगा

`<cite> -- </cite>`
       ↳ Text को Italic But author name likhne के लिए

`<bdo dir ="ltr"> Sunny </bdo>`
`dir ="rtl">`
Text की ymmus उलटी लिखाता के लिए

`<b>` ⇒ bold Display Text
`<I>` = itallic
`<U>` = underline
`<small>` = Text small.
`<big>` = Text big
`<ins>` = underline के लिए
`<Del>` = ~~sunny~~ st.
`<strong>` = bold टेक्स्ट के लिए
`<em>` – itallic के लिए
`<sub>` = subscript Text normal से निचे आता लिखता
`<sub>` = superscript = Text normal से upper चला जाता

☆ `<tt>` -- `</tt>` . Monospaced
font same width height of Text
`<dfn>` ⇒ Define टेक्स्ट के लिए

`<Font size ="12" color ="Red"`
`Face =" Times"> </FONT>`

`<div>` is block level
`<span>` inline level.

List
1 glossary List =
DL = Defination list
DT = " Term
DD = " Data

`<DL>`
`<DT> Coffee </DT>`
`<DD> Black hat coffee </DD>`
`</DL>`
output ⇒ Coffee:
Black hat coffee

2. Unordered list
type = circle, Disc, square,
`<UL type = "square">`
`<LI> country</LI>`
`<LI> India </LI>`
`<UL>`

3. Ordered list
type ="A", "a", "1"
style ="List-style-Type::
lower-roman
alpha
upper roman, alpha
`<OL type -- >`
`<LI> countr </LI>`
`<LI> India </LI>`
`<OL>`

Table
`<TABLE>`
`<TR>` = Table Row
`<TH>` = Table heading
`<TD>` Table Data.

Table Attributes
~~Table A~~
Alignment -- Left / Centre / right
Border, width,
cellspacing, cell padding

`<Table Alignment = "centre" Border=`
`"0" width ="100 cellspacing = "2"`
`cell padding = "4">`

Merging cell
Rowspan = used to merge cell in a rows
`<TD Rowspan = "2" >`
2 row column को Merge कर देता

calspan = used to merge cell in a column.
`<TD calspan = "2")`
2 color Row को merge कर देता

FORMS

Three main Attributes
1. Name = used to identify form.
2. Action = point towards the which page data is sent.
3. Method = get = To get the form details
   = post = do not show details of forms.

```
<fieldset>
  <legend> Personal Details </legend>
```
┌─ Personal Details ─────────┐
│   │
│   │
└────────────────────────────┘

```
<input>        <input id = "-.">
<label> = To give containt content name.
<label> First name </label>
<label for = "-.">  → Just like id in input
```

`<select>` dropdown list बनाने से
`<option>` value </option>

`<textarea>`

`<button>` click me </button>
`<input id = browser <button>`
`<datalist>` dropdown list बनाने
`<option>` value </option>
`<output>` To show output

── input type
`<input type = "button">` [Butto]
checkbox ☐
color [Red]
date [____ ⊡] phasedate
date time-local [- ⌄ ⊡]
email
file = choose file कोई भी option
hidden = data not shown to user but submitted to form

`<input type = "month"> month year.`
number [__⌄] ch
password
radio ○ ○ ○
range ┼──╫
reset
search
submit
tel 123-45-67.
text ⇒ [_____]
time
url
week

```
<input type = "text" name = "name"
required = "  "  placeholder = "
                 enter your name"
```
↑
इस field को जरूरी fill करना

autofocus = " "
↑
cursor को direct वहाँ पर ले जाता है

checked = " " ← default checked रहेगा.
↑
check box
Radio में

readonly = "welcome"
↑ सिर्फ Read कर सकते है edit नहीं

disable = field है तो disabled हो जाता है।

min, max = Range set कर सकते
maxlength = character की range set कर सकते
size
multiple

Frames
`<frameset rows =`
10% 80% 10%
┌────────────────────────┐
│   │
├────────────────────────┤
│   │ main
└────────────────────────┘ batto
`<frame name = "Tab" src =`

Favicon adding logo to the brouser

```
<title>My page title</title>
<link rel="icon" type="image/x-
   icon" href="/image/favicon
   .ico">
</head>
```

Emoji

```
<meta charset="UTF-8">
```

☺ 128516
♡ 128151

Media

HTML support MP4, WebM, Ogg video
            WAV, MP3, Ogg audio

video element

```
<video width="200" height="200">
<source src="Movie.MP4".
        type="video/MP4">
your brouser doesn't support
   video element.
<video      control - play
autoplay, muted   pause, download
                  volume.

<audio control>
<source src="horse.MP3" type=
        horse"
</audio>
      youTube.
<iframe width="200" height="200
src="url of youTube">
</iframe>
```

CSS

Cascading Style sheet

Syntax. property value .

h1 { color: Blue; }

↑ selector   declaration

Selector.

element selector.

P {
   text alignment: centr;
}

id selector.

#para1 {

}

class selector.

. class1 {   }

Universal Selector.

* {   सब के लिए काम करता है।
}

group selector.
h1, h2, P {

}

Types of CSS

Inline CSS

<h1 style = "color: blue, text-alignment:...

↳ Inline CSS ⇒ highest priority

External CSS

<style>
body { backgrou-color : Red}
</style>
</HTML>

External CSS

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

style.CSS
body {

}

comment
/* ---- */

colors

background-color: "---;

Text-color.

<h1 style color: ---;

 .border: 2px solid Tomato;

color values

rgb (255, 99, 77)

ff 63 47

hsl (9, 100% 64%)

BACKGROUND

background-color: lightblue;

opacity: 0.3; (transparency)

background-image: url("---")

background-repeat: repeat
                          repeat-x
                          repeat-y
                          Non-repeat

background-position: right top:
                          left Top
                          anything

shorthand

background: cyan url(: --)
                 no repeat right Top)

background-attachment: fixed
                     image fixe होगी
                          ↳ scroll करने पर
                     : scroll
                          scroll हो जायेगी

background-size: size of backgr u
  -origin   origin of image

Border Style

border-style: dotted

dashed

solid

double

groove ← ligh color.

ridge → ligh

inset

outset

hidden

dotted dashed solid double; mixed

border-width: 2px, medium, thick,

thin

: 5px 20px
Tab button.

5px 10px 15px 20px
Tab righ batto left

border-color: solid, green

border-tap-style:
right - style
botton - style
left - style

border-radius: 5px

Margin

margin-tap : 100px.
- right
- bottom
- left

margin: 10px 20px 30px 40px
Tab righ botto lef

10px 20px 30px
Tab righ botto
left

10px 20px
Tab right
batto lef

10px
all four side

: auto
contre st tedn

change the text
direction of element

Padding

same margin जैसा ही

padding - Tab
all property

div {
  heigh : 200px
  width : 200px or Max-width: 50%
}

Outline - Style.

outline-style : dotted
dashed
solid
double
groove
ridge
inset
outset
none
hidden

outline-width : 2px, medium, thick

outline - color : green; thin

→ same property

Offset → add space b/w an
outline and the edge / border of
element.

Outline-offset : 15px

SUNNY

CSS Text.

<h1> SUNNY >

h₁ {
  color : green; - Text color green
  background-color : --- ; so,
  text-align → horizontal alignme
  text-align : contro text left righ
  Justify equal width
  from all side
  text-align-last :
  how to align last line of text.

p{ direction : rtl
unicode-bidi : bidi-override;

vertical-align : baseline
              : text-top
              : text-bottom
              : sub
              : super

Text-decoration
text-decoration-line : overline
                     : line-through
                     : underline
                     : overline Underline

text-decoration-color : red
              ─── line का red करने
              के लिए.

Text-decoration-style : Solid
                        dotted
Sunny   wavy
Sunny

Text-decoration-thickness : 25%.
                            Auto
                            5px.

text-decoration : none.
text-transform : uppercase
               : lowercase.
               : capitalize

tex-indent : 50px.
First line of a text ot space
provide करना।
letter-spacing : 5px
     letter के बिच space.
line-height : 0.8;
     → दो line के बिच space.
word spacing : 10px
white-space : nowrap;
Shadow effect on text
text-shadow : 2px 2px 5px red;
          └──┘   │   │
        shadow blur color

a: links
a : visited
a: hover
a: active

font-family : Arial, Sans-serif;
font-style : normal
           : italic
           : oblique (similar to italic

font-weight : normal
            : bold
font-variant : normal
             : small-caps
        font को छोटा करने के लिए

font-size : 20px : size of text
google font

< link rel = " stylesheet " href = "
google font >

font-family : "Sofia"

position : static = is not position
                normal flow of the page.
relative = relative to normal
                    position.
fixed = एक जगह fix
Sticky = scroll करने पर उस fix रही
position : absolute

2-index
  -1 = Image text के पिछे चलाजाता है
   2 = तब अधिक ऊपर आजाता
   1 = normal

Overflow := control what happen
to content that is too big to fit into
area.
overflow : visible
         : hidden (content hide
                  हो जाता।
         : scroll] same type का
         : auto]
overflow -x : hidden
         -y : scroll

layout

float : right
: left
: none
: inherit (inherit float value to its par..

box - sizing : border-box;

display: inline (line के अन्दर

display: inline-block display का रूप)

display: block

Flex Box.

Flex Box make it simple to align items vertically and horizontally using rows and columns. item will flex to different size, space, direction. It make responsive design

display : flex

flex - direction : row, row-rev
column, column-rev

Justify - content : flex-start
flex-end
center
space-around

align - items : center
stretch
baseline
flex-end
start

align - self : start
end

MongoDB

To show all our mongo file
> show dbs;

To create database
> use name_of_database;

To check we are in which db
> db;

To create ~~sol~~ Table for collection in mongo
> db. createCollection ("demoo1")

To insert data
db. student. insert ({ "name":
"admin"})

MongoDB always follow JSON web structure.

To show all collection.
> show collection

To show what data is present inside any particular collection
> db. student. find();

• db. createCollection will provide some external capping but this is not possible in db. student. insert →method()
db. createCollection ('student'. {capped: true, size: 524888 BB, max: 5})
we can clarify max character, size etc in create collection but not possible in insert

---

Inserting Multiple Records in single short

var my Emb = [{ "Emd id": 1, "EmbNa":
sonn"},
{Emb#: 2, EmbN: R}]
db. newEmb. insert (my Emb);

Update Table
db. new Emb ({ "EmbName": "sonn",
{ $set: {"EmbName": "sonnylu}}
db. new Emb. find ();

• It will update only the first value ~~or~~ not all In order to update all values.

In order to apply multiple Upda
db. new Emb. update ({"EmbNam": "chang},
{ $set: {"Embname": "new Manager"}}
{multiple: true}).
It will update all data.

UPsert (It will update the existing one If it is not found then it create new object)
db. newemb. update ({{"EmpNam": "da"},
{$set: {"EmpNam": "new da"}}, {
upsert: True })

Remove (जितना भी EmbName admin now होगा वो सारे remove हो जायेगा |
db. new Emb. remove ({"EmbName": "admin"
सारे इस Name को remove हो जायेगा. })

To find single Name.
db. newEmb. find ({"EmpName": "admin"})

इस particular name के employee
दिखा देगा

---

db. newEmp. find (). limit (2);

2 employe का data ही दिखायेगा

---

db. newEmp. find(). sort ({"EmpName": -1"})

-1 ⇒ for descending order
+1 ⇒ for ascending order.

To find in nested object
db. newEmp. find ({"address.city": {
'bengaluru"})

Condition

gt ⇒ greater than
Lt = less than
db. newEmp. find ({ "Emp Id". {$gt : 1}};

In between.
db. newEmp. find ({ empId: {$in [3,6]}});