* css tutorial *

- css is the language we use to style a web page.
- cascading style sheets
-

* css syntax -

- A css rule consists of a selector and a declaration block.

```
                    declaration
      declaration
h1  { color: blue ; font-size: 12px ; }
        ↑      ↑
   property   value
```

* css selectors *

A css selector selects the HTML elements you want to style.

- simple selectors (select ele based on id, class, name)
- combinator selector (relationship bet① them)
- pseudo-class selectors (based on certain state)
- pseudo-elements selectors (select and style a part of an element)
- Attribute selectors (select elements based on an attribute or attribute value)

* → will select all html element on document.

* How to add css -

1) External <link rel = "stylesheet" href = "" >
2) internal
3) inline

* css comments -

```
/* . . . */
```

* css colors → rgb, Hexa, hsl,
0-255 # hsl(,)

color: red;

border: 1px solid green;

background-color: light-gray;

* css backgrounds *

The css background properties are used to add background effects for elements.

background-color - sets the background color of an elem

background-image - use an image which does not disturb ele

background-repeat - sets how a background image will be rep

background-attachement → bk image is fixed or scrolls.

background-position - sets the starting position of back gr

background (shorthand property) - sets all bak prop in one declaration

background-origin - specifies where the background images is/are positioned.

background-size - specifies the size of the backgrou image(s)

* css borders -

The css border properties allow you to specify the style, width, and color of an elements border.

- css Border style -

dotted
dashed
solid
double
groove
ridge } effect depends upon border-color value
inset
outset
none
hidden

- border-width
- border color
- border-radius

$$\begin{tikzcd} \text{top} & \text{right} & \text{bottom} & \text{left} \arrow[l, phantom] \end{tikzcd}$$

* CSS margins *

margin : top right bottom left

margin : auto; → horizontally center the element

margin : inherit; → inherited from parent element

Note – Negative values are ~~not~~ allowed.

* CSS margin collapse –

sometimes two margins collapse into a single margin.

margin collapse

* ~~CSS~~ CSS padding –

padding is used to create space around an element's content, inside of any defined borders.

padding : top right bottom left ;

length → px, pt, cm etc

% →

inherit.

Note – Negative values are not allowed.

does not use width & padding together.

* CSS height, width, max-width –

max-width ⟶ sets max(m) width

height + width values –

auto – default.

length

%

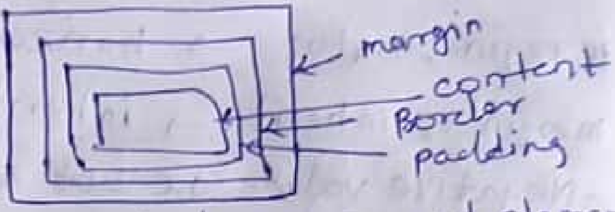
initial → sets the height/width to its default value.

inherit

* CSS Box Model -

All HTML Elements can be considered as boxes.

Imp - when you set the width and height properties of an element with css, you just set the width and height of the content area. To calculate the full size of an element, you must also add padding, border and margins.



* CSS outline -

An outline is a line that is drawn around elements, outside the borders, to make the element "stand out"

outline-style -
outline-color
outline-width
outline-offset
outline

Note - Outlines differs from borders! unlike border, the outline is drawn outside the element's border and may overlap other content. Also, the outline is Not a part of the element's dimensions; the element's total width and height is not affected by the width of the outline.

* CSS text →

color - Note - High contrast is very imp for people with vision problems. So, always ensure that the contrast bet^® the text color and the background color is good!

* Text alignment

text-align ⟶ set the horizontal alignment of the text.

text-align-last ⟶ property specifies how to align last line of a text.

direction and unicode-bidi properties can be used to change the text direction of an element.

vertical-align : sets the vertical alignment of an element.

baseline, text-top, text-bottom, sub, super.

* css text decoration -

text-decoration-line : overline, underline. line-through

text-decoration-color :

text- decoration-style : (none) - to avoid underline of link.

text - transform:- uppercase, lowercase, capitilized.

* css text spacing -

i) text-indent - property is used to specify the identation of first line of a text

ii) letter-spacing - spaces bet^n characters in a text

iii) line-height - line height

iv) word-spacing - space bet^n words in a text

v) white-space : (nowrap) ⟶ single line output

* css text shadow ⟶

text-shadow property adds shadow to text

* CSS Fonts

- choosing the right font for your website is Importan

* Generic font families →

serif
sans-serif
monospace
cursive
fantasy

font-family - specify the font of a text.

* font style -
font-style : normal , italic, oblique

* font weight -
font-weight : normal, bold ;

* font-variant → property specifies wheather or not a text should be displayed in a small-caps font.
↓
normal
small-caps

* font - size
↓
10px
1em = 16px.

Responsive
10 ~~[illegible]~~ vw viewport width :

* Google font
<link rel = "stylesheet" href = "googleapi" >

* CSS icons →

* CSS link

link Buttons

add padding.

a : link
a : hover
a : active
a : visited

* CSS list

list - style - type : circle, square, disc,

list - style - image : url('path');

list - style - position : outside, inside,

list - style - type : none; }
margin : 0 } Remove default
padding : 0 } settings.

list - style : type position image → Shorthand

* CSS Tables →

Table border

```
table, th, td {
        border : 1px solid green;
}
```

full width table → width : 100%

Double borders → table, th, td all have seperate border

border-collapse : collapse ;

* table size

width height

* table alignment.

text-align: left, right, center

vertical-align: bottom, top.

* css table style →

tr:hover {background-color: coral; }

tr:nth-child(even) {background-color: # }

* Responsive table –

Add a container element (like <div>) with $\boxed{\text{overflow-x: auto;}}$ around the <table> element to make it responsive.

caption-side → Specifies the placement of a table caption.

empty-cells → specifies whether or not to display borders and background on empty cells in a table

table-layout → sets the layout algorithm to be used for a table.

* css layout – the display property.

The display property is the most important css property for controlling layout.

- The display property specifies if/how an element is displayed.

- Every HTML Element has a default display value depending on what type of element it is. The default display value for most elements is block or inline.

Display: none ; block ; inline.

is commonly used with javascript to hide and show elements without deleting and recreating them. Take alook at our last example on this page if you want to.

inline →

visibility : ~~none~~ $\overset{\text{hidden}}{}$, → It takes space

display : none. ↳ It does not takes space

* css - max width → useful for small devices

* css layout - The position property

The position property specifies the type of positioning method used for an element (static, relative, fixed, absolute or sticky).

All css positioning properties -

*css z-index → $\boxed{\text{sets the stack order of an element}}$

*css overflow →

The cssoverflow property controls what happens to content that is too big to fit into an area.

$\boxed{\text{overflow: visible hidden scroll auto}}$

* css float + clear

float : left, right, none, inherit.

*The clear property

- when we use the float property, and we want the next element below (not on right or left), we will have to use the clear property

clear : none, left, right, both, inherit

* CSS - inline-block →

↑ display: inline, the major difference is that compared to

display: inline-block; allows to set a width and height of the element.

Also, with display: inline-block, the top and bottom margins / paddings are respected, but with display: inline they are not.

* CSS align →

1> Center Align Elements →

To horizontaly center a block element, use margin: auto;

Note - center aligning has no effect if the width property is not set.

2> Center an Image

To center an image, set left and right margin to auto and make it into a block element.

```
img {
  display: block;
  margin-left : auto;
  margin-right : auto;
  width : 40;
}
```

3> Left & Right align - using position

* css Combinators —

-A combinator is something that explains the relationship betⁿ the selectors

- descedant selector (space)
- child selector (>)
- adjacent sibling selector (+) [immediately following)
- general sibling (~)

* <u>css</u> <u>pseudo-classes</u> —

<u>What are pseudo-classes</u> →

-A pseudo-class is used to define a special state of an element.

Note - a:hover MUST come after a:link & a:visited in the css defination in order to be effective. a:active MUST come after a:hover in the css defination in order to be effective. Pseudo-class names are not case-sensitive.

* <u>css</u> <u>pseudo-elements</u>

-A css pseudo-element is used to style specified parts of an element.

syntax

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

::first-line ⟶ special style to first line of a text.

```
p::first-line {
      color:
      font-variant:
}
```

Scanned with CamScanner

* CSS opacity -:

opacity : 0.5; $\boxed{0 \text{ to } 1}$

* CSS position → property.

The position property specifies the type of positioning method used for an element (static, relative, fixed, absolute or sticky).

- There are five different position values -
  1) static →
  2) relative →
  3) fixed →
  4) absolute →
  5) sticky →

- Elements are then positioned using top, bottom, left and right properties. However, these properties will not work unless the position property set first. They also work differently depending on position value.

$\boxed{\text{position: static}}$;

HTML elements are positioned static by default.

.position: relative

An element with position: relative; is positioned relative to its normal position.

Setting the top, right, bottom, and left properties of a relatively-positioned element will cause it to be adjusted away from its normal position. Other content will not be adjusted to fit into any gap left by the element.

```
div.relative {
    position : relative;
    left     : 30px;
  border : 1px solid green; }
```

Scanned with CamScanner

position:-fixed

An element with position:-fixed; is positioned relative to the viewport, which means it always stays the same place even if the page is scrolled. The top, right, bottom and left properties are used to position the element.

- A fixed element does not leave a gap in the ~~top~~ page where it would normally have been located.

```
div.fixed {
      position: fixed;
       bottom: 0;
       right : 0;
       width : 300px;
       border : 3px solid green;

}.
```

position: absolute →.

- An element with position: absolute is positioned relative to the nearest positioned ancestor (instead of positioned relative to the viewport, like fixed).
- However; if an absolute positioned element has no positioned ancestors, it uses document body, and moves along with page scrolling.

Note - Absolute positioned elements are removed from the normal flow, and can overlap elements.

position: sticky ✗

- An element with position: sticky; is positioned based on the user's scroll position.
- A sticky element toggles between relative and fixed, depending on the scroll position. It is positioned relative until a given offset position is met in the viewport - then it 'sticks' in place (like position:-fixed).

```
div.sticky {
  position: sticky;
  top: 0;
  background-color: green
  border: 2px solid green;
}.
```

positioning text in an Image: