

CS S

Handwritten

-NOTES-

@cookwithimraan .

⇒ Introduction :-

HTML is a standard for "cascading style sheeting". It is generally used with HTML to change the style of the webpages and user interface.

It helps browser how to display

HTML Document [appearance, styling, formatting]

It is used with any kind of XML document including XML, XUV, SVG.

⇒ Adding CSS in HTML Doc :-

Basically there are 3 types to add CSS file in HTML document.

(i) inline CSS

(ii) internal CSS

(iii) external CSS

① Inline CSS :-

we use '`style`' attribute in HTML elements. writing style attribute we actually write styling properties. around of

Syntax :-

`<p style="style property"> </p>`

here we can see that

Example :-

```
<body>
  <p style="color: Blue">content</p>
```

`</body>`

This is inline CSS. (i)

② Internal CSS :- (ii)

we can use the internal CSS by integrating the `<style>` element in the `<head>` section of HTML Document.

Syntax :- Example → <https://codewithmoan.com>

22 <head> 22 22 (normal)

position: absolute; **style** border: 1px solid black; width: 100px; height: 100px; background-color: white; left: 50%; top: 50%; margin-left: -50px; margin-top: -50px; border-radius: 50%;

Color : Blue ~~orange~~

Algorithm for solving linear equations

Amada el ~~style~~ ~~style~~ ni van mane

</head>

the idea of the use of the term 'language' in the sense of the language of the people.

function with <body> inside based with of code

< p > code with imraan </p >

<body>

output

<heart>

Code with Pmraan

→ color changed to Blue.

③ External CSS :-

External CSS is separate CSS file that can be accessed by creating a link within the head section of the webpage.

multiple webpages can be use the same link to access the style sheet.

we need to link an external style sheet in the head section of the HTML document.

Example :-

Name of CSS file is written here.

<head>

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

</head>

<body>

<p> code with imraan </p>

</body>

HTML file

- Selectors -

→ Selectors :-

used to HTML element to which we want to style.

① Element selector :- selector which uses tag name as a selector for styling.

Example :- < p >

p { color : Red ; }

output

< p > My name is imraan < /p >

→ paragraph is in Red color.

② class selector :- we define selector as a class with class name and this class is assigned to required tags. Many classes can be assigned to one element as

(class = "yellow" style = " ")

→ class selector is represented as Dot (.) in CSS

Example :-

```
q { color: black; border: 1px solid black; padding: 5px; margin: 10px; }
#hero { color: pink; }
#hero p { color: pink; }
```

↓
output

```
<p> ... </p>
<p id="hero"> ... </p>
```

→ color is pink.

④ combining selectors :-

(i) element with class selector

```
p .big {
  font-size: 18px;
}
```

↓
output

```
<p> ... <p>
```

```
<span> ... </span>
```

From the above code, only p element size

is increased, others are unaffected.

(ii) child - selector :-

Every p of the article which is direct child gets the style,

article > p {
 color: red;

}
 ↓
 Selects
 ↓
 Output

<article>

<p> ... </p>

(iii) Descendant selector :-

article p {
 color: red; }
 ↓
 Output

<p> ... </p>

<div> ... </div>

<p> ... </p>

<div> ... </div>

Every p element inside

article element irrespective of whether it is direct element or not gets that style.

Various selector combinations :-

• colored p { }

color : red ;

{ X9 SS : } file - foot

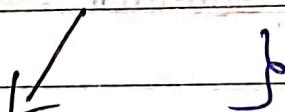


" Every element inside elements with class

• colored gets style "

• article > . colored { }

color : pink ;



{ X9 SS : } comp in vb

" Every element having class selector name

• colored and inside article and direct child gets style "

Lengths - Lengths

(Lengths) - Lengths

style tag inside tag will

Adjacent Sibling Selector

div + span {

Font-size: 22px;

div + span { font-size: 22px; }

"Adjacent sibling is element next to specific element at same level." (sibling)

General Sibling Selector :-

div ~ span {

div ~ span { font-size: 18px; }

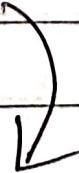
div ~ span { font-size: 18px; }

<div>

 ...

</div>

 ...



All the span element gets style.

Pseudo-class Selectors :-

A pseudo-class can be defined as a keyword which is combined with a selector that defines the special state of the selected elements.

Some popular pseudo-classes :-

① :active → An element is activated by the user.

② :hover → when the mouse pointer is over an element.

③ :Focus → An element receives focus.

④ :First-child → selects the first child element of its parent.

Syntax :-

selector : pseudo class {

property : value ;

}

CSS pseudo-elements :-

in both and ~~and~~ pseudo-class can be defined as a keyword which is combined into a selector that defines the special style of the selected elements.

The pseudo-elements are used to style the specific part of an element, whereas the pseudo-classes are used to style the element.

Some popular pseudo-elements :-

- ① ::first-line → It styles the first line of the text.
- ② ::before → It is used to add something before the element's content.
- ③ ::selection → Select the area of an element that is selected by the user.

— 20/02 —

Syntax :-

© cotwithmrcan

↳ value - selector : pseudo - element {

↳ property : Value ;

} frame

↳ attribute : value

Attribute selector :-

The CSS attribute selector is

used when we want to style multiple HTML elements that have the same attribute values.

(i) element name {

(ii) element name [attribute = "value"] {

element + [attribute = "value"] {

}

- Colors -

→ Colors :-

color property is mainly used for setting the color of an HTML element, color of fonts, background color, border color etc.

Syntax :-

element {

color : color name ;

}

Basically there are 4 ways to set a color.

(i) Keyword Value

(ii) Hexa decimal values

(iii) RGB values

(iv) HSL values

(i) keyword value :-

To set the keyword value, you just have to write down a name of the color in the color property. For example, if we want red color of our paragraph then we just type color name red.

Example :-

<style>

color: red;

}

<body>

color: red; ⇒ <h1>....</h1>

</body>

(model, mid, </style>) → output (x)

(ii) Hexadecimal value

Hexadecimal values can be defined

as 6 digit color representation

* It starts with # symbol

* 6 characters starts from 0 to 255

and A to F if also add

Example :-

<style>

color: #000000;

color: #FFFF00;

</style>

(iii) RGB value :— (Red, green, blue) @codewithimran

RGB is (red, green, blue).

* Range of RGB is 0 to any number.

* It can be in float or any integer.

Example :- `p{color: red;}`

`</style>`

```
<html>
  <head>
    <style>
      p { color: red; }
    </style>
  </head>
  <body>
```

`</body>`

(iv) HSL value :— (Hue, saturation, lightness)

* It is mainly used for select the color from color wheel.

* The value is in 0 to 360 degree.

* saturation means the visibility of color its value is from 0 to 100.

* lightness is used for masking glow the color. its value is in %.

- Paragraph -

Example :-

```
  <style> body {background-color: #f0f0f0; font-family: sans-serif; font-size: 16px; margin: 0; padding: 0;}
```

p {background-color: #fff; border: 1px solid #ccc; border-radius: 5px; color: hsl(202, 36, 40); padding: 10px;}

</style>

HTML :-

```
  <body>Hello World</body>
```

Output :-

- Background -

→ (i) Background - color property :-

The CSS background-color property specifies the background color of a container.

Example :-

```
p {  
    background-color: Red;  
}
```

(ii) Background - image property.

It is used to set an image in the Background

Example :-

```
body {  
    background-image: URL("../img/");  
}
```

(iii) Background - repeat property. (with image)

Normally, the image is by default repeated in x and y directions.

* 'repeat - x' is used for make image horizontally repeated.

* 'repeat - y' is used image vertically repeated.

* 'no-repeat' is used to show only single image.

Example :-

body {
background: no-repeat;

background: repeat;

}

(iv) Background - size property.

It can be important.

cover → fits and no empty space remains

contain → fits and image is visibly fully

auto → display in original size

{width} → set width

{width} {height} → set width and height.

3. Backgrounds

(vi) Background - position property.

We can set at anywhere. Here I set the starting position of a background image.

Example:- `background-position: x - longer;`

`div {`

`background-`

`background-position: left top;`

(vii) Background - attachment property.

It defines a scrollable and non-scrollable

`background-attachment: scroll;`

Example :-

`div {`

`background-attachment: fixed;`

Background shorthand @codewithimran

The single property to set multiple background properties

dir {

background: url('img') no-repeat fixed top left;

background-color image repeat attach position

background-color image repeat position

background-color image repeat position

background-color image repeat position

shorthand

background-color image repeat

background-color image repeat position

background-color image repeat position

↓
background-color image repeat

background-color image repeat

{}
<type>

- Border -

(2) border with imraan

→ Border :- Border is the important element in web design. It can be used to separate the content on a web page, making it easier for people to understand and take an action on your site.

Border have mainly 4 properties.

- (i) border - style
- (ii) border - color
- (iii) border - width
- (iv) border - radius.

(i) Border - Style :-

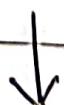
The 'border-style' property is used for specify the type of border which you want to display on screen.

Examples :-

</script>

{ border - style : none; , dotted,
dashed; , solid; , double; ,
groove; , inset; , outset; ,

</script> }



These are the border-styles.

(ii) Border-color :-

It is used to set the color of borders of an element.

Syntax :-

```
div {
```

border-style: dotted;

border-color: red;

};

(iii) Border-width :- with exactly border and border-color if the third main thing is set in styling web page is border-width.

Example :-

```
p {
```

border-width: 10px;

px is a number-based unit.

(iv) Border-radius :- (circle with inward)

It is used to make corners rounded. we can use 4 types of values single value, two value, three value, four value.

* we set values in pixels (px).

→ First value :- Write as :-

border-radius : 30px ;

This will set for all the corners.

→ Two values :- Write as :- border-radius : 20%, 10%;

Here 20% is round the top left corner and 10% is round the top right corner.

whereas 10% is round the bottom left corner and 20% is round the bottom right corner.

→ Three values :- Write as :- border-radius : 10%, 20%, 30%;

10% is for top left corner

20% is for top right and bottom left

30% is for bottom right corner.



©cockwith pocom

→ fair value :- write as :- border-radius: 10px, 20px, 30px, for;

border-radius: 10px, for top-left-corner

border-radius: 20px, for top-right-corner

border-radius: 30px, for bottom-right

40px, for bottom-left

border-radius: 50px, for rounded-circle

border-radius: 100px, for circle

border-radius: 50px, for circle (i)

border-radius: 50px, for circle (ii)

border-radius: 50px, for circle (iii)

border-radius: 50px, for circle (iv)

border-radius: 50px

border-radius: 50px, for circle (v)

border-radius: 50px

border-radius: 50px

{ 5 }

Available and used

memory

for (i) (ii) (iii) (iv) (v)

{ 5 }

- Font -

→ Font :- ~~which is the main part of the website~~ ^{① code with Pavan}

Font is the main part of the website so that's why choosing the right font can create a strong identity for your website.

→ CSS font's having these properties font size, font-style, font-weight,

(i) Font-size :- used for adjust the font size. There are 2 ways to define

- ① Keyword value
- ② Adjustable value.

P {

Font-size : smaller;

}

medium;

large ;

These are keyword values

P {

Font-size : 10%;

}

20px;

These are Adjustable values.

- Font-style :-

(i) Font-style :-

©codewithmango

It is a CSS property used for setting the style of your font. There are mainly 4 values, bold, italic, normal, oblique.

Example :-

P {

Font-style : bold ;

}

It can be italic, normal, oblique.

(ii) Font-weight :-

It is a CSS property used for setting the boldness of font.

Example :-

P {
font-style : italic ;
font-weight : bold ; } Keyword value.

P {

Font-weight : 200 ; } Adjustable value.

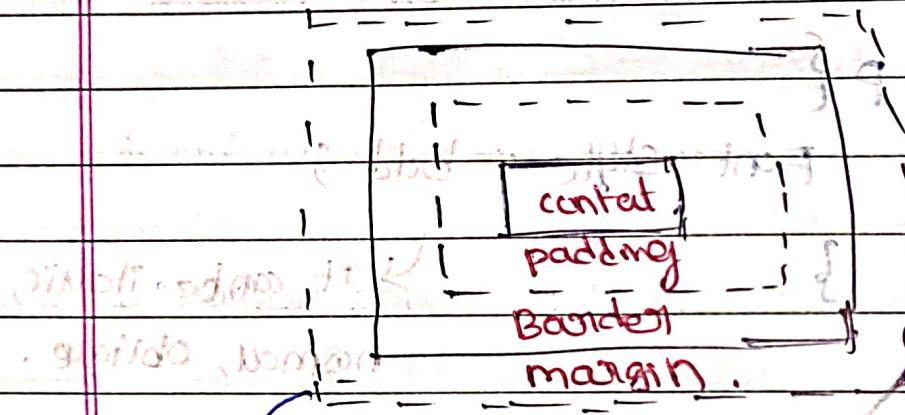
}

- Box model -

→ Box model :-

Very simple. It is used to develop the structure and design of making web pages.

According to Box model, browser supplies each element as square box.



With this, any web page we see is made up of elements wrapped in rectangular boxes and arranged in relation to each other. These elements consist of above, besides, within each other. depending on the type of element they are:

guru vishwakarma

↑ { 08 : typical tag }

(i) CSS - padding :-

With the help of CSS - padding property, it is used to define the space between the element and its content box and the element border.

More precisely, there are some sub properties to define :-

* padding :- It is used to set all the padding properties in one declaration.

* padding-left :- set left padding of an element

* padding-right :- set right padding of an element

* padding-top :- set top padding of an element

* padding-bottom :- set bottom padding of an element.

We can define the values in px, cm, or %.

Example :- `box { padding-top : 1cm
padding-right : 10px
padding-bottom : 1rem
padding-left : 120% }`

f

g

(ii) CSS - Margin :-

This property is used to define the space around elements. It is completely transparent and does not have any background or color. It creates area around the element.

* Margin :- setting the properties in one declaration

* Margin-left :- left margin of an element

* Margin-right :- right margin of an element

* Margin-top :- top margin of an element

* Margin-bottom :- bottom margin of an element.

Define the values in "auto, px, cm,

in % and inherit

Example :- .box { margin-top : 1em ;

margin-bottom : 20% ;

margin-right : auto ;

margin-left : 30px ;



- Position -

→ position :-

It is used for defines the position of an element on web page. This position works with the right, left, top, bottom properties to determine the final position of the element on the web page.

Following are the position values:-

* Static :- Default position for HTML elements.
top / bottom / left / right / z-index has no effect.

* relative :- when we assign the position: relative, an element follows the render flow of the page, but will be shift relative to its initial position.

* Fixed :- Element is positioned relative to the browser window.

* absolute :- The element is removed from the flow and is relatively positioned to the first non-static ancestor.

* sticky :-

① code with screenshot

It is positioned depending on the user scroll, it is depending on how far the user can scroll the page, and so on. All sticky element behaves like relative element until the viewport meets a specified position value.

state with no styling applied :- Note the following points of their behavior :-

1. when styling with position: absolute, the element is positioned relative to the nearest ancestor with position: relative or absolute, and is positioned in the middle of the left and right of

2. when styling with position: relative, the element is positioned relative to the nearest ancestor with position: relative

3. when styling with position: absolute, the element is positioned relative to the nearest ancestor with position: relative or absolute, and is positioned in the middle of the left and right of

— Overflow —

© code with raman

→ **Overflow** :- The `css: overflow` property specifies how to handle the content when it overflows its block level container.

Following are the overflow values :-

* **visible** :- It is the default value of CSS overflow property. It is not clipped and renders the element outside its box.

* **Hidden** :- The overflow is clipped, and scroll bar is used to see the rest of the content.

* **scroll** :- The overflow is clipped, and rest of the content will be invisible.

* **clip** :- The overflow is clipped, then the rest of its content will be clipped.

* **Auto** :- The overflow is clipped, a scroll bar is needed to see the rest of content.

- Hover Effect

④ click with mouse

→ Hover effect: Hover animation occurs when user uses their cursor and the element responds with any animated effect.

→ Many different and also combining

Syntax :-

22) `the styling like hover: { }` is called

box, is for all the different

elements and properties box

Decoration Property.

and

Now this box will affect all the elements

to the left and right then it will

function will

function box, box will affect all the elements

and this function will

and box will affect all the elements

so this function will

box

and now a box will affect all the elements

and this will not take if

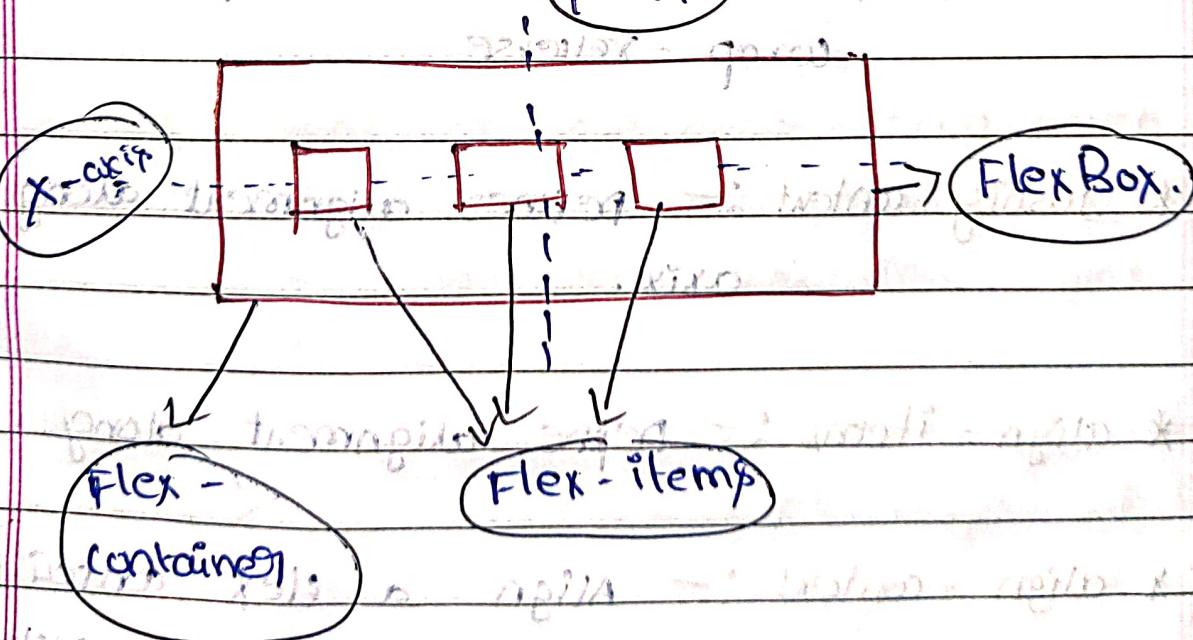
— Flex Box —

→ Flex Box in CSS :- We use Flex box, a layout model that offers lots of variety of ways to arrange the elements in CSS. With flex box help, we can design responsive and dynamic webpage.

Initialize a flex box :-

```
→ (display: flex;)  
or (display: flexbox;)  
or (display: flexbox; container {  
display: flex;}
```

• (display: flex; container {
display: flex; } flex {
display: flex; }



Flex-direction property :- ~~20/07/2017~~ @ code with Arman

1. row: defines the direction towards

at which the items are laid out in a line.

2. * Row (row) → default!

3. * row-reverse

* column (column) through has

* column-reverse.

Flex properties for parent (Flex-containers) :-

following properties for the flex parent

* Flex-wrap :- It can be wrap, nowrap.

.. wrap - reverse

* justify-content :- defines alignment along main axis.

* align-items :- define alignment along cross axis

* align-content :- Align a flex container lines when there is extra space in the cross axis.

→ ~~Flex~~

* Flex properties for children :-
The following are the properties for the flex children.

1. flex-grow :- It represents the size of the flex item.

2. flex-order :- It controls the order in which the items appear in the flex container.

* align-self :- Allows the default alignment to be overridden for the individual flex items.

* flex : property

* Flex-grow :- It is used for flex items to grow.

* Flex-shrink :- specifies how much a flex item will shrink relative to the rest of the flex items.

* flex-basis :- It is the initial size of the flex item.

* justify-content :- It represents the horizontal arrangement of the flex items.

- grid -

② code with program

→ CSS grid :- CSS grid is a two dimensional layout that helps web designers build complex and flexible web pages. It allows us to control all over the placement and the size of elements within the rows, columns.

in CSS grid styling with -

- container {

display: grid;

The grid - column-gap property :- It is used to adjust the space between the columns of a CSS grid.

The grid - row-gap property :- used to Adjust the space between the rows of a CSS grid.

The grid - gap property :-

Shorthand property for grid row-gap and grid - column gap.

• container {
display: grid;
grid-gap: 20px, 10px; }

• container {
grid-template-columns: 1fr 1fr 1fr; } row ← column.

properties for grid - container :-
grid-template-columns: 1fr 1fr 1fr;

* grid-template-columns :- used to specify the width of columns

Ex:- • container { display: grid;
grid-template-columns: 60px 10px auto; } row ← column

* grid-template-rows :- used to specify the height of each row.

Ex:- • container { display: grid;

grid-template-rows: 60px 10px; } row ← column

* justify-content :- align the whole grid inside container.

* align-content :- vertically align whole grid inside container.

properties for grid item :- \rightarrow \circ calculate in can

* grid-column - property :- It defines how many columns an item will span.

Ex :- \cdot grid item {
grid-column : 1/4 ;

* grid-row property :- how many rows can item span.
we can make an item to start on col1 and span 3 columns like this :

Ex :- \cdot grid item {
grid-column : 1/span 3 ;

\cdot grid item {
grid-column : 1/3 ;

grid-row : 1/3 ;

grid item {
grid-column : 1/3 ;

grid item {
grid-column : 1/3 ;

- Transform -

K Date
Page No. (41)

→ Transform - property : - ~~It is not @ code with in a class~~

~~Attribute of most Transform property changes the position, shape and the size of the element.~~

CSS Transform can be split into 2 categories:

* 2D Transformation

* 3D Transformation

* 2D Transformation

In 2D Transform, the element transform in two (x-axis, y-axis) and x-axis. we need mainly 5 Transform properties for make 2D Transform properties :

→ Translate ()

→ rotate ()

→ scale (x) ()

→ scale (y) ()

→ scale ()

→ skew ()

→ matrix ()

* 3D Transformation :-

5.1 (a) The element transform in 3 axis

frame i.e. x-axis by axis and z-axis we used mainly 3 properties for make 3D transform.

properties :-

→ rotate (x (n) degree) as *

→ rotate y ()

→ rotate z ()

transform with property as

→ perspective (n) and as transform

→ translate ()

→ rotate ()

Transition property :-

(1) statement

smoothly and gradually (2) changes from one position to another (3) position

(1) * slow

(2) wave

(3) vibration

- Transition

→ **Transition property** :-

It makes the element smoothly and gradually change from one position to another position.

It works like grease in the wheel.

Properties :-

* **Transition - property** :- used to transition.

* **Transition - Duration** :- Time taken for transition

* **Transition - Delay** :- specifies the delay for transition.

Ex :- `transition: width 2s easein 10s;`

↓
property

↓
duration
delay

↓
timing
function

- Animation

→ CSS animation :-

Animation keeps the movements changes
and actions like the elements to move a webpage.
making them makes a site live and dynamic. it
adds the better user experience to a page
since we human tends to interact
better with a dynamic environment.

we can use @keyframes rule to
change the animation from one style
to a new style.

@keyframes imran {

from {width: 10px}

to {width: 30px}

can change multiple

class, id, element properties.

↓
border
padding

{

properties to add Animations :-

Following are the properties used

to set animation in CSS.

* animation - name :- Name of the animation

* animation - duration :- How long does the animation

* animation - timing - function :- speed curve of the animation

* animation - delay :- Delay for the start of an animation

* animation - iteration - count :- No. of times an animation should run.

* animation - direction :- Direction of the animation.

Animation shorthand :-

animation : from 6s linear 1s infinite

1 2 3

4 5
reverse 6

using percentage values states it with animation.

but in jquery there is a problem

we can use % values to indicate what

should happen when a certain percent of
animation is completed.

animation will start with this code

@ keyframes imran {

0% {

width: 20px;

}

50% {

width: 40px;

100% {

100% {

width: 60px;

background-color: red;

color: black;

0 40

initial and small to start

20px

40 60

— Responsive Design —

→ Responsive Design :-

Responsive web design is basically used for resize, hide, move, shrink the content of the website for looking good when its screen size will be small, big, medium which is fit in Desktop, Tablet and mobiles.