**14 nov 2023**

1. Syntaxically awesome stylesheet
2. Deprecated
3. inline/internal/external - element/page/website
4. Relationship and location/ rel and href
5. Property:value
6. Selector: element or tag/id/class
7. Class name can be used multiple times while there should be only one id

Class comes from css and id comes from js

1. Attribute selector P[attribute]{property:value }
2. \* for all selector
3. Mdn web docs <https://developer.mozilla.org/en-US/docs/Web/CSS/named-color>
4. <https://colorhunt.co/>
5. <https://appbrewery.github.io/just-add-css/>
6. 1 px - pixel and 1 pt - point 1/96th inch and 1/72nd inch .26mm/.36mm
7. 1em - 100% of parent
8. 1rem - 100% of root

**15 nov 2023**

1. div.box#box\*4
2. ctrl+shift+I and f12
3. Complete details of developer tools
4. From top clockwise (if provide 4 values)
5. Top-bottom and right-left (if provide 2 values)
6. Position,specificity,type,importance
7. Position- Bottom up approach
8. Specification- tag<class<attribute<id
9. Type - external stylesheet <internal < inline
10. Importance - !important
11. **Position attribute in css - Static, absolute,relative , fixed and sticky**
12. **Position:static**

This is a by default position so, if content is in initial position there will be no any change

1. **Position: relative**

Relative means position the element relative to its normal position

Here you can set top,left.right etc. and create gap from its normal position

Agar apko purani position me gap chahia tb app log use krenge relatiive

1. **Position:absolute**

Here position the element be on the basis of first parent container

1. **Position:fixed**

This will fixed the element position on the basis of browser window

1. **position:sticky**

This is mixed of relative and fixed.

1. <https://appbrewery.github.io/css-positioning/>
2. **Display - Block, inline, inline-block and none**
3. Span has default display inline
4. You can not change size of a inline element
5. Inline block allow u to se theight and width
6. https://appbrewery.github.io/css-display/
7. **Float - Generally used in magazine and news paper**
8. <img>

<p>text</p>

Img{float:left}

1. Clear - used to avoid the float

<img>

<p>text</p>

<footer>text</footer>

Img{float:left}

footer{clear:left}

1. **Responsive website - refer to point 42**
2. **Media query**

@media(max-width:600px){

Css

}

1. **Css grid**

Create div on the basis of number of block in grid

In css

main container{

display:grid

Grid-template-columns:1fr 1fr; (columns division)

Grid-template-rows: 100px 200px 200px; (each rows height)

gap:30px;

}

First div{

Grid-column:span 2;

}

Second div{

Background-color:red;

}

1. **Css flexbox**

**<div class=”main-container>**

**<div class=”first-div common”>**

**<div class=”second-div common”>**

**<div class=” common”>**

**<div class=” common”>**

**</div>**

In css

main-container{

Display:flex;

gap:10px;

}

Common{

Background: blue;

Border: 30px solid black;

Height: 100px;

flex:1;

}

first-div{

flex:2;

}

second-div{

flex:0.5;

}

1. **External framework like bootstrap**

Col-12

Divide in 12 part accordingly

1. Inline take space according to itself
2. Block element take space of complete block and throw next thing in next line It starts on a new line, and takes up the whole width
3. In inline u can not set width Any height and width properties will have no effect
4. If u want to set width then use inline element
5. Inline-block - block will be inline

**16 Nov 2023**

1. Decendent

normal i.e. all s2 of s1

Selector1 selector2{} -

1. Child selector

selector1>selector2{}

this will select immediate decendent s2 in s1 and not all s2

1. Chaining

Apply where all selector are true(very specific)

Selectorselector{}

1. Other are +

**17 Nov 23**

1. @media (max-width:600px) and (max-width:600px) {

H1{

Font-size:15px;

}

}

1. 319 to 480 / 481 to 1200 / 1201 to 1600 / 1601+
2. Floats vs flexbox(used to create web layout)
3. Default order in flex box is zero

order:0;

1. **Flex**-wrap: nowrap; remove responsivenes

To do responshiv make nowrap to wrap

1. Flex-direction

<https://appbrewery.github.io/flex-layout/>

1. **justify-content - horizontal**
2. **Align-item- vertical**
3. **Align-item work with flex wrap:nowrap**
4. **Align-content work with flex wrap:wrap or reverse wrap**
5. **display:grid**

**Grid-template-rows: 100px auto;**

**Grid-template-column: 200px auto;**

[**https://appbrewery.github.io/grid-vs-flexbox/**](https://appbrewery.github.io/grid-vs-flexbox/)

[**https://appbrewery.github.io/grid-sizing/**](https://appbrewery.github.io/grid-sizing/)

1. **marquee**

**responsive**

**tables**

**css**

**frames**

**Media loading**

**Container writing**

1. **Flex**

[Flex-direction](https://www.w3schools.com/css/css3_flexbox_container.asp#flex-direction)

[flex-wrap](https://www.w3schools.com/css/css3_flexbox_container.asp#flex-wrap)

[flex-flow](https://www.w3schools.com/css/css3_flexbox_container.asp#flex-flow)

[justify-content](https://www.w3schools.com/css/css3_flexbox_container.asp#justify-content)

[align-items](https://www.w3schools.com/css/css3_flexbox_container.asp#align-items)

[Align-content](https://www.w3schools.com/css/css3_flexbox_container.asp#align-content)