9.0. Display : Flex :-

> What is it and how does it work?

→ earlier < table ></table > element is used to represent talubar data. Even now it is in use

Example :-

```
< table >
    < tr# class = 'row' >
        < td   class = 'col1' > .... </td >
        < td   class = 'col2 > ...... </td >
        < td   class = 'col3' > ..... </td >
    </tr>class
</table>
```

```
. col1 {
    width : 25%.;
}
. col 2 {
    width : 25%.;
}
. col 3 {
    width : 40%. ;
}
```

→ after table , display property evolved . then float was evolved.

→ float → Magazine - style Layouts

→ flexbox → overall page structure [simpler].

Example :-

index . html :-

```
< div class = "container" >
    < div class = "one" > < p> .... </p> </div>
    < div class = "two > < p > ... </p></div>
    < div class = "three"> < p > ... </p> < /div>
</div >
```
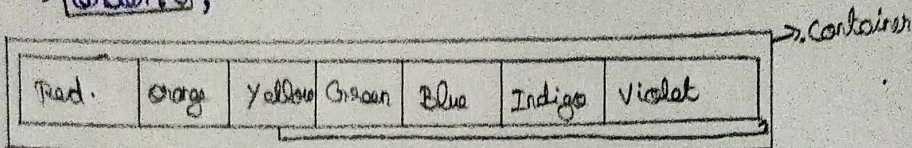
style . css :-

```
. container {
    display : flex;
    gap : 10px;        → gap b/w 2 flex box [columns]
}
```

9.1. Flex Direction :-

→ Row & Column Layout.

→ Default layout is stacked side by side.

→ If it is a block it is stacked one by one.

→ There is a property in flexbox : that is

flex - direction : row; row is default one [→]

if flex - direction : Column; [↓] → axis [Y]

→ flex - basis : n px; Right of flex is adjusted.

→ flex - basis adjust Right if direction : column else width is adjusted.

→ simply flexing on the Main axis.

9.2. Flexible Layout :-

→ align, justify and wrap.
→ creating flexible layout.
→ usually . container is parent inside parent is child
→ `order:0;`

Red.	Orange	Yellow	Green	Blue	Indigo	Violet

→ container

→ if we need to change the position of Yellow to right most, then

```
. Yellow {
    order : 1 ;   default value = 0
}
```

→ order is hierarchy. therefore the position of each element is based on ascending of order.

→ `flex - wrap : nowrap;`

→ it is useful when we run out of space.
→ flex - wrap property is normally set to nowrap.
→ we can change this property to wrap. but elements may not pushed out of screen. [next line]

→ so on ....