

14/25 * Display And Visibility → (Important Properties)

I Display Property in CSS defines how the components (div, hyperlink, heading etc) are going to be placed on the web page.

There are several values it can have →

- (1) block
- (2) inline
- (3) inline-block
- (4) none
- (5) flex
- (6) grid.

II Several values which visibility can have →

- (1) none
- (2) visible
- (3) initial
- (4) inherit
- (5) collapse.

→ —————— X ——————

* So starting with the properties of Display first →

Back ⇒

(1) display: block → This property implies that the content will be occupying the complete width of the web page and no other content will be coming adjacent to it apart from what is the size of the content.

(2) display: inline → This property implies that the content will be occupying only the width of the content. And if the width is not available in the same line it will automatically come in the next line.

Ex → {span, ~~img~~} they by default have display set as inline which can be further changed as per the requirement.

Note → The problem with inline elements is that whenever we try to change the width of them we will not be able to do that and same goes with the height as it is set by default.

So, the solution for this is that we use property "inline-block".

↳ This helps us with the elements to be in one line and let us change the width and height accordingly.

(4) display: none → It acts as if it was never there. No space is been occupied nor we can see the element, it behaves as if the element never existed.

VS

(1) visibility: hidden → It occupies the space of that element according to its height and width but we can't see the content.

Ex → .visible {

background-color: #blue;
display: inline-block;
visibility: hidden;
height: 50px;
width: 200px

}

(2) visibility: visible → To show the element as it is.

(3) visibility: initial → It means the default value of visibility will be set and default value of visibility is 'visible'.

(4) visibility: inherit → It inherits the value of parent class.

for ex - <div> inside a <div>

```
<div class="Parent">
```

```
  <div class="child"> Hello </div>
```

```
</div>
```

So, if the parent has value as 'invisible' then the child will also be invisible...

but if the child has its own value then irrespective of the fact what value parent has it will simply ignore, and display its own value or property.

* Positions - Static, Relative, Absolute...

Fixed, & Sticky...

(1) Static → By default what positioning the elements are having is said to be static. That is, the content written is displayed exactly in the similar manner as it was on the HTML page.

(2) Relative → The elements having relative positioning remain in the flow of the web page. No matter what top, bottom you give them other elements

are just bothered about it's original position.

So, for ex →

Ex → .div1 {

width: 100px;

height: 100 px;

background-color: # red;

margin-top: 30px;

}

So, what happens in here is all the elements will be shifted from top if we use margin-top.

So, if we not want all the elements to move ~~use~~ use →

Ex → .div1 {

width: 100 px;

height: 100 px;

position: relative;

top: 30px;

}

(3) Absolute → The element with absolute Positioning will Position itself according to the whole page. So, for this the complete page is the canvas.

→ So, in here the other elements position accordingly like for them the element with absolute positioning doesn't even exist.

Suppose, let's understand this with the help of an example →

.div1 {

width: 100px;

height: 100px;

background-color: red;

position: absolute;

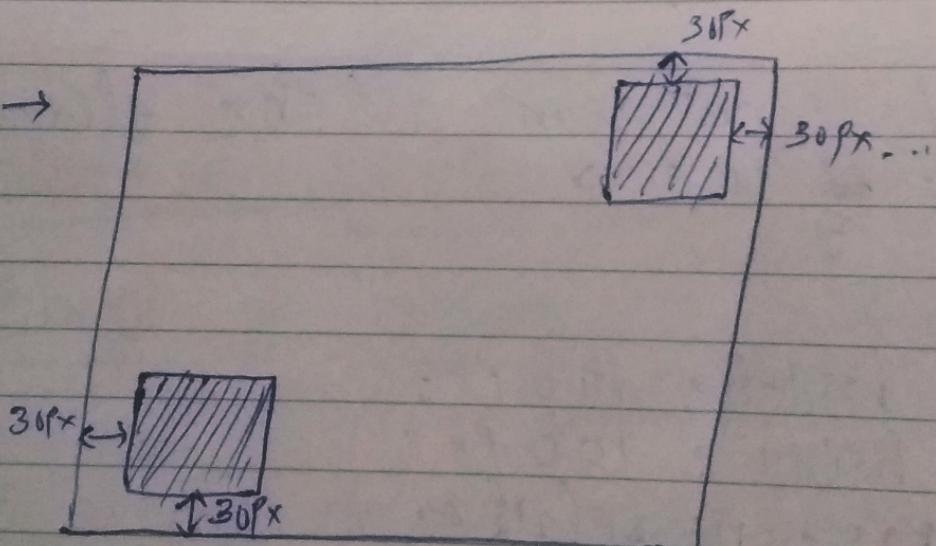
bottom: 30px;

left: 30px;

(if we do top), then

(if we do right)

Output →



→ We can understand this as a 2D plane with (x, y) coordinates and the element is placed accordingly.