

# 4. CSS DISPLAY, BOX MODEL & CSS UNITS

color → changes font color

background-color → changes the background color of the element.

background-image → An image can be used as background.

background-image: url( )

image link or relative path

↓

path from current

directory

Font-family: 'Times New Roman',  
Times, serif

Times is fallback to Times New Roman,  
& serif to Times.

If all specified fonts are not available,  
the browser will use the default  
serif font on user system.

font-weight → Intensity of the  
font. Ranges from  
100 to 800.

[fonts.googleapis.com](https://fonts.googleapis.com)

To add a new font,

Add the font imports in HTML head.

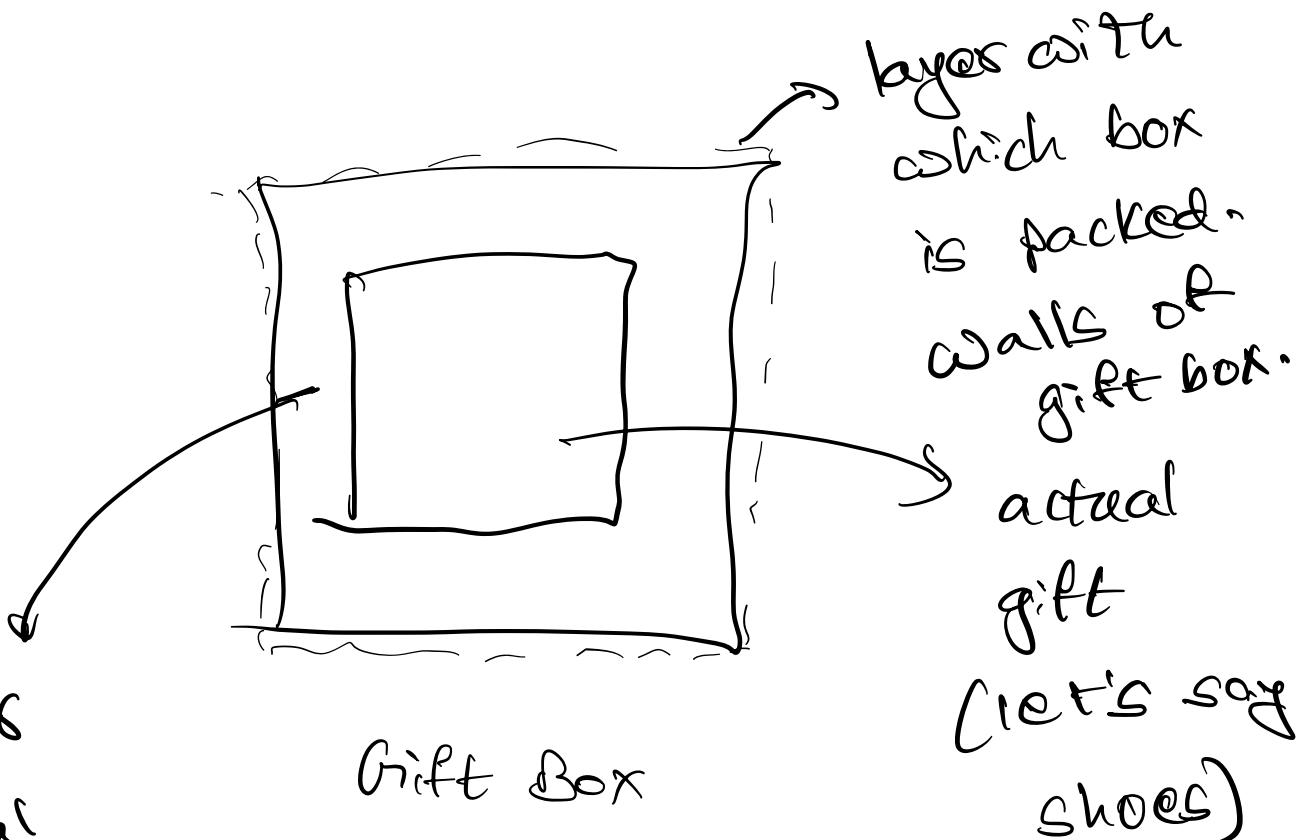
text-align → To change alignment  
word-spacing → To adjust the spacing between words

line-height → To adjust space between lines.

## Box Model:

Goto any web page → Inspect  
There is a box associated with each element.

Every element in CSS follows a box model by default.

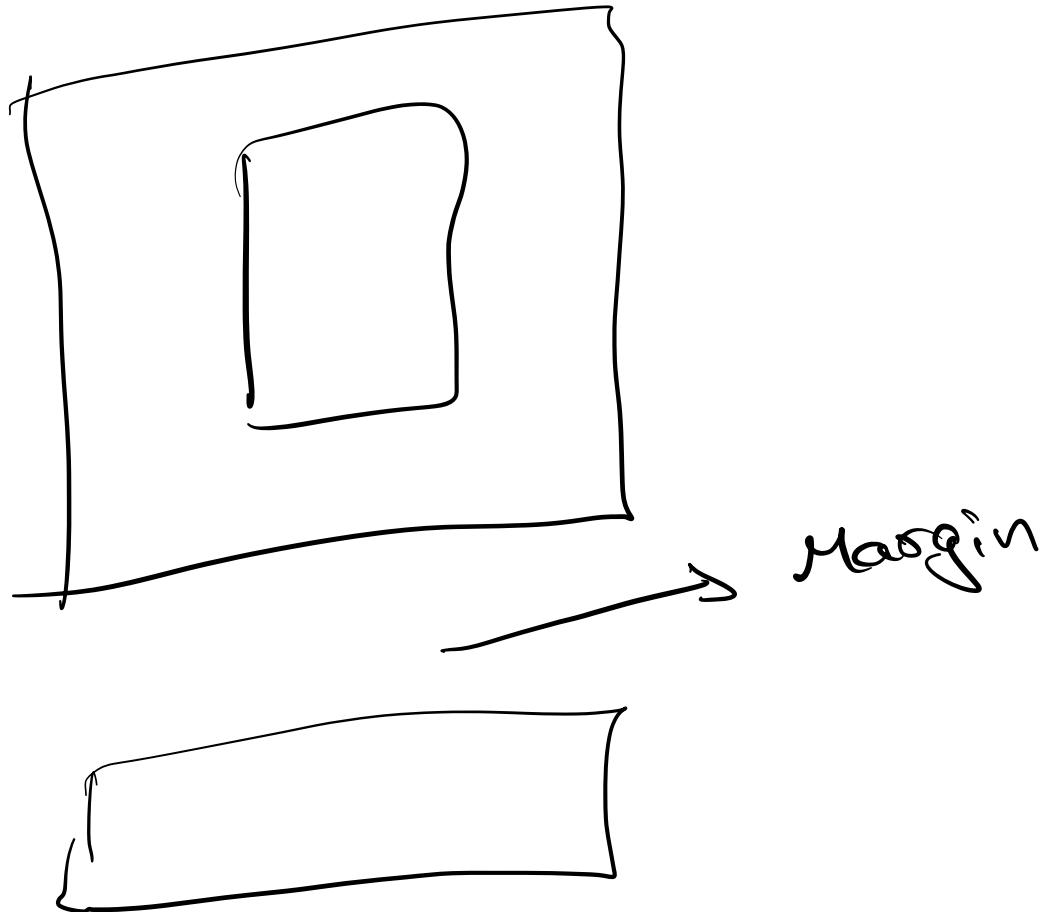


buffer  
material  
(like paper,  
styrofoam) or  
empty space  
to protect gift.

(padding)

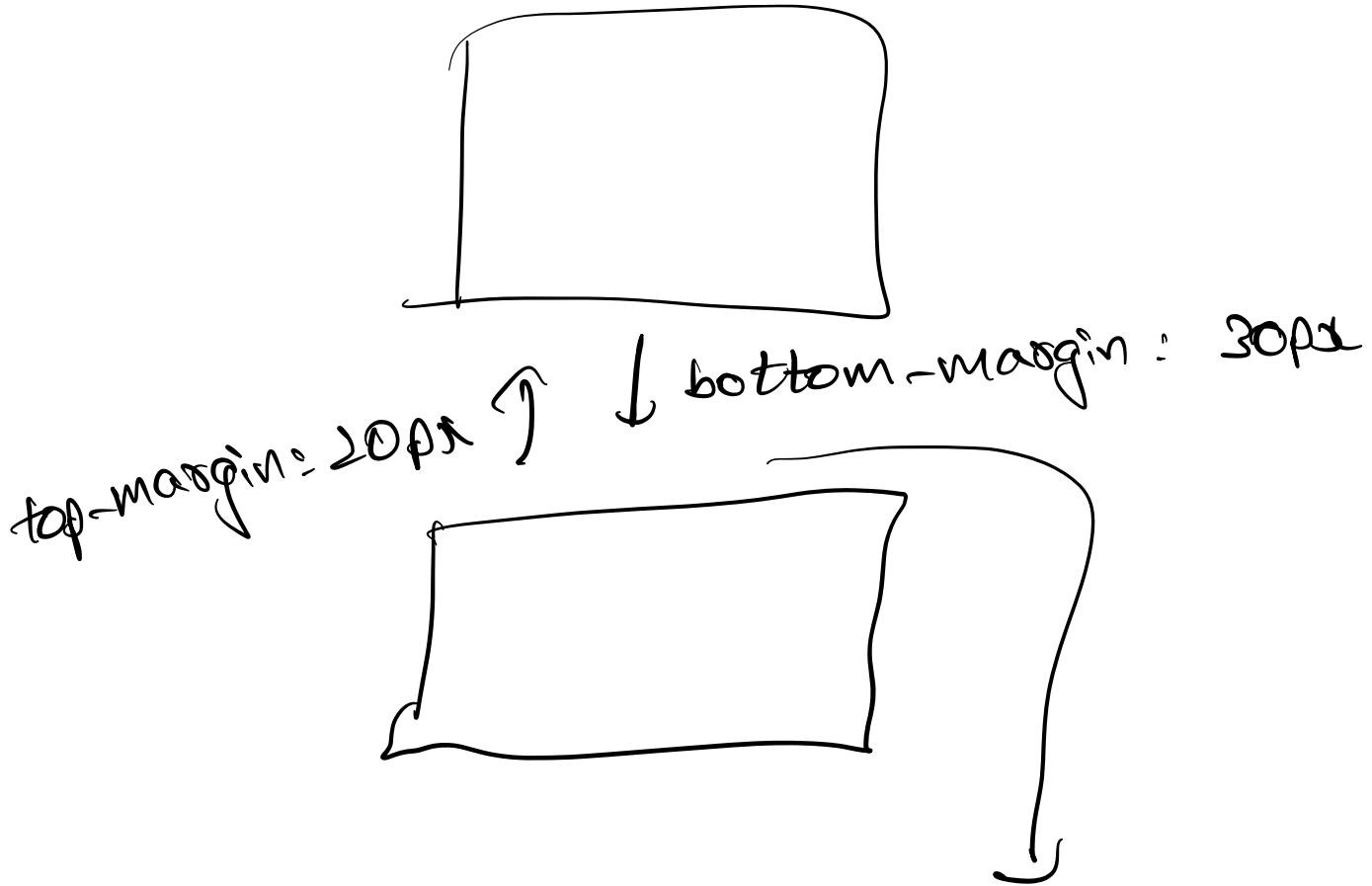
↓  
space between content & walls

Border → Boundary around padding.



Margin → Space outside the border  
between elements.

Let's say there are two elements,



The total margin  
between both elements  
is not 50px.

Instead the total margin between the  
two is max of both of them.

`margin: 30px;` → Applies margin in all four directions.

If margin needs to be provided separately,

`margin: top, right, bottom, left`

clockwise direction

`margin: 30px 20px 30px 20px;`

`margin: 20px 30px;` → By default for left & right

By default it will  
considered for both

top & bottom

margin: 10px 20px 30px;  
↓      ↓      ↓  
top    right    bottom  
&  
left

Same applies for padding as well.

Border is an extra element that you can create around your HTML elements.

border: 2px solid pink;  
↓      ↗      ↘  
thickness      type of border.      color  
of border      popular ones are  
→ solid  
→ dashed  
→ dotted

border-radius → To increase curves  
of borders corner.

Every HTML element should be styled  
imagining a boxed model.

### Box Sizing:

A property in CSS to determine how box  
model will be used.

Two possible values,

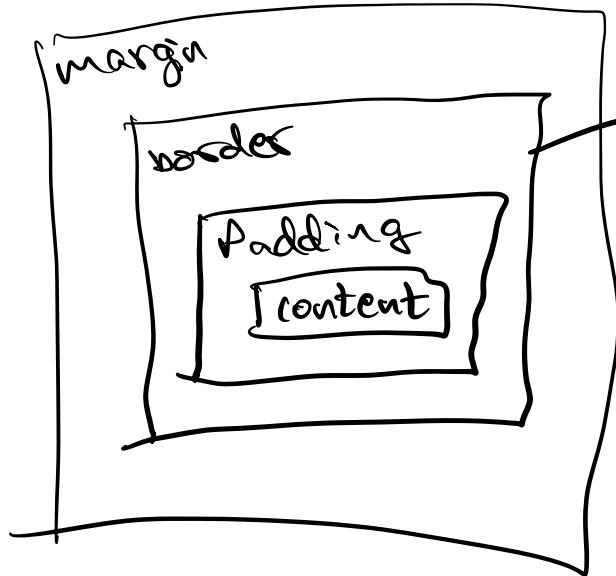
- Content box
- Border box

### Content Box:

There is difference between content & element.

Element contains content.

In box model,



All the layers  
including  
borders are  
element.

border + padding + content

=  
element

`<div class = "box" >   </div>`

-box {

width: 200px ] → applied to  
height: 100px content

border: 3px solid black;

padding: 9px;

box-sizing: content-box;

width of element = width of content + 2(padding) + 2(borders)

present on left & right side

present on left & right side

$$= 200 + 2(5) + 2(3)$$

$$= 200 + 10 + 6$$

$$= 216 \text{ px}$$

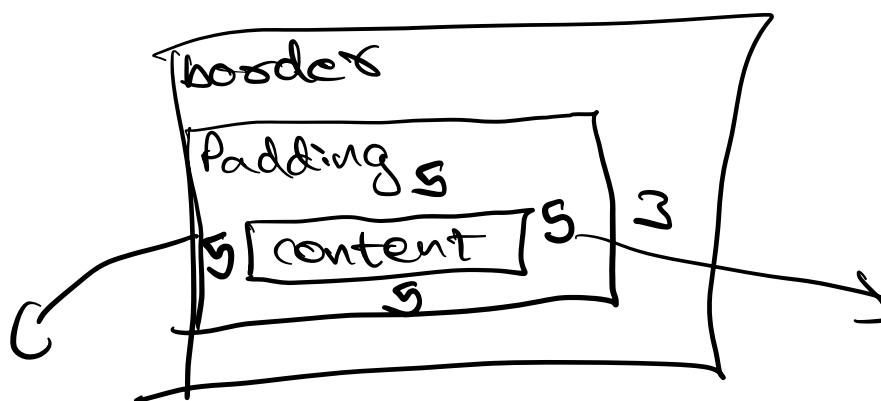
height of element = height of content + 2(padding) + 2(borders)

$$= 100 + 2(5) + 2(3)$$

$$= 116 \text{ px}$$

In this case 2 is multiplied to

borders & padding as they are applied on all sides i.e; if the border is applied only to right side, when we calculate width,



there is no border on this

side.

$$\text{width of element} = \text{width of content} + 2(\text{padding}) + \text{border}$$

$$= 200 + 2(5) + 3$$

$$= 213\text{px}$$

box-sizing: content-box;

width & height of content not element.

This is default property.

Adding/removing this property doesn't make any change.

box-sizing: border-box;

width & height of element.

. border box {

height: 100px

width: 200px

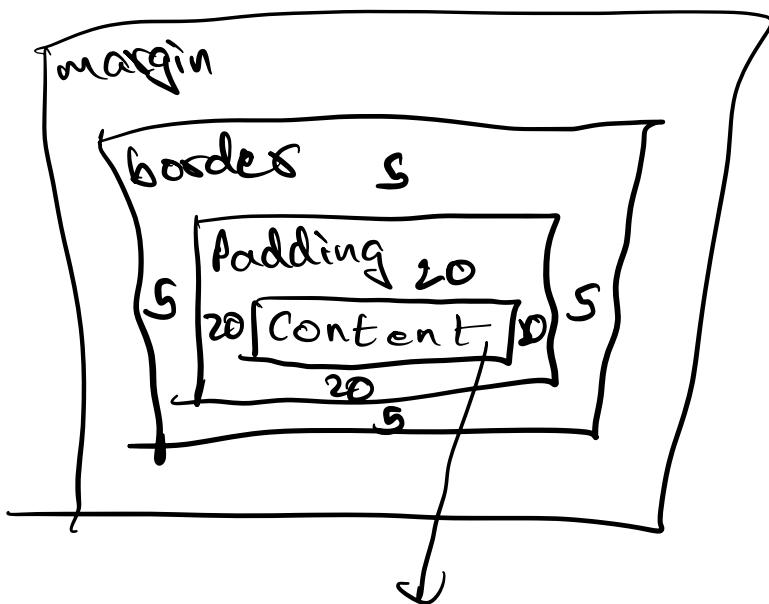
padding: 20px

→ height & width  
of actual  
element

`border: 5px solid red;`

`box-sizing: border-box;`

2



150x50

Padding is 20px all across, border is 5px all across as we defined, however the width & height we defined is shrunk.

Almost all the time we work with border-box.

When we define height & width, we want our element to take that space not our content.

Use,

content-box → when content area needs to have fixed size, & you don't mind element growing as padding & borders are added.

Useful when exact content dimensions are crucial, regardless of padding or borders.

borders-box → when you want element to maintain consistent size regardless of amount of padding or border.

Useful for responsive design, to ensure element fits within a specific space without having to manually calculate the total size.

### Overflow:

When we have fixed height & width the content might not always fit in, resulting in overflow.

To handle such scenarios,

overflow: hidden;

↳ 'visible' is  
the default  
value.

Other available values,

→ scroll

→ auto

overflow-x: hidden; → To hide  
horizontal  
overflowing  
content

Display:

Block Elements: Take the full available  
width in the container they

are.

`<p> para 1 </p>`

`<p> para 2 </p>`

`p {`

`border: 3px solid red;`

`}`

`<span> Span 1 </span>`

`<span> Span 2 </span>`

`Span {`

`border: 3px solid blue;`

`}`

Display property is used to handle how the elements are displayed.

`display: block;`

↳ For 'paragraphs'  
this is default  
value.

Block element will always start on a  
new line.

`<Span> Span 1 </span>`

`<p> Para </p>`

`display: inline;`

↓  
They will take the width  
required to display content.

They will start in the same line.

The display property can be overridden.

Span {

display: block

}

Inline elements can never be given a height & width property. You can give, but inline elements won't resize.

{ Block → Resize

  |  
  { Inline → Only take necessary  
    height & width

→ InlineBlock