CSS : Box model

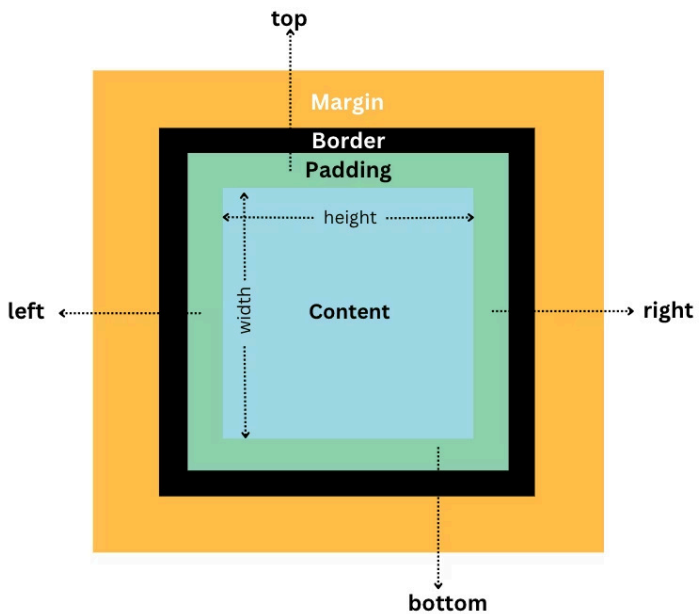
Lec-3

Made with GAMMA

CSS Box Model

- All elements on a web page are interpreted by the browser as **living inside of a box**.
- The box model comprises of the **set of properties** that define the parts of an element that take up space on a web page.
- The properties include:

1. **Width and Height of the content area.**
2. **Padding:** Amount of space between **content area** and **border.**
3. **Border:** The **thickness** and **style** of the border surrounding **content area** and **padding**.
4. **Margin:** The amount of **space between border and outside edge** of element.



Made with GAMMA

Height and Width Property

- An Element's content has two dimensions- **height** and **width.**
- By default the dimensions of an HTML box are **set to hold the raw contents of the box**.
- The CSS **height** and **width** property can be used to modify these default dimensions.
- **Eg:**

```
p{
  height: 80px;
  width: 240px;
}
```

Pixels (px) allows us to set the exact size of an element's content such as width and height of an content, such that it will be same on all devices i.e after giving value in pixel, an element that fills a laptop screen will **overflow (go beyond boundaries)** in a mobile screen.

Border Property

- A border is a line that surrounds an element, like a frame around a painting.
- Border can be set with a specific **width, style** and **color.**
- **Eg:**

```
p{
  border: 3px solid black;
}
```

- In order to modify the corners of an element's border, we use **border-radius** property.
- **Eg:**

```
div .container{
        border: 2px solid blue;
        border-radius: 5px;
      }
```

- In order to create a border that is **perfect circle**, first we create a element with same **width** and **height,** and then setting radius or **border-radius** as **50%.**
- **Eg:**

```
div .container{
        height: 60px;
        width: 60px;
        border: 3px solid blue;
        border-radius: 50%;
      }
```

Made with GAMMA

Padding Property

- The **Space between contents of a box and the borders of a box** is known as padding.
- **Eg:**

```
p .content-header{
                border:3px solid coral;
                padding: 10px;
              }
```

- The **Padding** property is often used to expand the background color and make content look less cramped.
- If you want to be more specific about the amount of padding on each side of the box content, you can use the following properties specifically.

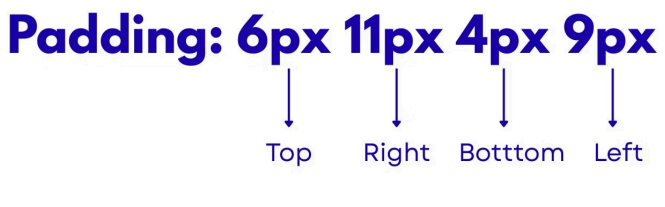
1. **padding-top**
2. **padding-right**
3. **padding-bottom**
4. **padding-left**

- Each property affects the padding on only one side of the box's contents, giving you more flexibility in customization.
- **Eg:**

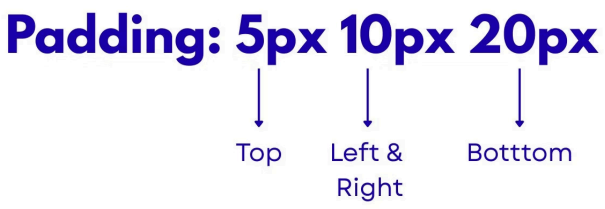
```
p .content-header{
                  border: 3px solid fuchsia;
                  padding-top: 10px;
                }
```

Shorthand Padding Property

1. **4 values:**



2. **3 values:**



3. **2 values:**

Made with GAMMA

Margin Property

- Margin refers to the **space directly outside of the box**. The **margin** property is used to specify the size of this space.
- It has specific properties-

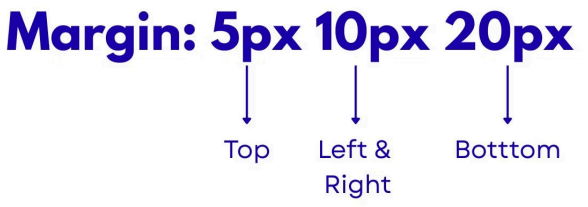
1. **margin-top**
2. **margin-left**
3. **margin-bottom**
4. **margin-right**

- Each property affects the margin on only one side of the box's contents, giving you more flexibility in customization.

Shorthand Margin Property

1. **4 values:**

2. **3 values:**



3. **2 values:**

Made with GAMMA

Margin with Auto

- The **margin** property also lets you center content. However, you must follow a few syntax requirement.
- **Eg:**

```
div .headline{
            width: 400px;
            margin: 0 auto;
          }
```

- **margin: 0 auto;** will center the divs in their containing elements. The **0 sets top and bottom margin to 0px** and **auto will be giving instructions to browser to adjust left and right margins until the element is centered within it's containing element.**
- In order to center an element, **a width must be set for that element.** Otherwise, the width of the div will be automatically set to the full width of it's containing element.

Margin Collapse

- **Margin collapse** is when **two margins collapse into a single margin**.
- Top and bottom margins of elements are sometimes collapsed into a single margin that is **equal to the largest of the two margins**.
- **Eg:**

```
h1 {
  margin-bottom: 50px;
}

h2 {
  margin-top: 20px;
}
```

- In above eg, the **<h1> element has a bottom margin of 50px** and the **<h2> element has a top margin of 20px**. So, the **vertical margin between the <h1> and the <h2> would be a total of 70px (50px + 20px)**. But due to **margin collapse**, the **actual margin ends up being 50px i.e largest margin among two.**

Margin collapse only happens with top and bottom margins! Not left and right margins!

Made with GAMMA

Minimum and Maximum Height and Width

- Because a web page can be viewed through the displays of **differing screen size** the **content on the web page can suffer from those changes in size.**
- To avoid this problem, CSS offers **two properties** that can limit **how narrow or how wide an element's box can be sized** to:

1. **min-width:** This property ensures a minimum width of an element's box.
2. **max-width:** This property ensures a maximum width of an element's box.

Eg:

```
p{
   min-width: 300px;
   max-width: 600px;
}
```

In above example, **the width of all paragraphs will not shrink below 300 pixels, nor will exceed 600 pixels.**

- Same property is for height as well:

Eg:

```
p{
  min-height: 150px;
  max-height: 300px;
}
```

In above example, **the height of all paragraphs will not shrink below 150px nor will the height exceed 300px.**

Made with GAMMA

Overflow in CSS (Parent Property)

- The overflow property **controls what happens to the content that spills or overflows outside it's box.**
- The most commonly used values are:

1. **visible:** Any content getting overflown will be displayed outside of the containing element.
2. **hidden:** Any content that overflow will be hidden from view.
3. **scroll:** A scrollbar get's added in order to view the rest of the content which is getting overflown.

- **Eg:**

```
p{
   overflow: scroll;
}
```

- In above eg, **if any of the paragraph content overflows (perhaps a user resizes their browser window), a scrollbar will appear so that users can view the rest of the content.**
- The **overflow property is set on a parent element to instruct a web browser on how to render a child elements.**
- For eg, if a div's overflow property is set to **scroll**, all children of this div will display overflowing content with a scrollbar.

Made with GAMMA

Visibility in CSS

- Elements can be **hidden from the view** with the visibility property
- The visibility property can be set to one of the following values:

1. **hidden:** Hides an element.
2. **visible:** Displays an element,
3. **collapse:** Collapses an element.

- **Eg:**

```
html

<ul>
  <li> Explore </li>
  <li> Connect </li>
  <li class ="future"> Donate </li>
</ul>
```

```
css

.future{
        visibility: hidden;
      }
```

- In this example, the list item with a class **future** will be hidden from view in the browser.
- Furthermore, the **web page will only hide the contents of the element**. It will **still leave an empty space where the element is intended to display.**
- An element with **display: none** will be completely removed from the web page.
- An element with **visibility: hidden** will not visible on the web page, but the space reserved for it will.

Made with GAMMA

Resetting Default CSS

- All major web browsers have a default stylesheet which they use in the absence of an external stylesheet.
- These default stylesheets are called as- **User Agent Stylesheet**
- Default stylesheets have default CSS ruleset that sets values for **margin** and **padding.** This affects how browser displays HTML elements, which can make difficult for a developer to design or style a web page.
- Many developers chose to reset these default values so that they can truly work with a clean slate.
- Default CSS is removed in a following way:

```
*{
  margin: 0;
  padding: 0;
  box-sizing: border-box;
}
```

Box-sizing property in CSS

- The CSS `box-sizing` property defines how to calculate the width and height of an element: should the calculation include padding and borders, or not.
- It takes two values:

1. **content-box**
2. **border-box**

Content-box

- The width and height you set apply **only to the content** — not the padding or border.
- **Eg:**

```
css

div {
  width: 200px;
  height: 100px;
  padding: 20px;
  border: 10px solid black;
  box-sizing: content-box;
}
```

- Now, let's calculate the **total actual size** of the box.

Calculation of width of the content

Width = Content Width + Left Padding + Right Padding + Left Border + Right Border

= 200px + 20px + 20px + 10px + 10px

Total width = 260px

Calculation of height of the content

Height = Content height + Top Padding + Bottom Padding + Top Border + Bottom Border

= 100px + 20px + 20px + 10px + 10px

Total Height = 160px

- So, Although you set width = 200px, the box will *actually occupy 260px* on the screen!
- Similarly, Although you set height = 100px, the box will *actually occupy 160px* on the screen!

Border-box (recomended)

- The width and height you set include the **content**, **padding**, and **border**.
- This means the total size of the element **stays fixed** — padding and border are *absorbed inside* the given width and height.
- **Eg:**

```
css

div {
  width: 200px;
  height: 100px;
  padding: 20px;
  border: 10px solid black;
  box-sizing: border-box;
}
```

- Now, calculation changes:

Calculation of width of the content

Total width = 200px (fixed)

→ content width = total width - (padding + border)

→ 200 - (20 + 20 + 10 + 10) = 140px

So here, **Total Width = 200px** and **Content Width = 140px**

Calculation of height of the content

Total height = 100px (fixed)

→ content height = total height - (padding + border)

→ 100 - (20 + 20 + 10 + 10) = 40px

So here, **Total height = 200px** and **Content height = 140px**

- So here, even though you set padding and border, the element still **takes only 200px width** and **100px height** in total.
- Since the result of using the `box-sizing: border-box;` is so much better, many developers want all elements on their pages to work this way.

Made with GAMMA