CSS Box Model

Topics Covered:

- Box model.
- CSS width and height.
- CSS Borders.
- CSS Padding.
- CSS margin.
- Maximum and minimum width and height.
- CSS overflow.

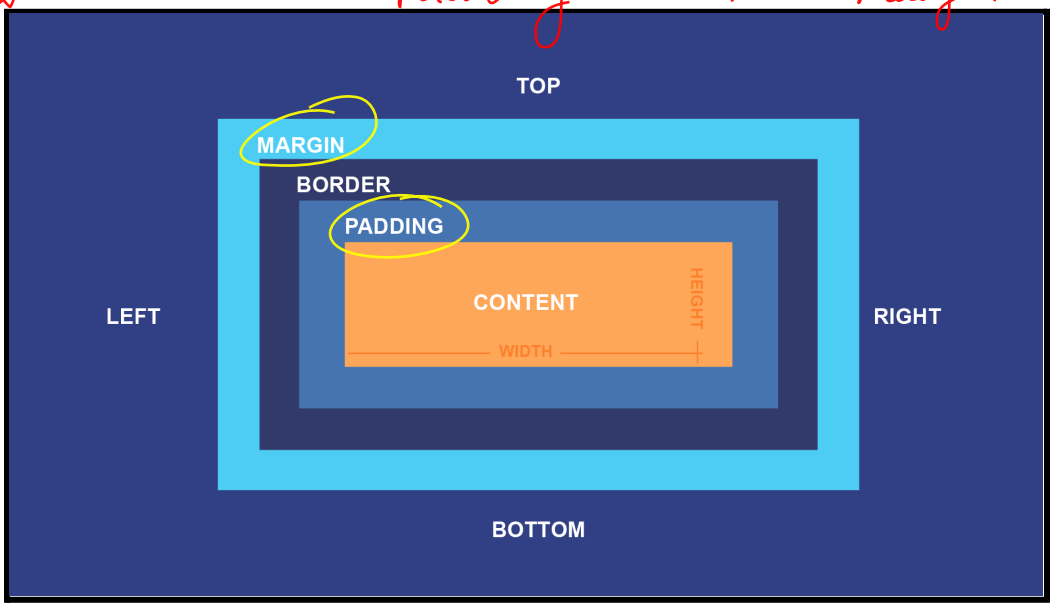
Topics in Detail:

CSS Box model:

- Box model helps you to understand how elements are positioned and displayed on a website.
- All elements on a web page are interpreted by the browser as “living” inside of a box. This is what is meant by the box model.
- For example:
    - If you have set the background color of an element, you may have noticed that the color was applied not only to the area directly behind the element but also to the area around the element.
    - when you change the background color of an element, you change the background color of its entire box.
- The box model includes:
    - The dimensions of an element’s box.
    - The borders of an element’s box. (border of element box is already learnt in the last topic)
    - The paddings of an element’s box.
    - The margins of an element’s box.

Box model properties:

The box model comprises a set of properties of elements that can take up space on a web page.



- Width and height: The width and height of content.
- Padding: The space between the content and the border.
- Border: The thickness, style of border surrounding the content and padding.
- Margin: The space between the border of the element and outside edge/ border of other elements.

Width and Height:

- The content of any HTML element has two dimensions width and height.
- The dimension of the element can be modified with the properties width and height.
- Example:

```
#main {
 width: 400px;
 height: 1000px;
}
```

CSS-Borders:

- A border is a line that surrounds an element, like a frame around a painting.
- Different styles, width, color, radius can be kept for HTML elements.
    - border-style.
    - border-width.
    - border-color.
    - border-radius.

CSS-Padding:

- The space between the content and the border of the box is known as the padding.
- For example, padding is a space between the picture and it's frame.
- The space can be modified by using the padding property in CSS.
- Example:

```
#main {
 padding: 40px;
}
```

- The padding property is often used to expand the background color and make the content look spacious.
- The padding property can be specifically used to all four sides:
    - padding-top: 40px;
    - padding-right: 30px;
    - padding-bottom: 20px;
    - padding-left: 10px;

Padding Shorthand property:

- Padding shorthand property let's you specify all the padding properties as values on a single line.
- 4 values- Example: padding: 6px 11px 4px 9px;
    - padding-top: 6px;
    - padding-right: 11px;
    - padding-bottom: 4px;
    - padding-left: 9px;

- 3 values - Example: padding: 5px 10px 20px;
  - padding-top: 5px;
  - padding-left: 10px; and padding-right: 10px;
  - padding-bottom: 20px;
- 2 values - Example: padding: 5px 10px;
  - padding-top: 5px; and padding-bottom: 5px;
  - padding-right: 10px; and padding-left: 10px;

CSS-Margin:

- Margin refers to the space directly outside the box, or margin can be referred to the space between the border of two elements.
- The margin property of the CSS is used to specify the size of the space between the borders of two elements.
- Example:

```
P{
margin:
20px;
}
```

- The margin property can be specifically used to all four sides:
  - margin-top: 40px;
  - margin-right: 30px;
  - margin-bottom: 20px;
  - margin-left: 10px;

Margin shorthand property:

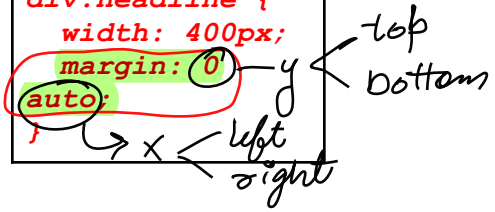
- Margin shorthand property let's you specify all the padding properties as values on a single line.
- 4 values- Example: padding: 6px 11px 4px 9px; where,
  - margin-top: 6px;
  - margin-right: 11px;
  - margin-bottom: 4px;
  - margin-left: 9px;

- 3 values - Example: margin: 5px 10px 20px; where,
  - margin-top: 5px;
  - margin-right: 10px; and margin-left: 10px;
  - margin-bottom: 20px;
- 2 values - Example: margin: 5px 10px;
  - margin-top: 5px; and margin-bottom: 5px;
  - margin-left: 10px; and margin-right: 10px;

margin: auto;

- The margin property also allows you to center content, with help of the value "auto".
- Example:

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



  - In the example above, the width of the div is set to 400 pixels, which is less than the width of most screens.
  - This will cause the div to center within a containing element that is greater than 400 pixels wide.
  - It's not possible to center an element that takes up the full width of the page, so the width is given for the div element.
  - The top and bottom margin of the div element will be set to 0.
  - The auto value instructs the browser to adjust the left and right margins until the element is centered within its containing element.

Margin collapse: top, bottom } margins

- padding is space added inside an element's border, while margin is space added outside an element's border.
- One additional difference is that top and bottom margins, also called vertical margins, **collapse**, while top and bottom padding does not.
- Horizontal margins (left and right), like padding, are always displayed and added together.

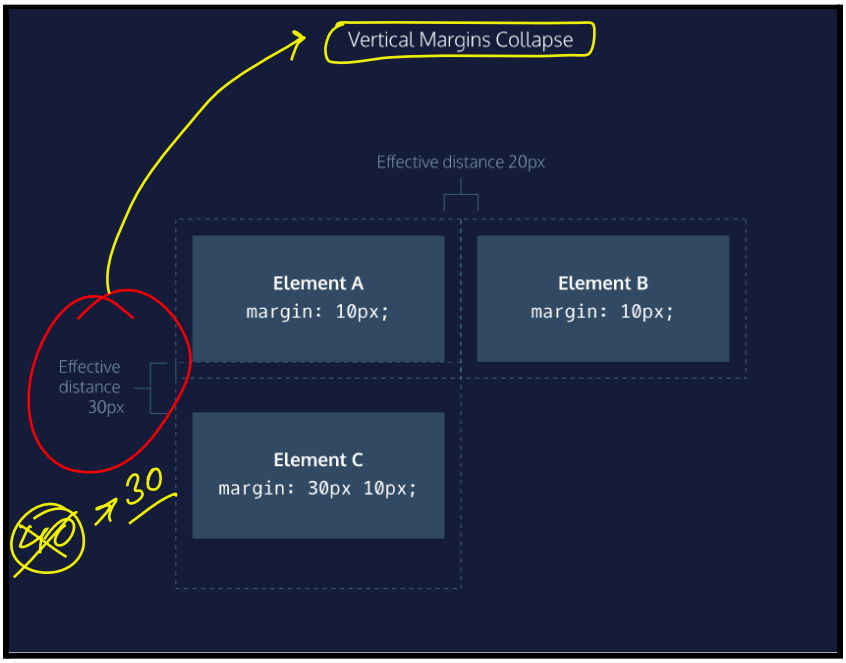
- Example:

```
#elementa {
  margin: 10px;
}

#elementb {
  margin: 10px;
}

#elementc {
  margin: 30px 10px;
}
```

- For elements A and B the all the margin spaces are 10px. For element C the top and bottom margin is 30px and left and right margin is 10px;
- The elements A and B are placed horizontally next to each other. So the space between A and B elements is (10 +10) 20px. Which will remain the same.
- The element A and C are placed vertically one below the other(element C is placed below the element A).
    - The bottom margin of element A is 10px;
    - The top margin of element C is 30px;
    - The total space is 40px but the vertical gap between element A and C will be 30px. This is known as the Margin collapse.



Margin collapse

Minimum and maximum Height and Width:

- The web page can be viewed through 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:
    - min-width—this property ensures a minimum width of an element's box.
    - max-width—this property ensures a maximum width of an element's box.

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

    - the width of all paragraphs will not shrink below 300 pixels, nor will the width exceed 600 pixels.
- Similar to width, You can also limit the minimum and maximum *height* of an element:
    - min-height — this property ensures a minimum height for an element's box.
    - max-height — this property ensures a maximum height of an element's box.
    - Example:

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

    - the height of all paragraphs will not shrink below 150 pixels and the height will not exceed 300 pixels.

Practice

Overflow:

The overflow property controls what happens to content that spills, or overflows, outside its box.

The most commonly used values are: auto, hidden, visible, scroll

- hidden—when set to this value, any content that overflows will be hidden from view.
- scroll—when set to this value, a scrollbar will be added to the element's box so that the rest of the content can be viewed by scrolling.
- visible—when set to this value, the overflow content will be displayed outside of the containing element. Note, this is the default value.
- Example:

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

Best

overflow: auto !important

overflow-y : scroll (Vertical)

overflow-x : scroll (Horizontal)