CSS Essentials

Box model

Overview

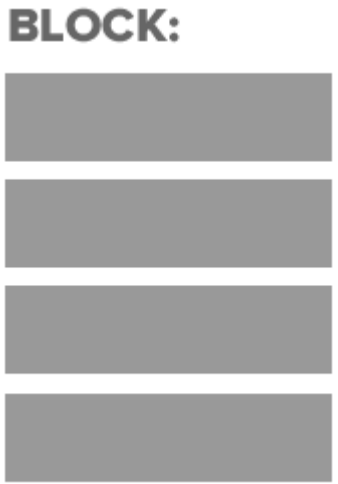
1. Block and Inline boxes
2. Inner and outer display types
3. The CSS box model
4. Use browser DevTools to view the box model
5. Margins, padding, and borders
6. The box model and inline boxes
7. Using display: inline-block

Section 1

Block and Inline boxes

1. Block and Inline boxes

- **Block box:**
  - The box will break onto a new line.
  - The width and height properties are respected.
  - Padding, margin and border will cause other elements to be pushed away from the box



1. Block and Inline boxes

➢ Inline box:

- The box will not break onto a new line.
- The width and height properties will not apply.
- Vertical padding, margins, and borders will apply but will not cause other inline boxes to move away from the box.
- Horizontal padding, margins, and borders will apply and will cause other inline boxes to move away from the box

Section 2

Inner and outer display types

2. Inner and outer display types

- **Inner display types:**
  - ***Table:*** These elements behave like HTML <table> elements. It defines a block-level box.
  - ***Flex:*** The element behaves like a block element and lays out its content according to the flexbox model.
  - ***Grid:*** The element behaves like a block element and lays out its content according to the grid model.

2. Inner and outer display types

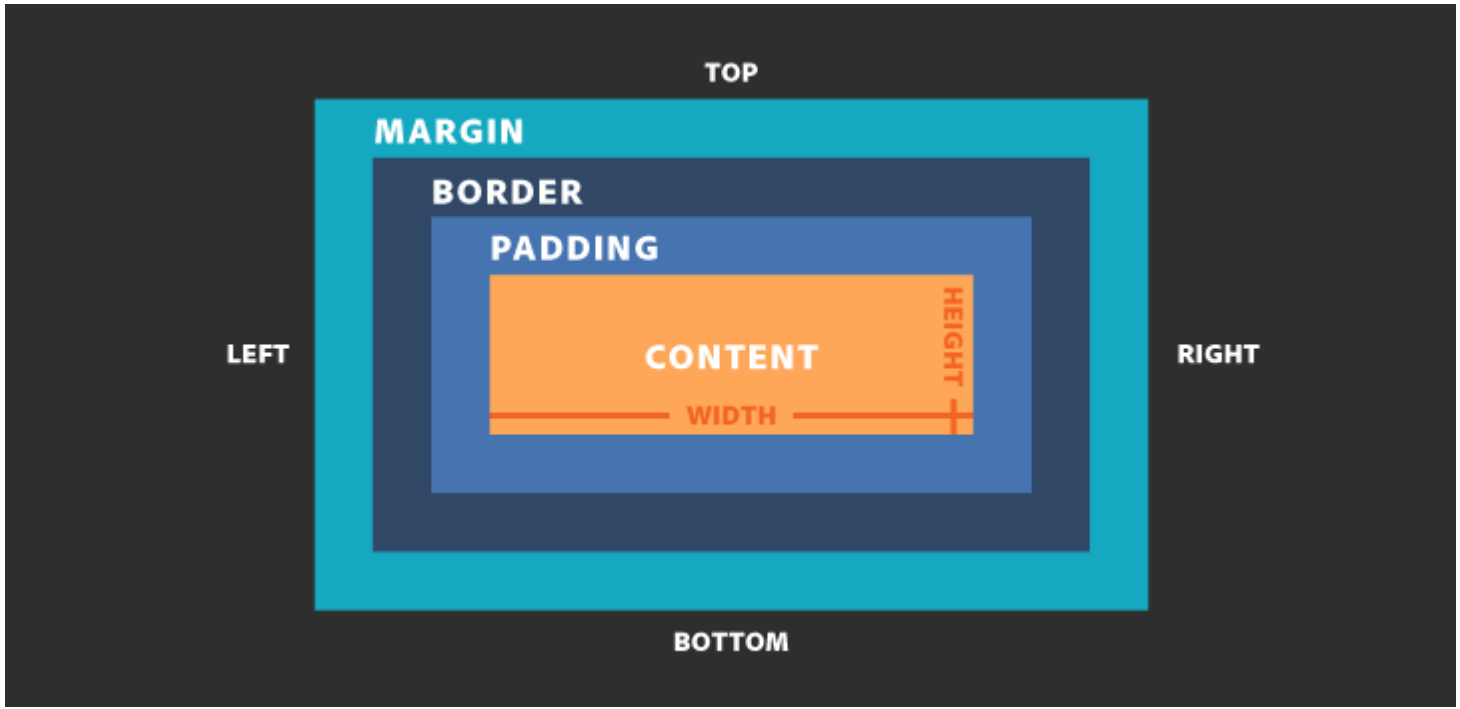
- **Outer display types:**
  - ***block:*** The element generates a block element box, generating line breaks both before and after the element when in the normal flow.
  - ***inline:*** The element generates one or more inline element boxes that do not generate line breaks before or after themselves. In normal flow, the next element will be on the same line if there is space.
  - ***run-in:*** The element generates a run-in box. If the adjacent sibling of the element defined as display: run-in box is a block box, the run-in box becomes the first inline box of the block box that follows it.

Section 3

The altenative CSS Box Model

3. The altenative CSS Box Model

- **Box model**



3. The CSS Box Model

- **Making up a block box in CSS we have:**
  - ***Content box:*** The area where your content is displayed, which can be sized using properties like width and height.
  - ***Padding box:*** The padding sits around the content as white space; its size can be controlled using padding and related properties.
  - ***Border box:*** The border box wraps the content and any padding. Its size and style can be controlled using border and related properties.
  - ***Margin box:*** The margin is the outermost layer, wrapping the content, padding and border as whitespace between this box and other elements. Its size can be controlled using margin and related properties.

3. The CSS Box Model

- In the **Standard box model,** if you give a box a width and a height attribute, this defines the width and height of the content box. Any padding and border is then added to that width and height to get the total size taken up by the box

The space taken up by our box using the standard box model will actually be 410px (350 + 25 + 25 + 5 + 5), and the height 210px (150 + 25 + 25 + 5 + 5), as the padding and border are added to the width used for the content box.

3. The CSS Box Model

- In the **Alternative box model,** any width is the width of the visible box on the page, therefore the content area width is that width minus the width for the padding and border.

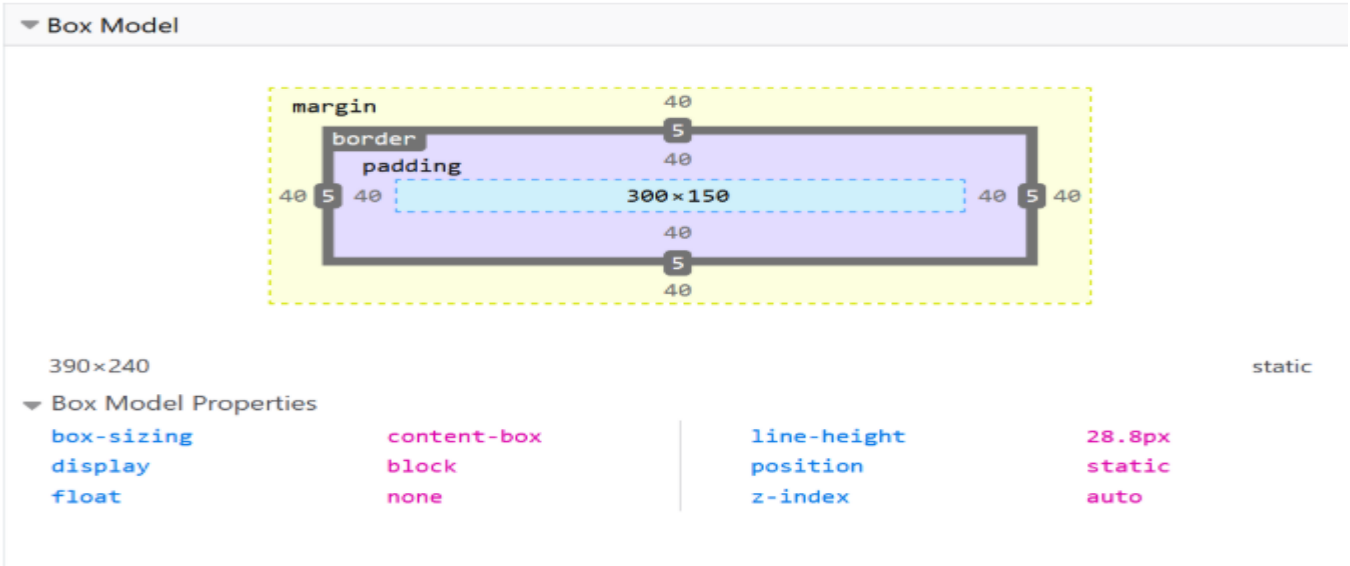
width = 350px, height = 150px

Section 4

Use browser DevTools to view the box model

4. Use browser DevTools to view the box model

- If you inspect an element in browser DevTools, you can see the size of the element plus its margin, padding, and border. Inspecting an element in this way is a great way to find out if your box is really the size you think it is

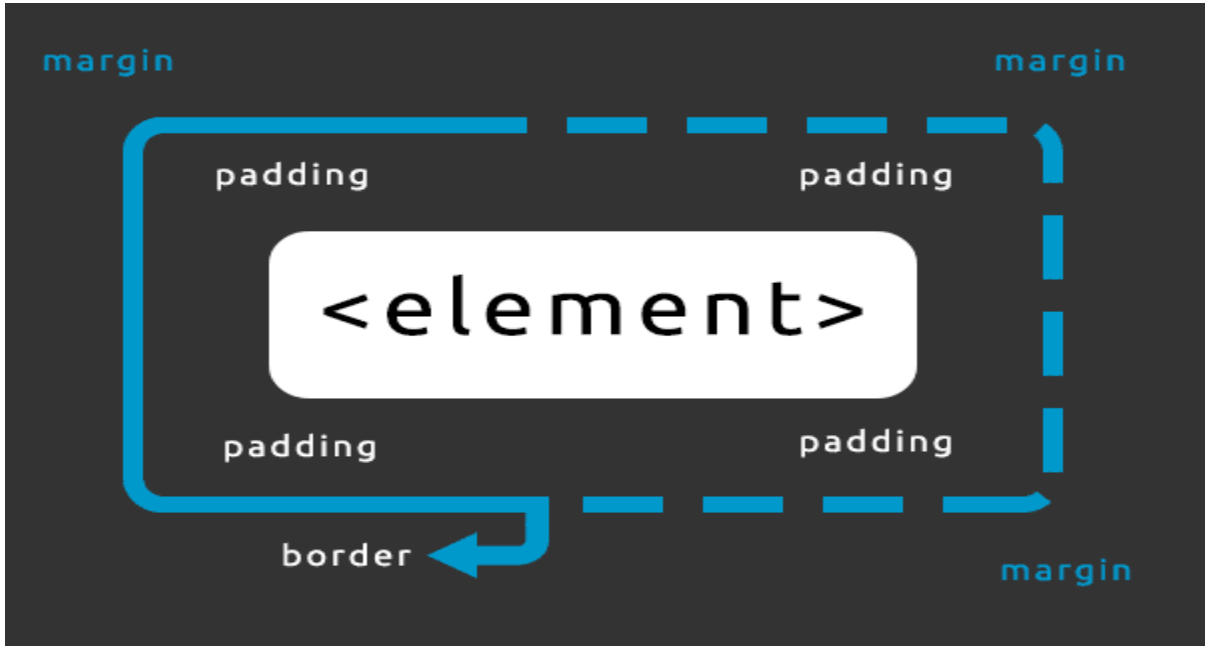


Section 5

Margins, paddings and borders

5. Margins, paddings and borders

- **Take a look**



5. Margins, paddings and borders

- **Margin:** Is an invisible space around your box. It pushes other elements away from the box. Margins can have positive or negative values. Setting a negative margin on one side of your box can cause it to overlap other things on the page
- We can control all margins of an element at once using the margin property, or each side individually using the equivalent longhand properties:
  - ***margin-top***
  - ***margin-right***
  - ***margin-bottom***
  - ***margin-left***

5. Margins, paddings and borders

- **Padding:** Sits between the border and the content area. Unlike margins you cannot have negative amounts of padding, so the value must be 0 or a positive value, it is typically used to push the content away from the border.
- We can control all margins of an element at once using the margin property, or each side individually using the equivalent longhand properties:
  - ***padding-top***
  - ***padding-right***
  - ***padding-bottom***
  - ***padding-left***

5. Margins, paddings and borders

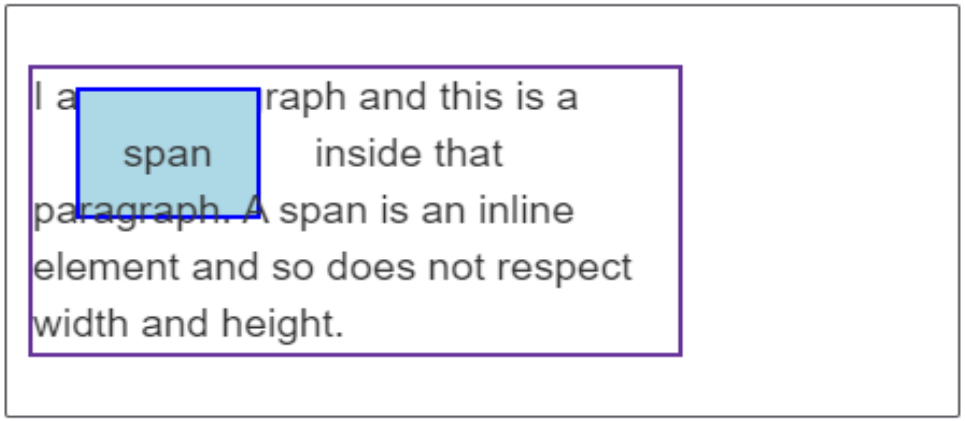
- **Border:** is drawn between the margin and the padding of a box.
- To set the properties of each side individually, you can use:
  - ***border-top***
  - ***border-right***
  - ***border-bottom***
  - ***border-left***
- To set the width, style, or color of all sides, use the following:
  - ***border-width***
  - ***border-style*** *(dotted, dashed, solid , double, groove, inset, outset,..)*
  - ***border-color***

Section 6

The box model and inline boxes

6. The box model and inline boxes

- With an **Inline box**, that the width and height are ignored. The margin, padding, and border are respected but they do not change the relationship of other content to our inline box and so the padding and border overlaps other words in the paragraph.



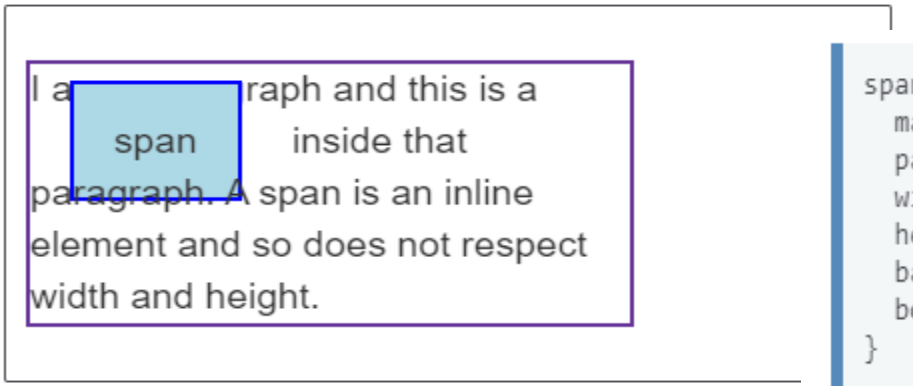

```
span {
  margin: 20px;
  padding: 20px;
  width: 80px;
  height: 50px;
  background-color: lightblue;
  border: 2px solid blue;
}
```

Section 7

Using display: inline-block

7. Using display: inline-block

- Value **“inline-block”** provides a middle ground between inline and block.
  - The *width* and *height* properties are respected
  - *padding*, *margin*, and *border* will cause other elements to be pushed away from the box.




```
span {
  margin: 20px;
  padding: 20px;
  width: 80px;
  height: 50px;
  background-color: lightblue;
  border: 2px solid blue;
}
```

8. Quiz

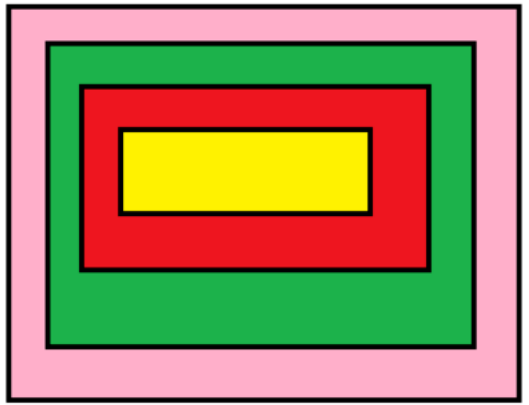
- **Which of the following is not a part of the box model?**

A. Radius
B. Padding
C. Margin
D. content

8. Quiz

- **Assuming the yellow area is the content, what is the area between the content and the border?**

A. padding
B. margin
C. height
D. width



Box Model

Thank you