

Box Model & Layout Basics

Web Development Essentials - Session 6

Session Overview

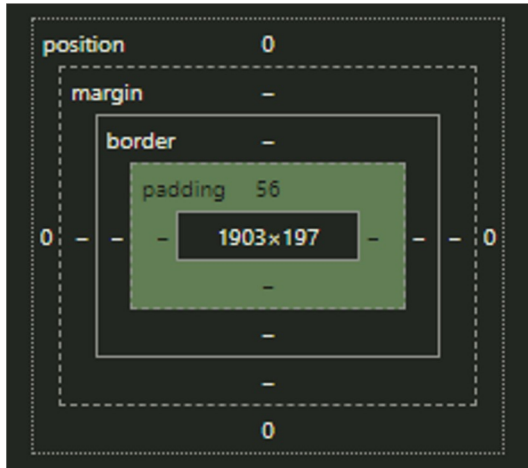
Learning Goals for Today:

- Understand the CSS box model and how it defines element spacing
- Explore different CSS positioning methods
- Learn about display properties and their role in layout

What is the Box Model?

Definition: The box model describes the structure of every HTML element as a rectangular box, consisting of:

- **Content:** The actual content inside the box (text, image, etc.).
- **Padding:** Space between the content and the border.
- **Border:** Surrounds the padding and content.
- **Margin:** Space outside the border, separating the element from others.



Understanding the Box Model

Width: The content width (200px).

Padding: Adds 10px around the content inside the box.

Border: A 5px solid border surrounds the padding.

Margin: Creates 20px of space outside the box.

```
.box {  
  width: 200px;  
  padding: 10px;  
  border: 5px solid black;  
  margin: 20px;  
}
```

Margin and Padding

Margin:

- Controls the space outside the element's border.

```
.element {  
  margin: 20px;  
}
```

Padding:

- Controls the space between the content and the element's border.

```
.element {  
  padding: 15px;  
}
```

Border Properties

Border: Defines the outline around an element

```
.box {  
  border: 3px solid blue;  
}
```

Customizing Borders:

- **Width:** `border-width: 3px;`
- **Style:** `border-style: solid;`
- **Color:** `border-color: blue;`

CSS Positioning Overview

Purpose: CSS positioning allows you to define how an element is placed on the page.

Four Types of Positioning:

1. **Static** (default)
2. **Relative**
3. **Absolute**
4. **Fixed**

Static Positioning

Static Positioning (Default):

- All elements are positioned statically by default, meaning they appear in the natural flow of the document.
- No special positioning is applied.

```
.static-element {  
  position: static;  
}
```


Relative Positioning

Relative Positioning:

- Moves the element relative to its original static position.
- Does not remove the element from the document flow.

```
.relative-element {  
  position: relative;  
  top: 10px;  
  left: 20px;  
}
```

- **Result:** The element is shifted 10px down and 20px to the right.

Absolute Positioning

Absolute Positioning:

- Removes the element from the document flow and positions it relative to the nearest positioned ancestor (non-static element) or the viewport.

```
.absolute-element {  
  position: absolute;  
  top: 50px;  
  right: 30px;  
}
```

- **Result:** The element is placed 50px from the top and 30px from the right of its positioned parent or the browser window.

Fixed Positioning

Fixed Positioning:

- Similar to absolute, but the element is positioned relative to the viewport (browser window) and stays in place even when scrolling.

```
.fixed-element {  
  position: fixed;  
  bottom: 0;  
  left: 0;  
}
```

- **Use Case:** Often used for sticky headers or footers.

Display Properties

CSS Display: Defines how an element is displayed on the page.

- **Block:**

- Takes up the full width of the container.
- Starts on a new line.
- Example: `<div>`, `<p>`.

```
div {  
  display: block;  
}
```

- **Inline:**

- Takes up only as much width as the content.
- Does not start on a new line.
- Example: ``, `<a>`

```
span {  
  display: inline;  
}
```

- **Inline-Block:**

- Behaves like an inline element but allows setting width and height like a block element.

```
.inline-block-element {  
  display: inline-block;  
  width: 100px;  
  height: 50px;  
}
```

Display Properties in Action

```
.block {  
  display: block;  
  background-color: lightblue;  
  width: 100%;  
}  
.inline {  
  display: inline;  
  background-color: lightgreen;  
}
```

Block Element: Spans the entire width of the container.

Inline Element: Only takes up as much width as its content.

Hands-On Activity

Goal: Practice applying the box model and positioning elements using CSS.

- Use margin, padding, and border to style a container.
- Experiment with different positioning types.
- Explore block, inline, and inline-block displays.

Instructions:

- Open your HTML file and apply different layout techniques using CSS.
- Style a few elements with borders, margins, and positioning.

Summary

What We Learned Today:

- How the CSS box model works with margin, padding, border, and content
- Different CSS positioning methods: static, relative, absolute, and fixed
- Display properties and their role in layout: block, inline, and inline-block

Questions?

Q&A Session

- Any questions before we wrap up?

Thank You & See You in the Next Class!