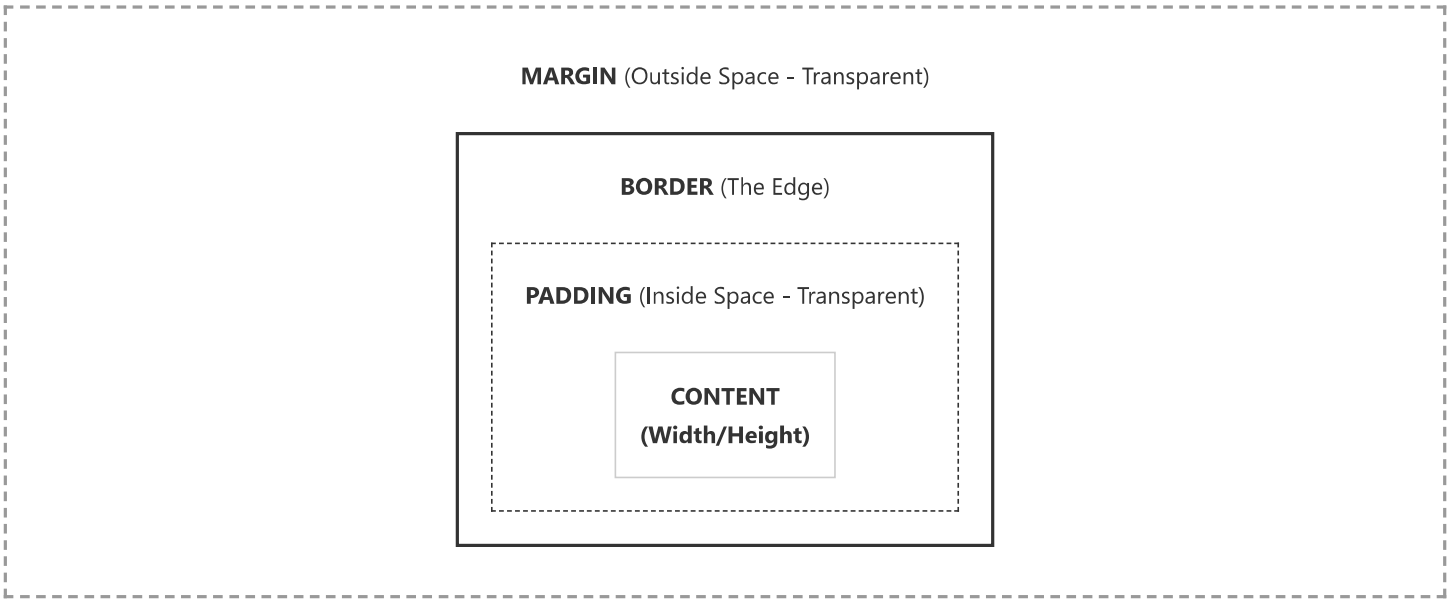
CSS FUNDAMENTALS: BOX MODEL & DISPLAY

Instructor: Saurabh Sir

Part 1: The CSS Box Model

The **Box Model** is the foundation of layout in CSS. It states that every element on a webpage (Heading, Image, Div) is essentially a rectangular box consisting of four layers.



The 4 Components Explained:

1. **Content:** The innermost part where text or images appear.
2. **Padding:** Space *inside* the border. It pushes the border away from the content. It takes the background color of the element.
3. **Border:** The line around the padding. You can control its style, width, and color.
4. **Margin:** Space *outside* the border. It separates the element from its neighbors. Margins are always transparent.

Part 2: CSS Display Properties

The `display` property defines how an element behaves on the page (whether it sits on a new line or the same line).

Display Type	New Line?	Width/Height?	Top/Bottom Margin?	Examples
Block	✅ Yes	✅ Works	✅ Works	`<div>`, `<p>`, `<h1>`
Inline	❌ No	❌ Ignored	❌ Ignored	`<span>`, `<a>`
Inline-Block	❌ No	✅ Works	✅ Works	Buttons, Images

Part 3: Box Sizing (`border-box`)

This is a critical concept for modern layouts.

Default Behavior (`content-box`):
If you set Width = 100px and Padding = 20px, the actual width becomes **140px**. This often breaks the layout.

The Fix (`border-box`):
If you set Width = 100px and Padding = 20px, the actual width stays **100px**. The padding is adjusted *inside* the box.

💻 Practical Coding Exercises

Copy the code below into VS Code to demonstrate these concepts live to students.

Task 1: Padding vs. Margin (Visual Difference)

Goal: Show that Padding adds internal space (making the box fat), while Margin adds external space (pushing boxes apart).

```
<style> .box { border: 2px solid blue; background-color: lightblue; display: inline-block; } /* Adds space INSIDE the border */ .pad-box { padding: 30px; } /* Adds space OUTSIDE the border */ .mar-box { margin: 30px; } </style> <h3>Example 1</h3> <div class="box">Normal</div> <div class="box pad-box">Padding (Inside)</div> <div class="box mar-box">Margin (Outside)</div>
```

Task 2: Inline vs. Inline-Block (The Height Issue)

Goal: Show why height/width fails on span tags unless display is changed.

```
<style> .test-span { background: orange; border: 1px solid black; width: 100px; height: 100px; color: white; } /* Default Inline: Height/Width are IGNORED */ .inline { display: inline; } /* Inline-Block: Height/Width are RESPECTED */ .block { display: inline-block; } </style> <h3>Example 2</h3> <span class="test-span inline">I am Inline (Size fails)</span> <br><br> <span class="test-span block">I am Inline-Block (Size works)</span>
```

Task 3: The Layout Fix (Border-Box)

Goal: Demonstrate how to keep element sizes consistent.

```
<style> .container { width: 300px; height: 100px; background: pink; border: 5px solid red; padding: 20px; margin-bottom: 20px; } /* Total Width becomes 350px (Bad) */ .bad { box-sizing: content-box; } /* Total Width stays 300px (Good) */ .good { box-sizing: border-box; } </style> <h3>Example 3</h3> <div class="container bad">Without Fix (Too Big)</div> <div class="container good">With Border-Box (Perfect) </div>
```

Prepared by Saurabh Sir