

# Day 7 – Display Types, Box Model & Positioning

---

## 1. Display Types

The `display` property controls how an element is rendered in the document flow.

### a. Common display values

1. **block** – Takes up full width, starts on a new line (`<div>`, `<p>`, `<h1>` by default).
2. **inline** – Fits in with text, only takes up as much width as its content (`<span>`, `<a>` by default).
3. **inline-block** – Like inline, but allows setting width and height.
4. **flex** – Turns the element into a flex container for flexible layouts.
5. **grid** – Creates a grid layout container.

### Example:

```
.block-example {  
  display: block;  
  background: lightblue;  
}  
  
.inline-example {  
  display: inline;  
  background: pink;  
}  
  
.inline-block-example {  
  display: inline-block;  
  width: 100px;  
  height: 50px;  
  background: lightgreen;  
}
```

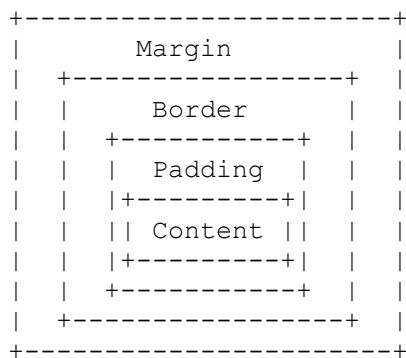
---

## 2. The Box Model

Every HTML element is treated as a box, made up of 4 layers:

1. **Content** – The actual text or image inside.
2. **Padding** – Space between the content and the border.
3. **Border** – The line around the padding.
4. **Margin** – Space outside the border (between elements).

## 📌 Diagram:



## Example:

```
.box {  
  width: 200px;  
  padding: 20px;  
  border: 5px solid black;  
  margin: 15px;  
}
```

---

## 3. box-sizing

By default, `element width = content width + padding + border`.  
You can change this behavior:

- **content-box** (*default*) – Width is for content only.
- **border-box** – Width includes padding and border.

## Example:

```
.content-box {  
  box-sizing: content-box; /* default */  
  width: 200px;  
  padding: 20px;  
  border: 5px solid black;  
}  
  
.border-box {  
  box-sizing: border-box;  
  width: 200px;  
  padding: 20px;  
  border: 5px solid black;  
}
```

---

## 4. Positioning

The `position` property determines how an element is placed.

- **static** – Default flow (no positioning).
- **relative** – Positioned relative to itself.
- **absolute** – Positioned relative to nearest positioned ancestor.
- **fixed** – Stays in place relative to the viewport.
- **sticky** – Switches between relative and fixed based on scroll.

### Example:

```
.relative {  
  position: relative;  
  top: 20px;  
  left: 10px;  
}  
  
.absolute {  
  position: absolute;  
  top: 50px;  
  right: 20px;  
}  
  
.fixed {  
  position: fixed;  
  bottom: 10px;  
  right: 10px;  
}
```

---

## 5. Practice Quiz

1. Which `display` value allows an element to set width & height but still flow inline?
2. In the box model, which property adds space **outside** the border?
3. What does `box-sizing: border-box` do?
4. What's the difference between `absolute` and `fixed` positioning?