

# Web Development Notes: Class 5

## CSS Layouts: Flexbox & Positioning

### 1. The `display` Property

This is the most important property for controlling layout.

- `block`: Takes up the full width (starts on a new line). Example: `<div>`, `<h1>`, `<p>`.
- `inline`: Takes up only as much width as necessary (does not start on a new line). Example: `<span>`, `<a>`.
- `none`: Hides the element completely.

### 2. Position Property

- `Static`: Default. Elements appear in order.
- `Relative`: Positioned relative to its normal position.
- `Absolute`: Positioned relative to the nearest positioned ancestor (removed from flow).
- `Fixed`: Stays in the same place even when the page is scrolled (like a navbar).

### 3. Flexbox (Flexible Box Layout)

Flexbox is designed for laying out items in a single dimension (row or column).

**The Container:**

## CSS

```
.container {  
display: flex;  
justify-content: center; /* Aligns horizontally */  
align-items: center; /* Aligns vertically */  
}
```

### Common Properties:

- `flex-direction`: row (default) or column.
- `justify-content`: flex-start, flex-end, center, space-between.
- `flex-wrap`: wrap (allows items to go to next line if there is no space).