Training Day 13 Report

11 July 2025

FLEXBOX (Flexible Box Layout)

Q. What is Flexbox?

Flexbox is a 1-dimensional layout system in CSS used to arrange items in rows or columns, and distribute space efficiently between them.

It makes it easier to build responsive layouts, like navbars, cards, or galleries, where items can grow, shrink, or align based on the container.

SYNTAX

HTML:
display: flex;
flex-direction: row | column;
gap: 10px;
justify-content: ...;
align-items: ...;

CSS:

<nav style="display: flex; gap: 20px;">
 Home
 Gallery
 Contact
</nav>

MAIN AXIS vs CROSS AXIS

Property	flex-direction: **row**	flex-direction: **column**
Main Axis	Left → Right	Top → Bottom
Cross Axis	Top ↓ Bottom	Left → Right
justify-content	Aligns along **main axis**	Aligns Top → Bottom
align-items	Aligns along **cross axis**	Aligns Left → Right

flex-direction

Controls the direction of items in the flex container.

CSS:

flex-direction: row;   /* default */

flex-direction: column;

Example:

.container {

 display: flex;

 flex-direction: column;

}

justify-content

Aligns items along the main axis.

Example:

.container {

 display: flex;

 justify-content: space-between;

}

align-items

Aligns items along the cross axis.

Example:

.container {

 display: flex;

 align-items: center;

}

VISUAL UNDERSTANDING

If flex-direction: row:

->Main Axis = horizontal (left ↔ right)

->justify-content: controls left ↔ right

->align-items: controls top ↕ bottom

.container {

 display: flex;

 flex-direction: row;

 justify-content: center;   /* left to right */

 align-items: flex-start;   /* top aligned */

}

If flex-direction: column:

- Main Axis = vertical (top ↕ bottom)
- justify-content: controls top ↕ bottom
- align-items: controls left ↔ right

```css
.container {
 display: flex;
 flex-direction: column;
 justify-content: space-evenly;   /* top to bottom */
 align-items: center;         /* horizontally center */
}
```

gap Property

Adds space between flex items (replaces need for margins).

```css
.container {
 display: flex;
 gap: 20px;
}
```

Example:

```html
<div style="
   display: flex;
   flex-direction: row;
   justify-content: space-between;
   align-items: center;
   gap: 15px;
   background-color: #f2f2f2;
   padding: 10px;
">
 <div style="background: #2196f3; color: white; padding: 10px;">Box 1</div>
 <div style="background: #4caf50; color: white; padding: 10px;">Box 2</div>
 <div style="background: #ff5722; color: white; padding: 10px;">Box 3</div>
</div>
```