**CSS Flexbox**

**1. What is Flexbox?**

**Definition:**

* Flexbox (Flexible Box Layout) is a **CSS layout module** for designing **one-dimensional layouts** (either row or column).
* It **distributes space** among items and **aligns them** within a container efficiently, even when sizes are unknown.
* Modern alternative to floats and inline-block for layout management.

**2. Container & Items**

* **Flex Container:** The parent element with display: flex or display: inline-flex.
* **Flex Items:** The direct children of the flex container.

**Example:**

<div class="flex-container">

<div>Item 1</div>

<div>Item 2</div>

<div>Item 3</div>

</div>

.flex-container {

display: flex;

}

* Item 1, 2, 3 are now **flex items**.

**3. Key Flex Container Properties**

**A. display: flex / inline-flex**

| **Property** | **Description** |
| --- | --- |
| display: flex | Makes the container a block-level flex container |
| display: inline-flex | Makes the container an inline-level flex container |

**B. flex-direction**

* Defines the **main axis** (direction of flex items).
* Values:

| **Value** | **Description** |
| --- | --- |
| row (default) | Left → Right |
| row-reverse | Right → Left |
| column | Top → Bottom |
| column-reverse | Bottom → Top |

**Example:**

.flex-container {

display: flex;

flex-direction: column;

}

**C. justify-content**

* Aligns **flex items along the main axis**.
* Common values:

| **Value** | **Effect** |
| --- | --- |
| flex-start | Align items to the start (default) |
| flex-end | Align items to the end |
| center | Center items |
| space-between | Equal space between items, first and last at edges |
| space-around | Equal space around each item |
| space-evenly | Equal space between and around items |

**Example:**

.flex-container {

display: flex;

justify-content: space-between;

}

**D. align-items**

* Aligns **flex items along the cross axis** (perpendicular to main axis).
* Values:

| **Value** | **Effect** |
| --- | --- |
| stretch (default) | Items stretch to fill container |
| flex-start | Align to top (row) or left (column) |
| flex-end | Align to bottom (row) or right (column) |
| center | Center items on cross axis |
| baseline | Align items’ text baseline |

**Example:**

.flex-container {

display: flex;

align-items: center;

}

**E. gap**

* Adds **space between flex items**.
* Replaces the need for margins on individual items.
* Supports row-gap and column-gap individually.

**Example:**

.flex-container {

display: flex;

gap: 20px; /\* 20px space between items \*/

}

**4. Flex Item Properties (Optional Basics)**

| **Property** | **Description** |
| --- | --- |
| flex-grow | Specifies how much an item will grow relative to others |
| flex-shrink | Specifies how much an item will shrink |
| flex-basis | Default size of the item before growing/shrinking |
| align-self | Overrides container’s align-items for individual items |

**Example:**

.flex-item {

flex: 1; /\* shorthand for grow 1, shrink 1, basis auto \*/

align-self: flex-end; /\* override container alignment \*/

}

**5. Practical Example**

<div class="flex-container">

<div class="item">1</div>

<div class="item">2</div>

<div class="item">3</div>

</div>

.flex-container {

display: flex;

flex-direction: row;

justify-content: space-around;

align-items: center;

gap: 15px;

height: 150px;

border: 2px solid #333;

}

.item {

background-color: lightblue;

padding: 20px;

text-align: center;

}

**Explanation:**

* **Direction:** Horizontal (row)
* **Justify:** Items spaced evenly with space-around
* **Align-items:** Center vertically
* **Gap:** 15px between each item
* Container height defines **cross-axis** alignment

**6. Quick Tips / Interview Points**

1. **Flexbox is 1D layout** → either row or column.
2. justify-content → **main axis** alignment, align-items → **cross axis** alignment.
3. gap simplifies spacing without extra margin.
4. flex-direction: column flips main axis → justify-content and align-items swap roles visually.
5. align-self → override container alignment for a specific item.
6. Flexbox is responsive-friendly → items adjust automatically in container.

**7. Common Interview Questions**

1. Difference between **block, inline, and flex** layouts.
2. Explain **main axis vs cross axis** in flexbox.
3. Difference between **justify-content** and **align-items**.
4. How to **center an element both horizontally and vertically** using flexbox.
5. Difference between **gap vs margin** in flex items.
6. When to use **flex-direction: column** vs row