Flexbox: Mastering One-Dimensional Layouts

Flexbox is a powerful CSS layout model that arranges elements in a single direction—either rows or columns—while giving you precise control over alignment, spacing, and size distribution.

One-Dimensional

Elements flow in one direction: horizontal or vertical

Flexible Sizing

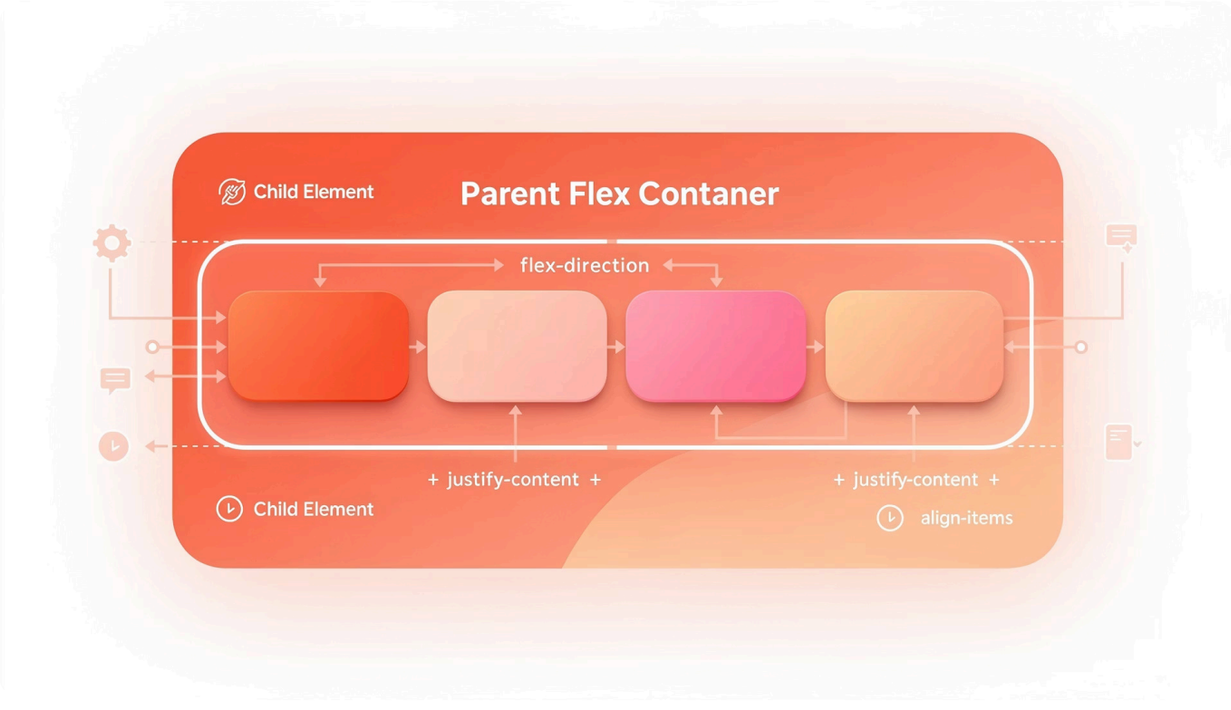
Items grow, shrink, and adapt to available space

Alignment Control

Precise positioning along both axes

What is Layout? Layout determines how elements are arranged and positioned on a webpage—where they appear, how they align, the space they occupy, and how they relate to each other.

The Two-Level System



Flexbox operates on two distinct levels that work together to create flexible layouts.

Flex Container

The parent element that holds flex items

Flex Items

Direct children of the container that flex and flow

Understanding Flexbox Axes

1

Main Axis

Direction set by `flex-direction`

Horizontal (row) or vertical (column)

2

Cross Axis

Always perpendicular to main axis

Handles alignment in opposite direction

Essential Flexbox Properties

1

flex-direction

Defines layout direction

```
flex-direction: row | column |
row-reverse | column-reverse;
```

- `row`: left to right (main), top to bottom (cross)
- `column`: top to bottom (main), left to right (cross)

2

justify-content

Aligns items **along the main axis**

```
justify-content: flex-start | center |
flex-end | space-between | space-around;
```

Controls horizontal spacing in rows, vertical in columns

3

align-items

Aligns items **along the cross axis**

```
align-items: flex-start | center |
flex-end | stretch | baseline;
```

Controls vertical alignment in rows, horizontal in columns

4

align-content

Aligns flex lines along the cross axis (multi-line only)

```
align-content: flex-start | center |
flex-end | space-between | space-around | stretch;
```

Only works when flex-wrap is enabled and there are multiple lines of flex items

align-self: Individual Item Control

The `align-self` property allows individual flex items to override the `align-items` value set on the container, aligning along the cross axis.

Priority Concept

`align-self` overrides `align-items` for individual flex items.

```
align-self: flex-start | flex-end | center | baseline | stretch;
```

- `flex-start`: aligns item to the start of the cross axis.
- `flex-end`: aligns item to the end of the cross axis.
- `center`: centers item on the cross axis.
- `baseline`: aligns item's baseline with other items.
- `stretch`: stretches to fill the container (default).

Flex Sizing: Controlling Item Dimensions

Flexbox provides three properties that control how flex items size themselves within the container, allowing dynamic and responsive layouts.

1

flex-basis

Sets the initial size of a flex item before space distribution

```
flex-basis: auto | <length> | <percentage>;
```

Defines the default size along the main axis (width for row, height for column). Default is auto.

2

flex-grow

Controls how much an item grows relative to others

```
flex-grow: <number>; (default: 0)
```

Determines how remaining space is distributed. Higher values claim more space. 0 means no growth.

3

flex-shrink

Controls how much an item shrinks when space is limited

```
flex-shrink: <number>; (default: 1)
```

Determines how items reduce size when container is too small. 0 prevents shrinking.

Shorthand: `flex: <grow> <shrink> <basis>;`

Example: `flex: 1 1 200px;`