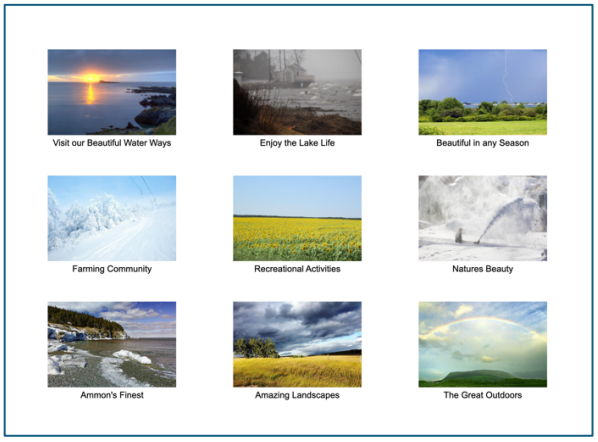
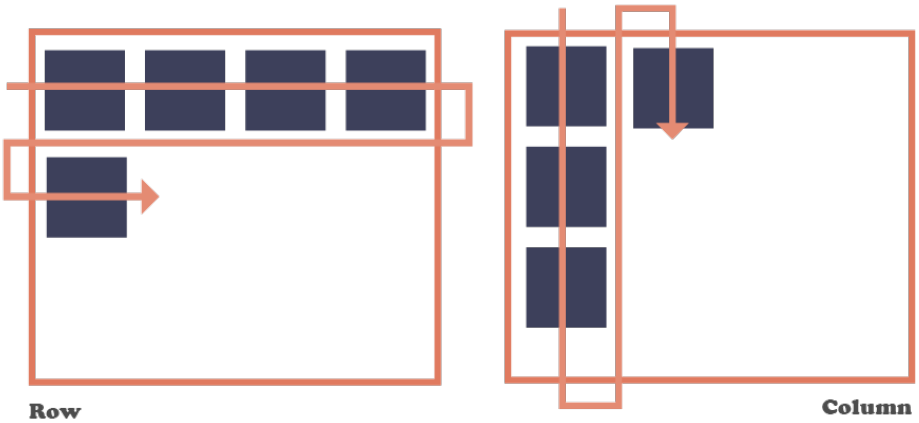
Flexbox

Flexbox (short for **Flexible Box Layout**) is a powerful CSS layout module designed for distributing space and aligning items within a container. Flex containers can expand items to fill available space or shrink them to prevent overflow, making them ideal for responsive layouts.



Flexbox includes a comprehensive set of properties—some apply to the container (called the **flex container**), while others apply to its direct children (**flex items**). Unlike traditional layouts that rely on **block** (vertical) and **inline** (horizontal) flow, Flexbox follows a **flex-flow direction** that can be either horizontal (**row**) or vertical (**column**), depending on the flex-direction property.



Key difference: Flexbox lays items out in a **single axis (row or column)**, while **Grid** handles both axes at once. Use **Grid** for full-page or two-dimensional layouts.

Flex Container and Flex Items

To use Flexbox, apply `display: flex` to a parent element. This defines a **flex container** and makes all direct children into **flex items**.

HTML:

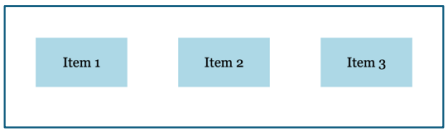
```
<div class="container">
  <div>Item 1</div>
  <div>Item 2</div>
  <div>Item 3</div>
</div>
```

CSS:

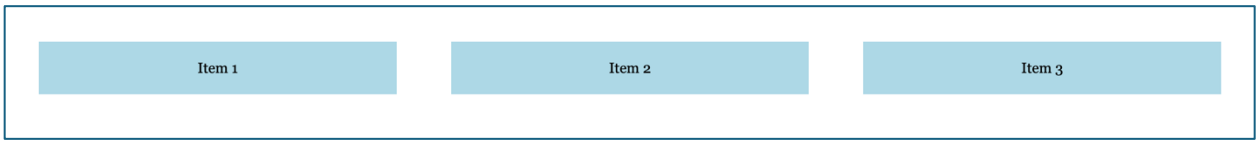
```
.container {
  display: flex;
}
```

The default layout is a **row**, with items arranged horizontally along the **main axis**.

The main idea behind the **Flexbox layout** is to give a container the ability to adjust the size and order of its items to best fit the available space—making it ideal for responsive design across different screen sizes and devices.



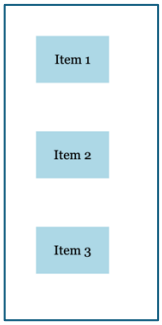
Smaller device



Larger device with a percentage width on each item.

Flex Direction and Wrapping

You can change the flex-direction to column. This is a nice way to make your site responsive on smaller screens.

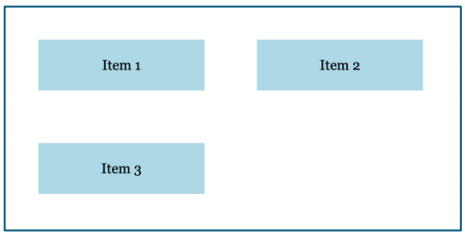


flex-direction: column;

Use **flex-wrap** to allow items to wrap onto new lines, useful for smaller screens:

```
.container {
  display: flex;
  flex-wrap: wrap;
}
```

If the device or screen size is smaller the items will automatically wrap.



Smaller device with flex-wrap: wrap;

Justifying and Aligning Items

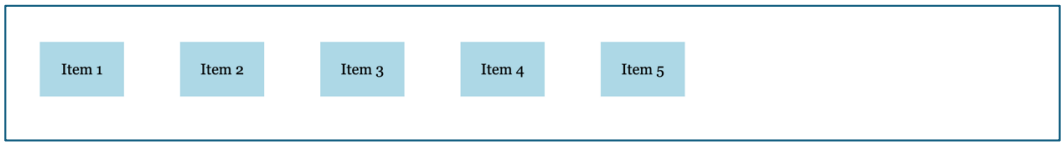
Flexbox simplifies alignment on both axes:

Property	Description
`justify-content`	Aligns items horizontally (if row)

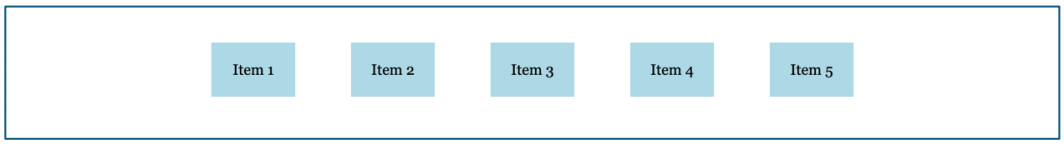
Property	Description
`align-items`	Aligns items vertically
`align-content`	Aligns multiple rows (when wrapped)

`justify-content` Examples:

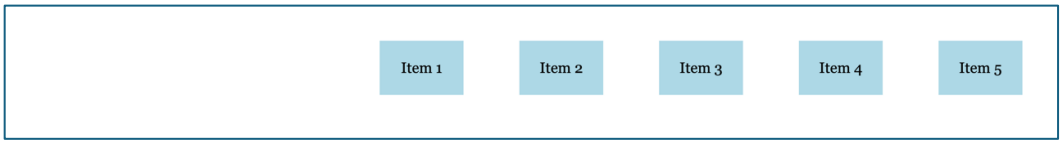
- `flex-start`



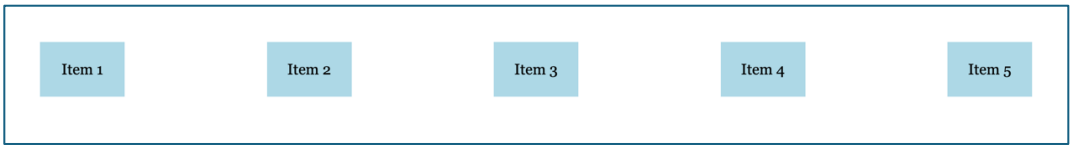
- `center`



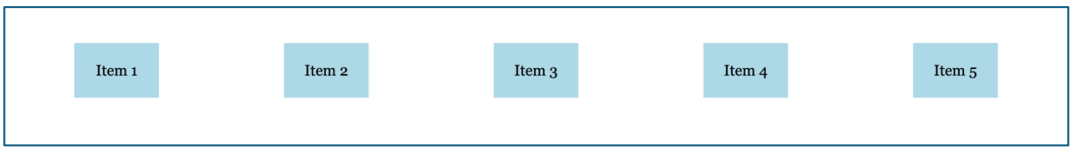
- `flex-end`



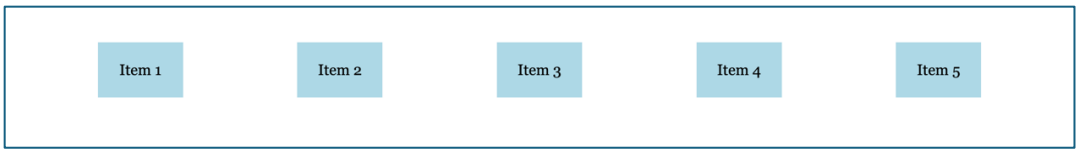
- `space-between`



- `space-around`

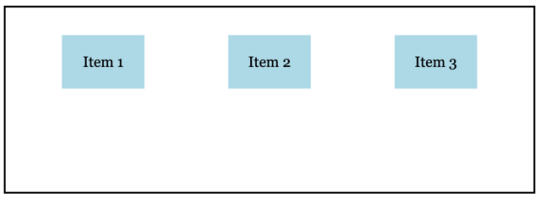


- `space-evenly`

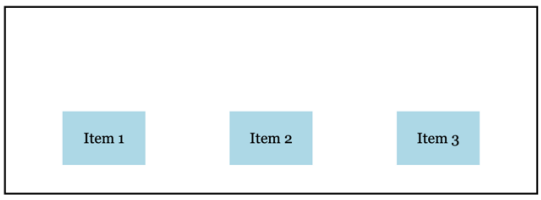


`align-items` Examples:

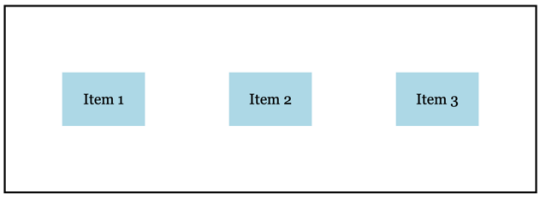
- `flex-start (default)`



- `flex-end`



- `center`



Flex Item Properties

Flex items (children of a flex container) can grow, shrink, and define their base size using the following:

Property	Description
`flex-grow`	Allows an item to grow to fill available space
`flex-shrink`	Allows an item to shrink when necessary
`flex-basis`	Sets the initial size of an item
`flex`	Shorthand for all three: flex: grow shrink basis
`align-self`	Overrides align-items for a single item

Responsive Layout with Flexbox

Flexbox makes it easy to create layouts that adapt to different screen sizes without media queries in many cases.

```
.item {
  flex: 1 1 200px;
}
```

This sets an initial size of `200px`, but allows the item to grow and shrink based on space.

Flexbox makes it easier to build clean, flexible, and accessible interfaces that respond to various screen sizes and content needs.