

CSS Flexbox

Flexbox is a **one-dimensional layout system** in CSS designed to arrange items in **rows OR columns** and **distribute space efficiently**, even when item sizes are unknown or dynamic.

A flexbox always consists of:

- **A Flex Container** - The parent (container) element, where the display property is set to flex or inline-flex
- **One or more Flex Items** - The direct children of the flex container automatically becomes flex items

Example: .container{

```
    display: flex;
```

```
}
```

- **display** - Must be set to flex or inline-flex
- **flex-direction** - Sets the display-direction of flex items
 - row (default)
 - column
 - row-reverse
 - column-reverse
- .flex-container {
 display: flex;
 flex-direction: row;
}
- **flex-wrap** - Specifies whether the flex items should wrap or not
 - nowrap (default)
 - wrap
 - wrap-reverse
- .flex-container {
 display: flex;
 flex-wrap: nowrap;
}
- **flex-flow** - Shorthand property for flex-direction and flex-wrap
 - .flex-container {
 display: flex;

- ```
 flex-flow: row wrap;
}
• justify-content - Aligns the flex items when they do not use all available space on the main-axis (horizontally)

 - center
 - flex-start (default)
 - flex-end
 - space-around
 - space-between
 - space-evenly
○ .flex-container {
 display: flex;
 justify-content: center;
}
• align-items - Aligns the flex items when they do not use all available space on the cross-axis (vertically)

 - center
 - flex-start
 - flex-end
 - stretch
 - baseline
 - normal (default)
○ .flex-container {
 display: flex;
 height: 200px;
 align-items: center;
}
• align-content - Aligns the flex lines when there is extra space in the cross axis and flex items wrap

 - center
 - stretch (default)
 - flex-start
 - flex-end
 - space-around
 - space-between
 - space-evenly
○ .flex-container {
 display: flex;
```

```
 height: 400px;
 flex-wrap: wrap;
 align-content: center;
}
```

## CSS Flex Items

The direct child elements of a flex container automatically become flex items.

- **order** - Specifies the display order of the flex items inside the flex container
  - The first flex item in the source code does not have to appear as the first item in the layout.
  - The order value must be a number, and the default value is 0.
  - <div class="flex-container">  
 <div style="order: 3">1</div>  
 <div style="order: 2">2</div>  
 <div style="order: 4">3</div>  
 <div style="order: 1">4</div>  
 </div>
- **flex-grow** - Specifies how much a flex item will grow relative to the rest of the flex items
  - <div class="flex-container">  
 <div style="flex-grow: 1">1</div>  
 <div style="flex-grow: 1">2</div>  
 <div style="flex-grow: 4">3</div>  
 </div>
- **flex-shrink** - Specifies how much a flex item will shrink relative to the rest of the flex items
  - <div class="flex-container">  
 <div>1</div>  
 <div>2</div>  
 <div style="flex-shrink: 2">3</div>  
 <div>4</div>  
 <div>5</div>  
 <div>6</div>  
 </div>
- **flex-basis** - Specifies the initial length of a flex item

- <div class="flex-container">  
    <div>1</div>  
    <div>2</div>  
    <div style="flex-basis: 250px">3</div>  
    <div>4</div>  
  </div>
- **flex** - Shorthand property for flex-grow, flex-shrink, and flex-basis
  - <div class="flex-container">  
    <div>1</div>  
    <div>2</div>  
    <div style="flex: 1 0 150px">3</div>  
    <div>4</div>  
  </div>
- **align-self** - Specifies the alignment for the flex item inside the flex container
  - This property overrides the default alignment set by the container's align-items property.
  - In these examples we use a 200 pixels high container, to better demonstrate the align-self property:
  - <div class="flex-container">  
    <div>1</div>  
    <div>2</div>  
    <div style="align-self: center">3</div>  
    <div>4</div>  
  </div>