



Flexbox

Layout and Structure



Definition

The Flexible Box Module, usually referred to as flexbox, was designed as a one-dimensional layout model, and as a method that could offer space distribution between items in an interface and powerful alignment capabilities.

When we describe flexbox as being one dimensional we are describing the fact that flexbox deals with layout in one dimension at a time — either as a row or as a column.



Direction and wrap

flex-direction is a property that sets how items in the container are placed and defines the axis and the direction:

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

flex-wrap decides what happens when space runs out:

- **nowrap** forces the items to stay in one line;
- **wrap** allows them to wrap into multiple lines;
- **wrap-reverse** wraps them in the opposite direction;



Justify and align

The CSS `justify-content` property defines how the browser distributes space between and around content items along the main-axis of a flex container:

- `justify-content: start;`
- `justify-content: end;`
- `justify-content: center;`

The CSS `align-items` in Flexbox, controls the alignment of items on the Cross Axis:

- `align-items: start;`
- `align-items: end;`
- `align-items: center;`



Grow, shrink, basis

The elements can grow or shrink depending on the size of the container:

- **flex-grow** is used to determine if an element takes up additional space;
- **flex-shrink** is the same as flex-grow but determines how much an element can shrink;
- **flex-basis** sets the default size of a node along the main axis;
- **flex** is the abbreviation used to grow, shrink, etc;



Example 1

Home

News

Gallery

Login

HTML

```
<nav>
  <ul>
    <li>Home</li>
    <li>News</li>
    <li>Gallery</li>
    <li>Login</li>
  </ul>
</nav>
```

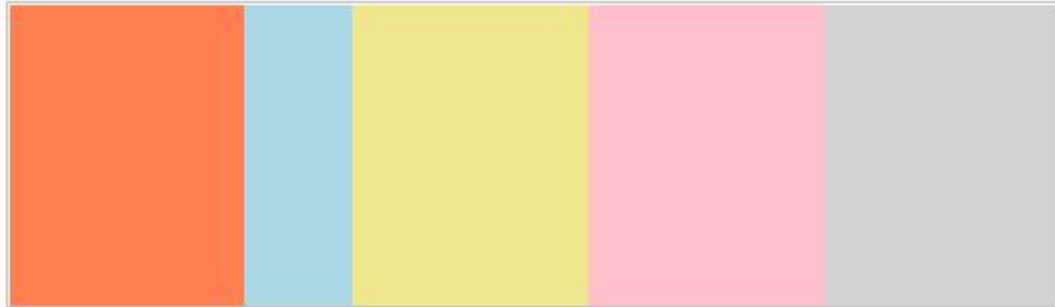
CSS

```
nav {
  background-color: black;
  color: white;
  height: 80px;
  display: flex;
  align-items: center;
}
```

```
nav ul {
  width: 100%;
  list-style-type: none;
  display: flex;
  flex-direction: row;
  justify-content: space-around;
  align-items: center;
}
```



Example 2





Example 2

HTML

```
<div id="main">
  <div style="background-color:coral;"></div>
  <div style="background-color:lightblue;"></div>
  <div style="background-color:khaki;"></div>
  <div style="background-color:pink;"></div>
  <div style="background-color:lightgrey;"></div>
</div>
```

CSS

```
#main {
  width: 350px;
  height: 100px;
  border: 1px solid #c3c3c3;
  display: flex;
}

#main div:nth-of-type(2) {
  flex-shrink: 3;
}
```




Example 3

HTML

```
<ul>
  <li class="flex-item">1</li>
  <li class="flex-item">2</li>
  <li class="flex-item">3</li>
  <li class="flex-item">4</li>
</ul>
```

CSS

```
ul {
  display: flex;
  list-style-type: none;
  flex-direction: row;
}

li:nth-of-type(1) { order: 2; }
li:nth-of-type(3) { order: 4; }
li:nth-of-type(4) { order: -1; }
```

—————→ 4213



Example 4





Example 4

HTML

```
<main>
  <div></div>
  <div></div>
  <div></div>
  <div></div>
  <div></div>
</main>
```

CSS

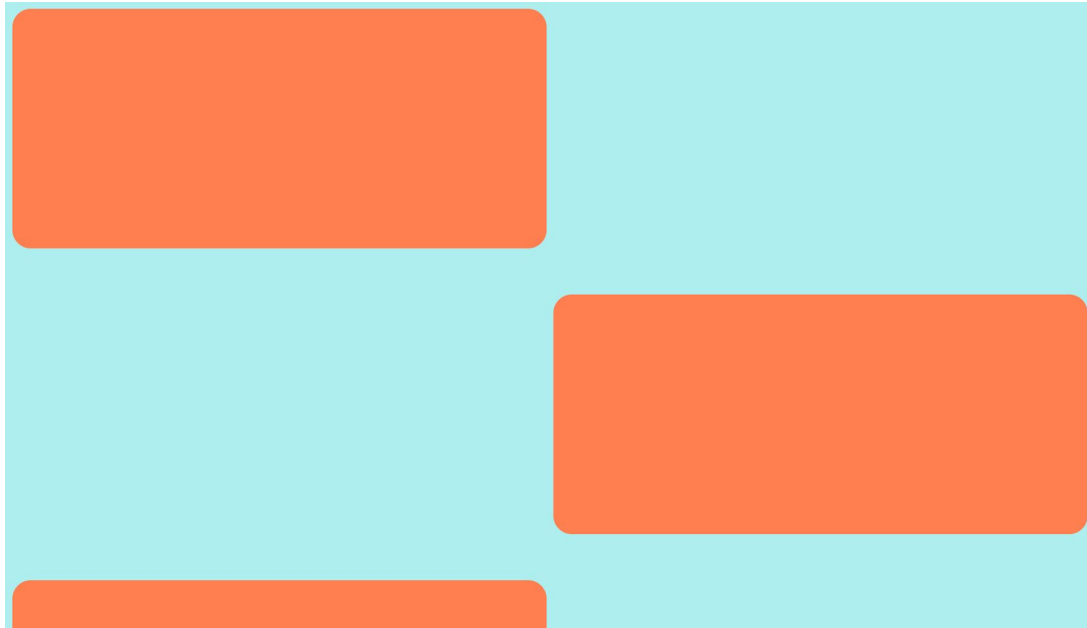
```
main {
  display: flex;
  flex-direction: row;
  justify-content: space-around;
  gap: 20px;
  height: 420px;
}

main div:nth-of-type(3) {
  flex-grow: 2;
  transition: flex-grow 2s;
}
```

```
main div {
  height: 200px;
  background-color: coral;
  border-radius: 20px;
  flex-grow: 1;
  flex-shrink: 1;
}
```



Example 5





Example 5

HTML

```
<main>
  <div></div>
  <div></div>
  <div></div>
  <div></div>
  <div></div>
</main>
```

CSS

```
main {
  display: flex;
  flex-direction: column;
  justify-content: space-around;
  gap: 50px;
  height: 1500px;
  width: 100%;
}

main div:nth-of-type(even) {
  align-self: flex-end;
}
```

```
main div {
  width: 580px;
  background-color: coral;
  border-radius: 20px;
  flex-grow: 1;
  flex-shrink: 1;
}
```



The end

Made by Adrian, Emanuele, Martina, Samuele