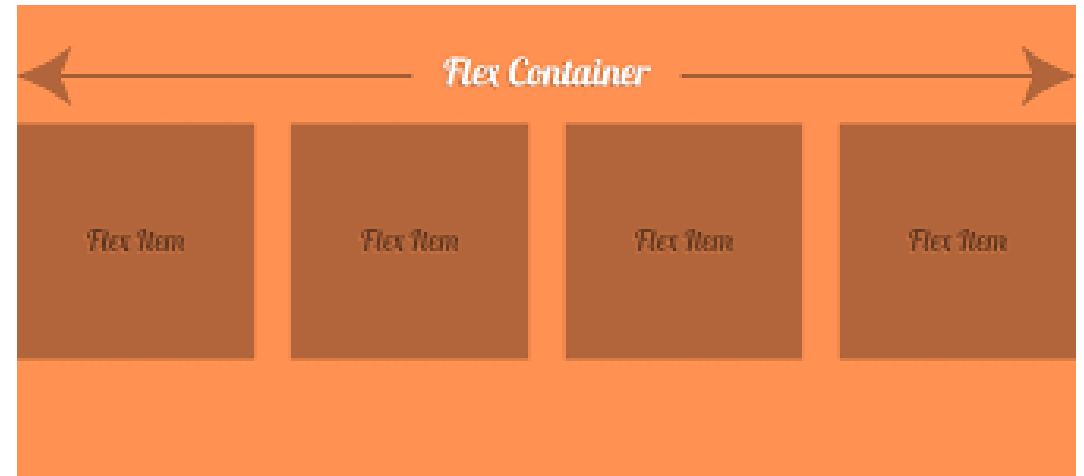


Layout with Flexbox

- **Flexbox** is a layout mode in CSS that provides a **flexible way to distribute space and align content within a container.**
- Used to make the **elements behave predictably when they are used with different screen sizes and different display devices.**

The CSS3 flexbox contains **flex containers** and **flex items**.

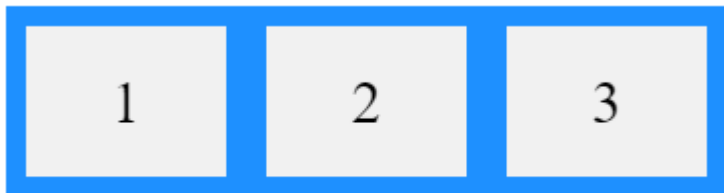
- **Flex container:** specifies the properties of the parent.
 - **display: flex or inline-flex.**
- **Flex items:** specify properties of the children.
 - There may be **one or more flex items inside a flex container.**



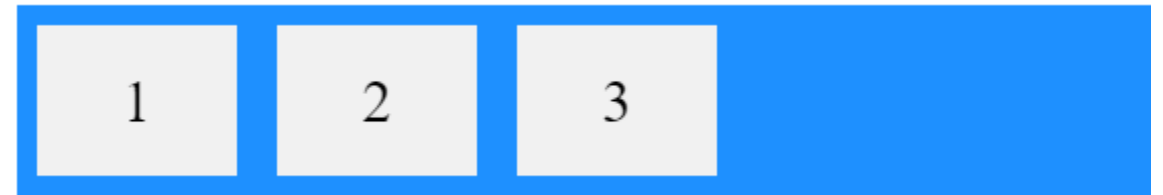
Creating flex display: flex /inline-flex.

- A flex box can be created as.
display: **flex** and display: **inline-flex**
- When you apply **display: flex** or **display: inline-flex** to an element, that element becomes a flex container, and its child elements become flex items.
- **display: flex** creates a **block-level** flex container.
- **display: inline-flex** creates an **inline-level** flex container

```
.flex-container {  
  display: inline-flex;  
  background-color: DodgerBlue;  
}
```



```
.flex-container {  
  display: flex;  
  background-color: DodgerBlue;  
}
```



Layout with Flexbox- creating a flex

```
<html>
<head>
<style>
.flex-container {
  display: flex;
  width: 400px;
  height: 150px;
  background-color: lightpink;
}
.flex-item {
  background-color: gray;
  width: 100px;
  height: 80px;
  margin: 10px;
  padding: 20px;
  font-size: 20pt;
}
</style>
</head>
```

```
<body>
<div class="flex-container">
  <div class="flex-item">Flex Item 1</div>
  <div class="flex-item">Flex Item 2</div>
  <div class="flex-item">Flex Item 3</div>
</div>
</body>
</html>
```



CSS flex-direction Property -

- The **flex-direction** property specifies the **direction of the flexible items.**

CSS Syntax:

flex-direction: row

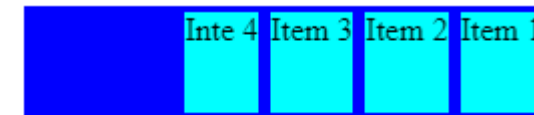
Property Values:

Value	Description
row	Default value. The flexible items are displayed horizontally , as a row
column	The flexible items are displayed vertically, as a column
column-reverse	Same as column, but in reverse order
row-reverse	Same as row, but in reverse order

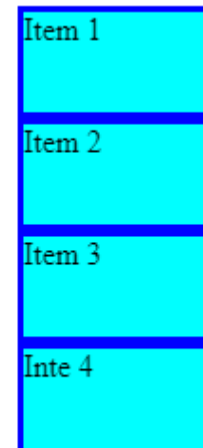
The flex-direction property



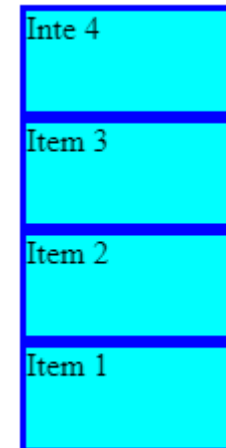
flex-direction:row;



flex-direction:row-reverse;



flex-direction:column;



flex-direction:column-revrse;

CSS flex-direction Property -row

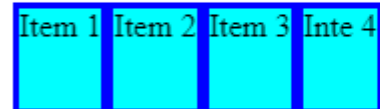
- The **flex-direction** property specifies the **direction of the flexible items**.

```
<html>
<head>
<style>
.container{
  display: flex;
  flex-direction: row;
  background-color: blue;
}
.item{
  background-color: aqua;
  margin: 3px;
  height: 50px;
}
</style>
</head>
<body>
<h1>The flex-direction property</h1>
```

```
<div class="container">
  <div class="item">Item 1</div>
  <div class="item">Item 2</div>
  <div class="item">Item 3 </div>
  <div class="item">Inte 4</div>
</div>

</body>
</html>
```

The flex-direction property



CSS flex-direction Property column

- The **flex-direction** property specifies the **direction of the flexible items**.

```
<html>
<head>
<style>
.container{
  display: flex;
  flex-direction: column;
  background-color: blue;
}
.item{
  background-color: aqua;
  margin: 3px;
  height: 50px;
}
</style>
</head>
<body>
<h1>The flex-direction property</h1>
```

```
<div class="container">
  <div class="item">Item 1</div>
  <div class="item">Item 2</div>
  <div class="item">Item 3 </div>
  <div class="item">Inte 4</div>
</div>

</body>
</html>
```

The flex-direction property



display: justify-content.

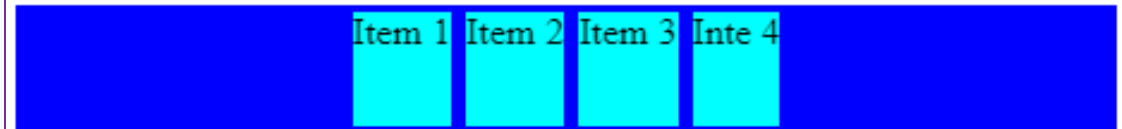
- **justify-content** is a CSS property that is used to **align and distribute flex items along the main axis of the flex container**.

```
<html>
<head>
<style>
.container {
  display: flex;
  justify-content: center;
  background-color: dodgerblue;
}
.item{
  background-color: aqua;
  margin: 3px;
  height: 50px;
}
```

```
</style>
</head>
<body>

  <div class="container">
    <div class="flex-item">Item 1</div>
    <div class="flex-item">Item 2</div>
    <div class="flex-item">Item 3</div>
  </div>
</body>
</html>
```

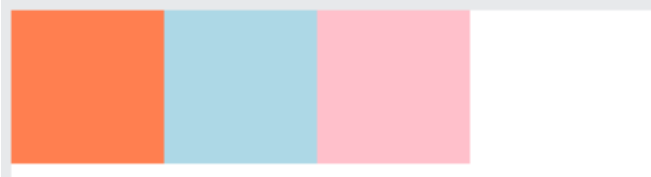
justify-content: center;



display: justify-content.

Property Values

Value	Description
flex-start	Default value. Items are positioned at the beginning of the container
flex-end	Items are positioned at the end of the container
center	Items are positioned in the center of the container
space-between	Items will have space between them
space-around	Items will have space before, between, and after them
space-evenly	Items will have equal space around them



flex-start



flex-end



center

CSS flex-grow Property

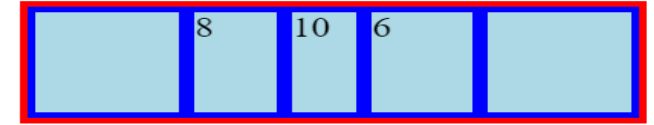
- The **flex-grow property** specifies **how much the item will grow** relative to the rest of the flexible items inside the same container.

```
<html>
<head>
<style>
.container {
display: flex;
background-color: green;
}
.flex-item {
background-color: lightgreen;
text-align: center;
font-size: 25px;
width: 100px;
height: 100px;
padding-top: 20px;
margin: 5px;
}
</style> </head>
```

```
<body>
<h1> flex-grow: 0; </h1>
<div class="container">
<div class="flex-item" style = "flex-grow: 0;">
flex-item 1 </div>
<div class="flex-item" style = "flex-grow: 0;">
flex-item 2 </div>
<div class="flex-item" style = "flex-grow: 0;">
flex-item 3 </div>
</div>
<h1> flex-grow: 1; </h1>
<div class="container">
<div class="flex-item" style = "flex-grow: 1;">
flex-item 1 </div>
<div class="flex-item" style = "flex-grow: 1;">
flex-item 2 </div>
<div class="flex-item" style = "flex-grow: 1;">
flex-item 3 </div>
</div>
</div>
</body>
</html>
```

CSS flex-shrink Property

Example of the flex-shrink property



- The **flex-shrink** property specifies **how the item will shrink** relative to the rest of the flexible items inside the same container.

```
<html>
<head>
<style>
.container {
width: 400px;
height: 100px;
border: 5px solid red;
display: flex;
background-color: blue;
margin: 30px;
}
.flex-item{
background-color: lightblue;
font-size: 25px;
margin: 5px;
flex-grow: 1;
flex-shrink: 1;
flex-basis: 100px;
}
```

```
</style>
</head>
<body>
<h1> Example of the flex-shrink property </h1>
<div class="container">
<div class = "flex-item"></div>
<div class = "flex-item"></div>
<div class = "flex-item" style = "flex-shrink: 5;"> 5 </div>
<div class = "flex-item" style = "flex-shrink: initial;"> initial
</div>
<div class = "flex-item" style = "flex-shrink: inherit;"> inherit
</div>
</div>
<div class="container">
<div class = "flex-item"></div>
<div class = "flex-item" style = "flex-shrink: 8;"> 8 </div>
<div class = "flex-item" style = "flex-shrink: 10;"> 10 </div>
<div class = "flex-item" style = "flex-shrink: 6;"> 6 </div>
<div class = "flex-item"></div>
</div></body></html>
```

CSS flex-basis Property

The flex-basis property specifies the initial length of a flexible item.

```
<!DOCTYPE html>
<html>
<head>
<style>
.container {
display: flex;
background-color: lightblue;
}
.flex-item {
background-color: white;
text-align: center;
line-height: 40px;
font-size: 25px;
margin: 5px;
}
</style>
</head>
```

```
<body>
<h1> Example of the flex-basis property </h1>
<div class = "container">
<div class = "flex-item" style = "flex-basis: auto;"> auto
</div>
<div class = "flex-item" style = "flex-basis: initial;">
initial </div>
<div class = "flex-item" style = "flex-basis: inherit;">
inherit </div>
<div class = "flex-item" style = "flex-basis: 150px;">
150px </div>
<div class = "flex-item" style = "flex-basis: auto"> auto
</div>
</div>
</body>
</html>
```

Example of the flex-basis property

auto	initial	inherit	150px	auto
------	---------	---------	-------	------

CSS flex-wrap Property

- The **flex-wrap** property specifies whether the flexible **items should wrap or not**.

```
<!DOCTYPE html>
<html>
<head>
<style>
#main {
  width: 200px;
  height: 200px;
  border: 1px solid #c3c3c3;
  display: flex;
  flex-wrap: wrap;
}
```

```
#main div {
  width: 50px;
  height: 50px;
}
</style>
</head>
<body>
```

```
<h1>The flex-wrap Property</h1>
```

```
<div id="main">
  <div style="background-color:coral;">A</div>
  <div style="background-color:lightblue;">B</div>
  <div style="background-color:khaki;">C</div>
  <div style="background-color:pink;">D</div>
  <div style="background-color:lightgrey;">E</div>
  <div style="background-color:lightgreen;">F</div>
</div>

</body>
</html>
```

The flex-wrap Property



CSS flex-direction Property

The **flex-direction** property specifies the direction of the flexible items.

```
<html>
<head>
<style>
#main {
  width: 400px;
  height: 400px;
  border: 1px solid #c3c3c3;
  display: flex;
  flex-direction: row-reverse;
}
```

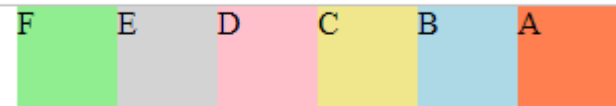
```
#main div {
  width: 50px;
  height: 50px;
}
</style>
</head>
<body>
```

```
<h1>The flex-direction Property</h1>
```

```
<div id="main">
  <div style="background-color:coral;">A</div>
  <div style="background-color:lightblue;">B</div>
  <div style="background-color:khaki;">C</div>
  <div style="background-color:pink;">D</div>
  <div style="background-color:lightgrey;">E</div>
  <div style="background-color:lightgreen;">F</div>
</div>

</body>
</html>
```

The flex-direction Property



CSS flex-flow Property

This CSS property is shorthand for **flex-direction** and **flex-wrap** properties.

```
<!DOCTYPE html>
<html>
<head>
<style>
#main {
  width: 200px;
  height: 200px;
  border: 1px solid #c3c3c3;
  display: flex;
  flex-flow: row-reverse wrap;
}
```

```
#main div {
  width: 50px;
  height: 50px;
}
</style>
</head>
<body>
```

```
<h1>The flex-flow Property</h1>
```

```
<div id="main">
  <div style="background-color:coral;">A</div>
  <div style="background-color:lightblue;">B</div>
  <div style="background-color:khaki;">C</div>
  <div style="background-color:pink;">D</div>
  <div style="background-color:lightgrey;">E</div>
  <div style="background-color:lightgreen;">F</div>
</div>
</body>
</html>
```

The flex-flow Property



Nested Flexboxes

- Flex items within a flex container can be laid out either **horizontally or vertically**, but not both.
- If you want to lay out items in both dimensions, you'll need to **nest a flex container inside another one.**

```
<!doctype html>
<title>Example</title>
<style>
.container {
  display: flex;
}
.red {
  background: orangered;
  display: flex;
  flex-direction: column;
}
.green {
  background: yellowgreen;
}
.blue {
  background: steelblue;
}
```

```
.container div {
  font-size: 5vw;
  padding: .5em;
  color: white;
  flex: 1;
}
</style>
<div class="container">
  <div class="red">1
    <div class="green">1a</div>
    <div class="blue">1b</div>
  </div>
  <div class="green">2</div>
  <div class="blue">3</div>
</div>
```

