CSS Flexbox Layout Module

Before the Flexbox Layout module, there were four layout modes:

- Block, for sections in a webpage
- Inline, for text
- Table, for two-dimensional table data
- Positioned, for explicit position of an element

The Flexible Box Layout Module, makes it easier to design flexible responsive layout structure without using float or positioning.

CSS Flex Container

Parent Element (Container)

Like we specified in the previous chapter, this is a flex **container** (the blue area) with three flex **items**:

The flex container becomes flexible by setting the `display` property to `flex`:

Example :-

```
<!DOCTYPE html>

<html>

<head>

<style>

.flex-container {

  display: flex;

  background-color: DodgerBlue;
```

```
}

.flex-container > div {
  background-color: #f1f1f1;
  margin: 10px;
  padding: 20px;
  font-size: 30px;
}
</style>
</head>
<body>

<h1>Create a Flex Container</h1>

<div class="flex-container">
  <div>1</div>
  <div>2</div>
  <div>3</div>
</div>

<p>A Flexible Layout must have a parent element with the <em>display</em> property set to <em>flex</em>.</p>

<p>Direct child elements(s) of the flexible container automatically becomes flexible items.</p>

</body>
```

</html>

The flex container properties are:

- `flex-direction`
- `flex-wrap`
- `flex-flow`
- `justify-content`
- `align-items`
- `align-content`

The flex-direction Property

The `flex-direction` property defines in which direction the container wants to stack the flex items.

Example

The `column` value stacks the flex items vertically (from top to bottom):

<!DOCTYPE html>

<html>

<head>

<style>

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

.flex-container > div {
  background-color: #f1f1f1;
  width: 100px;
  margin: 10px;
  text-align: center;
  line-height: 75px;
  font-size: 30px;
}
</style>
</head>
<body>

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

<p>The "flex-direction: column;" stacks the flex items vertically (from top to bottom):</p>

<div class="flex-container">
```

```
  <div>1</div>

  <div>2</div>

  <div>3</div>

</div>


</body>

</html>
```

Example

The `column-reverse` value stacks the flex items vertically (but from bottom to top):


```
<!DOCTYPE html>

<html>

<head>

<style>

.flex-container {

  display: flex;
```

```
  flex-direction: column-reverse;
  background-color: DodgerBlue;
}

.flex-container > div {
  background-color: #f1f1f1;
  width: 100px;
  margin: 10px;
  text-align: center;
  line-height: 75px;
  font-size: 30px;
}
</style>
</head>
<body>

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

<p>The "flex-direction: column-reverse;" stacks the flex items vertically (but from bottom to top):</p>

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

```
</body>

</html>
```

Example

The `row` value stacks the flex items horizontally (from left to right):


```
<!DOCTYPE html>

<html>

<head>

<style>

.flex-container {

  display: flex;

  flex-direction: row;

  background-color: DodgerBlue;

}


.flex-container > div {

  background-color: #f1f1f1;

  width: 100px;

  margin: 10px;

  text-align: center;

  line-height: 75px;
```

```
  font-size: 30px;
}
</style>
</head>
<body>

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

<p>The "flex-direction: row;" stacks the flex items horizontally (from left to right):</p>

<div class="flex-container">
  <div>1</div>
  <div>2</div>
  <div>3</div>
</div>

</body>
</html>
```

Example

The `row-reverse` value stacks the flex items horizontally (but from right to left):

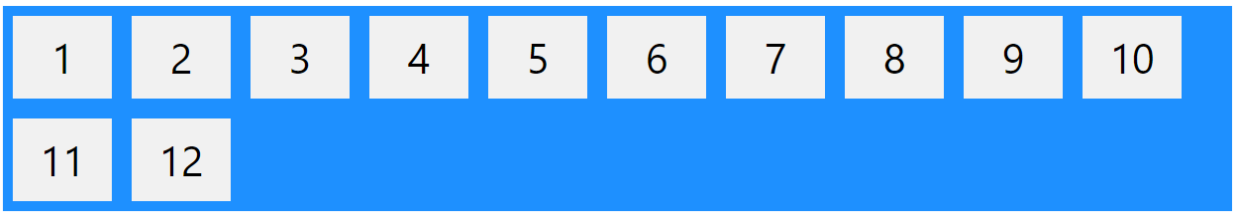

```
<!DOCTYPE html>
<html>
<head>
<style>
.flex-container {
  display: flex;
  flex-direction: row-reverse;
  background-color: DodgerBlue;
}

.flex-container > div {
  background-color: #f1f1f1;
  width: 100px;
  margin: 10px;
  text-align: center;
  line-height: 75px;
  font-size: 30px;
}
</style>
</head>
```

```
<body>

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

<p>The "flex-direction: row-reverse;" stacks the flex items horizontally (but from right to left):</p>

<div class="flex-container">
  <div>1</div>
  <div>2</div>
  <div>3</div>
</div>

</body>
</html>
```

The flex-wrap Property

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

The examples below have 12 flex items, to better demonstrate the `flex-wrap` property.



Example

The `wrap` value specifies that the flex items will wrap if necessary:

```
<!DOCTYPE html>

<html>

<head>

<style>

.flex-container {

  display: flex;

  flex-wrap: wrap;

  background-color: DodgerBlue;

}


.flex-container > div {

  background-color: #f1f1f1;

  width: 100px;

  margin: 10px;

  text-align: center;

  line-height: 75px;

  font-size: 30px;

}

</style>

</head>

<body>


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

```
<p>The "flex-wrap: wrap;" specifies that the flex items will wrap if necessary:</p>

<div class="flex-container">
  <div>1</div>
  <div>2</div>
  <div>3</div>
  <div>4</div>
  <div>5</div>
  <div>6</div>
  <div>7</div>
  <div>8</div>
  <div>9</div>
  <div>10</div>
  <div>11</div>
  <div>12</div>
</div>

<p>Try resizing the browser window.</p>

</body>
</html>
```

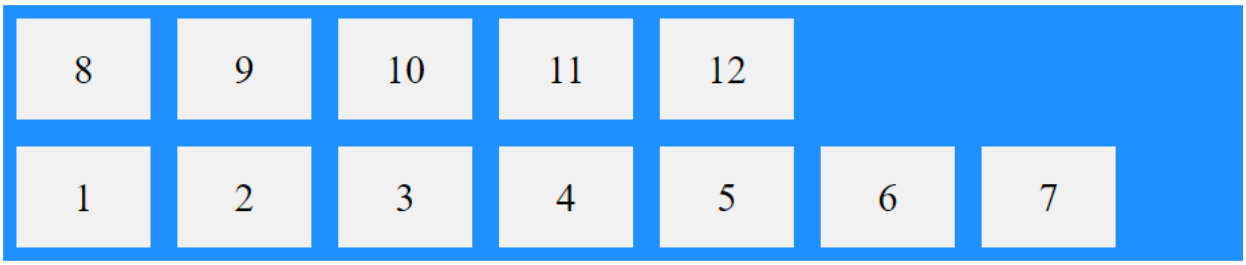
Example

The `nowrap` value specifies that the flex items will not wrap (this is default):

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

Example

The `wrap-reverse` value specifies that the flexible items will wrap if necessary, in reverse order:



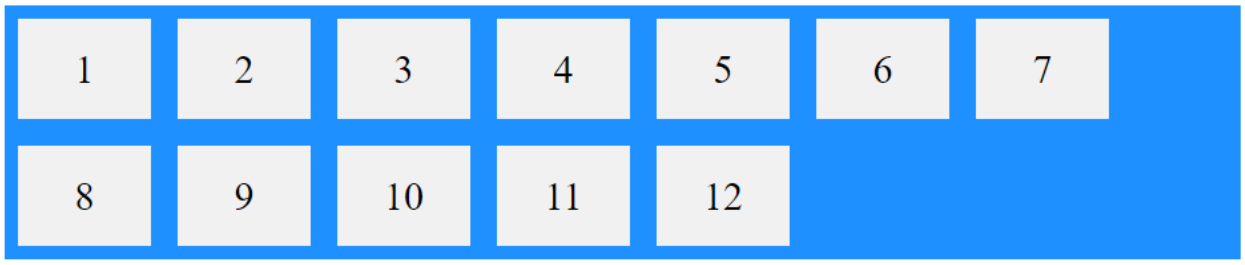

```
<!DOCTYPE html>

<html>

<head>

<style>

.flex-container {

 display: flex;

 flex-wrap: wrap-reverse;

 background-color: DodgerBlue;

}
```

```
.flex-container > div {
  background-color: #f1f1f1;
  width: 100px;
  margin: 10px;
  text-align: center;
  line-height: 75px;
  font-size: 30px;
}
</style>
</head>
<body>

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

<p>The "flex-wrap: wrap-reverse;" specifies that the flex items will wrap if necessary, in reverse order:</p>

<div class="flex-container">
  <div>1</div>
  <div>2</div>
  <div>3</div>
  <div>4</div>
  <div>5</div>
  <div>6</div>
  <div>7</div>
  <div>8</div>
```

```
  <div>9</div>

  <div>10</div>

  <div>11</div>

  <div>12</div>

</div>


<p>Try resizing the browser window.</p>


</body>

</html>
```

The flex-flow Property

The `flex-flow` property is a shorthand property for setting both the `flex-direction` and `flex-wrap` properties.



Example

```
<!DOCTYPE html>

<html>
```

```
<head>
<style>
.flex-container {
  display: flex;
  flex-flow: row wrap;
  background-color: DodgerBlue;
}

.flex-container > div {
  background-color: #f1f1f1;
  width: 100px;
  margin: 10px;
  text-align: center;
  line-height: 75px;
  font-size: 30px;
}
</style>
</head>
<body>
<h1>The flex-flow Property</h1>

<p>The flex-flow property is a shorthand property for the flex-direction and the flex-wrap properties.</p>

<div class="flex-container">
  <div>1</div>
```

```
  <div>2</div>
  <div>3</div>
  <div>4</div>
  <div>5</div>
  <div>6</div>
  <div>7</div>
  <div>8</div>
  <div>9</div>
  <div>10</div>
  <div>11</div>
  <div>12</div>
</div>

<p>Try resizing the browser window.</p>

</body>
</html>
```

The justify-content Property

The `justify-content` property is used to align the flex items:

Example

The `center` value aligns the flex items at the center of the container:

```
<!DOCTYPE html>
<html>
<head>
<style>
.flex-container {
  display: flex;
  justify-content: center;
  background-color: DodgerBlue;
}

.flex-container > div {
  background-color: #f1f1f1;
  width: 100px;
  margin: 10px;
  text-align: center;
  line-height: 75px;
  font-size: 30px;
}
</style>
</head>
<body>

<h1>The justify-content Property</h1>
```

```
<p>The "justify-content: center;" aligns the flex items at the center of the container:</p>


<div class="flex-container">
  <div>1</div>
  <div>2</div>
  <div>3</div>
</div>


</body>
</html>
```

Example

The `flex-start` value aligns the flex items at the beginning of the container (this is default):


```
<!DOCTYPE html>
<html>
<head>
<style>
.flex-container {
  display: flex;
  justify-content: flex-start;
```

```
  background-color: DodgerBlue;
}

.flex-container > div {
  background-color: #f1f1f1;
  width: 100px;
  margin: 10px;
  text-align: center;
  line-height: 75px;
  font-size: 30px;
}
</style>
</head>
<body>

<h1>The justify-content Property</h1>

<p>The "justify-content: flex-start;" aligns the flex items at the beginning of the container (this is default):</p>

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

```
</body>

</html>
```

Example

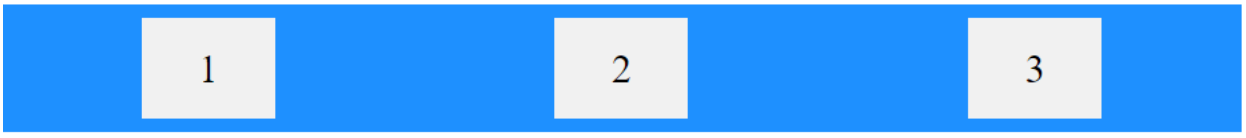
The `flex-end` value aligns the flex items at the end of the container:

```
.flex-container {
  display: flex;
  justify-content: flex-end;
}
```

Example

The `space-around` value displays the flex items with space before, between, and after the lines:

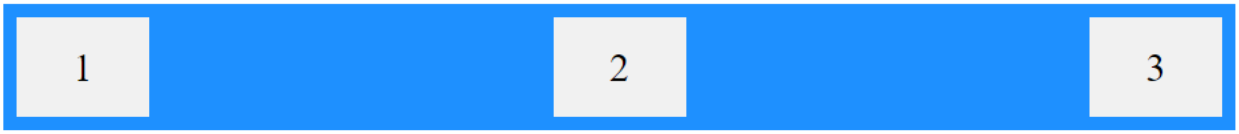
```
.flex-container {
  display: flex;
  justify-content: space-around;
}
```



Example

The `space-between` value displays the flex items with space between the lines:

```
.flex-container {
  display: flex;
  justify-content: space-between;
}
```



The align-items Property

The `align-items` property is used to align the flex items.

In these examples we use a 200 pixels high container, to better demonstrate the `align-items` property.

Example

The `center` value aligns the flex items in the middle of the container:

```
<!DOCTYPE html>

<html>

<head>

<style>

.flex-container {
```

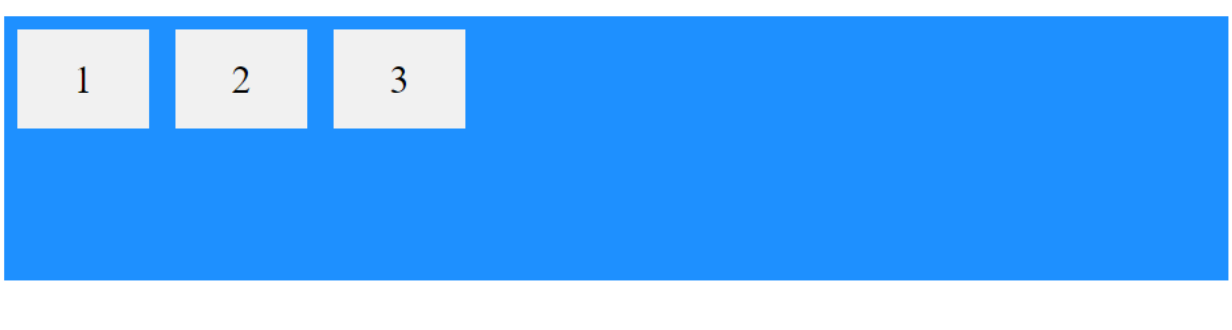
```
  display: flex;
  height: 200px;
  align-items: center;
  background-color: DodgerBlue;
}

.flex-container > div {
  background-color: #f1f1f1;
  width: 100px;
  margin: 10px;
  text-align: center;
  line-height: 75px;
  font-size: 30px;
}
</style>
</head>
<body>

<h1>The align-items Property</h1>

<p>The "align-items: center;" aligns the flex items in the middle of the container:</p>

<div class="flex-container">
  <div>1</div>
  <div>2</div>
```

```
 <div>3</div>

</div>


</body>

</html>
```

Example

The `flex-start` value aligns the flex items at the top of the container:

```
.flex-container {
  display: flex;
  height: 200px;
  align-items: flex-start;
}
```



Example

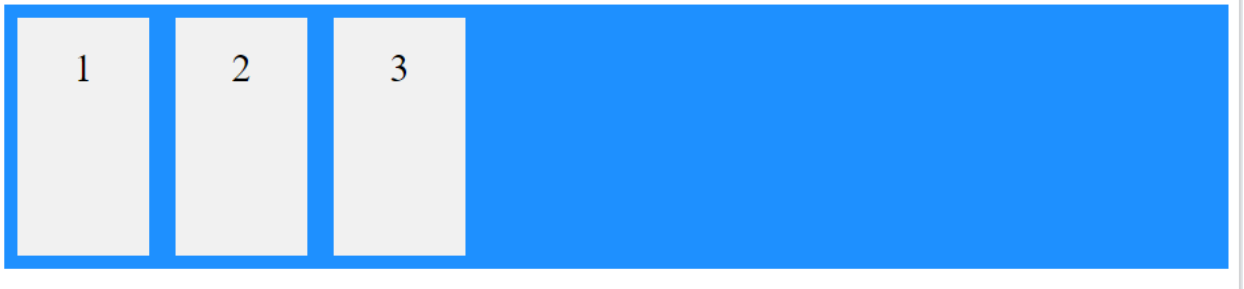
The `flex-end` value aligns the flex items at the bottom of the container:

```
.flex-container {
  display: flex;
  height: 200px;
  align-items: flex-end;
}
```

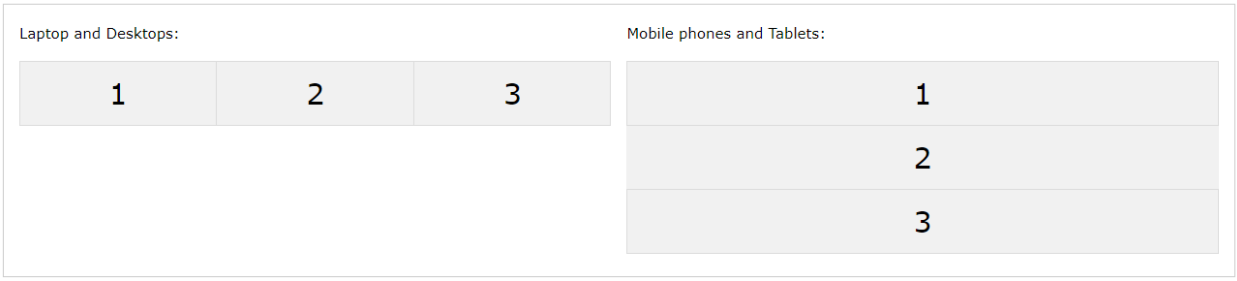
Example

The `stretch` value stretches the flex items to fill the container (this is default):

```
.flex-container {
  display: flex;
  height: 200px;
  align-items: stretch;
}
```



CSS Flex Responsive



For example, if you want to create a two-column layout for most screen sizes, and a one-column layout for small screen sizes (such as phones and tablets), you can change the `flex-direction` from `row` to `column` at a specific breakpoint (800px in the example below):

Example

```
.flex-container {
  display: flex;
  flex-direction: row;
}

/* Responsive layout - makes a one column layout instead of a two-column
layout */
@media (max-width: 800px) {
  .flex-container {
    flex-direction: column;
  }
}
```

```
<!DOCTYPE html>

<html>

<head>

<style>

* {
```

```
  box-sizing: border-box;
}

.flex-container {
  display: flex;
  flex-direction: row;
  font-size: 30px;
  text-align: center;
}

.flex-item-left {
  background-color: #f1f1f1;
  padding: 10px;
  flex: 50%;
}

.flex-item-right {
  background-color: dodgerblue;
  padding: 10px;
  flex: 50%;
}

/* Responsive layout - makes a one column-layout instead of two-column layout */
@media (max-width: 800px) {
  .flex-container {
```

```
   flex-direction: column;
  }
}
</style>
</head>
<body>

<h1>Responsive Flexbox</h1>

<p>The "flex-direction: row;" stacks the flex items horizontally (from left to right).</p>
<p>The "flex-direction: column;" stacks the flex items vertically (from top to bottom).</p>
<p><b>Resize the browser window to see that the direction changes when the
screen size is 800px wide or smaller.</b></p>

<div class="flex-container">
  <div class="flex-item-left">1</div>
  <div class="flex-item-right">2</div>
</div>

</body>
</html>
```

Another way is to change the percentage of the `flex` property of the flex items to create different layouts for different screen sizes. Note that we also have to include `flex-wrap: wrap;` on the flex container for this example to work:

Example

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

.flex-item-left {
  flex: 50%;
}

.flex-item-right {
  flex: 50%;
}

/* Responsive layout - makes a one column layout instead of a two-column
layout */
@media (max-width: 800px) {
  .flex-item-right, .flex-item-left {
    flex: 100%;
  }
}
```

```
<!DOCTYPE html>

<html>

<head>

<style>

* {

 box-sizing: border-box;

}



.flex-container {

 display: flex;

 flex-wrap: wrap;
```

```
  font-size: 30px;
  text-align: center;
}

.flex-item-left {
  background-color: #f1f1f1;
  padding: 10px;
  flex: 50%;
}

.flex-item-right {
  background-color: dodgerblue;
  padding: 10px;
  flex: 50%;
}

/* Responsive layout - makes a one column-layout instead of a two-column layout */
@media (max-width: 800px) {
  .flex-item-right, .flex-item-left {
    flex: 100%;
  }
}
</style>
</head>
<body>
```

```
<h1>Responsive Flexbox</h1>


<p>In this example, we change the percentage of flex to create different layouts for different screen sizes.</p>

<p><b>Resize the browser window to see that the direction changes when the

screen size is 800px wide or smaller.</b></p>


<div class="flex-container">

  <div class="flex-item-left">1</div>

  <div class="flex-item-right">2</div>

</div>


</body>

</html>
```