

CSS Flexbox Container

 [w3schools.com/css/css3_flexbox_container.asp](https://www.w3schools.com/css/css3_flexbox_container.asp)



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**:

1

2

3

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

Example

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

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):

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

Example

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

```
.flex-container {  
  display: flex;  
  flex-direction: column-reverse;  
}
```

Example

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

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

Example

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

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

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:

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

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:

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

The flex-flow Property

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

Example

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

The justify-content Property

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

1

2

3

Example

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

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

Example

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

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

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:

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

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;  
}
```

Example

The **baseline** value aligns the flex items such as their baselines aligns:

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

Note: the example uses different font-size to demonstrate that the items gets aligned by the text baseline:

The align-content Property

The **align-content** property is used to align the flex lines.

In these examples we use a 600 pixels high container, with the **flex-wrap** property set to **wrap**, to better demonstrate the **align-content** property.

Example

The **space-between** value displays the flex lines with equal space between them:

```
.flex-container {  
  display: flex;  
  height: 600px;  
  flex-wrap: wrap;  
  align-content: space-between;  
}
```

[Try it Yourself »](#)

Example

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

```
.flex-container {  
  display: flex;  
  height: 600px;  
  flex-wrap: wrap;  
  align-content: space-around;  
}
```

Example

The **stretch** value stretches the flex lines to take up the remaining space (this is default):

```
.flex-container {  
  display: flex;  
  height: 600px;  
  flex-wrap: wrap;  
  align-content: stretch;  
}
```

Example

The **center** value displays the flex lines in the middle of the container:

```
.flex-container {  
  display: flex;  
  height: 600px;  
  flex-wrap: wrap;  
  align-content: center;  
}
```

Example

The **flex-start** value displays the flex lines at the start of the container:

```
.flex-container {  
  display: flex;  
  height: 600px;  
  flex-wrap: wrap;  
  align-content: flex-start;  
}
```

Example

The **flex-end** value displays the flex lines at the end of the container:

```
.flex-container {  
  display: flex;  
  height: 600px;
```

```
flex-wrap: wrap;
align-content: flex-end;
}
```

Perfect Centering

In the following example we will solve a very common style problem: perfect centering.

SOLUTION: Set both the `justify-content` and `align-items` properties to `center`, and the flex item will be perfectly centered:

Example

```
.flex-container {
  display: flex;
  height: 300px;
  justify-content: center;
  align-items: center;
}
```

The CSS Flexbox Container Properties

The following table lists all the CSS Flexbox Container properties:

Property	Description
<u>align-content</u>	Modifies the behavior of the flex-wrap property. It is similar to align-items, but instead of aligning flex items, it aligns flex lines
<u>align-items</u>	Vertically aligns the flex items when the items do not use all available space on the cross-axis
<u>display</u>	Specifies the type of box used for an HTML element
<u>flex-direction</u>	Specifies the direction of the flexible items inside a flex container
<u>flex-flow</u>	A shorthand property for flex-direction and flex-wrap
<u>flex-wrap</u>	Specifies whether the flex items should wrap or not, if there is not enough room for them on one flex line
<u>justify-content</u>	Horizontally aligns the flex items when the items do not use all available space on the main-axis

