
Session 8

Mohamed Emary

March 31, 2024

1 Flex Display

`display: flex` is a CSS property that makes the element a flex container. The flex container is the parent element that contains the flex items.

1.1 Flex Direction

`flex-direction` property specifies the direction of the flexible items. It can be set to:

- `row` (*default*)
- `row-reverse`
- `column`
- `column-reverse`

The `row` will be from left to right if the page direction is `ltr` (left to right) and from right to left if the page direction is `rtl` (right to left).

`row-reverse` will be from right to left if the page direction is `ltr` and from left to right if the page direction is `rtl`.

The same applies to `column` and `column-reverse`.

1.2 Flex Wrap

`flex-wrap` property specifies whether the flexible items should wrap or not. It can be set to:

- `nowrap` (*default*)
- `wrap`
- `wrap-reverse`

1.3 Flex Flow

`flex-flow` property is a shorthand property for the `flex-direction` and `flex-wrap` properties. It takes two values: `flex-direction` and `flex-wrap`, so it is written as `flex-flow:`

`flex-direction flex-wrap`.

1.4 Order

`order` property specifies the order of the flexible items. It takes a number as a value. The default value is 0. The items are ordered based on the value of the `order` property. The items with the lower values will be placed before the items with the higher values.

1.5 Placing flex items

`justify-content` property specifies how the flexible items are placed in the flex container. It can be set to :

- `flex-start`
- `flex-end`
- `center`
- `space-between`
- `space-around`
- `space-evenly`

`align-items` property specifies how the flexible items are placed in the flex container. It can be set to:

- `stretch` (*default*)
- `flex-start`
- `flex-end`
- `center`
- `baseline` - aligns the items based on their baselines which is the line that the letters sit on

`justify-content` property works on the main axis, while `align-items` property works on the cross axis.

The main axis is the axis defined by the `flex-direction` property, while the cross axis is the axis perpendicular to the main axis.

`align-content` property specifies how the flexible items are placed in the flex container **when there is extra space in the cross axis**. It can be set to:

- `stretch` (*default*)
- `flex-start`
- `flex-end`
- `center`
- `space-between`
- `space-around`

`align-content` property only works if the flex items are wrapped.

1.6 Row & Column Gaps

`row-gap` property specifies the space between the rows of a grid layout. `column-gap` property specifies the space between the columns of a grid layout.

The `gap` property is a shorthand property for the `row-gap` and `column-gap` properties. It takes two values: `row-gap` and `column-gap`, so it is written as `gap: row-gap column-gap`.

All these properties only work if the parent element has the `display: flex` property.

1.7 Grow & Shrink

`flex-grow` property specifies how much the item will grow relative to the rest of the flexible items. It takes a number as a value.

`flex-shrink` property specifies how much the item will shrink relative to the rest of the flexible items. It takes a number as a value.

The default value for `flex-grow` is 0, and the default value for `flex-shrink` is 1.

For example if you have 4 items inside a flex container and you have extra space of 300px and you set the `flex-grow` property of one item to 1 and the `flex-grow` property of the other item to 2, the first item will take 100px and the other items will take 200px each.

Another example, if you have 4 items inside a flex container and you lack space of 300px and you set the `flex-shrink` property of one item to 1 and the `flex-shrink` property of the other item to 2, and you set the rest of the items to 0, the first item will shrink by 100px and the other items will shrink by 200px each.

1.8 Flex Basis

`flex-basis` property specifies the initial size of the item before the remaining space is distributed.

The value of `flex-basis` can be a `width` or `height` depending on the `flex-direction` property, so if the `flex-direction` is `row`, the value of `flex-basis` will be a `width`, and if the `flex-direction` is `column`, the value of `flex-basis` will be a `height`.

1.9 Flex Shorthand

`flex` is a CSS property that allows us to create a flexible layout. It is a shorthand property for the `flex-grow`, `flex-shrink`, and `flex-basis` properties.