CSS Flexbox

CSS Flexbox is a one-dimensional layout model that allows you to design complex web layouts more efficiently and intuitively. It provides a flexible way to distribute space among elements in a container and align them in various ways.

Parent properties

1. Flex Container (display: flex):

- To enable Flexbox layout for a container, set its display property to flex.
- The children of a flex container become flex items.

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

2. Flex Direction (flex-direction):

- Determines the main axis direction along which flex items are laid out.
- Values: row (default), row-reverse, column, column-reverse.

```
.container {
    flex-direction: row;         /* Default: horizontal */
    flex-direction: row-reverse; /* Horizontal reversed */
    flex-direction: column;   /* Vertical */
    flex-direction: column-reverse; /* Vertical reversed */
  }
```

3. Justify Content (justify-content):

- Aligns flex items along the main(horizontal) axis of the flex container.
- Values: flex-start, flex-end, center, space-between, space-around, space-evenly.

```
.container {
    justify-content: flex-start; /* Default: aligned at the start */
    justify-content: flex-end;   /* Aligned at the end */
    justify-content: center;    /* Centered */
    justify-content: space-between; /* Evenly distributed with space between */
    justify-content: space-around; /* Evenly distributed with space around */
    justify-content: space-evenly; /* Evenly distributed with equal space around them */
}
```

4. Align Items (align-items):

- Aligns flex items along the cross axis of the flex container.
- Values: stretch (default), flex-start, flex-end, center, baseline.

```
.container {
    align-items: stretch;       /* Default: stretch to fill container */
    align-items: flex-start;    /* Align items at the start */
    align-items: flex-end;      /* Align items at the end */
    align-items: center;        /* Center items */
    align-items: baseline;      /* Align items along the baseline */
}
```

5. Align Content (align-content):

- Aligns multiple lines of flex items when there's extra space in the cross axis.
- Only applicable when there are multiple lines of flex items.
- Values: flex-start, flex-end, center, space-between, space-around, space-evenly.

```
.container {
align-content: stretch;     /* Default: lines stretch to fill space */
align-content: flex-start;  /* Lines packed to the start */
align-content: flex-end;    /* Lines packed to the end */

align-content: center;      /* Lines centered */

align-content: space-between; /* Lines evenly distributed with    space
between them */

align-content: space-around; /* Lines evenly distributed with space
around them */

align-content: space-evenly; /* Lines evenly distributed with equal
space around them */

            }
```

6. Flex wrap (flex-wrap):

- property specifies whether the flex items should wrap or not.
- Values: wrap or no-wrap or wrap-reverse.

```
.container {

   flex-wrap: nowrap;        /* Default: no wrap */

   flex-wrap: wrap;          /* Wrap items */

   flex-wrap: wrap-reverse;/* Wrap items in reverse order */

}
```

7. row-gap | column-gap | gap

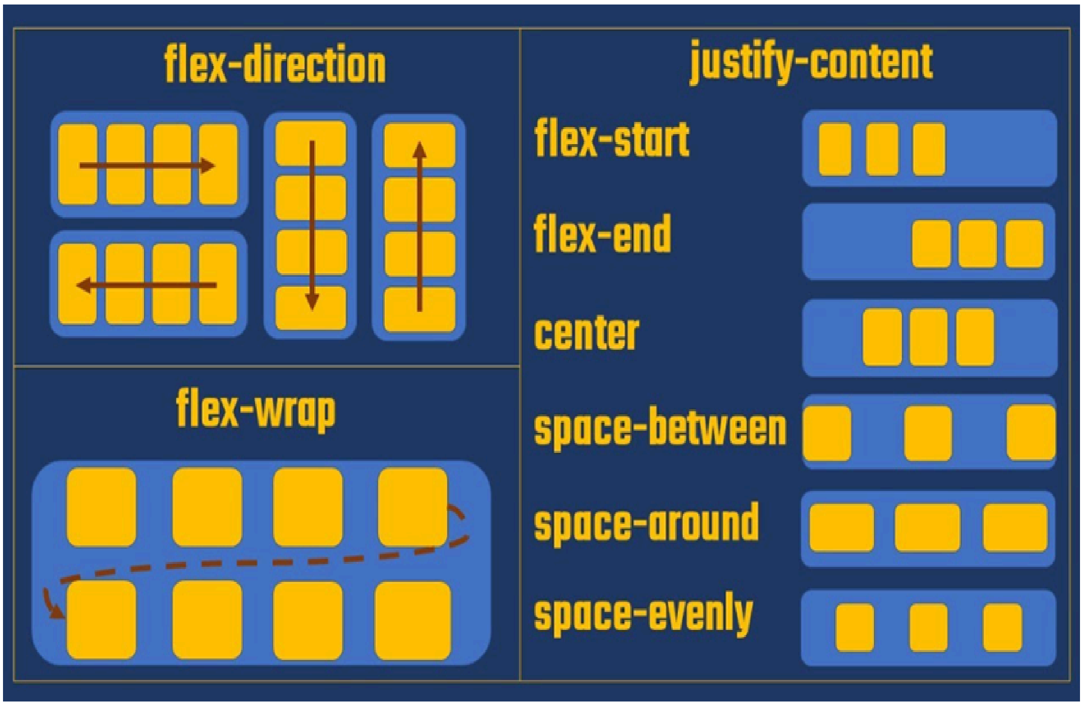
- row-gap creates the space between the rows
- column-gap creates the space between the columns
- gap is a shorthand for setting both column-gap and row-gap simultaneously

```
.container {
    gap: 20px;       /* creates a gap of 20px on both rows and columns*/
    row-gap:10px;       /*creates a gap of 10px between the rows */
    column-gap:10px;  /* creates a gap of 10px between the column */
}
```

8. flex-flow

A shorthand for flex-direction and flex-wrap.

```
.container {
        flex-flow: row wrap;      /* Combines direction and wrapping */
}
```



Child properties

1. Flex Basis (flex-basis):

- Specifies the initial size of a flex item before free space is distributed.
- Default value is auto, which means the item's size is based on its content size.

```
.item {
    flex-basis: 200px; /* Default is auto */
}
```

2. Flex Grow (flex-grow):

- Defines how much a flex item can grow relative to other flex items.

- A value of 0 means the item won't grow.
- A higher value means it will grow more compared to other flex items.

```
.item {
  flex-grow: 1; /* Default is 0, no growth */
}
```

3. Flex Shrink (flex-shrink):

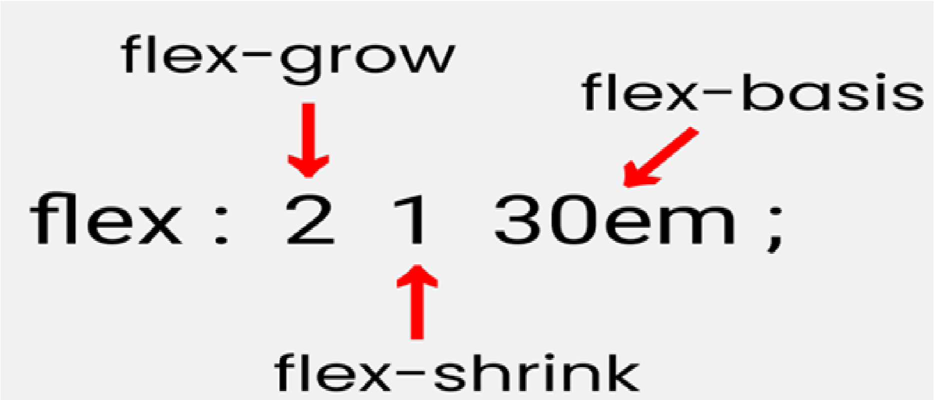
- Defines how much a flex item can shrink relative to other flex items.
- A value of 0 means the item won't shrink.
- A higher value means it will shrink more compared to other flex items.

```
.item {
    flex-shrink: 1; /* Default is 1, items shrink */
}
```

4. Flex (flex shorthand):

- Combines flex-grow, flex-shrink, and flex-basis properties into a single shorthand property.

```
.item {
    flex: 1 1 200px; /* flex-grow: 1; flex-shrink: 1;
flex-basis: 200px; */
}
```



5. Order (order):

- Specifies the order in which a flex item appears in the flex container.
- By default, items have an order of 0.
- Lower values come first, and higher values come later.

```
.item {
        order: 1;   /* Default is 0 */
   }
```