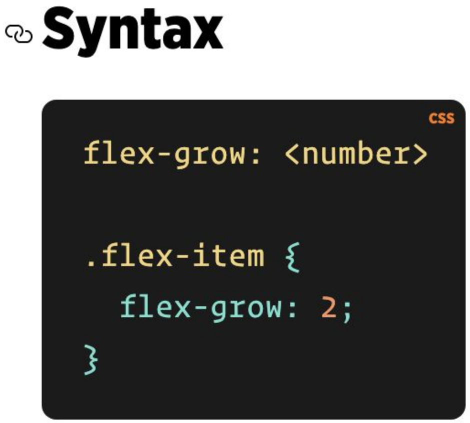
FLEXBOX ON CHILD ELEMENTS

FLEX-GROW

The flex-grow property defines the ability for a flex item to grow if necessary. It dictates what amount of the available space inside the flex container the item should take up.

For example, if all items have flex-grow set to 1, every child will set to an equal size inside the container. If you were to give one of the children a value of 2, that child would take up twice as much space as the others.

Negative values are considered invalid.



Syntax

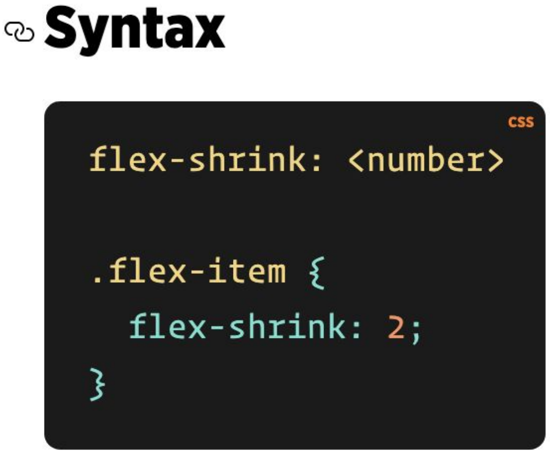
```css
flex-grow: <number>

.flex-item {
  flex-grow: 2;
}
```

FLEX-SHRINK

It specifies the “flex shrink factor”, which determines how much the flex item will shrink relative to the rest of the flex items in the flex container when there isn’t enough space on the row.

When omitted, it is set to 1 and the flex shrink factor is multiplied by the flex basis when distributing negative space.

Negative values are considered invalid.



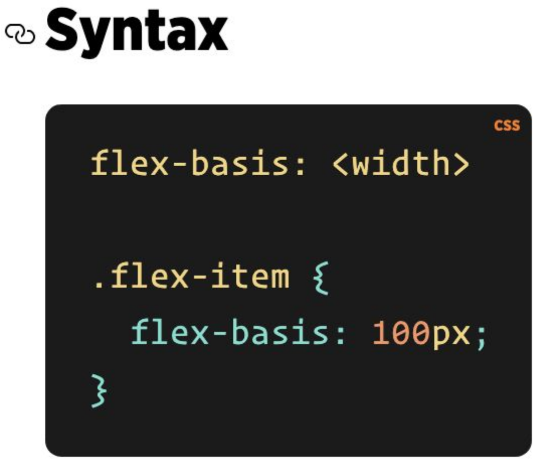
Syntax

```css
flex-shrink: <number>

.flex-item {
  flex-shrink: 2;
}
```

FLEX-BASIS

It specifies the initial size of the flex item, before any available space is distributed according to the flex factors.

When omitted from the flex shorthand, its specified value is the length zero. A flex-basis value set to auto sizes the element according to its size property (which can itself be the keyword auto, which sizes the element based on its contents).



Syntax

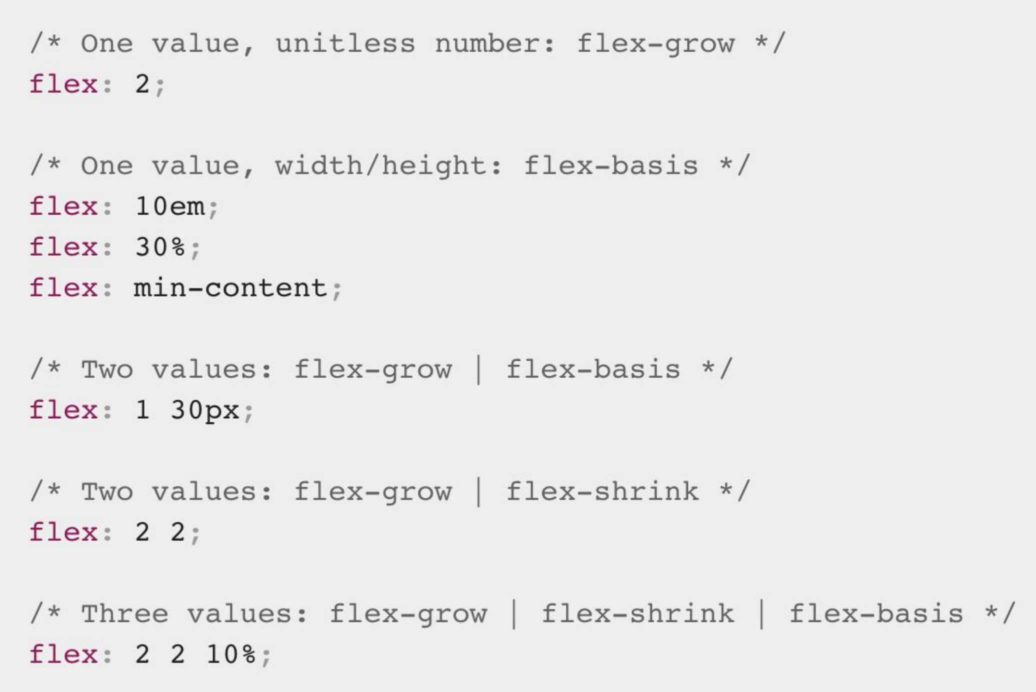
```css
flex-basis: <width>

.flex-item {
  flex-basis: 100px;
}
```

FLEX

The flex CSS shorthand property sets how a flex item will grow or shrink to fit the space available in its flex container.

flex: flex-grow flex-shrink flex-basis;

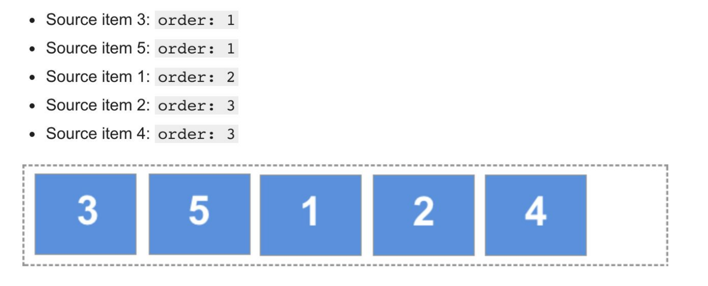
The second and third parameters (flex-shrink and flex-basis) are optional.



```
/* One value, unitless number: flex-grow */
flex: 2;

/* One value, width/height: flex-basis */
flex: 10em;
flex: 30%;
flex: min-content;

/* Two values: flex-grow | flex-basis */
flex: 1 30px;

/* Two values: flex-grow | flex-shrink */
flex: 2 2;

/* Three values: flex-grow | flex-shrink | flex-basis */
flex: 2 2 10%;
```

ORDER

The order property specifies the order of a flexible item relative to the rest of the flexible items inside the same container.

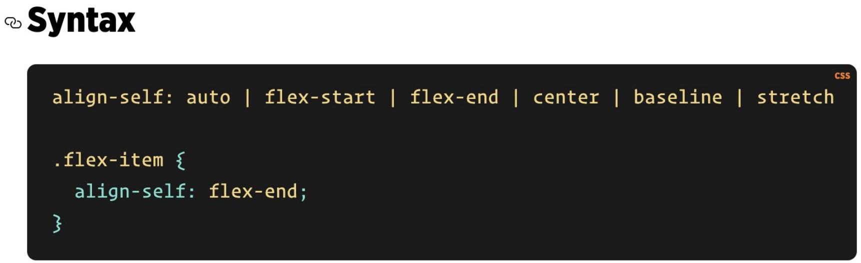


- Source item 3: `order: 1`
- Source item 5: `order: 1`
- Source item 1: `order: 2`
- Source item 2: `order: 3`
- Source item 4: `order: 3`

If you want to make one item display first, and leave the order of all other items unchanged, you can give that item an order of -1.

Avoid order for accessibility since people with screen readers will receive the order from the html document instead.

ALIGN-SELF



Syntax

```css
align-self: auto | flex-start | flex-end | center | baseline | stretch

.flex-item {
  align-self: flex-end;
}
```

align-self makes possible to override the align-items value for specific flex items.

The **align-self** property accepts the same 5 values as the align-items: flex-start, flex-end, center, baseline, stretch.

ANY QUESTIONS?