

# Web Technologies

Kamran

Lecture 9

# Flexbox

CSCI 1720



CSCI 1720  
Intermediate Web Design

# CSS3 Flexbox

Flexible boxes, or flexbox, is a new layout mode in CSS3

Use of flexbox ensures that elements behave predictably when the page layout must accommodate different screen sizes and different display devices

For many applications, the flexible box model provides an improvement over the block model in that it does not use floats, nor do the flex container's margins collapse with the margins of its contents



# CSS3 Flexbox - Concepts

Flexbox consists of flex containers and flex items

A flex container is declared by setting the display property of an element to either flex (rendered as a block) or inline-flex (rendered as inline)

Inside a flex container there is one or more flex items



# CSS3 Flexbox - Concepts

Note: Everything outside a flex container and inside a flex item is rendered as usual

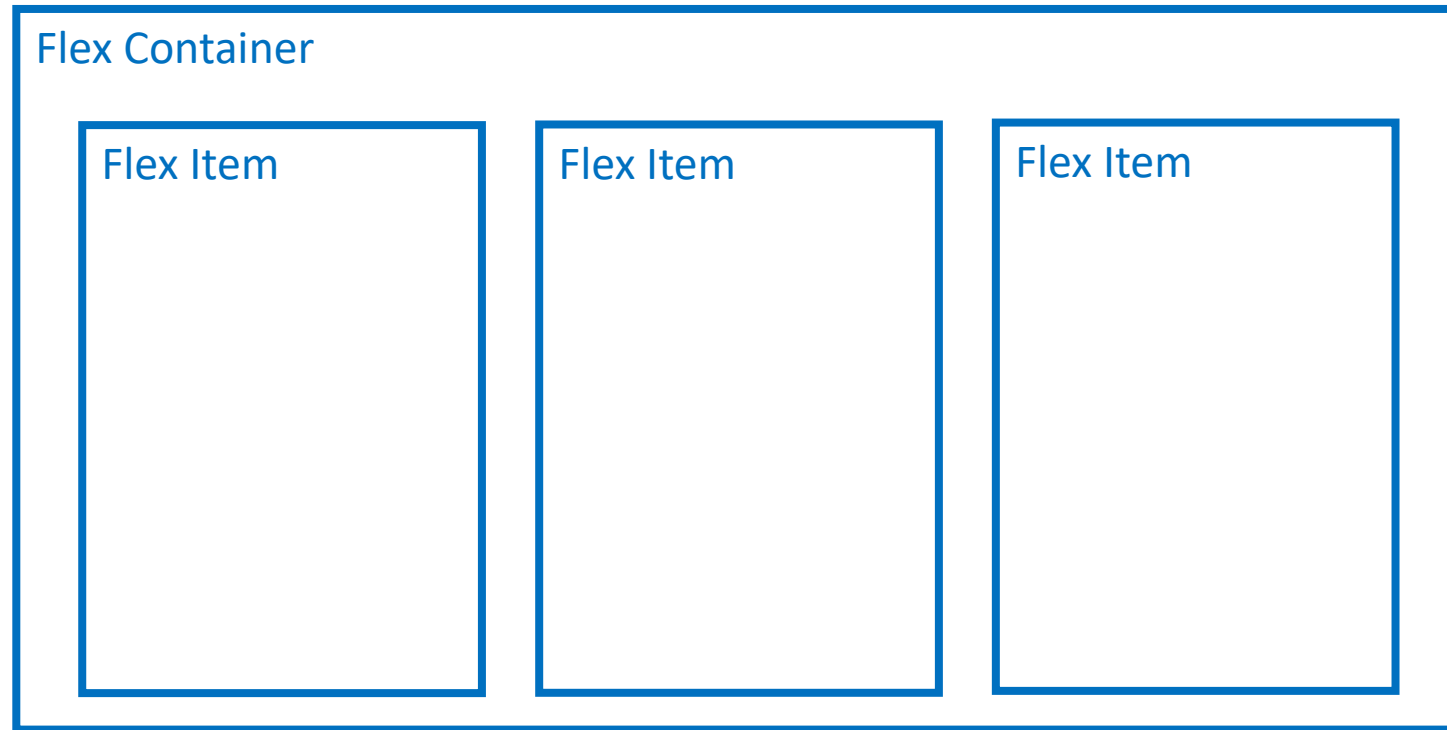
Flexbox defines how flex items are laid out inside a flex container

Flex items are positioned inside a flex container along a flex line

By default there is only one flex line per flex container



# CSS3 Flexbox - Concepts



# CSS3 Flexbox - Code

CSS:

```
.flex-container {  
    display: -webkit-flex;  
    display: flex;  
    width: 400px;  
    height: 250px;  
    background-color: lightgrey;  
}  
  
.flex-item {  
    background-color: cornflowerblue;  
    width: 100px;  
    height: 100px;  
    margin: 10px;  
}
```

Why do we need  
this?



# CSS3 Flexbox - Code

HTML:

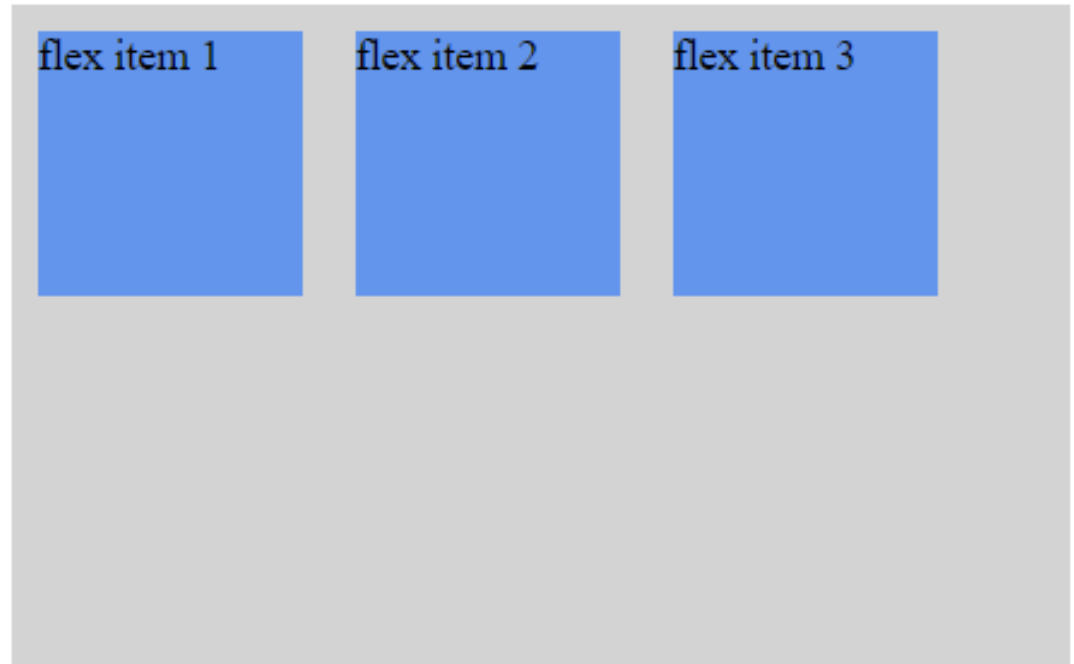
```
<div class="flex-container">  
  <div class="flex-item">flex item 1</div>  
  <div class="flex-item">flex item 2</div>  
  <div class="flex-item">flex item 3</div>  
</div>
```





# CSS3 Flexbox - Code

Result:



# CSS3 Flexbox - Code

It is also possible to change the direction of the flex line

If we set the direction property to rtl (right-to-left), the text is drawn right to left, and also the flex line changes direction, which will change the page layout

```
body {  
    direction: rtl;  
}
```

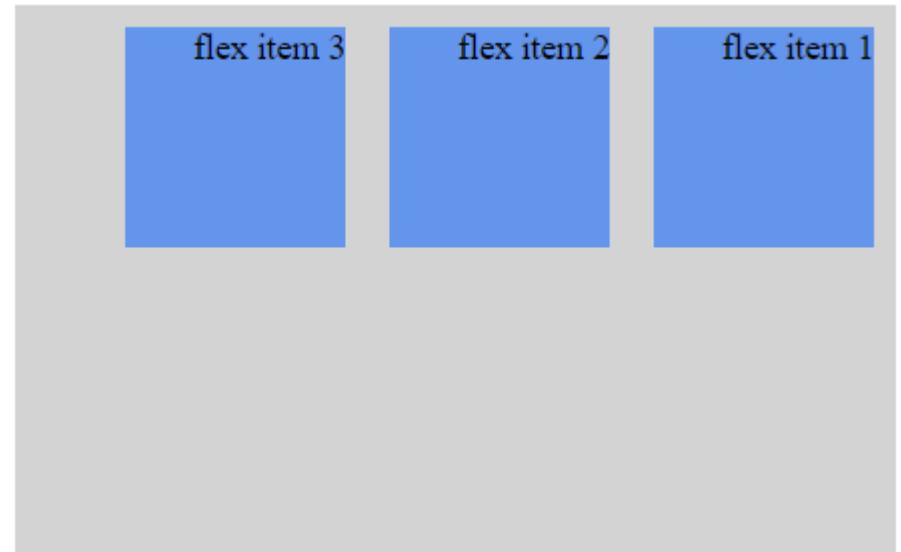
```
.flex-container {  
    display: -webkit-flex;  
    display: flex;  
    width: 400px;  
    height: 250px;  
    background-color: lightgrey;  
}
```

```
.flex-item {  
    background-color: cornflowerblue;  
    width: 100px;  
    height: 100px;  
    margin: 10px;  
}
```



# CSS3 Flexbox - Code

Result:



# CSS3 Flexbox - Flex Direction

The flex-direction property specifies the direction of the flexible items inside the flex container. The default value of flex-direction is row (left-to-right, top-to-bottom)

The other values are as follows:

**row-reverse** - If the writing-mode (direction) is left to right, the flex items will be laid out right to left

**column** - If the writing system is horizontal, the flex items will be laid out vertically

**column-reverse** - Same as column, but reversed



# CSS3 Flexbox - Flex Direction

Example:  
(Row-reverse)

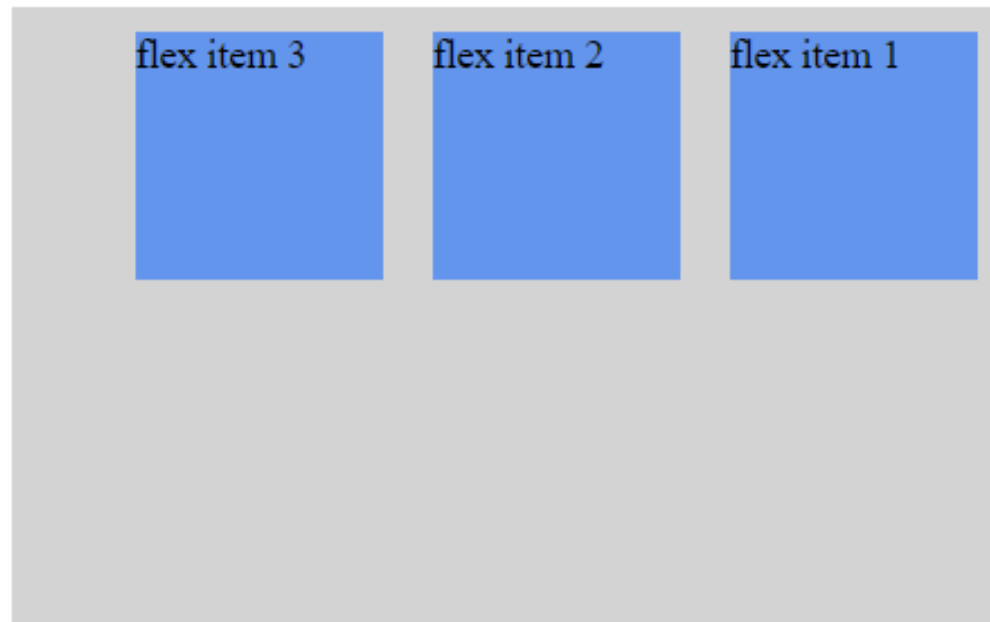
```
.flex-container {  
    display: -webkit-flex;  
    display: flex;  
    -webkit-flex-direction: row-reverse;  
    flex-direction: row-reverse;  
    width: 400px;  
    height: 250px;  
    background-color: lightgrey;  
}
```

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



# CSS3 Flexbox - Flex Direction

Example:



# CSS3 Flexbox - Flex Direction

Example:  
(Column)

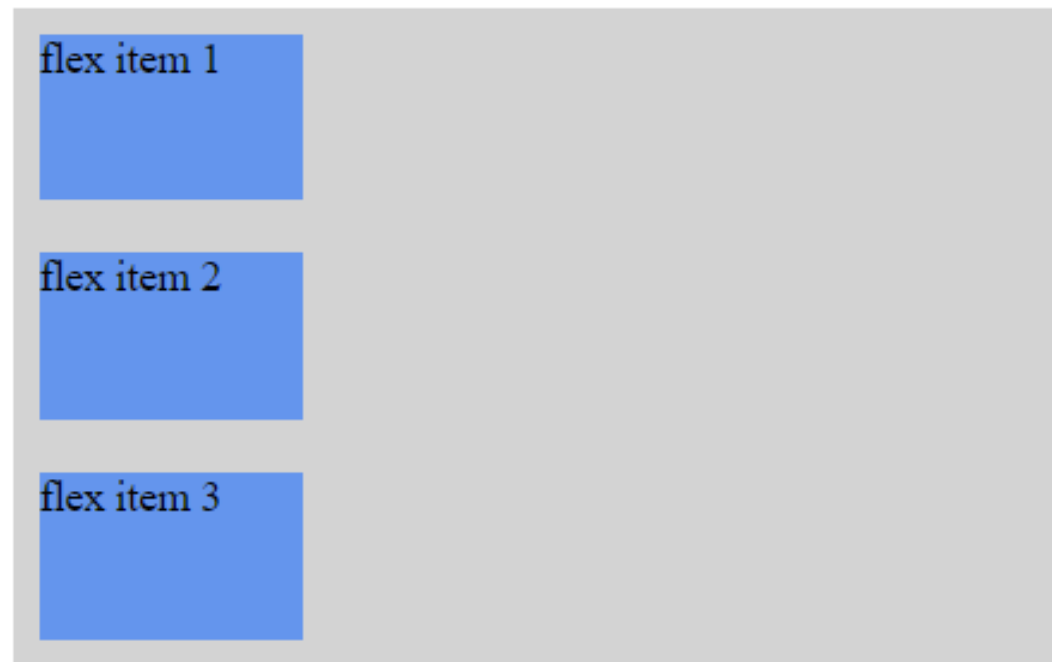
```
.flex-container {  
    display: -webkit-flex;  
    display: flex;  
    -webkit-flex-direction: column;  
    flex-direction: column;  
    width: 400px;  
    height: 250px;  
    background-color: lightgrey;  
}
```

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



# CSS3 Flexbox - Flex Direction

Example:





# CSS3 Flexbox - Flex Direction

Example:  
(Column-  
reverse)

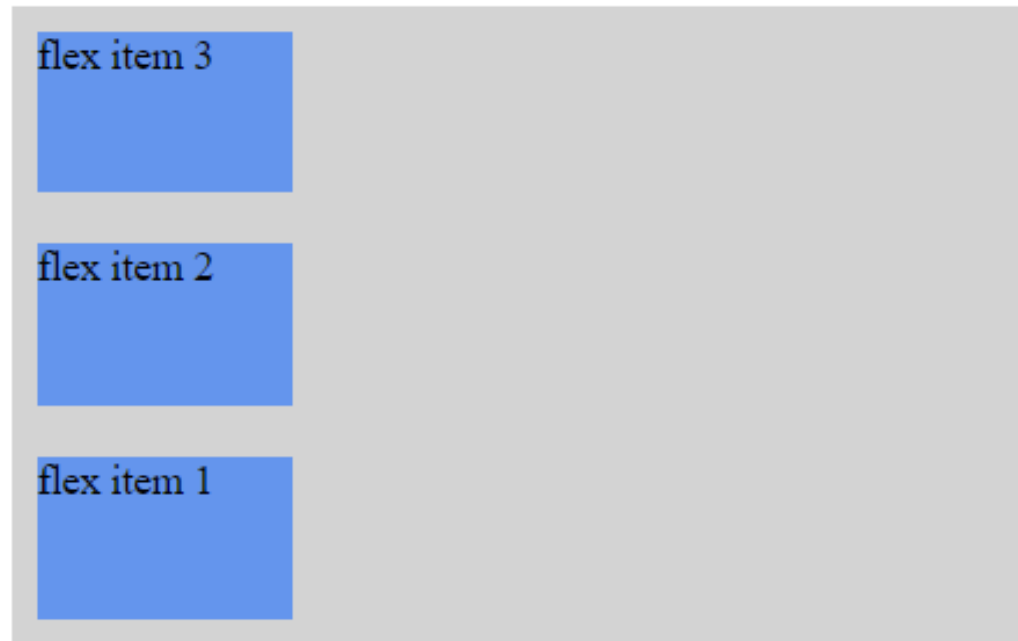
```
.flex-container {  
    display: -webkit-flex;  
    display: flex;  
    -webkit-flex-direction: column-reverse;  
    flex-direction: column-reverse;  
    width: 400px;  
    height: 250px;  
    background-color: lightgrey;  
}
```

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



# CSS3 Flexbox - Flex Direction

Example:



## justify-content Property

The `justify-content` property horizontally aligns the flexible container's items when the items do not use all available space on the main-axis

The possible values are as follows:

**flex-start** - Default value. Items are positioned at the beginning of the container

**flex-end** - Items are positioned at the end of the container

**center** - Items are positioned at the center of the container

**space-between** - Items are positioned with space between the lines

**space-around** - Items are positioned with space before, between, and after the lines



# justify-content Property

Example:  
(Flex-end)

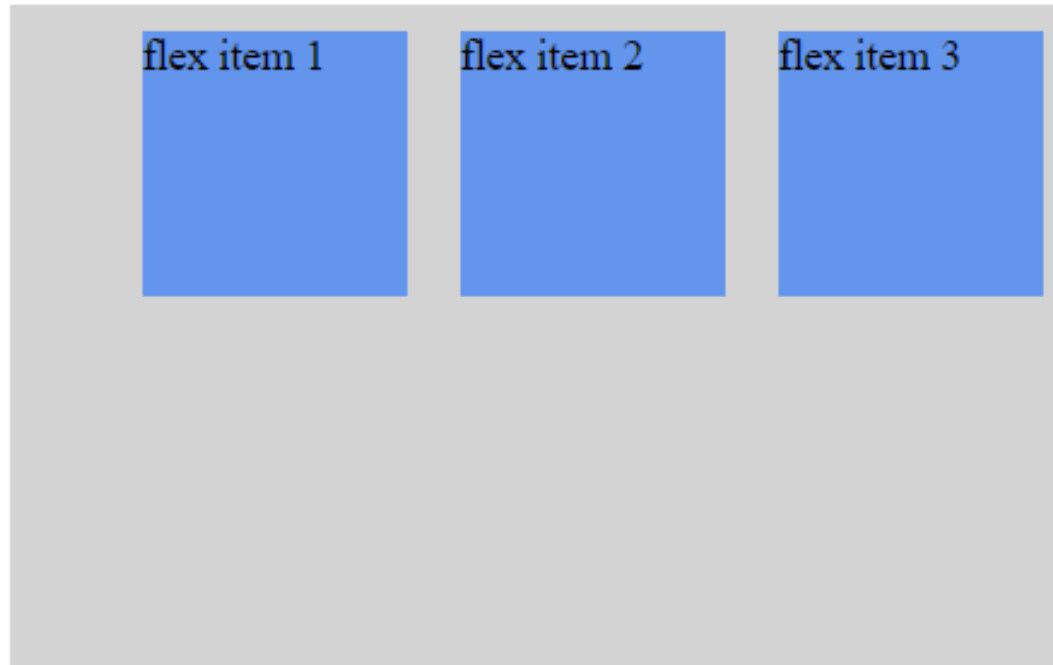
```
.flex-container {  
    display: -webkit-flex;  
    display: flex;  
    -webkit-justify-content: flex-end;  
    justify-content: flex-end;  
    width: 400px;  
    height: 250px;  
    background-color: lightgrey;  
}
```

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



# `justify-content` Property

Example  
(Flex-end):



# justify-content Property

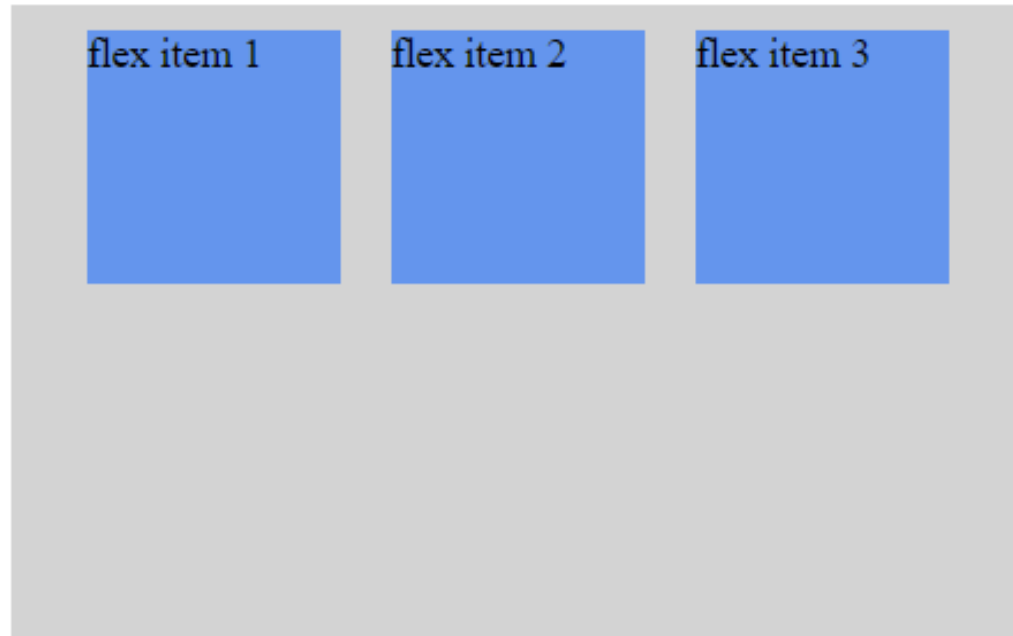
Example:  
(center)

```
.flex-container {  
    display: -webkit-flex;  
    display: flex;  
    -webkit-justify-content: center;  
    justify-content: center;  
    width: 400px;  
    height: 250px;  
    background-color: lightgrey;  
}
```

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

# `justify-content` Property

Example:  
(center)



# justify-content Property

Example:  
(space-  
between)

```
.flex-container {  
  display: -webkit-flex;  
  display: flex;  
  -webkit-justify-content: space-between;  
  justify-content: space-between;  
  width: 400px;  
  height: 250px;  
  background-color: lightgrey;  
}
```

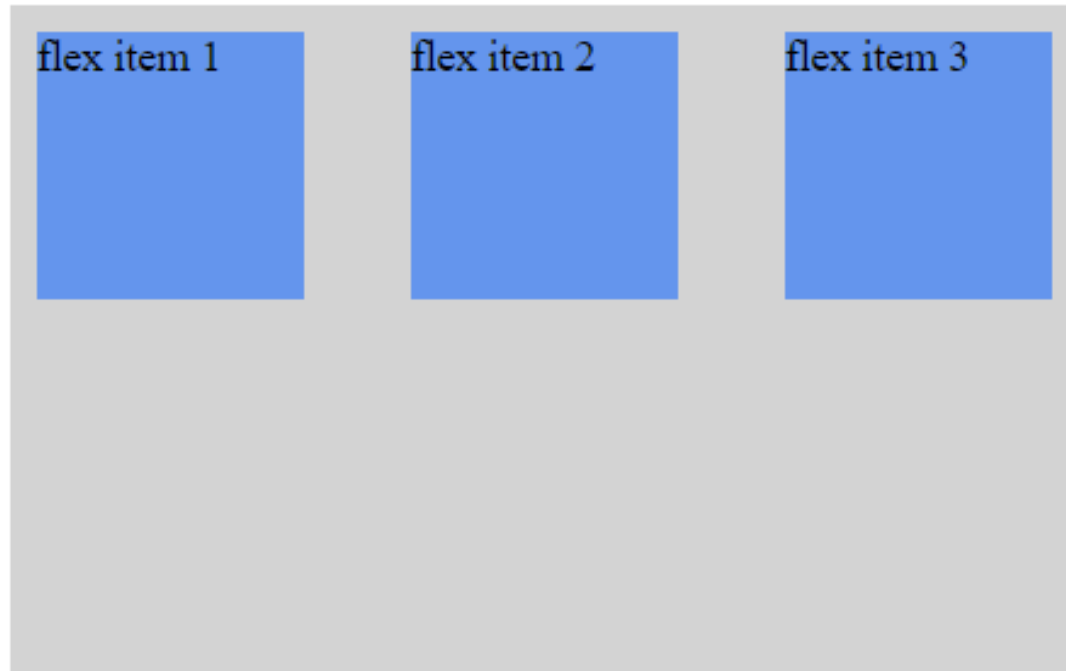
```
<div class="flex-container">  
  <div class="flex-item">flex item 1</div>  
  <div class="flex-item">flex item 2</div>  
  <div class="flex-item">flex item 3</div>  
</div>
```





# `justify-content` Property

Example:  
(space-between)



## `align-items` Property

The `align-items` property vertically aligns the flexible container's items when the items do not use all available space on the cross-axis

The possible values are as follows:

**stretch** - Default value. Items are stretched to fit the container

**flex-start** - Items are positioned at the top of the container

**flex-end** - Items are positioned at the bottom of the container

**center** - Items are positioned at the center of the container (vertically)

**baseline** - Items are positioned at the baseline of the container



## align-items Property

Example:  
(Stretch)

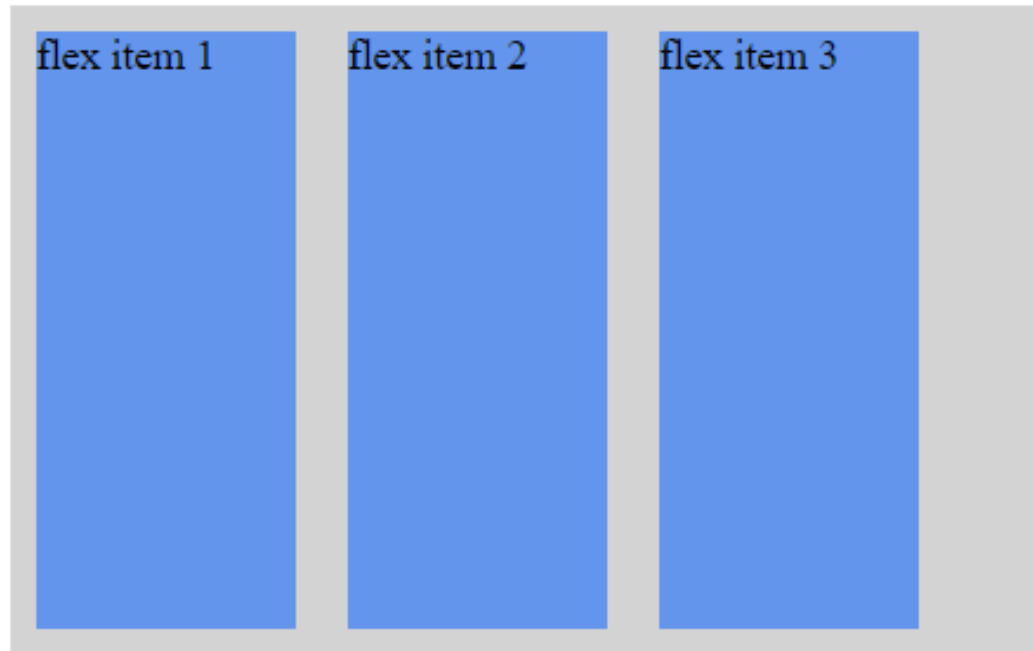
```
.flex-container {  
  display: -webkit-flex;  
  display: flex;  
  -webkit-align-items: stretch;  
  align-items: stretch;  
  width: 400px;  
  height: 250px;  
  background-color: lightgrey;  
}
```

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



# `align-items` Property

Example:  
(Stretch)



## align-items Property

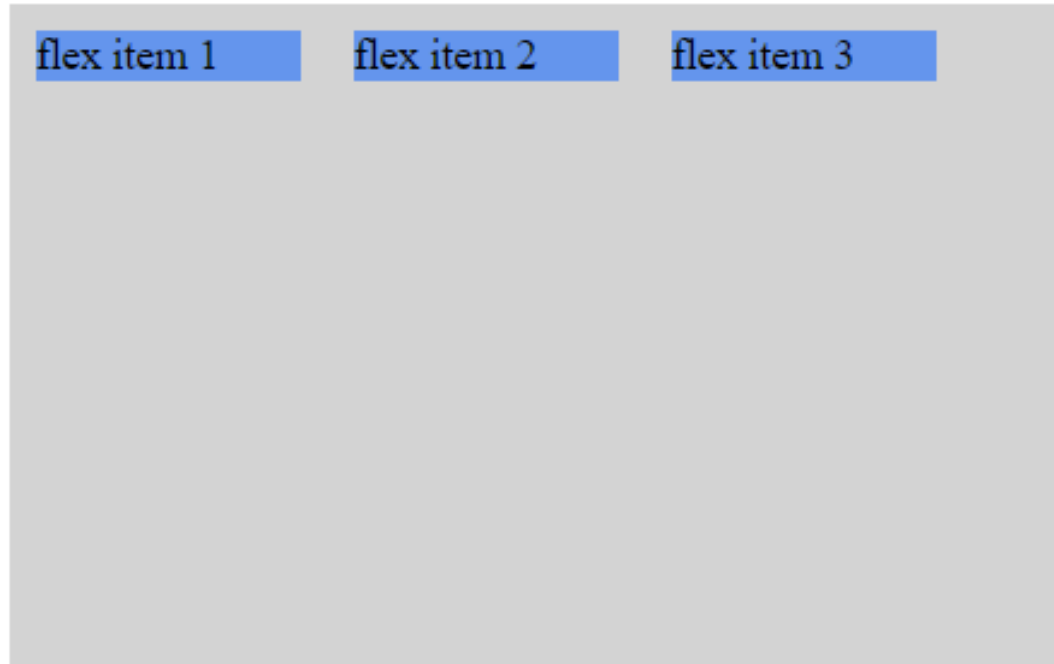
Example:  
(Flex-start)

```
.flex-container {  
  display: -webkit-flex;  
  display: flex;  
  -webkit-align-items: flex-start;  
  align-items: flex-start;  
  width: 400px;  
  height: 250px;  
  background-color: lightgrey;  
}  
  
<div class="flex-container">  
  <div class="flex-item">flex item 1</div>  
  <div class="flex-item">flex item 2</div>  
  <div class="flex-item">flex item 3</div>  
</div>
```



# `align-items` Property

Example:  
(Flex-start)



## align-items Property

Example:  
(Flex-end)

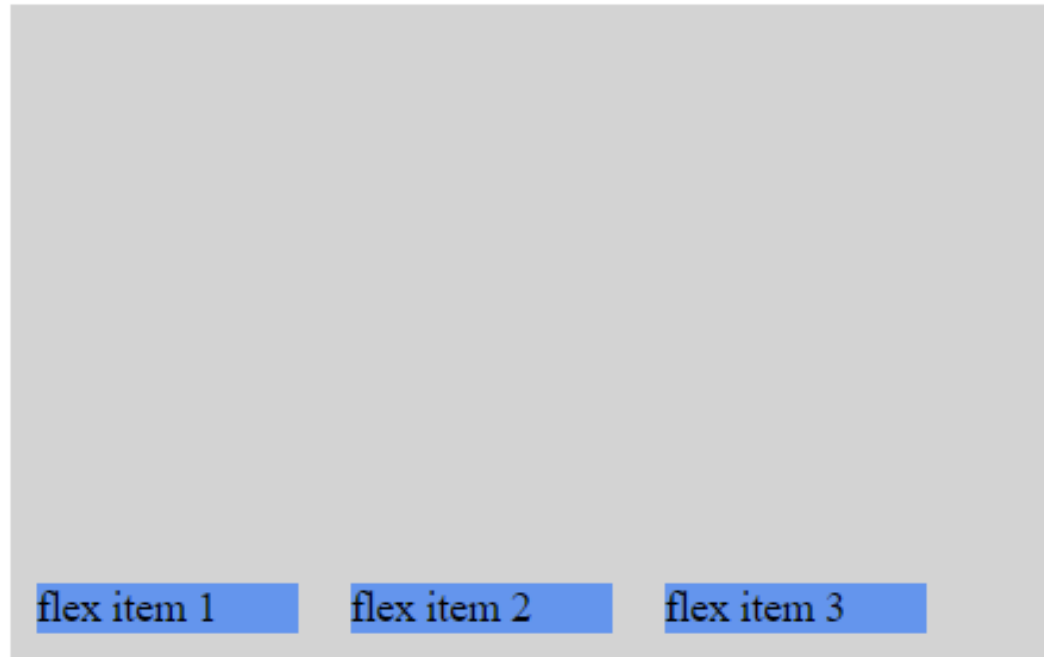
```
.flex-container {  
  display: -webkit-flex;  
  display: flex;  
  -webkit-align-items: flex-end;  
  align-items: flex-end;  
  width: 400px;  
  height: 250px;  
  background-color: lightgrey;  
}
```

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



# `align-items` Property

Example:  
(Flex-end)





## align-items Property

Example:  
(Center)

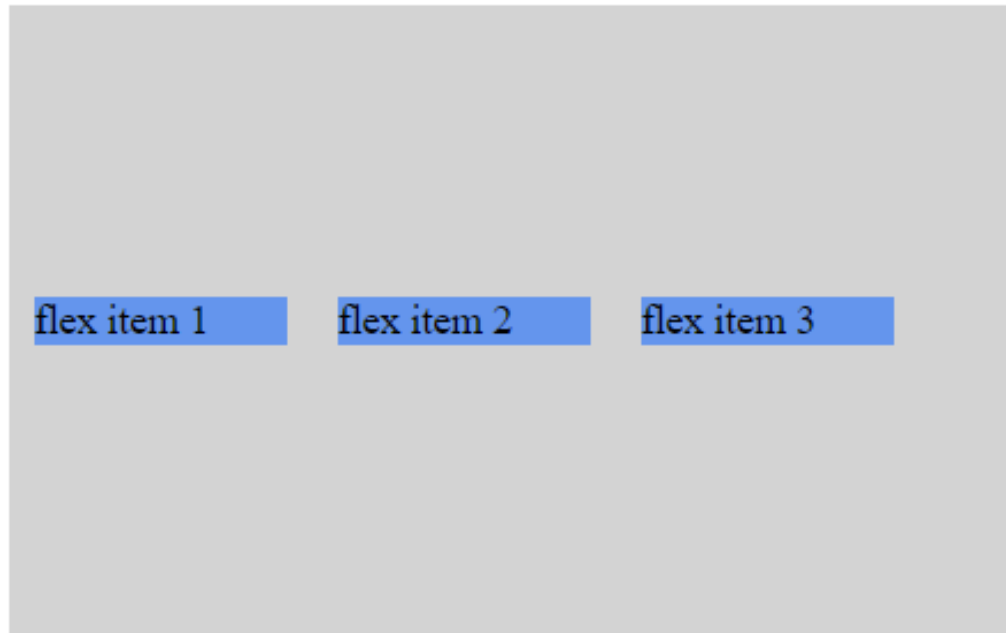
```
.flex-container {  
  display: -webkit-flex;  
  display: flex;  
  -webkit-align-items: center;  
  align-items: center;  
  width: 400px;  
  height: 250px;  
  background-color: lightgrey;  
}
```

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



# `align-items` Property

Example:  
(Center)



## align-items Property

Example:  
(Baseline)

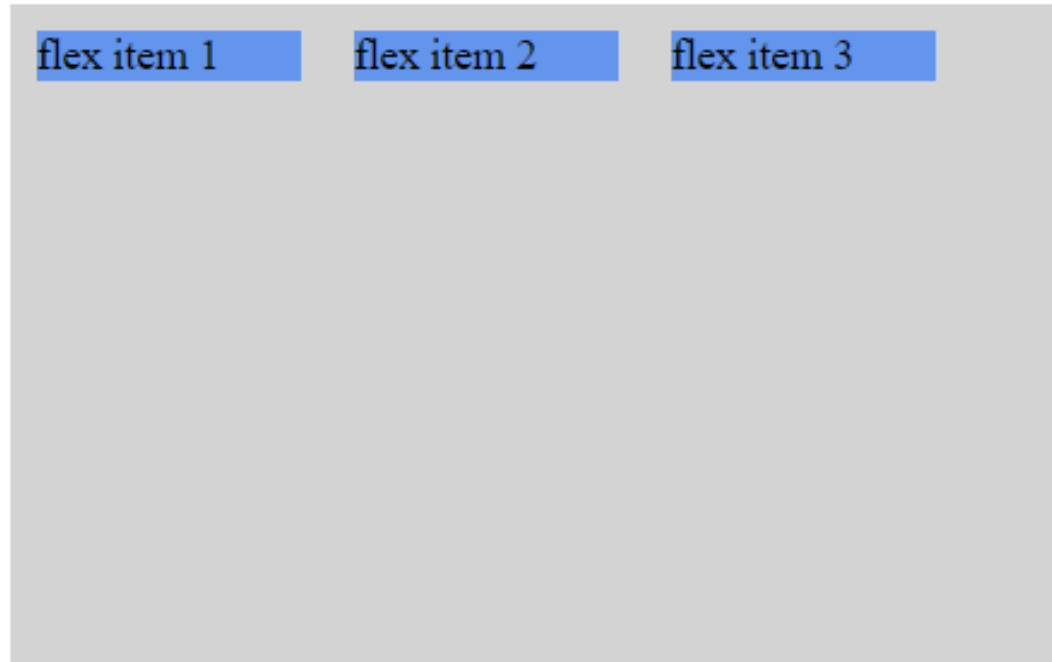
```
.flex-container {  
  display: -webkit-flex;  
  display: flex;  
  -webkit-align-items: baseline;  
  align-items: baseline;  
  width: 400px;  
  height: 250px;  
  background-color: lightgrey;  
}
```

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



# `align-items` Property

Example:  
(Baseline)



## flex-wrap Property

The flex-wrap property specifies whether the flex items should wrap or not, if there is not enough room for them on one flex line

The possible values are as follows:

**nowrap** - Default value. The flexible items will not wrap

**wrap** - The flexible items will wrap if necessary

**wrap-reverse** - The flexible items will wrap, if necessary, in reverse order



## flex-wrap Property

Example:  
(Nowrap)

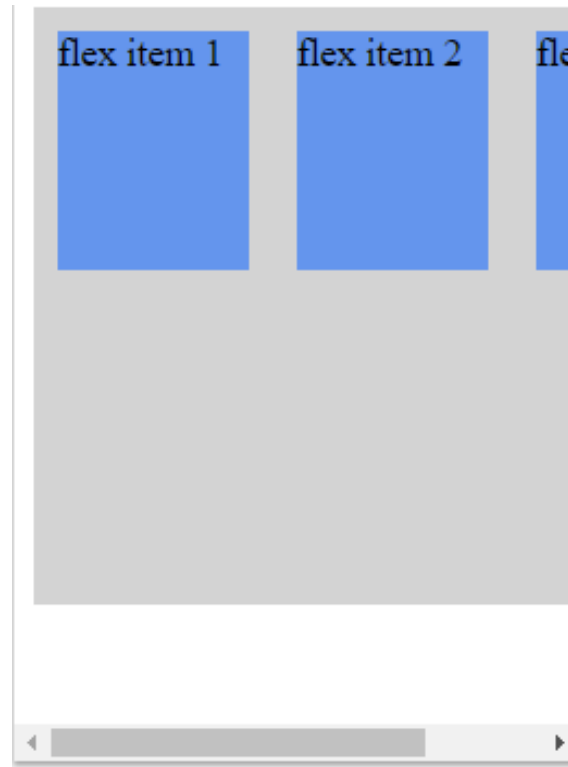
```
.flex-container {  
  display: -webkit-flex;  
  display: flex;  
  -webkit-flex-wrap: nowrap;  
  flex-wrap: nowrap;  
  width: 300px;  
  height: 250px;  
  background-color: lightgrey;  
}
```

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



# `flex-wrap` Property

Example:  
(Nowrap)



## flex-wrap Property

Example:  
(Wrap)

```
.flex-container {  
  display: -webkit-flex;  
  display: flex;  
  -webkit-flex-wrap: wrap;  
  flex-wrap: wrap;  
  width: 300px;  
  height: 250px;  
  background-color: lightgrey;  
}
```

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





## `flex-wrap` Property

Example:  
(wrap)



## `align-content` Property

The `align-content` property modifies the behavior of the `flex-wrap` property. It is similar to `align-items`, but instead of aligning flex items, it aligns flex lines

The possible values are as follows:

**`stretch`** - Default value. Lines stretch to take up the remaining space

**`flex-start`** - Lines are packed toward the start of the flex container

**`flex-end`** - Lines are packed toward the end of the flex container



## `align-content` Property

The align-content property modifies the behavior of the flex-wrap property. It is similar to align-items, but instead of aligning flex items, it aligns flex lines

The possible values are as follows:

**center** - Lines are packed toward the center of the flex container

**space-between** - Lines are evenly distributed in the flex container

**space-around** - Lines are evenly distributed in the flex container, with half-size spaces on either end



# align-content Property

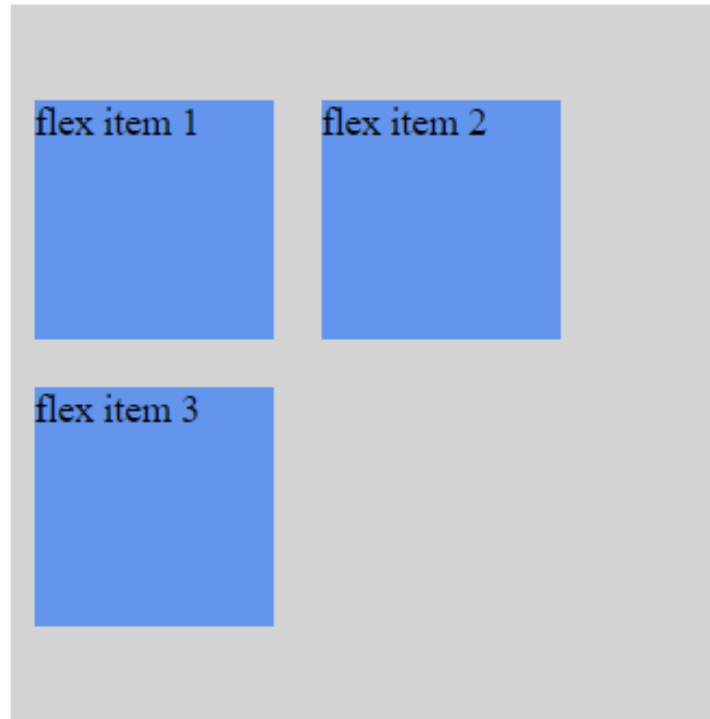
Example:  
(Center)

```
.flex-container {  
  display: -webkit-flex;  
  display: flex;  
  -webkit-flex-wrap: wrap;  
  flex-wrap: wrap;  
  -webkit-align-content: center;  
  align-content: center;  
  width: 300px;  
  height: 300px;  
  background-color: lightgrey;  
}  
  
<div class="flex-container">  
  <div class="flex-item">flex item 1</div>  
  <div class="flex-item">flex item 2</div>  
  <div class="flex-item">flex item 3</div>  
</div>
```



# `align-content` Property

Example:  
(Center)



# align-content Property

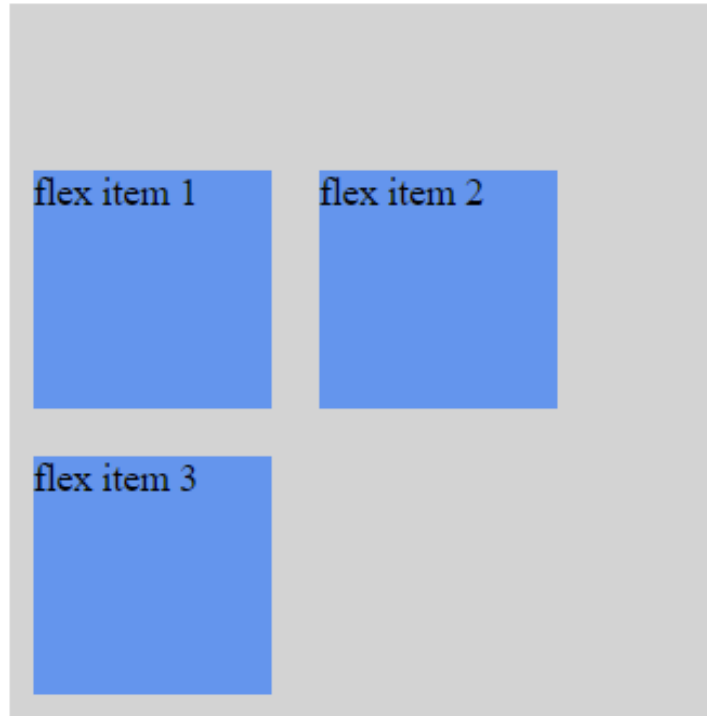
Example:  
(Flex-end)

```
.flex-container {  
  display: -webkit-flex;  
  display: flex;  
  -webkit-flex-wrap: wrap;  
  flex-wrap: wrap;  
  -webkit-align-content: flex-end;  
  align-content: flex-end;  
  width: 300px;  
  height: 300px;  
  background-color: lightgrey;  
}  
  
<div class="flex-container">  
  <div class="flex-item">flex item 1</div>  
  <div class="flex-item">flex item 2</div>  
  <div class="flex-item">flex item 3</div>  
</div>
```



# `align-content` Property

Example:  
(Flex-end)



# Flex Item Properties - Ordering

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





# Flex Item Properties - Ordering

Example:  
(Ordering)

```
.flex-item {  
    background-color: cornflowerblue;  
    width: 100px;  
    height: 100px;  
    margin: 10px;  
}
```

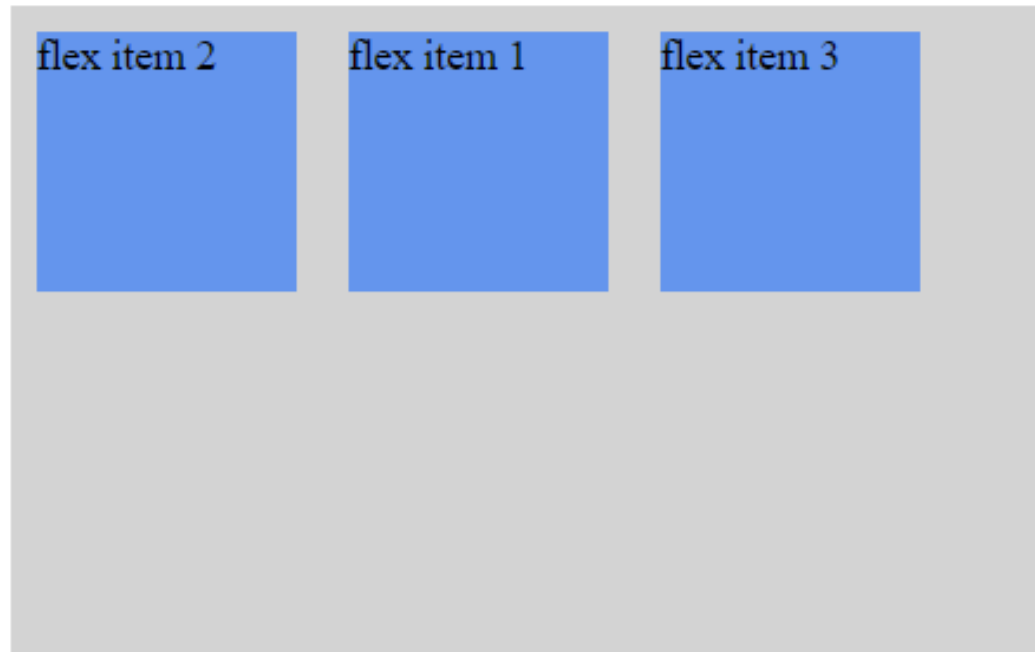
```
.first {  
    -webkit-order: -1;  
    order: -1;  
}
```

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



# Item Properties - Ordering

Example:  
(Ordering)



# Flex Item Properties - Margin

Setting margin: auto; will absorb extra space

It can be used to push flex items into different positions



# Flex Item Properties - Margin

Example:  
(Margin)

```
.flex-item {  
    background-color: cornflowerblue;  
    width: 75px;  
    height: 75px;  
    margin: 10px;  
}
```

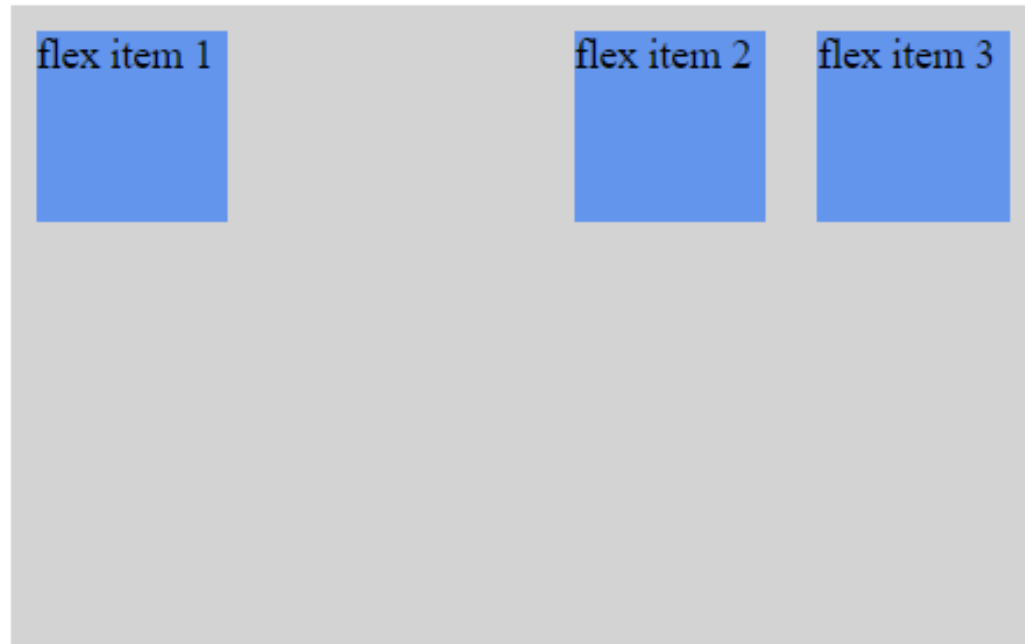
```
.flex-item:first-child {  
    margin-right: auto;  
}
```

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



# Flex Item Properties - Margin

Example:  
(Margin)



# Flex Item Properties - Perfect Centering

Setting `margin: auto;` will make the item perfectly centered in both axis



# Flex Item Properties - Margin

Example:  
(Margin)

```
.flex-item {  
    background-color: cornflowerblue;  
    width: 75px;  
    height: 75px;  
    margin: auto;  
}
```

```
<div class="flex-container">  
    <div class="flex-item">Perfect centering!</div>  
</div>
```



# Flex Item Properties - Margin

Example:  
(Margin)





# Flex Item Properties - Align Self

The `align-self` property of flex items overrides the flex container's `align-items` property for that item  
It has the same possible values as the `align-items` property



# Flex Item Properties - Align Self

Example:  
(Align self)

```
<div class="flex-container">
  <div class="flex-item item1">flex-start</div>
  <div class="flex-item item2">flex-end</div>
  <div class="flex-item item3">center</div>
  <div class="flex-item item4">baseline</div>
  <div class="flex-item item5">stretch</div>
</div>
```

```
.flex-item {
  background-color: cornflowerblue;
  width: 60px;
  min-height: 100px;
  margin: 10px;
}

.item1 {
  -webkit-align-self: flex-start;
  align-self: flex-start;
}

.item2 {
  -webkit-align-self: flex-end;
  align-self: flex-end;
}

.item3 {
  -webkit-align-self: center;
  align-self: center;
}

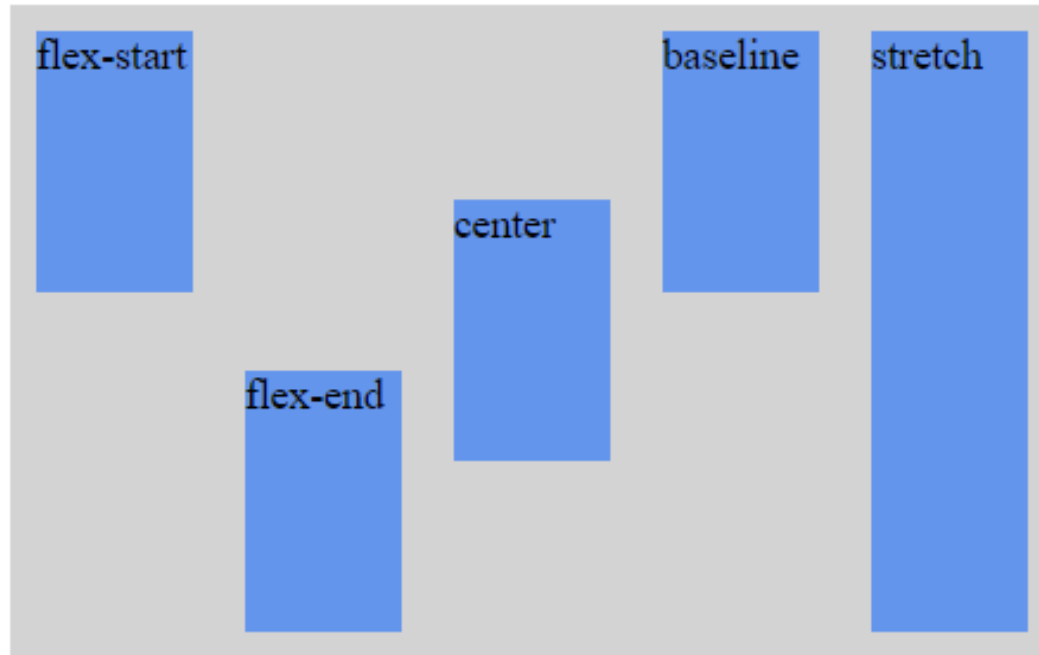
.item4 {
  -webkit-align-self: baseline;
  align-self: baseline;
}

.item5 {
  -webkit-align-self: stretch;
  align-self: stretch;
}
```



# Flex Item Properties - Align Self

Example:  
(Align self)



# Flex Item Properties - Flex

The `flex` property specifies the length of the flex item, relative to the rest of the flex items inside the same container



# Flex Item Properties - Flex

Example:  
(Flex)

```
<div class="flex-container">
  <div class="flex-item item1">flex item 1</div>
  <div class="flex-item item2">flex item 2</div>
  <div class="flex-item item3">flex item 3</div>
</div>
```

```
.flex-item {
  background-color: cornflowerblue;
  margin: 10px;
}

.item1 {
  -webkit-flex: 2;
  flex: 2;
}

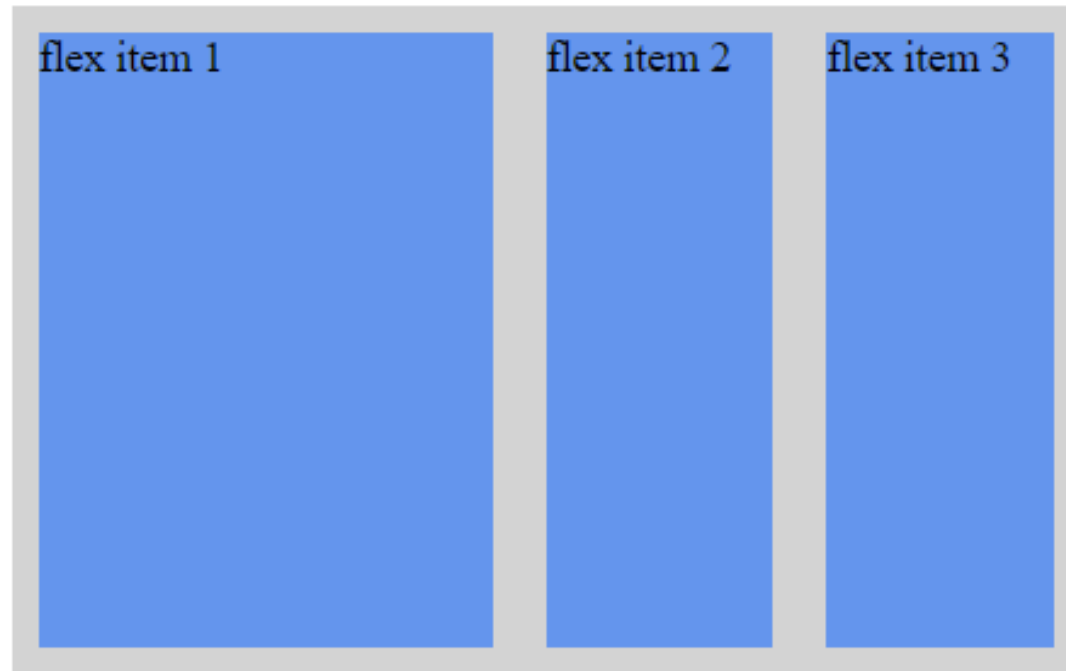
.item2 {
  -webkit-flex: 1;
  flex: 1;
}

.item3 {
  -webkit-flex: 1;
  flex: 1;
}
```



# Flex Item Properties - Flex

Example:  
(Flex)



# CSS3 Flexbox Properties

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



# Sources

[https://www.w3schools.com/css/css3\\_flexbox.asp](https://www.w3schools.com/css/css3_flexbox.asp)

