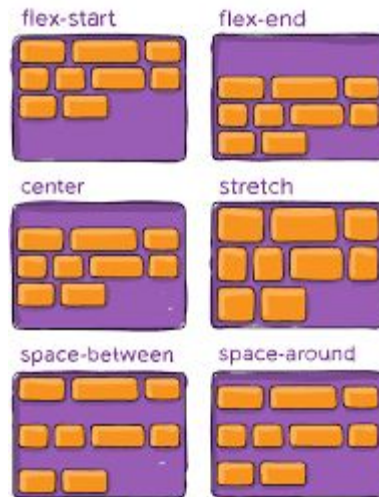
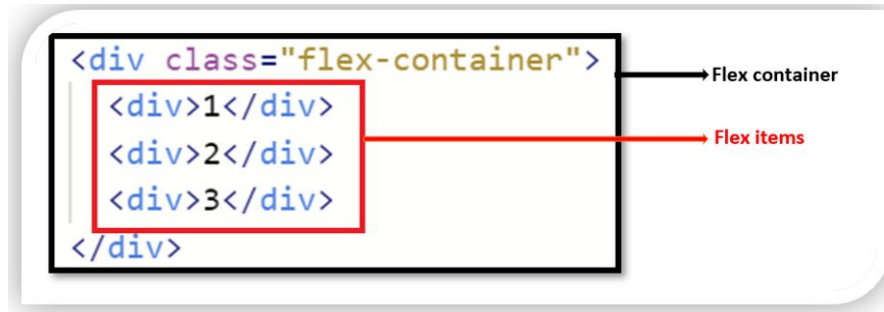


# CSS Flex

By Nitesh Gupta



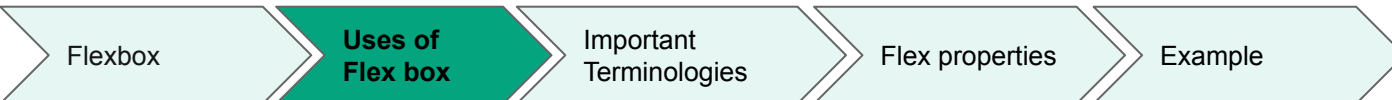
- Flexbox is a one-dimensional layout model, where flex means flexible length on flexible items.
- The flexbox layout must have a **parent element(container)** with the **display** property “**flex**”. So that the child elements of the container automatically become **flex items**.
- Flexbox provides an easy way to arrange items within a container.



- Flexbox **aligns and distributes space** among the items in the container even when the size of the items is unknown or dynamic.
- Flexbox expands/shrinks to fill available space/prevent overflow and distributes space among the items.
- Flexbox is used for creating small-scale layouts.
- Flexbox aligns the items vertically and horizontally using rows and columns.



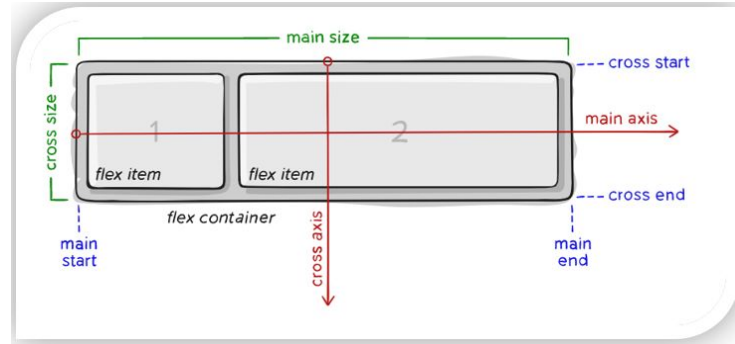
- The flexbox makes the website/web app behave in a similar manner even when they are viewed on different devices that have different screen sizes.
- To create responsive and mobile-friendly websites/web apps flexbox can be used.



- The two main components of flexbox are flex container and flex item.
  - **Flex container:** Property of parent element that is declared with display property flex.
  - **Flex item:** Properties of child elements. There may be one or many flex items.

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

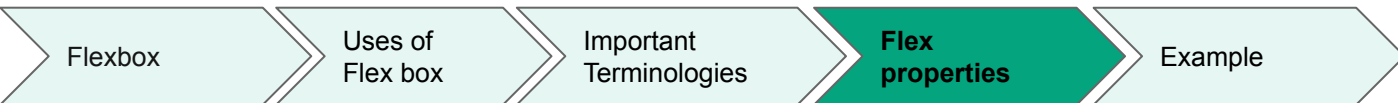
- Flex layout is based on **flex-flow directions**, which are used in displaying the flex items. Flexbox has two axes.
  - **Main axis:** The primary axis along with the flex items are laid out.
  - **Cross axis:** The axis perpendicular to the main axis is called cross axis.



Property	Value and Description	
display	flex	Enables flex context to all direct child of the container.

```
display: flex;
```

Property	Value and Description
flex-direction	<p>Specifies the direction of flex items in the main axis.</p> <ul style="list-style-type: none"><li>• row</li><li>• row-reverse</li><li>• column</li><li>• column-reverse</li></ul>





# Flex properties - Direction



Flex items

row

Flex items



column



Flex items

row-reverse

Flex items



column-reverse

Flexbox

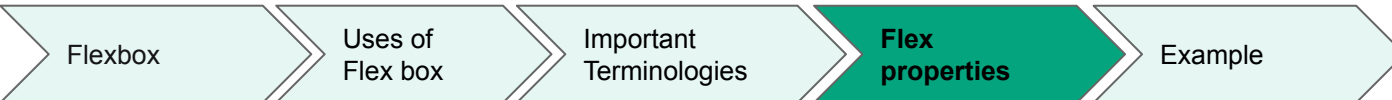
Uses of  
Flex box

Important  
Terminologies

**Flex  
properties**

Example

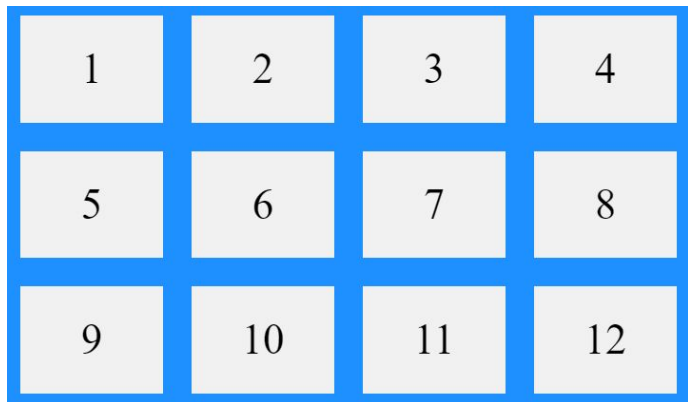
Property	Value and Description
flex-wrap	<p>Specifies whether the flex items should wrap or not when there is not enough space in a line.</p> <ul style="list-style-type: none"><li>• nowrap</li><li>• wrap</li><li>• wrap-reverse</li></ul>



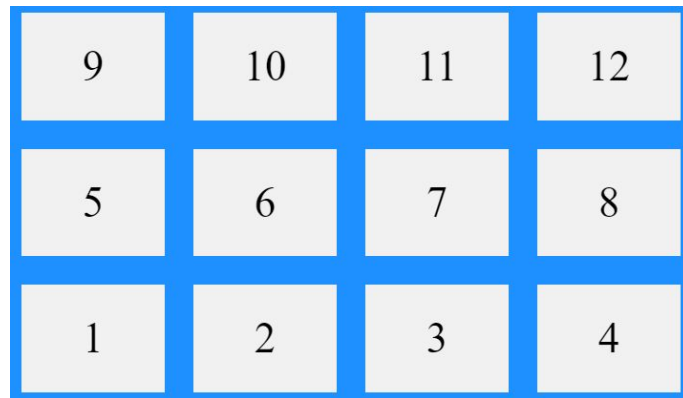
# Flex properties - Wrap



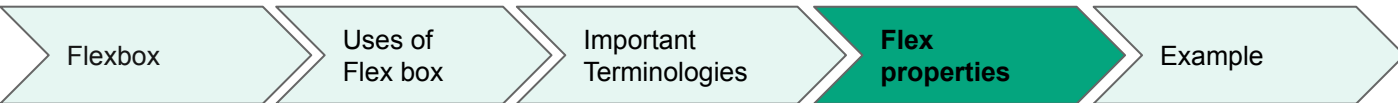
nowrap



wrap

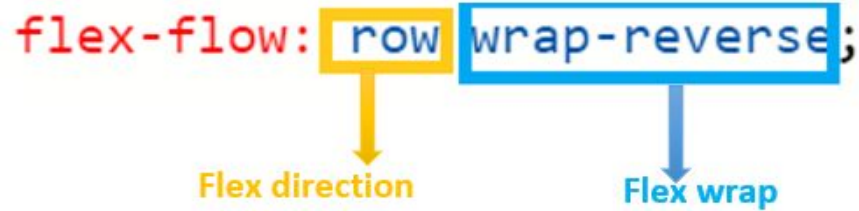


wrap-reverse



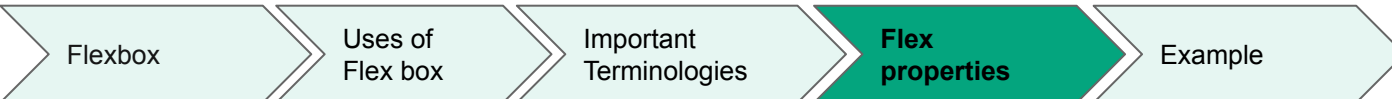
Property	Value and Description
flex-flow	Shorthand property for flex-direction and flex-wrap.

`flex-flow: row wrap-reverse;`

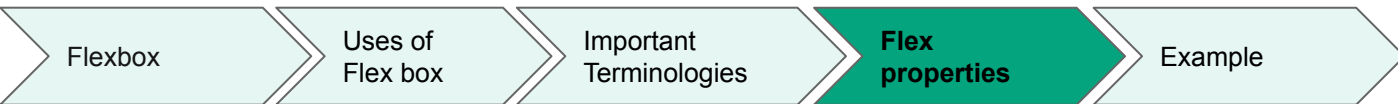


Flex direction

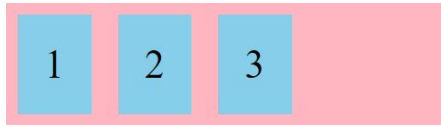
Flex wrap



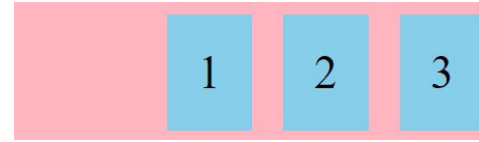
Property	Value and Description
justify-content	<p>To align flex items on the main axis.</p> <ul style="list-style-type: none"><li>• flex-start</li><li>• flex-end</li><li>• center</li><li>• space-between</li><li>• space-around</li><li>• space-evenly</li></ul>



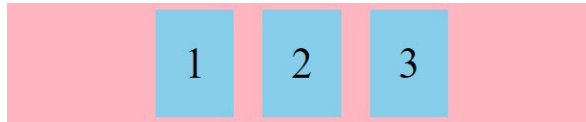
# Flex properties - Justify



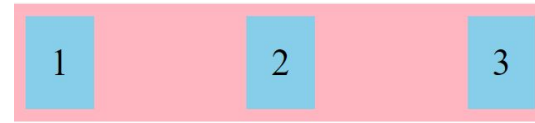
flex-start



flex-end



center



space-between

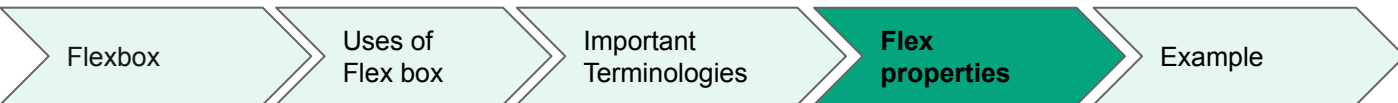


space-around

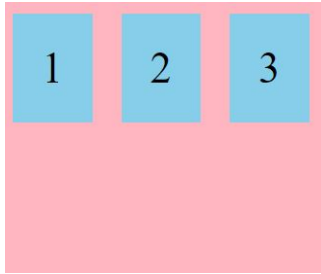


space-evenly

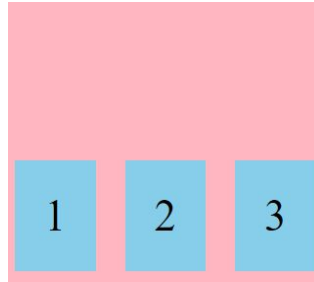
Property	Value and Description
align-items	<p>To align the flex items on the cross axis.</p> <ul style="list-style-type: none"><li>• flex-start</li><li>• flex-end</li><li>• center</li><li>• stretch</li><li>• baseline</li></ul>



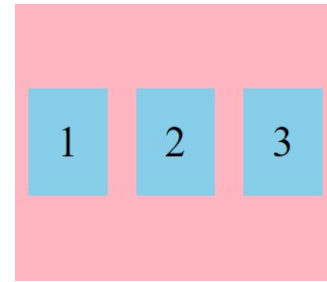
# Flex properties - Align



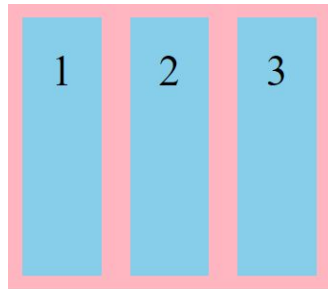
flex-start



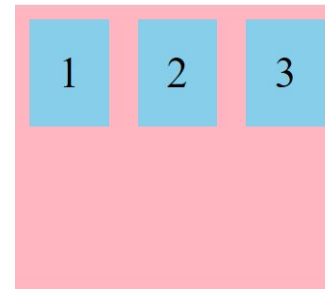
flex-end



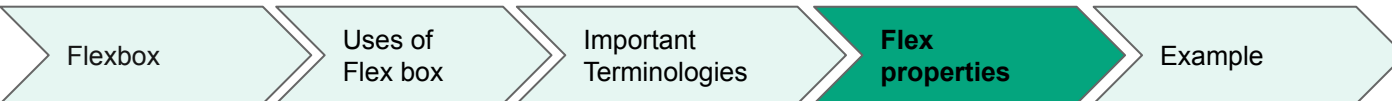
center



stretch

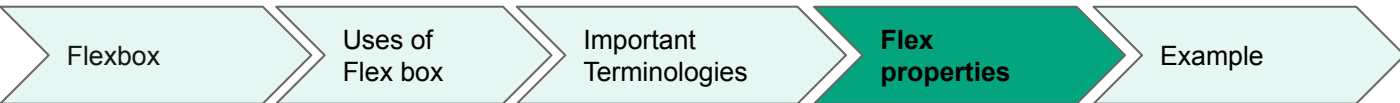


baseline

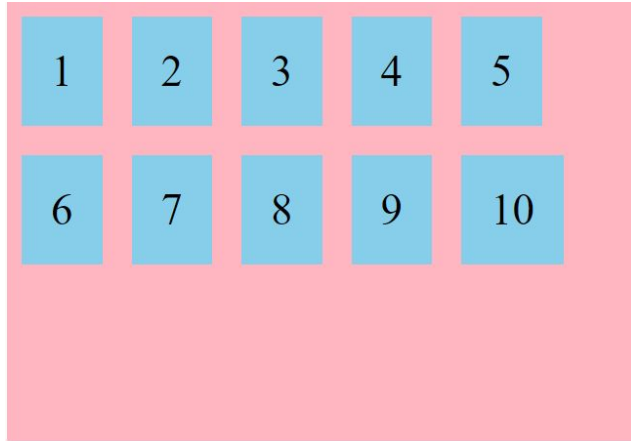




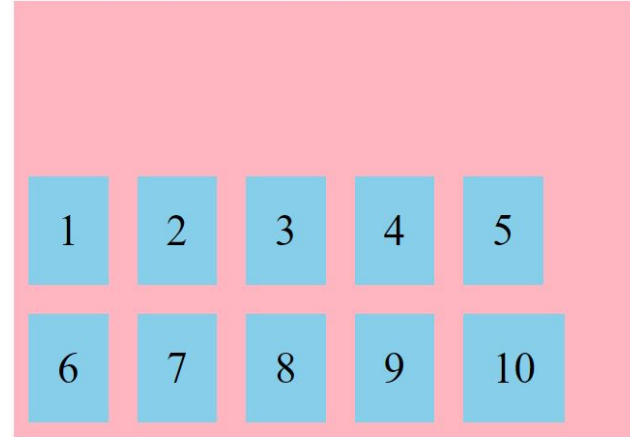
Property	Value and Description
align-content	<p>To align the flex lines. It will be effective only on multi-line flex containers where the flex-wrap property is either wrap or wrap-reverse. Similar to align-items but here the flex lines are aligned.</p> <ul style="list-style-type: none"><li>• flex-start</li><li>• flex-end</li><li>• center</li><li>• stretch</li><li>• space-between</li><li>• space-around</li></ul>



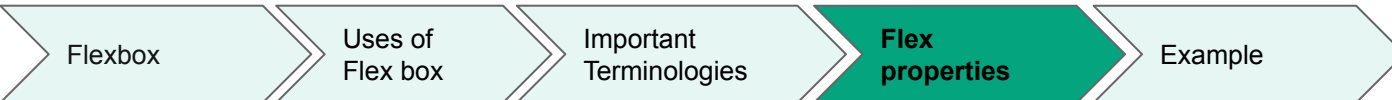
# Flex properties - Align



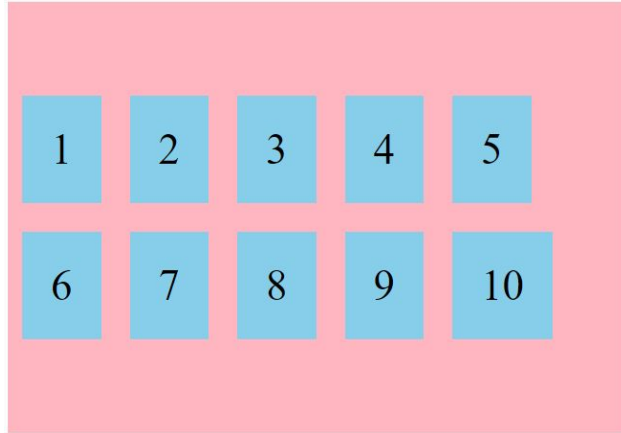
flex-start



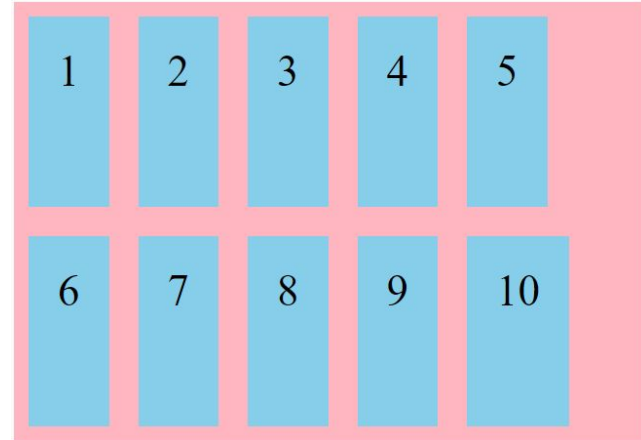
flex-end



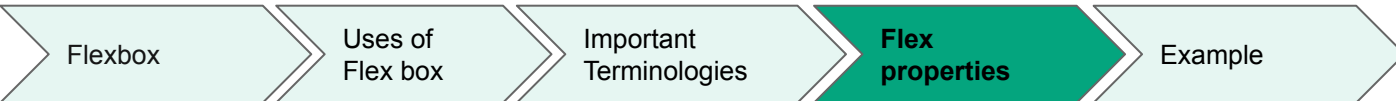
# Flex properties - Align



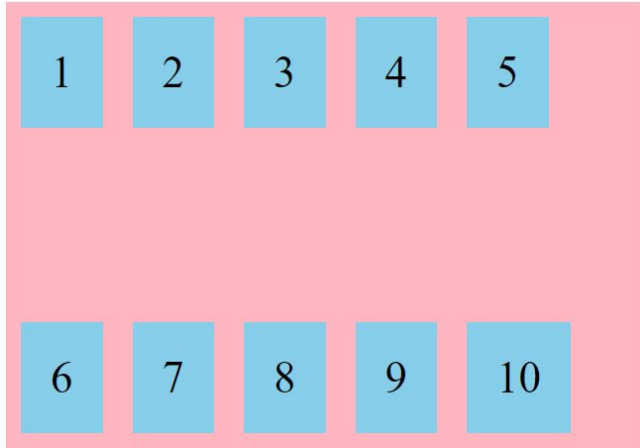
center



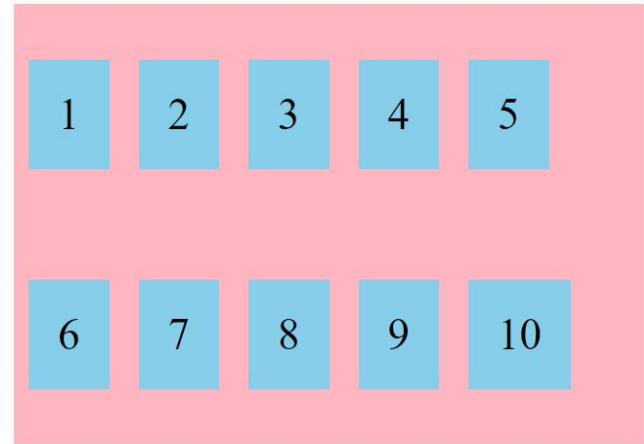
stretch



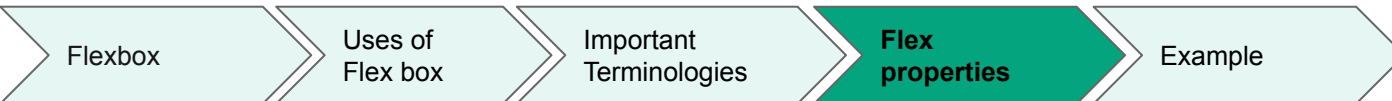
# Flex properties - Align



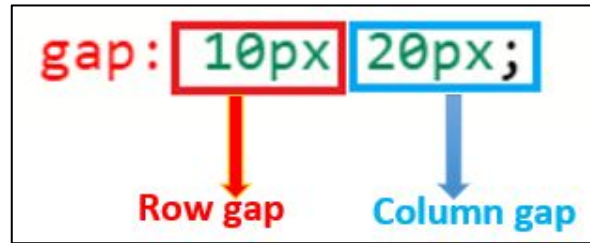
space-between



space-around

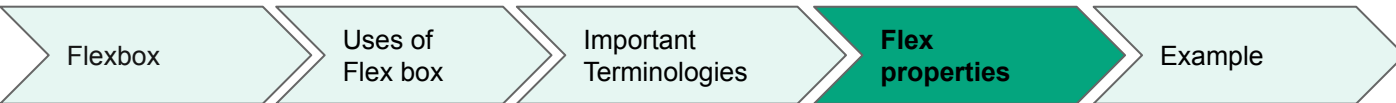


Property	Value and Description
gap	To control space between flex items.



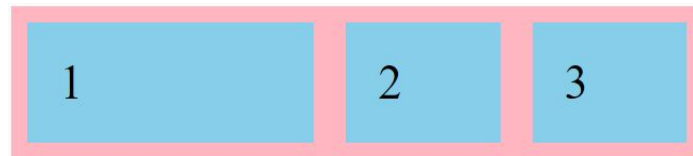
Property	Value and Description
order	<ul style="list-style-type: none"><li>• Value must be a number. Default value is 0.</li><li>• Specifies the order of flex items.</li></ul>

```
<div style="order: 2"> to</div>
<div style="order: 3">CSS Flex</div>
<div style="order: 1">Welcome</div>
```



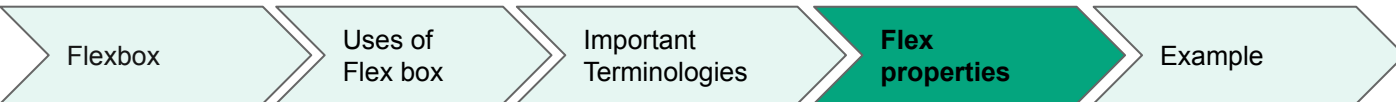
Property	Value and Description
flex-grow	<ul style="list-style-type: none"><li>Value must be a number. Default value is 0.</li><li>To make the flex item grow with respect to given value.</li></ul>

```
<div style="flex-grow: 3">1</div>  
<div style="flex-grow: 1">2</div>  
<div style="flex-grow: 1">3</div>
```



Property	Value and Description
flex-shrink	<ul style="list-style-type: none"><li>Value must be a number. Default value is 1. If the value is given as 0, it won't let the flex item shrink.</li><li>To make the flex item shrink with respect to given value.</li></ul>

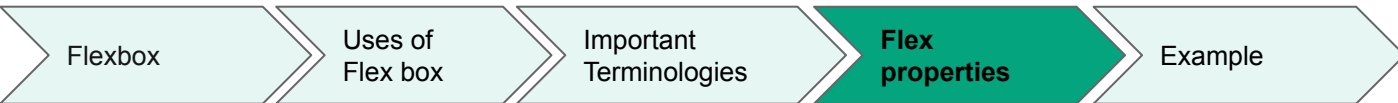
```
<div style="flex-shrink: 8">1</div>
<div style="flex-grow: 1">2</div>
<div style="flex-grow: 1">3</div>
```





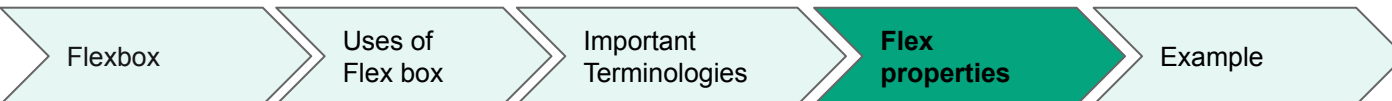
Property	Value and Description
flex-shrink	<ul style="list-style-type: none"><li>Length (px, %, rem, etc...)</li><li>The value will be assigned as the default size of the flex item and the remaining space will be distributed.</li></ul>

```
<div style="flex-basis: 200px">1</div>  
<div>2</div>  
<div>3</div>
```

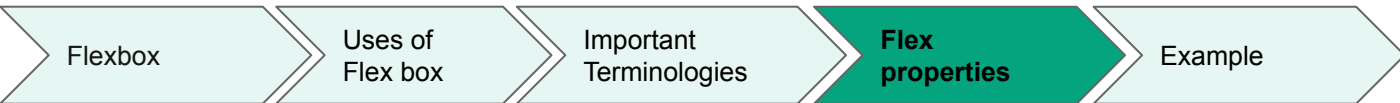


Property	Value and Description
flex	Shorthand property for flex-grow, flex-shrink, flex-basis.

```
<div>1</div>  
<div style="flex: 0 8 0px">2</div>  
<div>3</div>
```

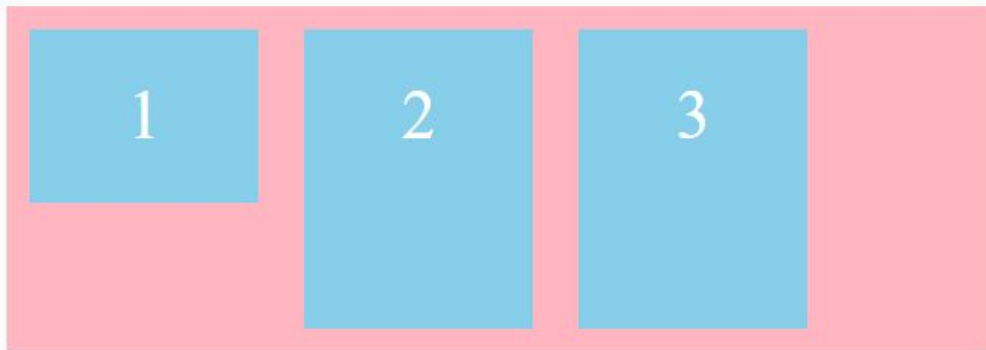


Property	Value and Description
align-self	<p>Specifies the alignment for selected flex items inside the container.</p> <ul style="list-style-type: none"><li>• flex-start</li><li>• flex-end</li><li>• center</li><li>• baseline</li><li>• stretch</li></ul>



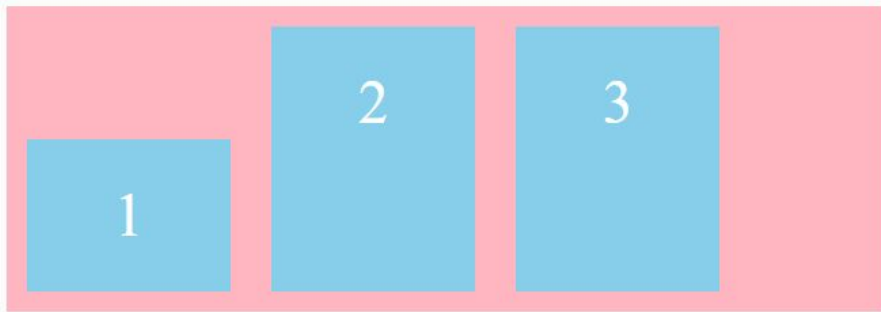
# Flex properties - Self align : flex-start

```
<div style="align-self: flex-start">1</div>  
<div>2</div>  
<div>3</div>
```



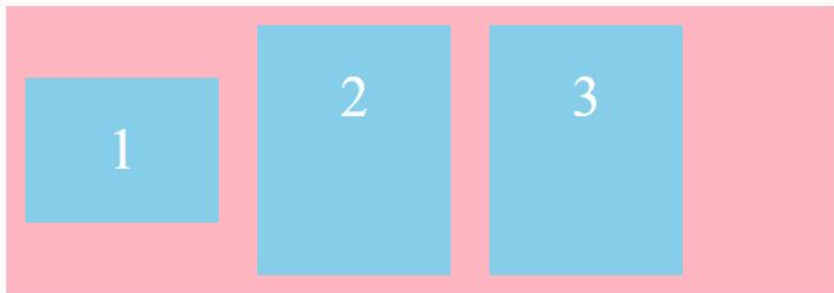
# Flex properties - Self align : flex-end

```
<div style="align-self: flex-end">1</div>  
<div>2</div>  
<div>3</div>
```



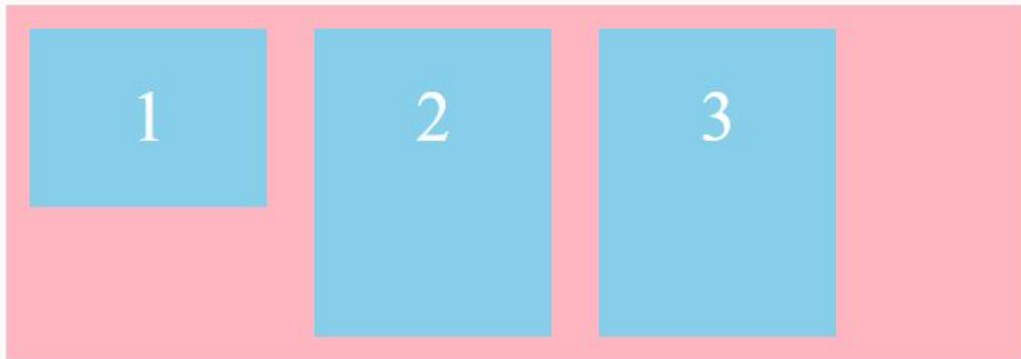
# Flex properties - Self align : center

```
<div style="align-self: center">1</div>  
<div>2</div>  
<div>3</div>
```



# Flex properties - Self align : baseline

```
<div style="align-self: baseline">1</div>  
<div>2</div>  
<div>3</div>
```



Flexbox

Uses of  
Flex box

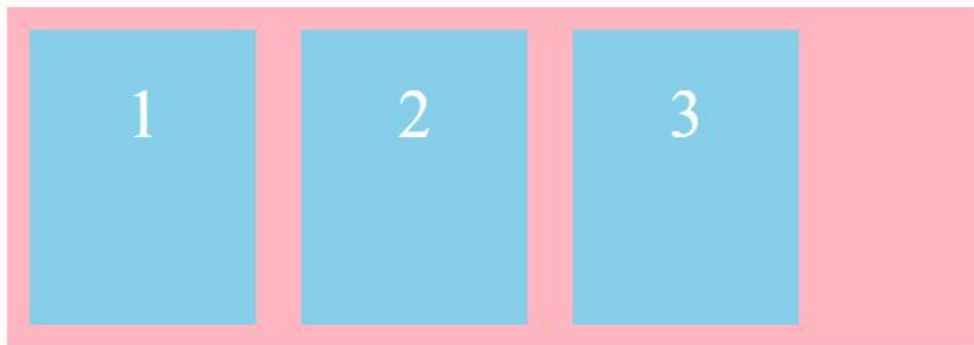
Important  
Terminologies

**Flex  
properties**

Example

# Flex properties - Self align : stretch

```
<div style="align-self: stretch">1</div>  
<div>2</div>  
<div>3</div>
```





```
<body>
  <div class="flex-container">
    <div>1</div>
    <div>2</div>
    <div>3</div>
    <div>4</div>
    <div>5</div>
    <div>6</div>
    <div>7</div>
    <div>8</div>
  </div>
</body>
```

code

```
.flex-container {
  display: flex;
  flex-flow: row wrap;
  justify-content: space-around;
  background: skyblue;
}

.flex-container > div {
  background: lightcoral;
  padding: 5px;
  width: 100px;
  height: 150px;
  margin-top: 10px;
  color: white;
  font-weight: bold;
  text-align: center;
}
```

style

Flexbox

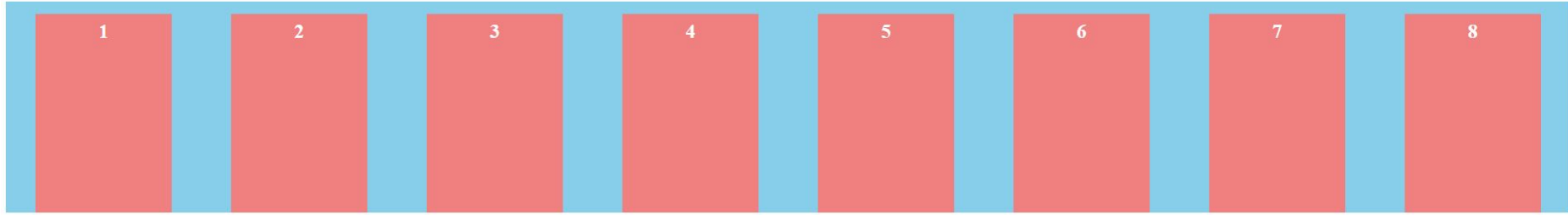
Uses of  
Flex box

Important  
Terminologies

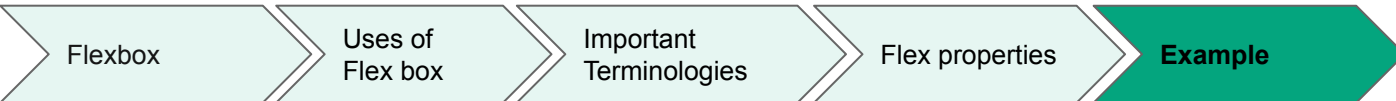
Flex properties

**Example**

# Example



Output



Flex properties in our LinkedIn project

```
.input_blocks{  
  display: flex;  
  flex-direction: row;  
  justify-content: space-around;  
  align-items:center;  
  flex-wrap: wrap;  
}
```

```
.nav_link{  
  box-sizing: content-box;  
  display: flex;  
  flex-direction: column;  
  justify-content: center;  
  align-items: center;  
  color: ■ #6e6558;  
  padding:0px 18px;  
}
```