

COMP 1950

Web Development and Design 2

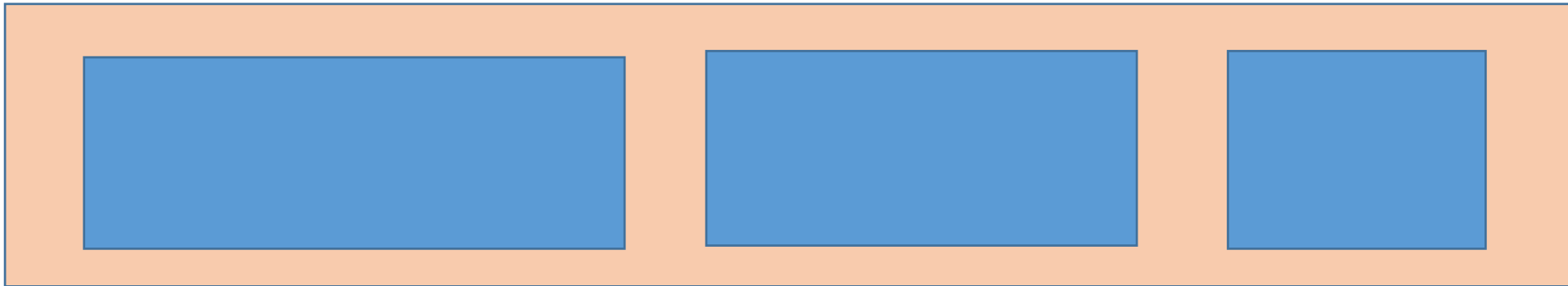
Day 03

Agenda

- Flexbox

What is Flexbox?

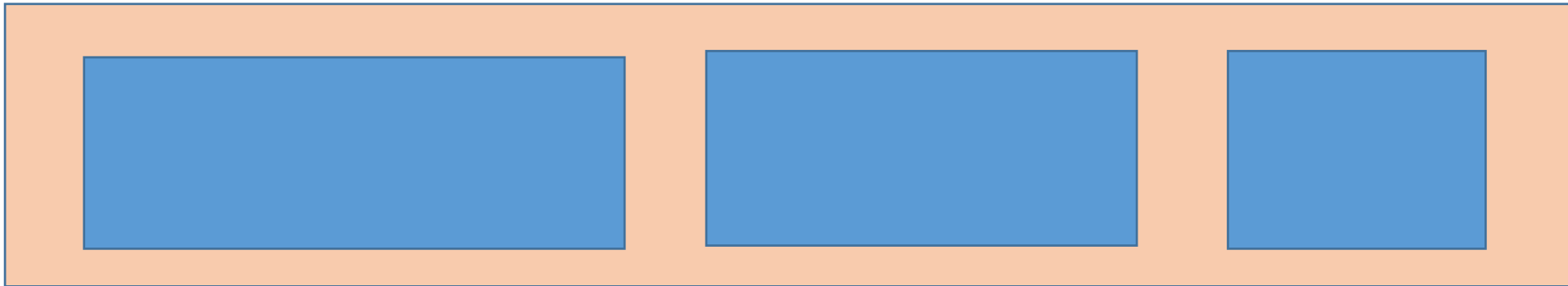
- The Flexbox Layout (Flexible Box) module aims at providing a more efficient way to lay out, align and distribute space among items in a container, even when their size is unknown and/or dynamic (thus the word "flex")¹



1. <https://css-tricks.com/snippets/css/a-guide-to-flexbox/>

What is Flexbox?

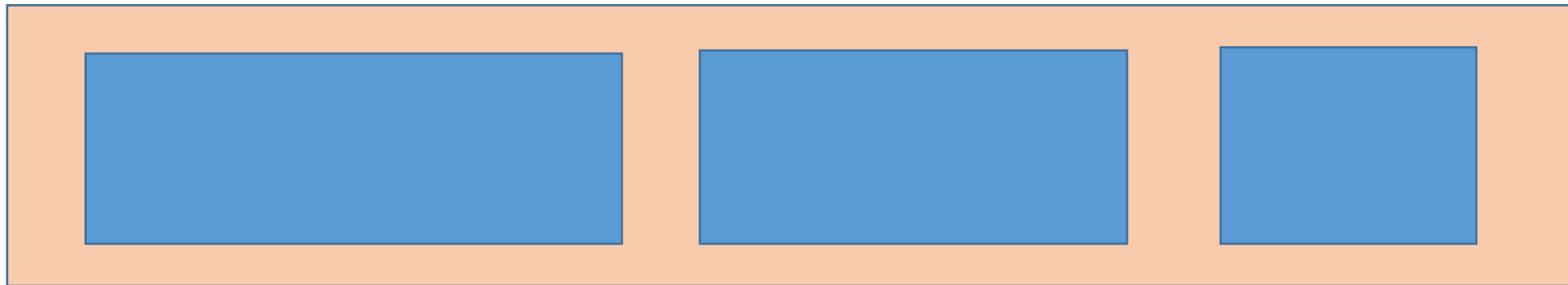
- The main idea behind the flex layout is to give the container the ability to alter its items' width/height (and order) to best fill the available space (mostly to accommodate to all kind of display devices and screen sizes). A flex container expands items to fill available free space, or shrinks them to prevent overflow¹



1. <https://css-tricks.com/snippets/css/a-guide-to-flexbox/>

What is Flexbox?

- The flexbox layout is direction-agnostic as opposed to the regular layouts (block which is vertically-based and inline which is horizontally-based). While those work well for pages, they lack flexibility (no pun intended) to support large or complex applications (especially when it comes to orientation changing, resizing, stretching, shrinking, etc.)¹



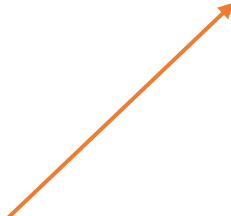
1. <https://css-tricks.com/snippets/css/a-guide-to-flexbox/>

Creating a Flexbox Area?

- To create a flexbox area you simply create a display property on a container element and set its value to "flex"
- Once the container has been set to display: flex then its immediate children become flex-items

```
<div class="container">  
  <div class="box box1">1</div>  
  <div class="box box2">2</div>  
  <div class="box box3">3</div>  
</div>
```

Set parent element to:
display: flex



Direct child elements to become flex-items

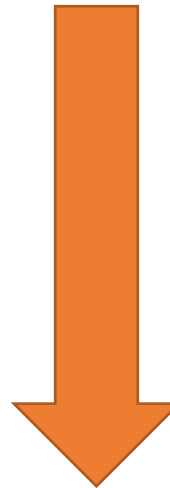


Flex Direction

- flex box layout is direction agnostic. It can be set to have flex-items flow horizontally (row (default)) or vertically (column)
- The order of flex items can even be reversed by setting the flex-direction property to row-reverse or column-reverse



flex-direction: row (default)



flex-direction: column

Flex Direction

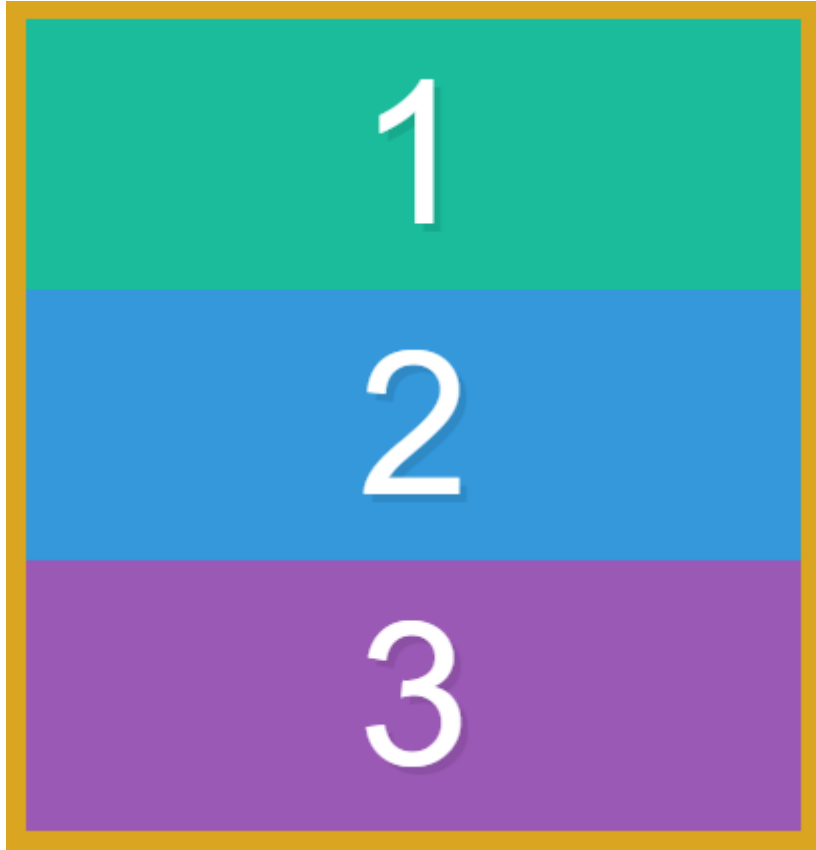


row (default)



row-reverse

Flex Direction



column

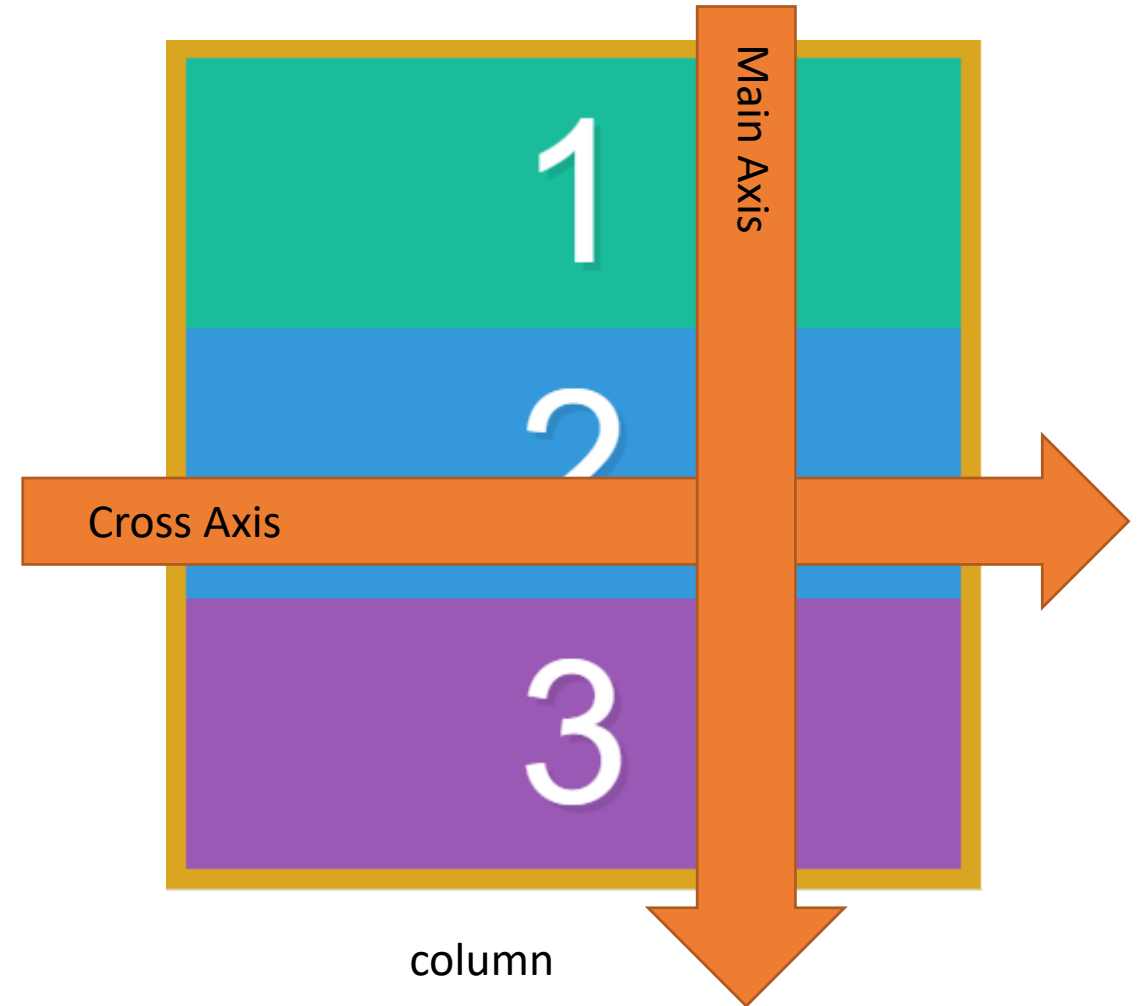
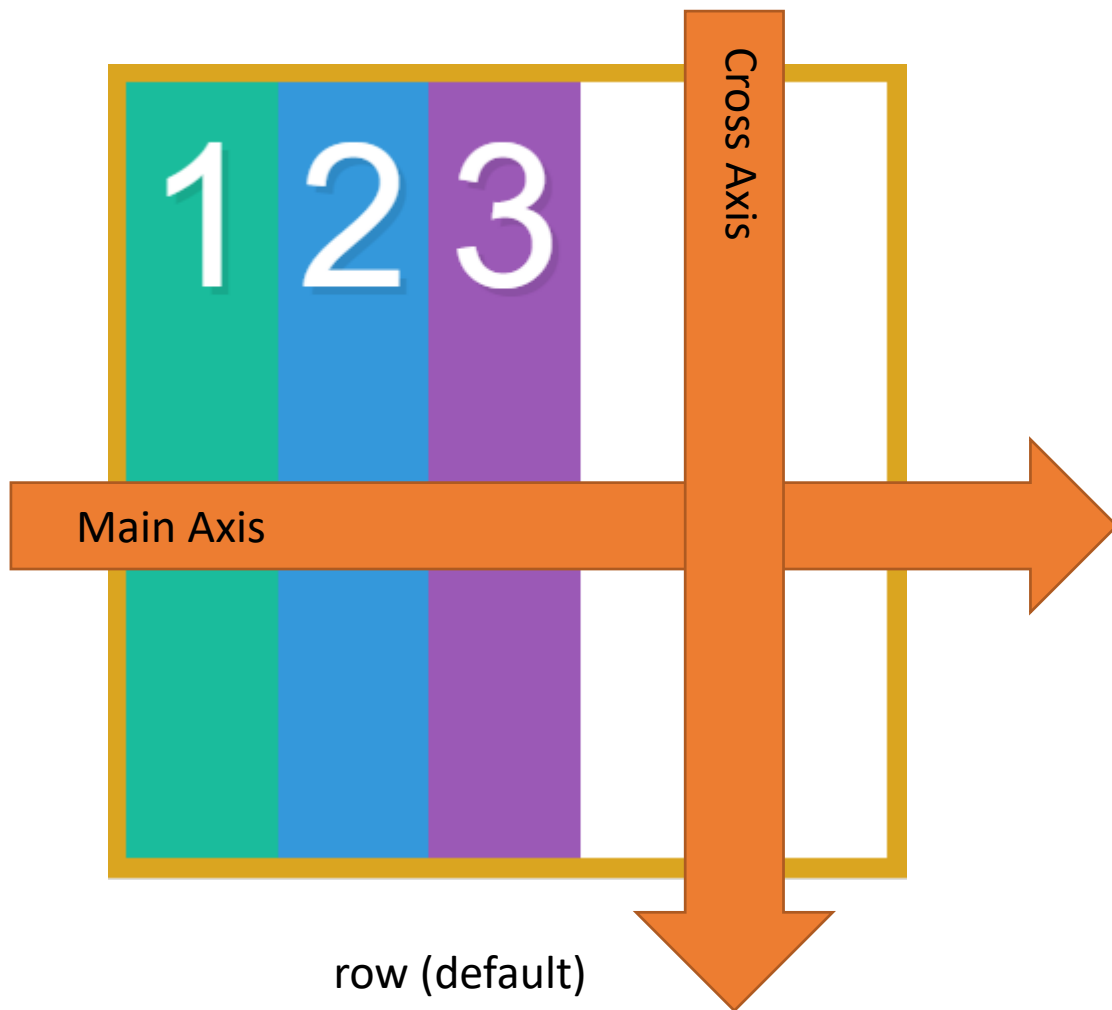


column-reverse

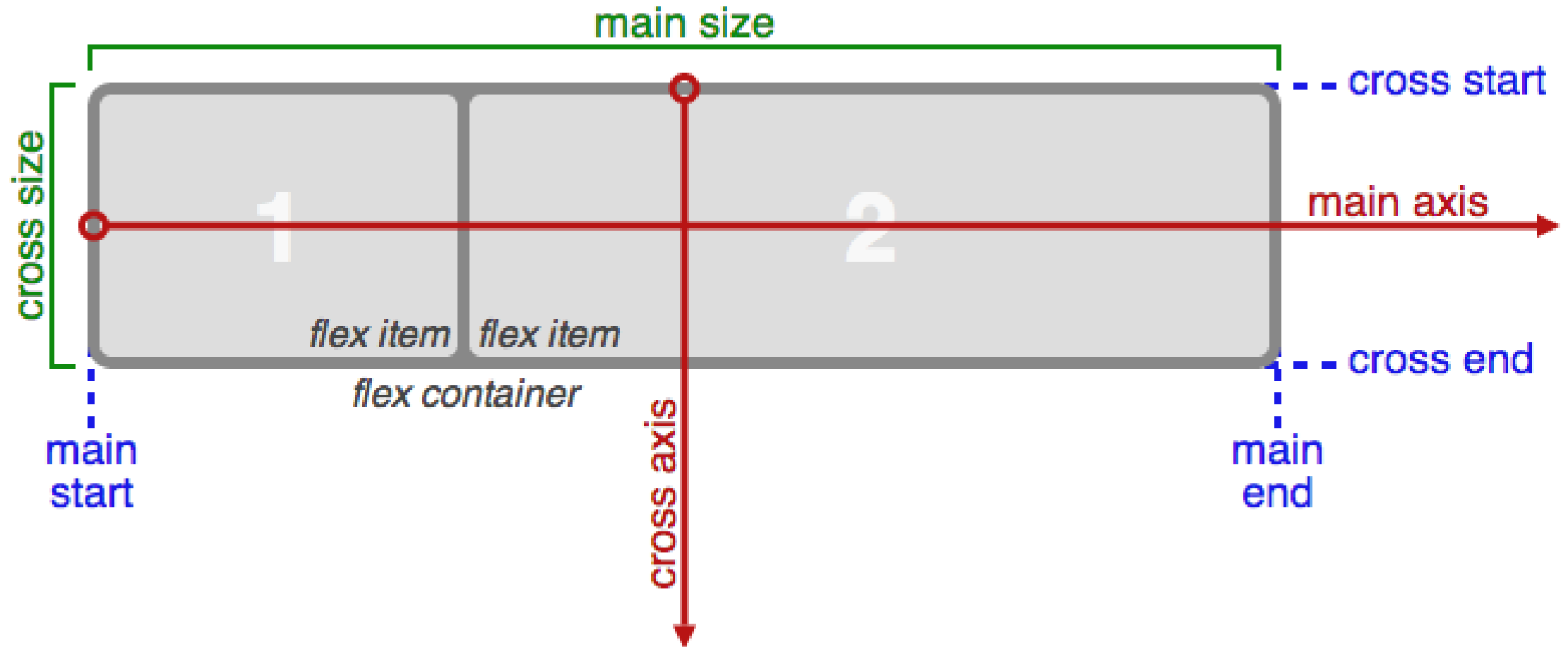
Flex Direction - Main Axis and Cross Axis

- Flex Box has some properties which determine how flex-items are laid out on the main axis and some which determine how flex-items are laid out on the cross axis
- The Flex Direction property determines which way the main axis and cross axis go
- See the next slide for details

Flex Direction - Main Axis and Cross Axis



Flexbox Layout



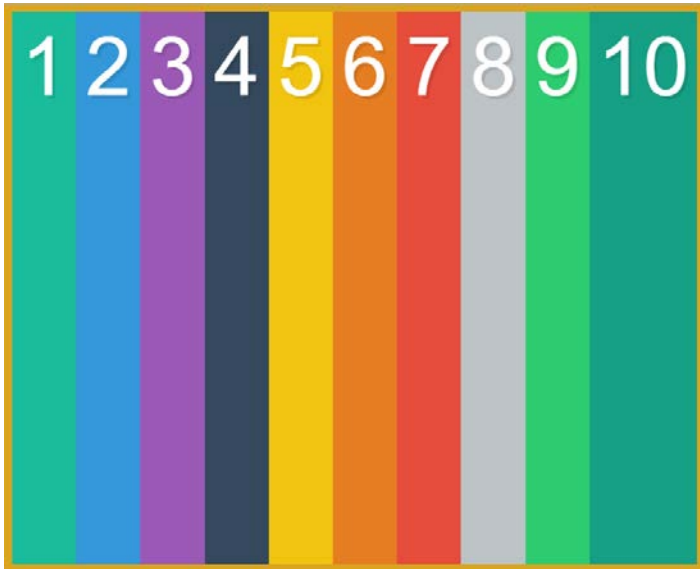
The above image from: <https://css-tricks.com/snippets/css/a-guide-to-flexbox/>

Flex Wrap

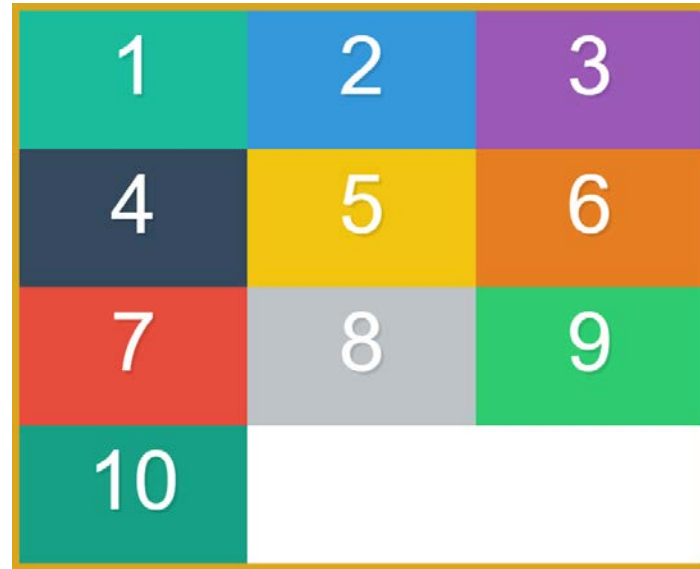
- The flex wrap property determines if the flex items will wrap to a new line if they run out of available space. The default is "nowrap" which means the flex items will shrink to fit on a single line
- See diagram on next page

Flex Wrap

- Each box has a width set at 33.33%. Notice how with the nowrap (default) version the flexbox layout engine shrinks the items to fit on a single line



flex-wrap: nowrap

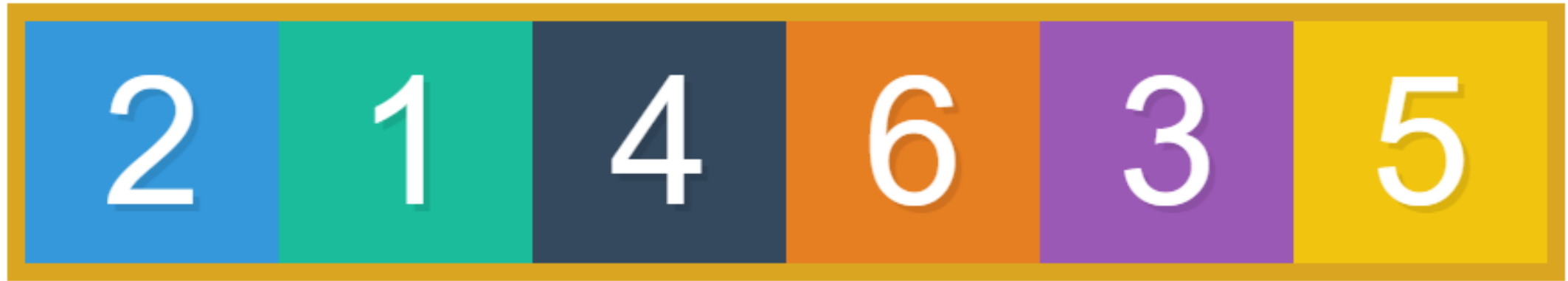


flex-wrap: wrap

Order

- With flexbox you can change the order that flex-items appear on the page by setting the order property
- By default all flex-items have their order property set to "0". So if you set a flex-item to have an order value higher than zero it will appear on the screen after all the other flex-items.
- The order property takes a unit less number. Higher numbers get placed last in the layout and lower numbers get displayed first
- Negative values are allowed
- See next page for a diagram explaining the order property

Order



order property set to -1

order property at the
default value of "0"

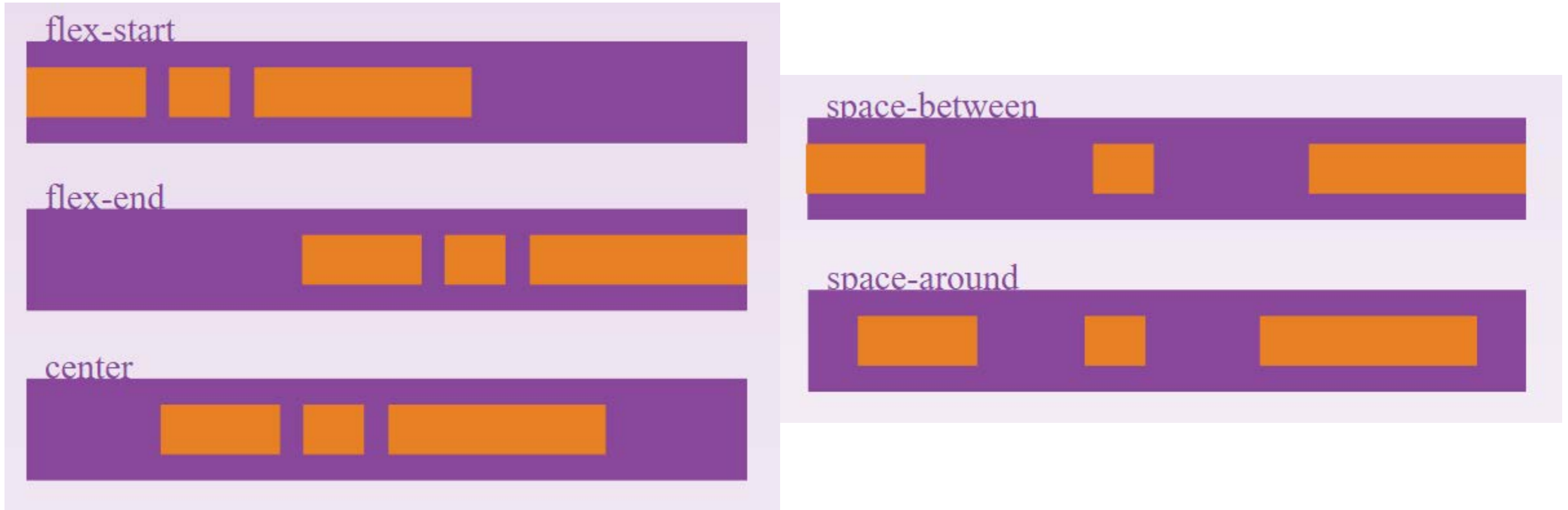
order property set to 1

order property set to 2

Justify Content

- The justify-content property handles how flex-items are aligned along the main axis
- The justify-content property can have 5 different values
 - flex-start
 - Aligns the flex items at the start of the main axis
 - flex-end
 - Aligns the flex items at the end of the main axis
 - center
 - Aligns the flex items in the center of the main axis
 - space-between
 - Places the first item at the start and the last item at the end and divides the space between the other items evenly
 - space-around
 - Evenly puts equal space around each item
- See diagrams on the next page

Justify Content

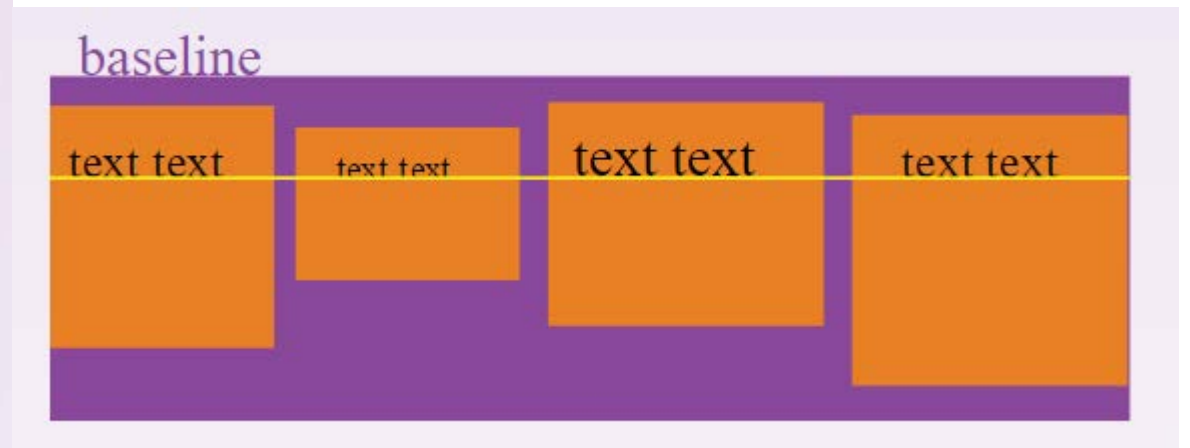
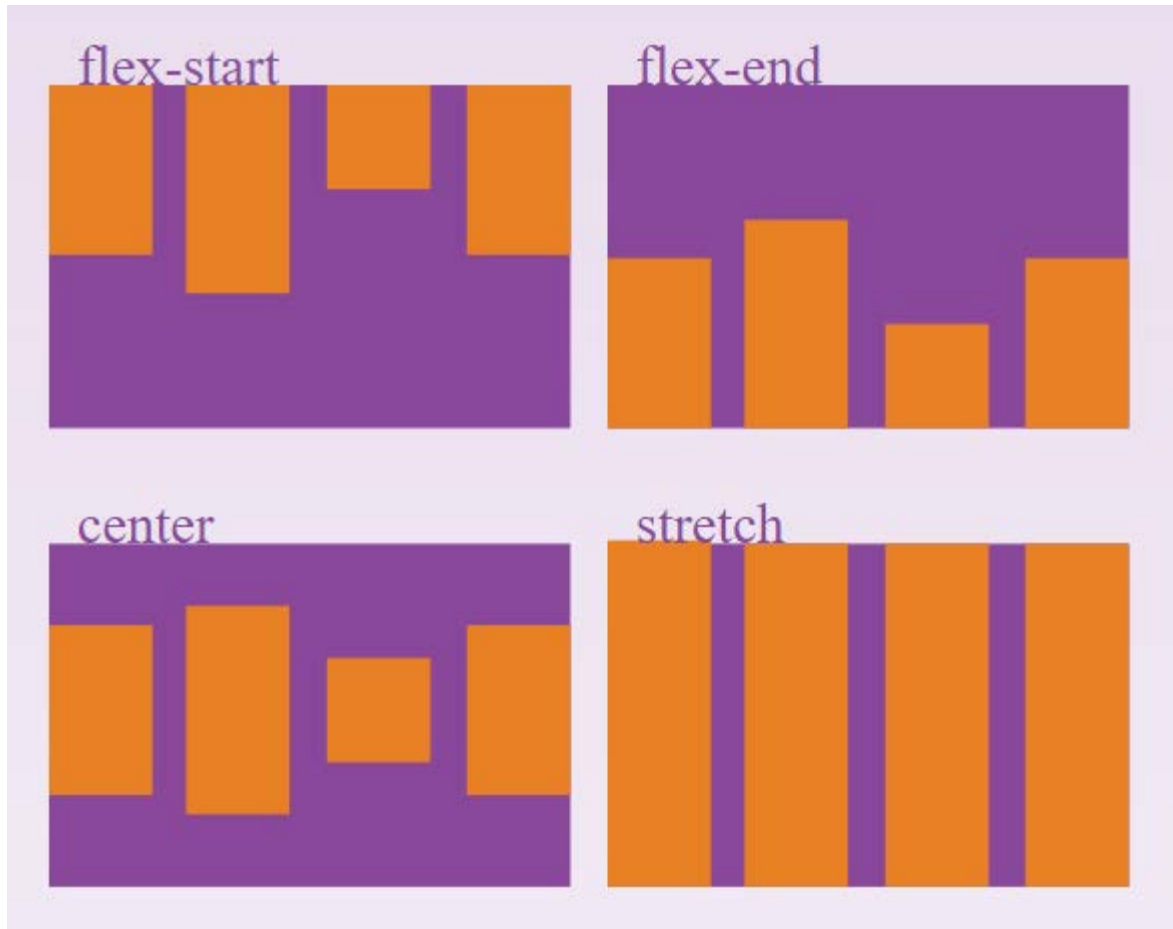


The above images are modified images from: <https://css-tricks.com/snippets/css/a-guide-to-flexbox/>

Align Items

- The align-items property determines how flex-items are aligned along the cross-axis on the current line
- The align-items property can have 5 different values
 - stretch
 - This is the default. All items are stretched to fill all the available space in the cross axis
 - flex-start
 - Flex items start at the start of the cross-axis
 - flex-end
 - Flex items are pushed to the end of the cross-axis
 - center
 - Flex items are placed in the center of the cross axis
 - baseline
 - All the flex items are aligned along the cross axis based on the baseline of their text content
- See Diagram on the next page

Align Items

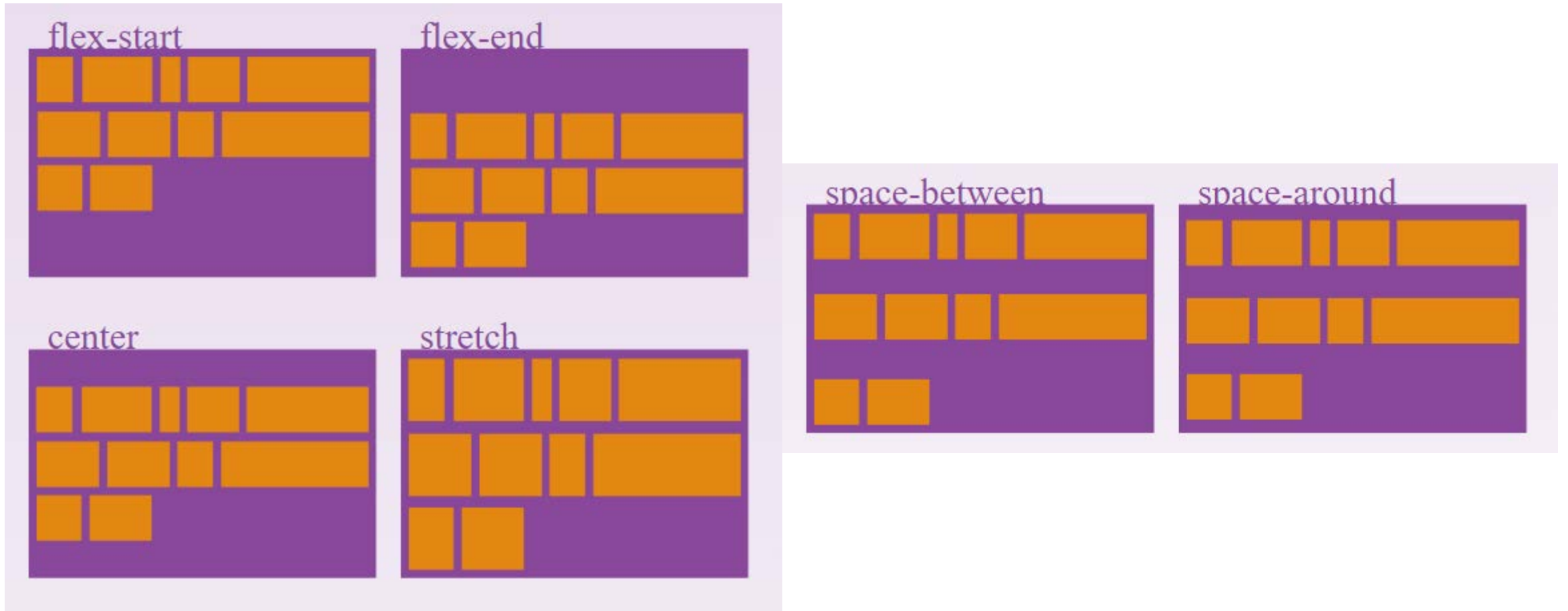


The above images are modified images from: <https://css-tricks.com/snippets/css/a-guide-to-flexbox/>

Align Content

- The align-content property works similarly to the justify-content property except that it works along the cross axis
- For the align-content property to work the flex-wrap property must be set to wrap
- The align content property takes the same values as the justify-content property. The only difference being it applies those values to the cross-axis
- See diagram on next page

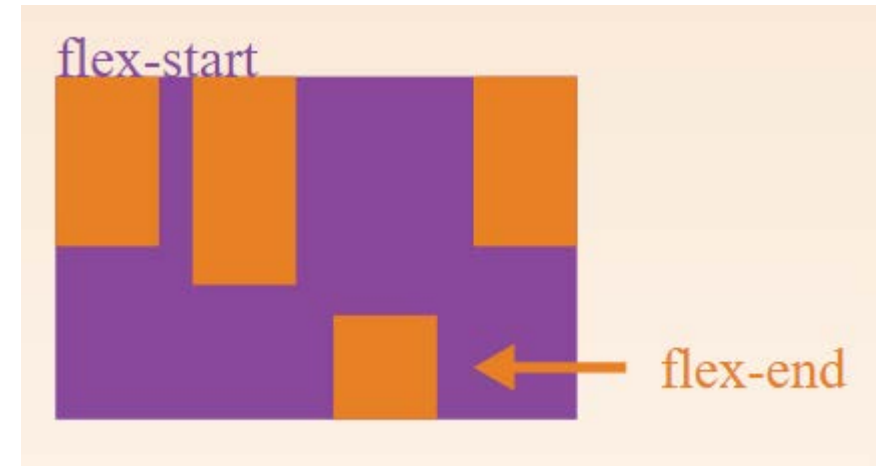
Align Content



The above images are modified images from: <https://css-tricks.com/snippets/css/a-guide-to-flexbox/>

Align Self

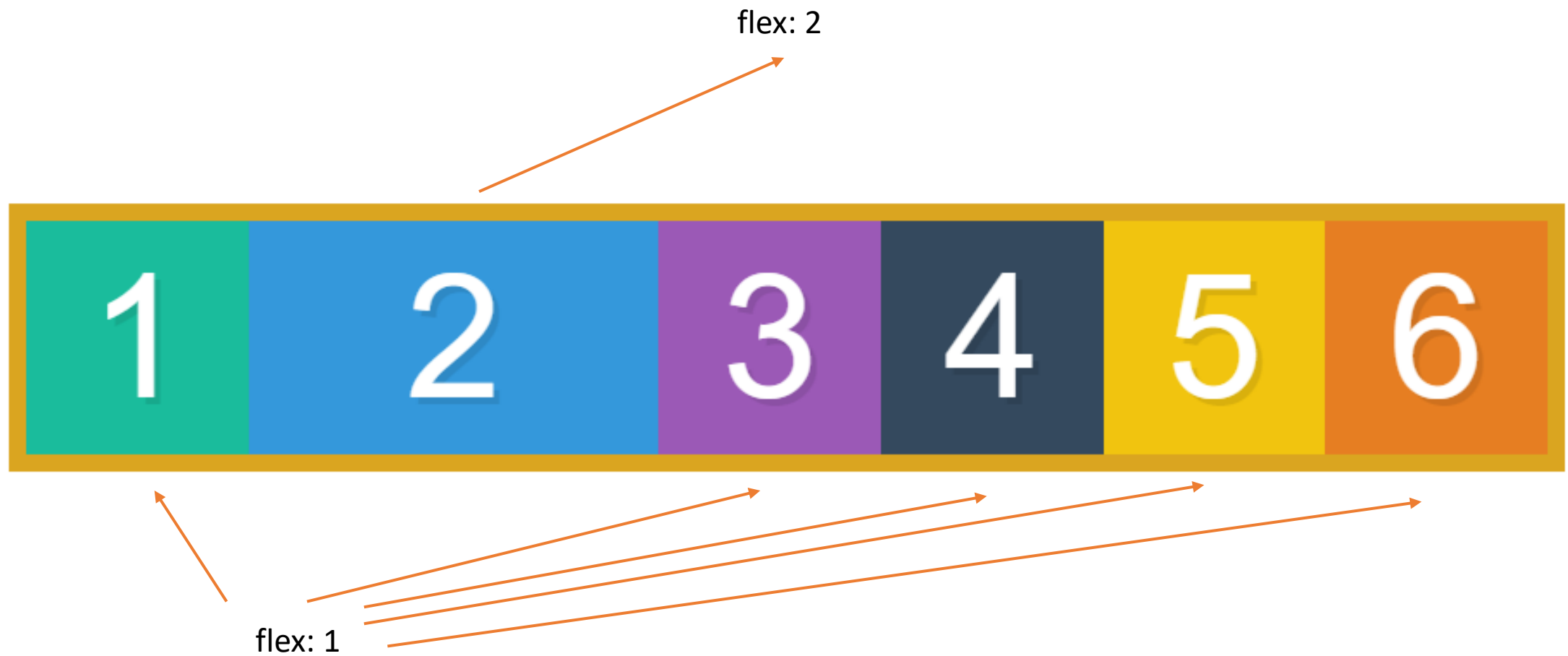
- The align-self property is set on a flex item and not on a flex container
- The align-self property allows you to override the align-items property on individual flex-items
- The align-self property takes the same values as align-items



Flex Property

- The flex property is set on flex items
- The flex property handles how to divvy up any extra white space between all the flex items or how shrink the items if there is not enough space
- The flex property takes a unit-less number (decimal numbers are fine). The numbers are simply proportional
 - For example
 - box1, box3, box4, box5 and box6 has a flex property set at 1
 - box2 has a flex property set at 2
 - This means that box2 will get twice the amount of space as the other boxes because 2 is twice as large as 1
- See diagram on the next page

Flex Property



Flex Grow, Shrink and Basis

- The flex property can actually take three values
 - The first value sets the grow value
 - This is a proportional value and determines how of the available free space a flex item will take up when there is free space available
 - The second value sets the shrink value
 - This is the opposite to the grow value and determines how much an item will shrink when there is not enough available space
 - The third value sets the basis value
 - The basis determines how space is given to a flex item before any shrinking or growing of a flex item

Flex Grow, Shrink and Basis

```
.box1 {  
  flex: 10 10 300px;  
}
```

flex-grow



The diagram consists of three orange arrows pointing from labels at the bottom to values in the CSS code above. The first arrow points from 'flex-grow' to the first '10'. The second arrow points from 'flex-shrink' to the second '10'. The third arrow points from 'flex-basis' to '300px'.

flex-shrink

flex-basis

Flex Grow, Shrink and Basis

```
.box1 {  
  flex: 10 10 300px;  
}  
  
.box2 {  
  flex: 1 1 300px;  
}
```

