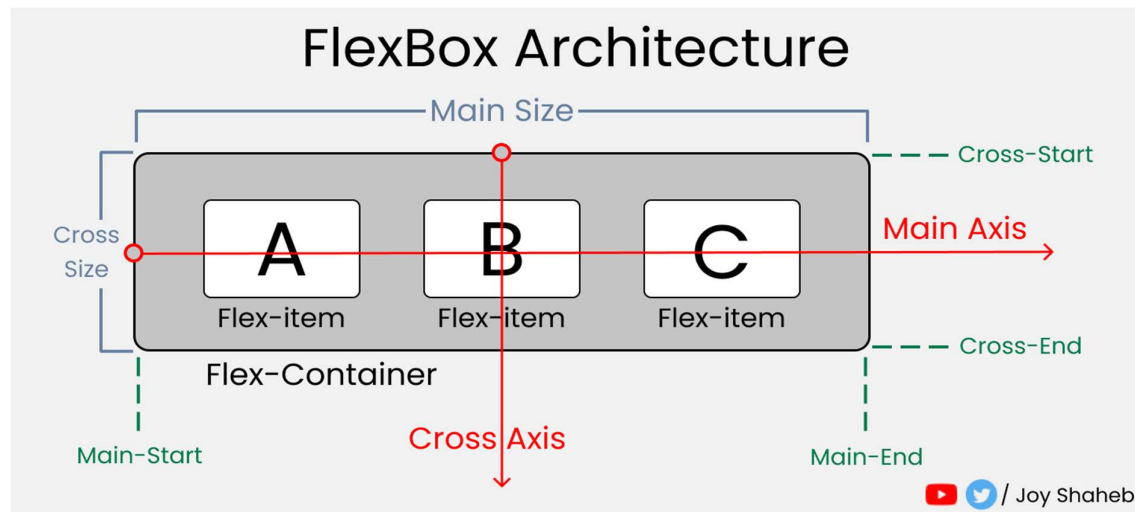


## Flexbox Architecture

So how does Flexbox architecture work?



The flex-items [Contents] are distributed along the **main axis** and **cross axis**. And, depending on the **flex-direction** property, the layout position changes between rows and columns.



## Flexbox Chart

This chart contains every possible property and value you can use when you're working with Flexbox.


Flex Box Chart	
Property	Value(s)
#1 display	flex
#2 flex-direction	row    column    column-reverse    row-reverse
#3 justify-content	flex-start    flex-end    center    space-between    space-around    space-evenly
#4 align-items	flex-start    flex-end    center    stretch    baseline
#5 align-content	flex-start    flex-end    center    stretch    space-between    space-around
#6 align-self	auto    flex-start    flex-end    center    baseline    stretch
#7 Order	/* Any positive Value */
#8 flex-grow	/* Any positive Value */
#9 flex-shrink	/* Any positive Value */
#10 flex-wrap	nowrap    wrap    wrap-reverse


  / Joy Shaheb

## flex-direction property


This property allows us to set the **direction** and **orientation** in which our **flex-items** should be distributed inside the flex-container.


flex-direction :

row → 

row-reverse → 

flex-direction

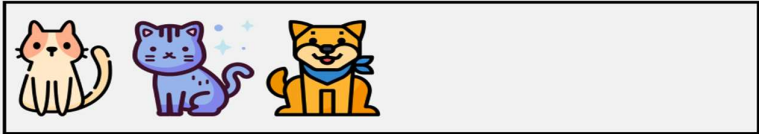
column 

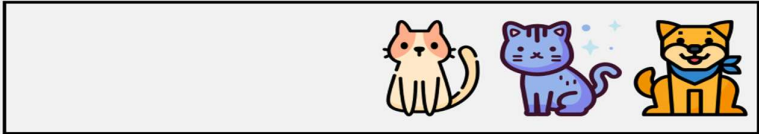
column-reverse 


## justify-content property

This property arranges flex-items along the **MAIN AXIS** inside the flex-container.

justify-content

flex-start 

flex-end 

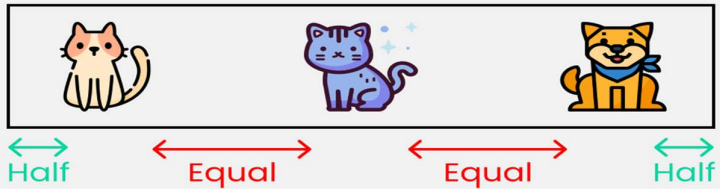
center 

## justify-content

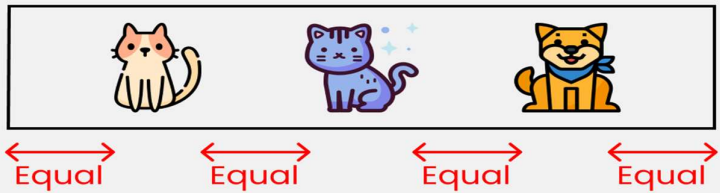
space-between



space-around



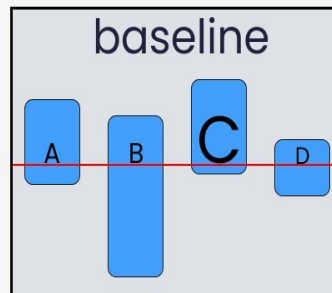
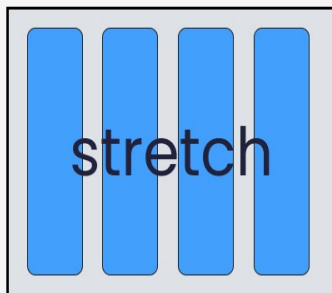
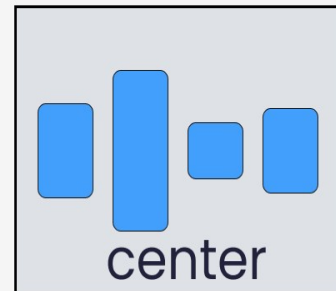
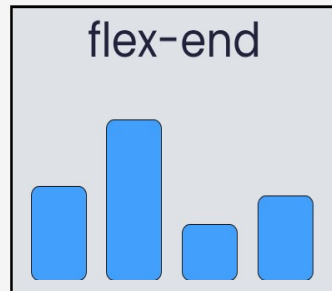
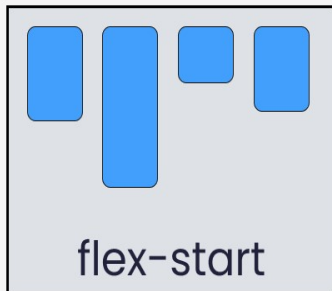
space-evenly



## align-items property

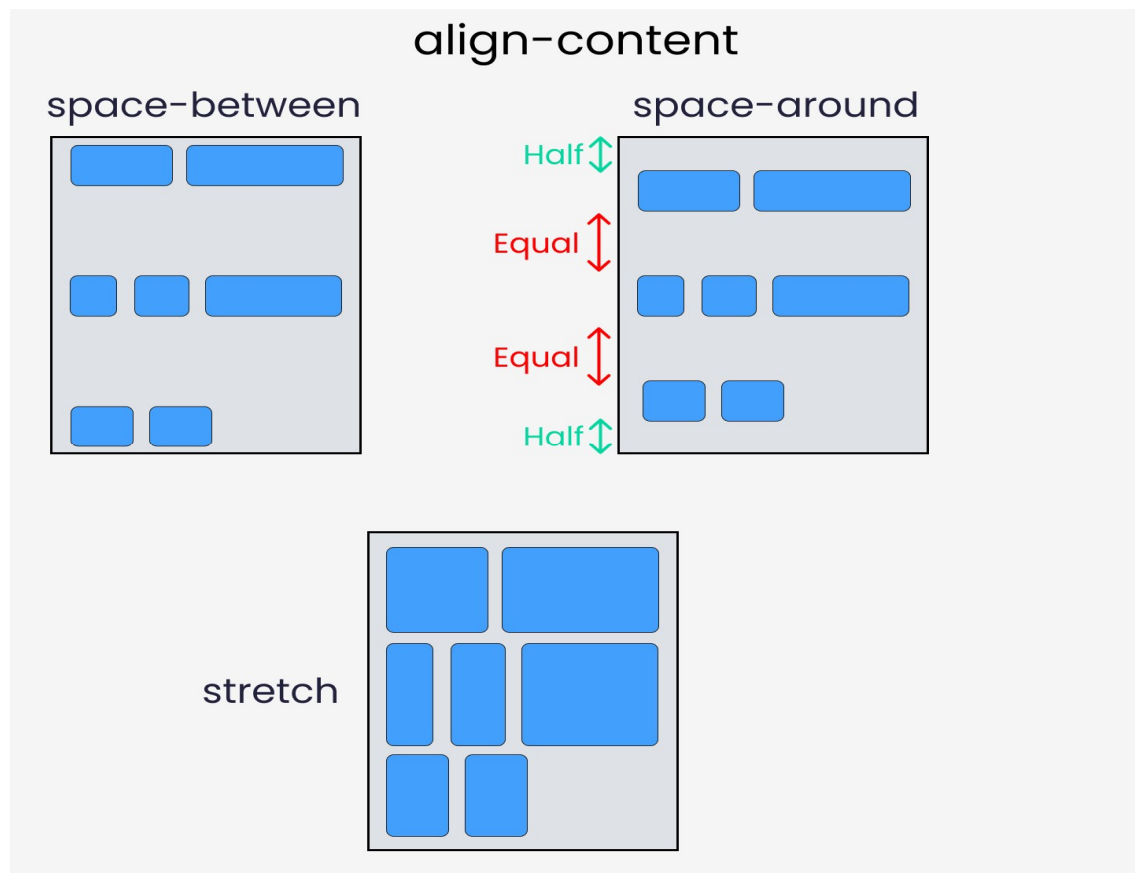
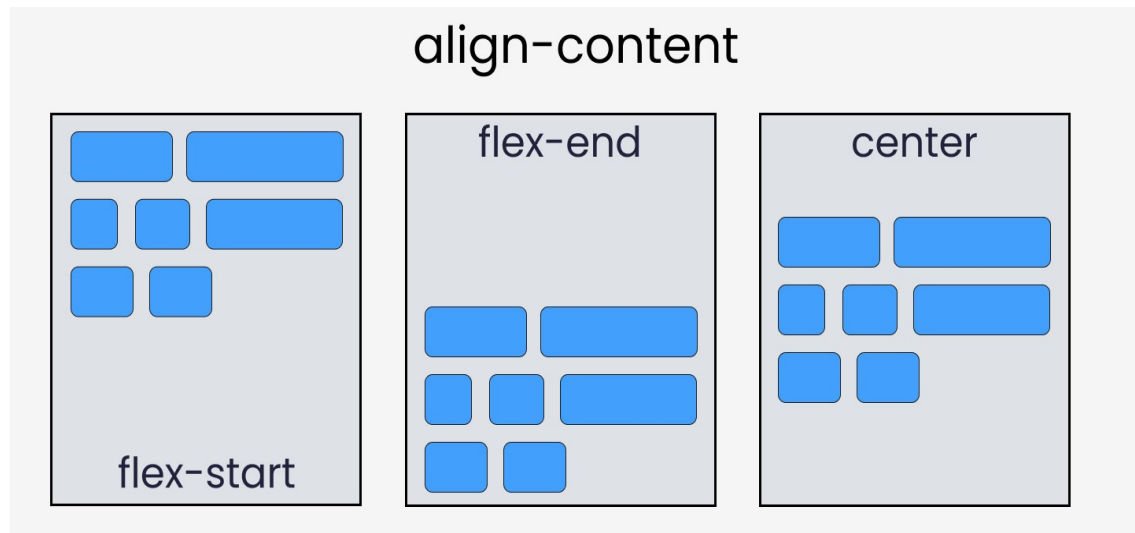
This property distributes Flex-items along the **Cross Axis**.

## align-item



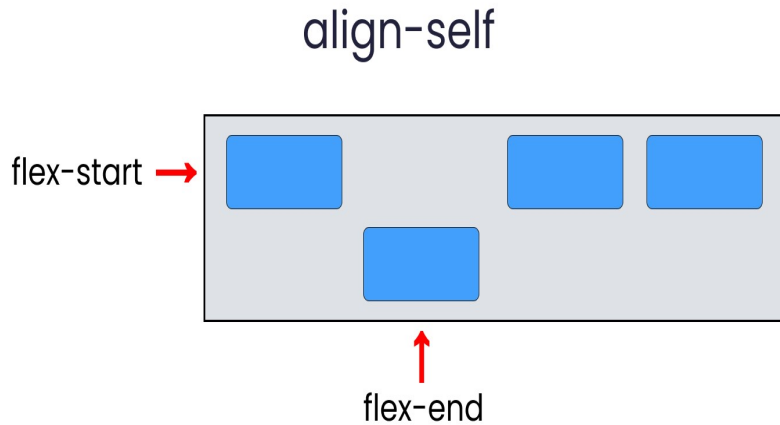
## align-content property

This property arranges flex-items along the **CROSS AXIS** inside the flex-container.



## align-self property

This property works on the child classes. It positions the selected item along the **Cross Axis**.



There are 6 values for this prop.

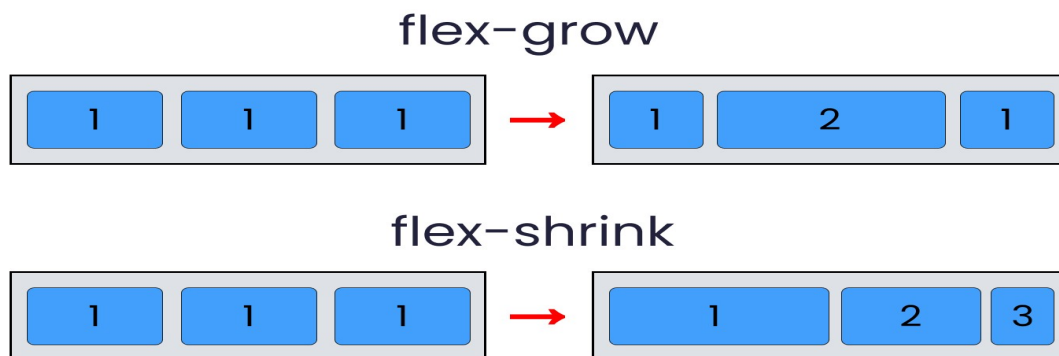
flex-start, flex-end, center, baseline, stretch, auto

## flex-grow

This property grows the size of a flex-item based on the width of the flex-container.

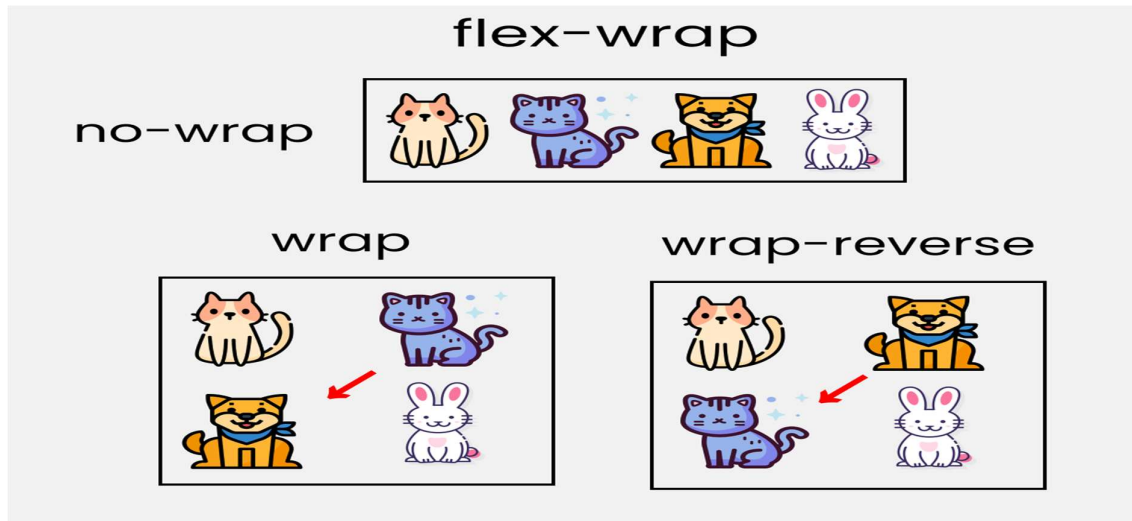
## flex-shrink

This property helps a flex item shrink based on the width of the flex-container. It's the opposite of flex-grow.



## flex-wrap

This property helps you set the number of flex-items you want in a line or row.



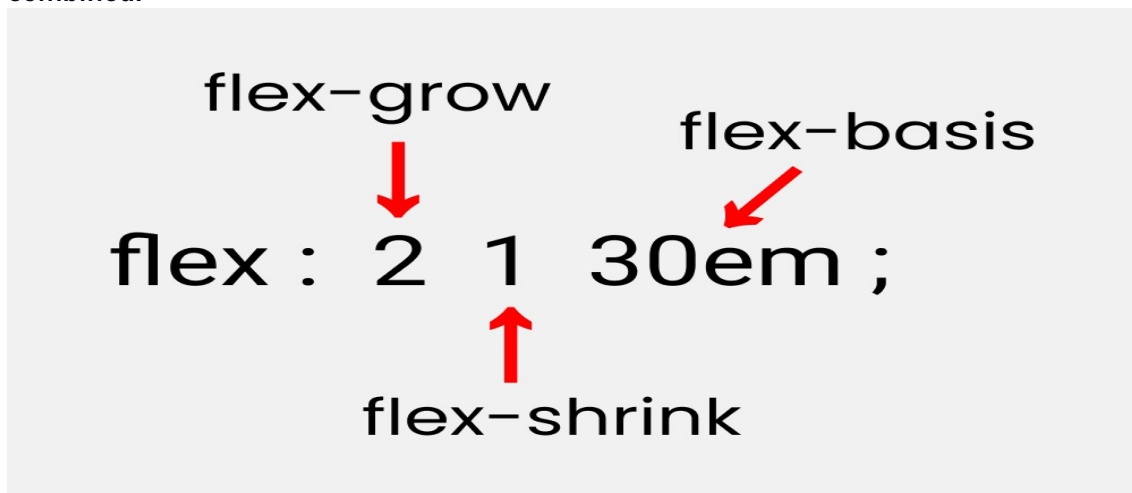
## flex-basis

This is similar to adding width to a flex-item, but only more flexible.

**flex-basis:** 10em, for example, will set the initial size of a flex-item to 10em. Its final size will be based on the available space, flex-grow, and flex-shrink.

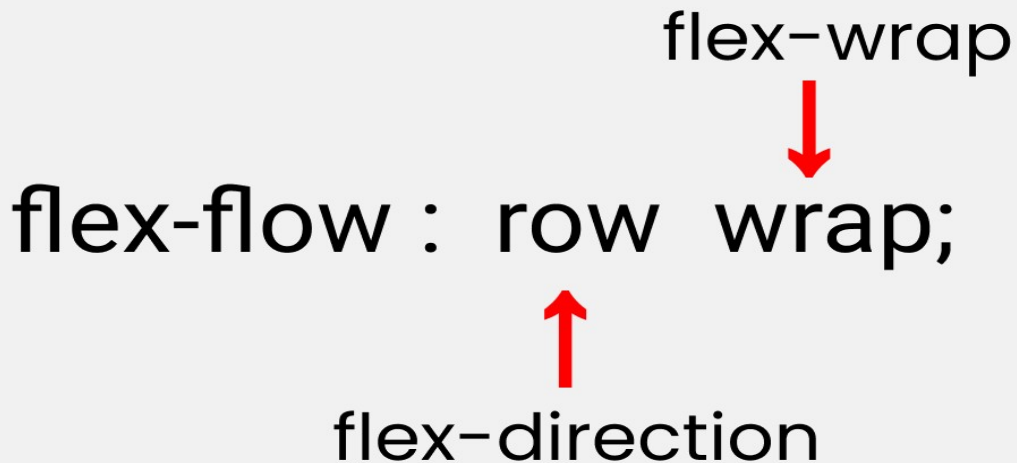
## flex shorthand

This is the shorthand for the **flex-grow**, **flex-shrink** and **flex-basis** properties combined.



## flex-flow

This is the shorthand for the **flex-direction** and **flex-wrap** properties



flex-wrap

flex-flow : row wrap;

flex-direction

The diagram illustrates the shorthand property `flex-flow` with the value `row wrap`. A red arrow points from the text `flex-wrap` above to the word `wrap` in the shorthand. Another red arrow points from the text `flex-direction` below to the word `row` in the shorthand.

## place-content

This is the shorthand for the **justify-content** and **align-content** properties



justify-content

place-content : center flex-end;

align-content

The diagram illustrates the shorthand property `place-content` with the value `center flex-end`. A red arrow points from the text `justify-content` above to the word `flex-end` in the shorthand. Another red arrow points from the text `align-content` below to the word `center` in the shorthand.