

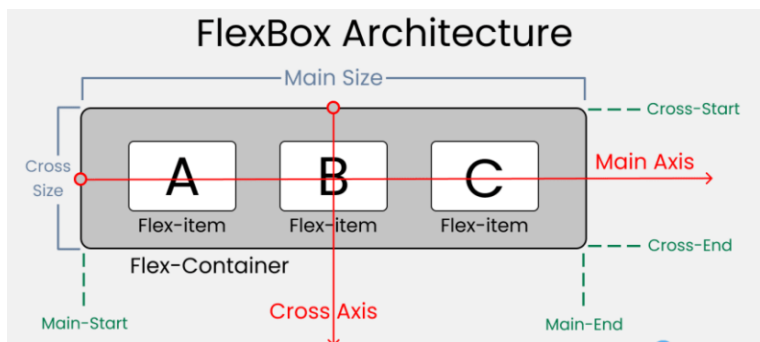
## CSS Flexbox:

Flexbox is a fundamental layout module in CSS that allows you to create flexible and responsive layouts for elements within a container.

Flexbox is a one-dimensional layout system that allows elements to be arranged along a single axis - either horizontally or vertically. The main advantage of using flexbox is its ability to align and distribute elements within a container flexibly and dynamically. Flexbox is particularly useful for creating responsive layouts, where elements need to adapt to different screen sizes and devices.

To use flexbox, you need to apply the `display: flex` property to the container element. This turns the container into a flex container and enables you to use various flexbox properties to manipulate the layout of its child elements.

1. Centering items horizontally and vertically
2. Creating a responsive navigation menu
3. Creating a flexible grid
4. Creating a sticky footer



## Common Flexbox Properties:

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

flex-direction :

row



row-reverse



column

column-reverse

flex-direction



justify-content

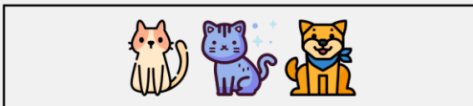
flex-start



flex-end

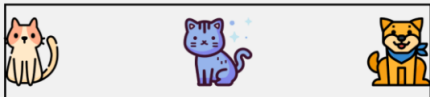


center

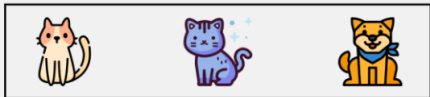


justify-content

space-between

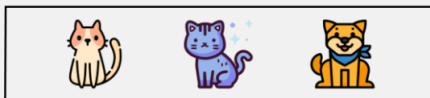


space-around



Half Equal Equal Half

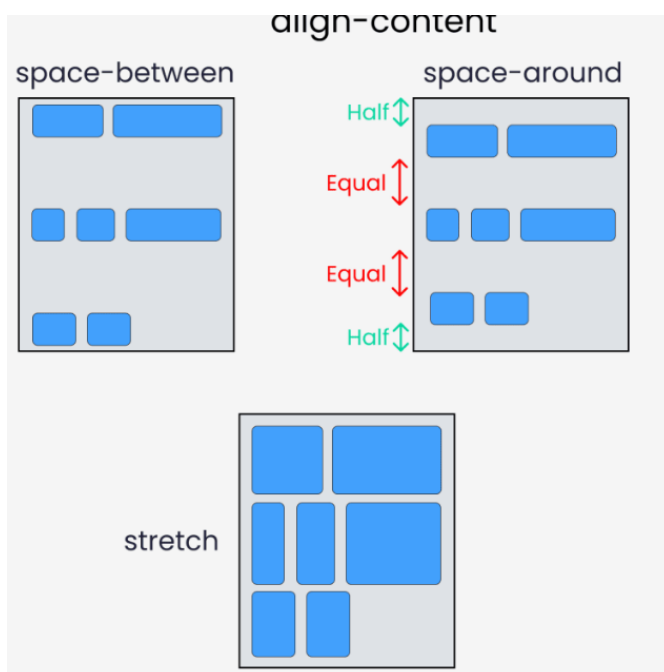
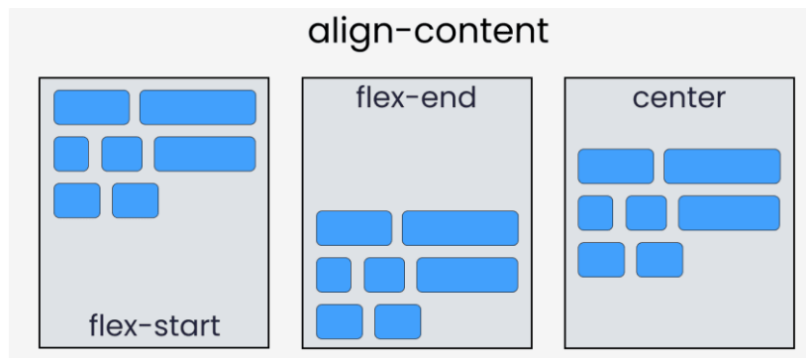
space-evenly



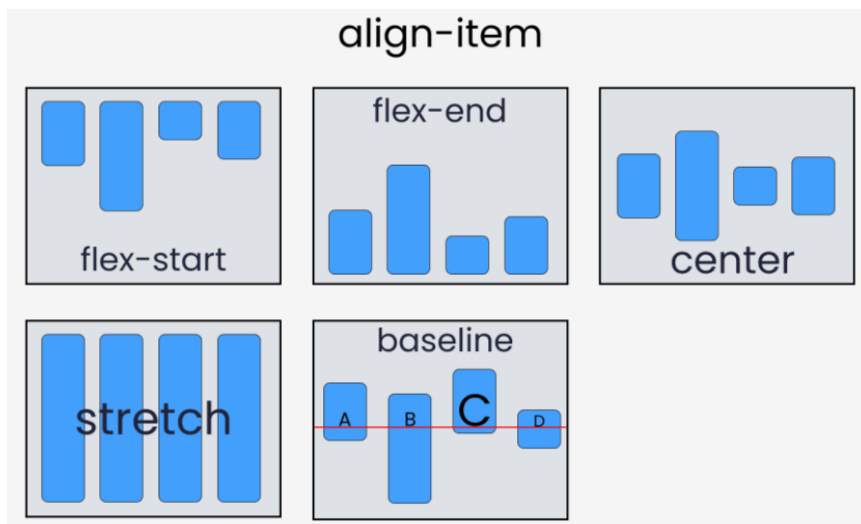
Equal Equal Equal Equal

## Align-Content:

Please note that, without the **flex-wrap** property, the align-content property doesn't work.



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



**flex-grow:** grows the size of a flex-item based on width of the flex-container.

**flex-shrink:** The ability for a flex item to shrink based on width of the flex-container.

Opposite of flex-grow.

flex-grow

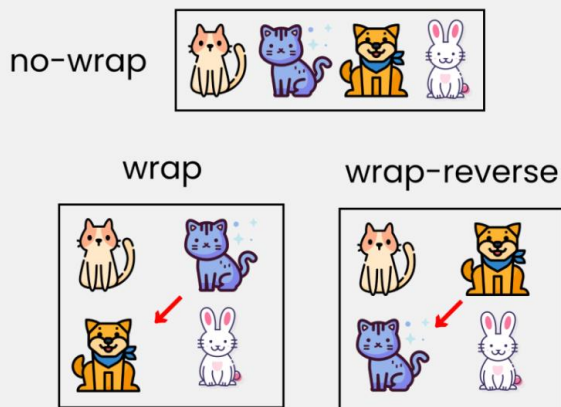


flex-shrink



**flex-wrap :** Amount of Flex-item you want in a line/row.

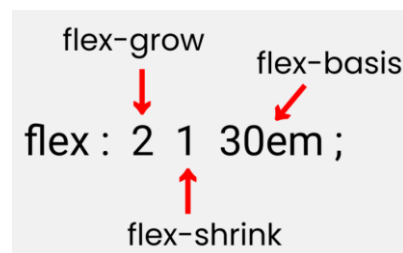
flex-wrap



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

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

**flex :** It is a shorthand to **flex-grow**, **flex-shrink** and **flex-basis** combined.



**@media rule in CSS:**

The @media rule in CSS allows you to apply different styles to your webpage based on different screen sizes and other media types. When combined with CSS Flexbox, @media queries can be used to create responsive layouts that adapt to different device sizes.

```
.container {  
  display: flex;  
  flex-wrap: wrap;  
}  
  
.item {  
  flex: 1 0 25%; /* 4 columns on desktop */  
}  
  
@media screen and (max-width: 767px) {  
  .item {  
    flex-basis: 50%; /* 2 columns on mobile */  
  }  
}
```