

# FLEXBOX IS EASY

Flexbox, short for "Flexible Box," is a layout model in CSS designed for building dynamic and responsive user interfaces.

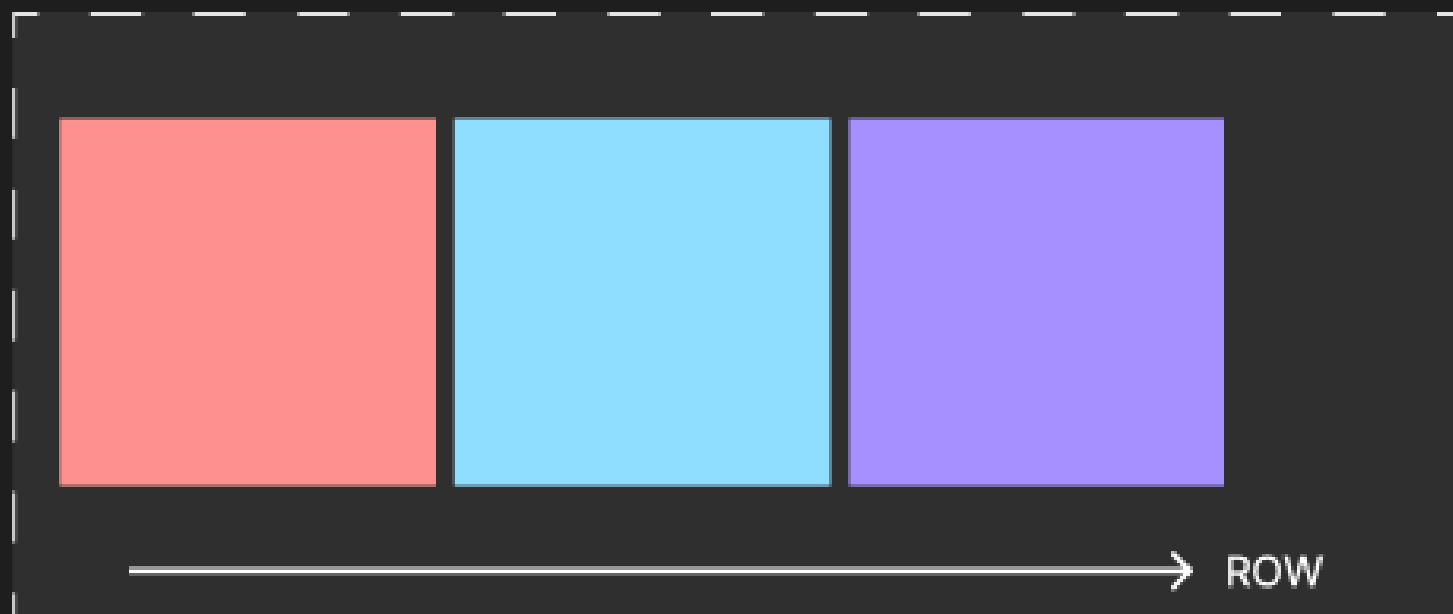


# FLEX-DIRECTION



determining the direction in which its  
flex items are placed

**flex-direction: row**



**flex-direction: column**



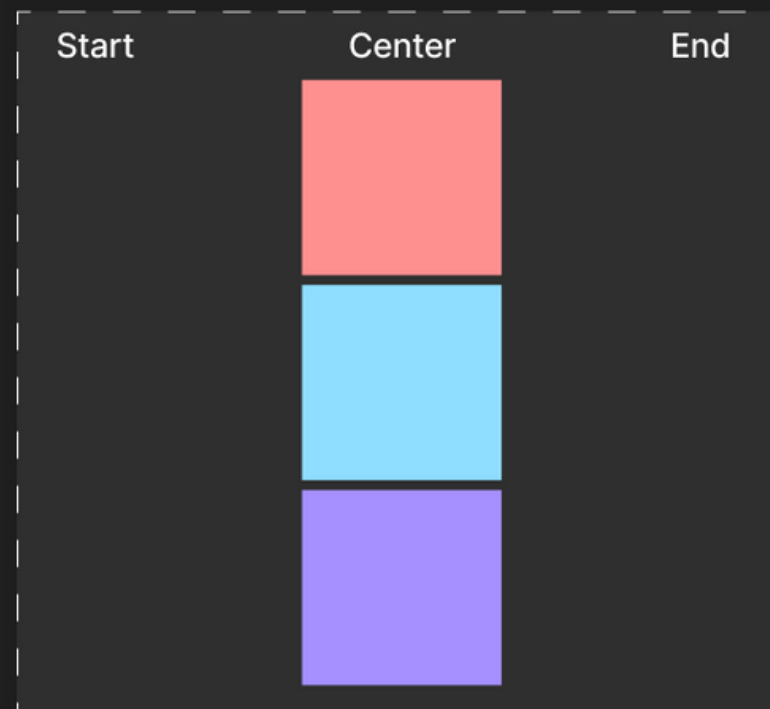
# ALIGN-ITEMS

define how flex items are aligned along  
the **X axis**

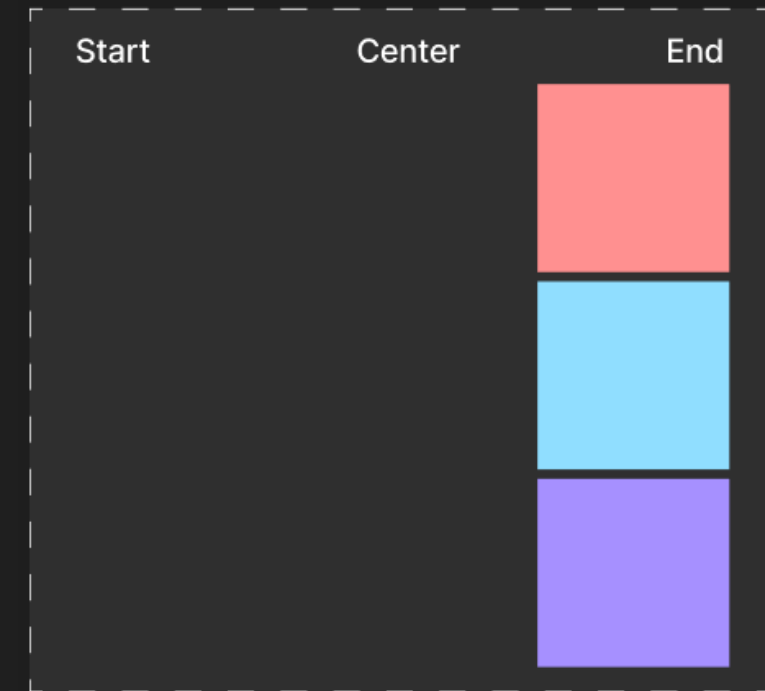
**align-items: start**



**align-items: center**



**align-items: end**

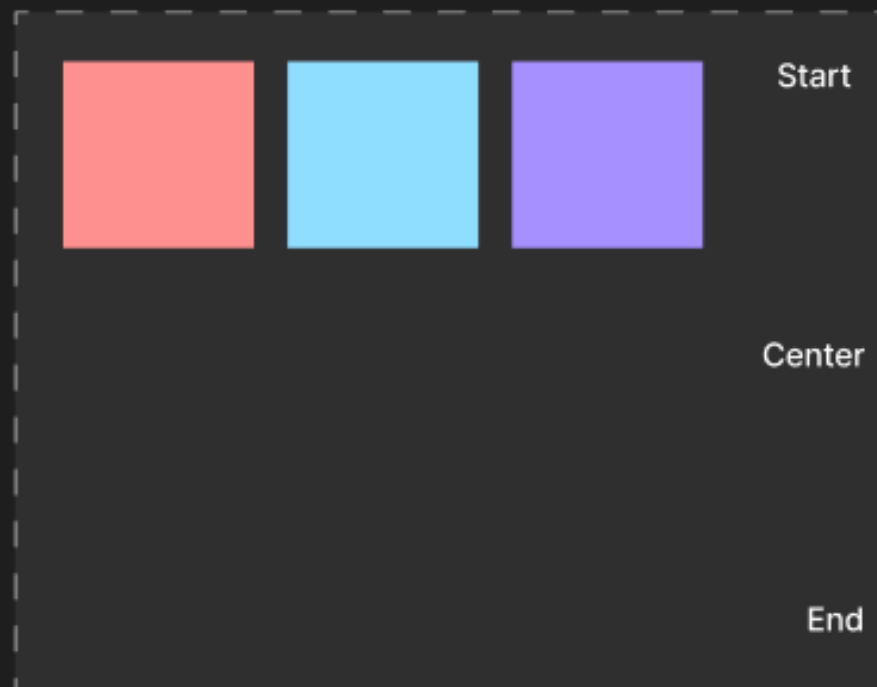


## COLUMN

# ALIGN-ITEMS

define how flex items are aligned along  
the Y axis

`align-items: start`



`align-items: center`



`align-items: end`



## ROW

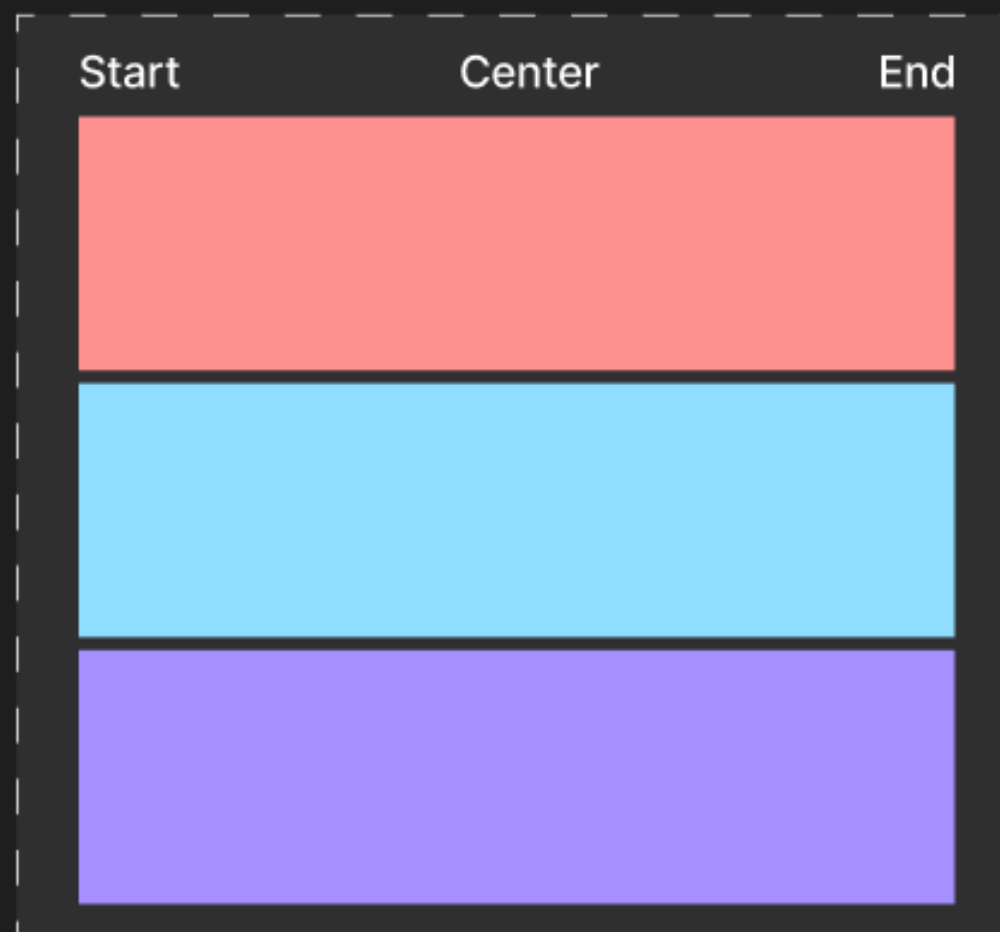


# ALIGN-ITEMS



define how flex items are aligned along  
the **cross axis**

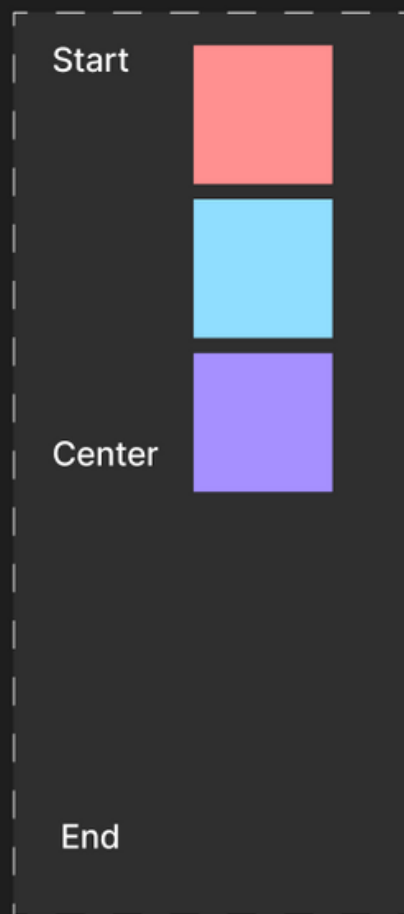
**align-items: stretch**



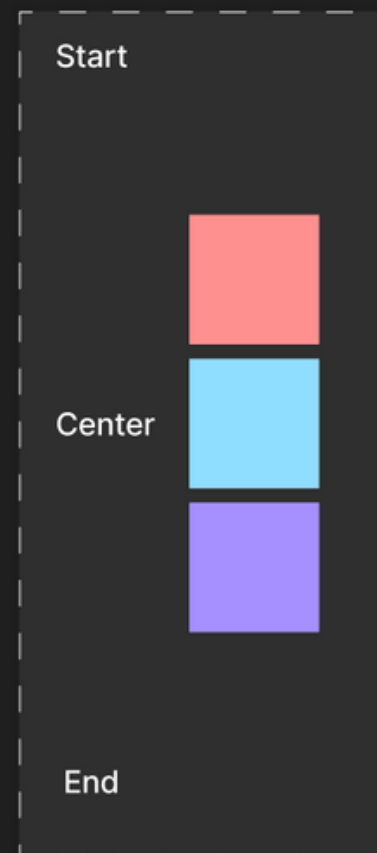
# JUSTIFY-CONTENT

define how flex items are aligned along  
the Y axis

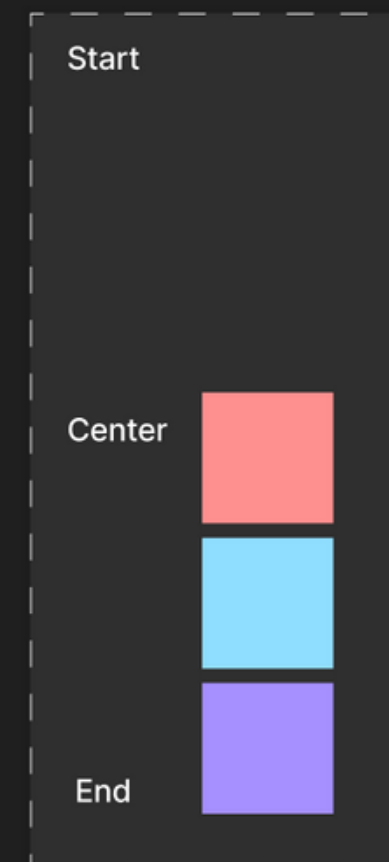
`justify-content: start`



`justify-content: center`



`justify-content: end`

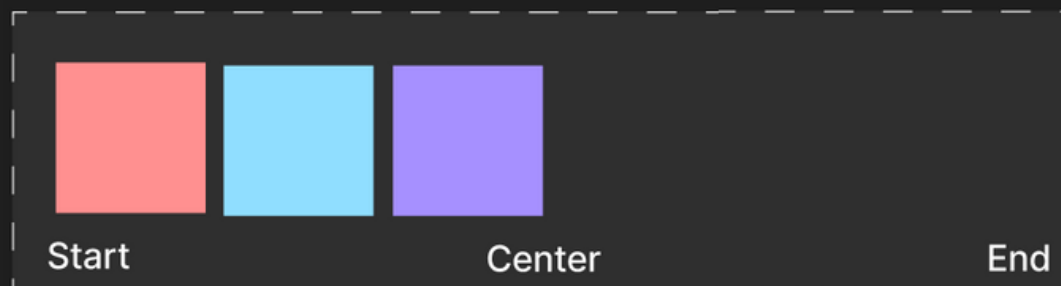


## COLUMN

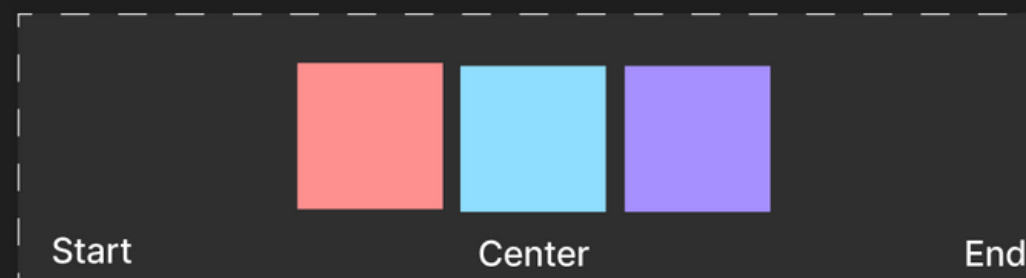
# JUSTIFY-CONTENT

define how flex items are aligned along  
the **X axis**

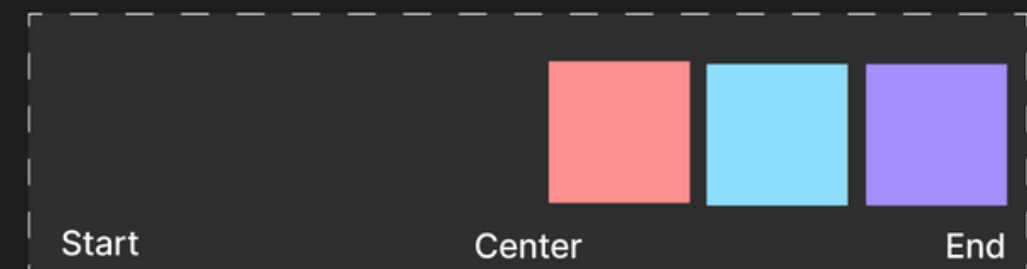
`justify-content: start`



`justify-content: center`



`justify-content: end`



## ROW

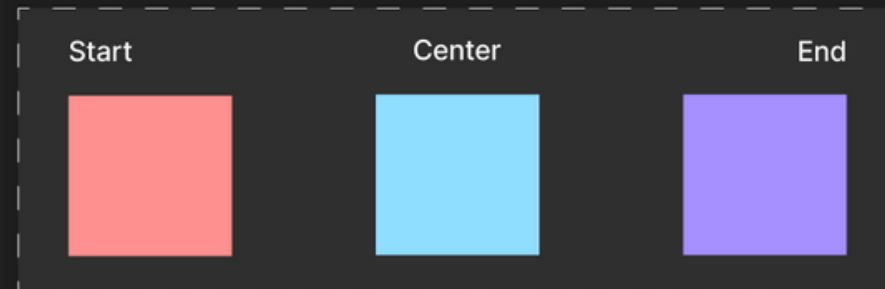


# JUSTIFY-CONTENT

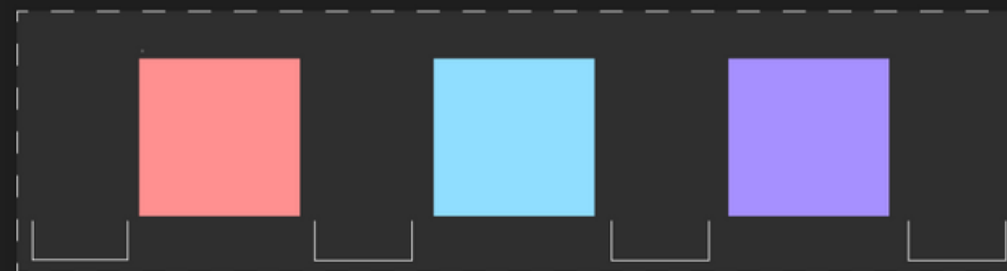


evenly distribute the available space  
along the main axis

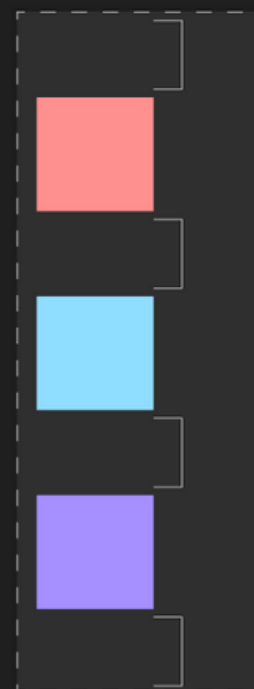
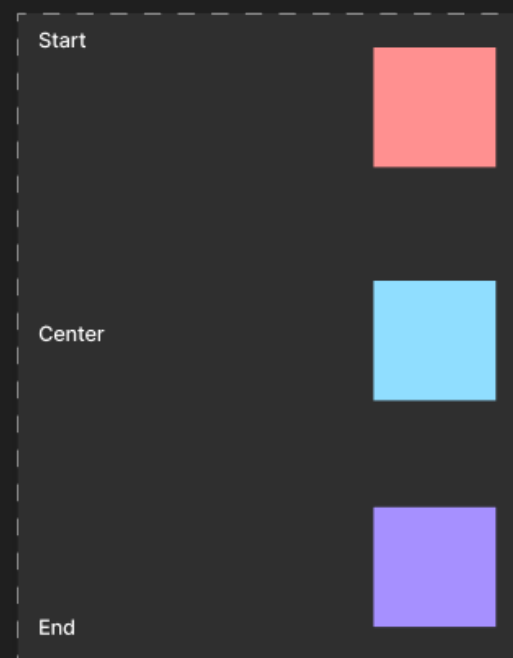
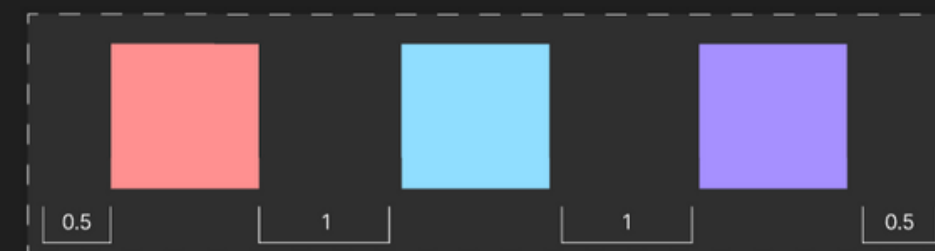
**justify-content:**  
**space-between**



**justify-content:**  
**space-evenly**



**justify-content:**  
**space-around**

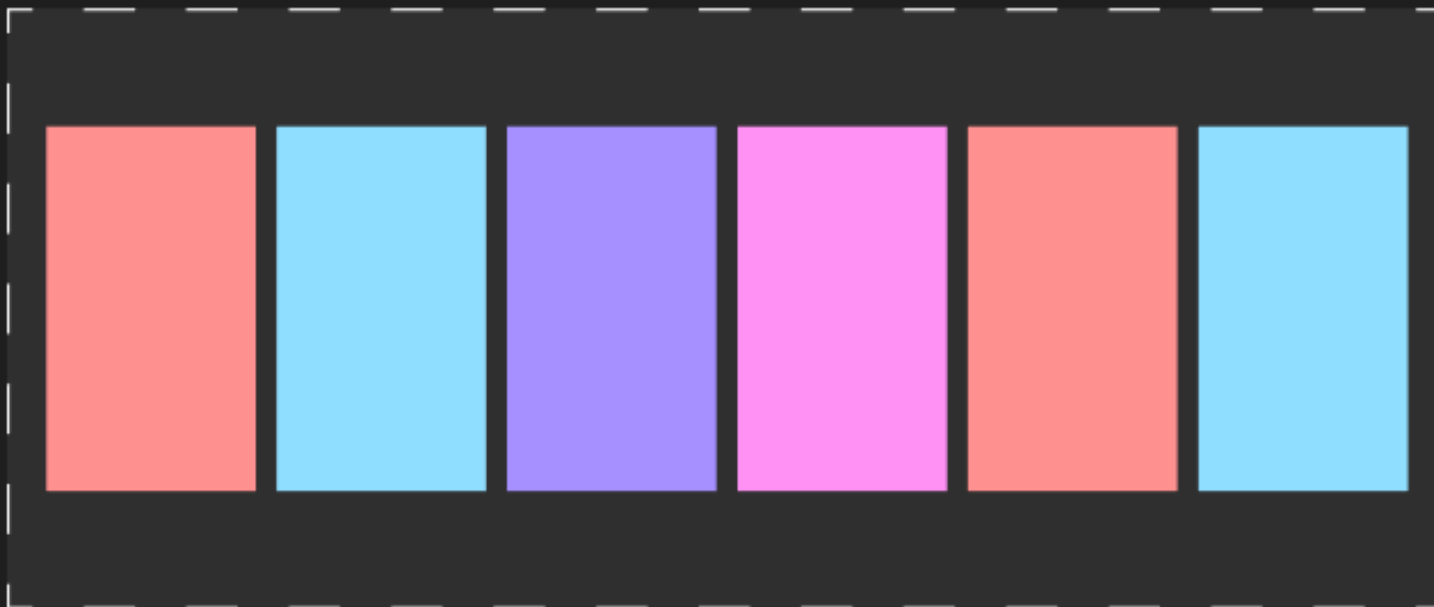




# FLEX-WRAP

control whether the flex container should wrap its items onto multiple lines or not

**flex-wrap: no-wrap**



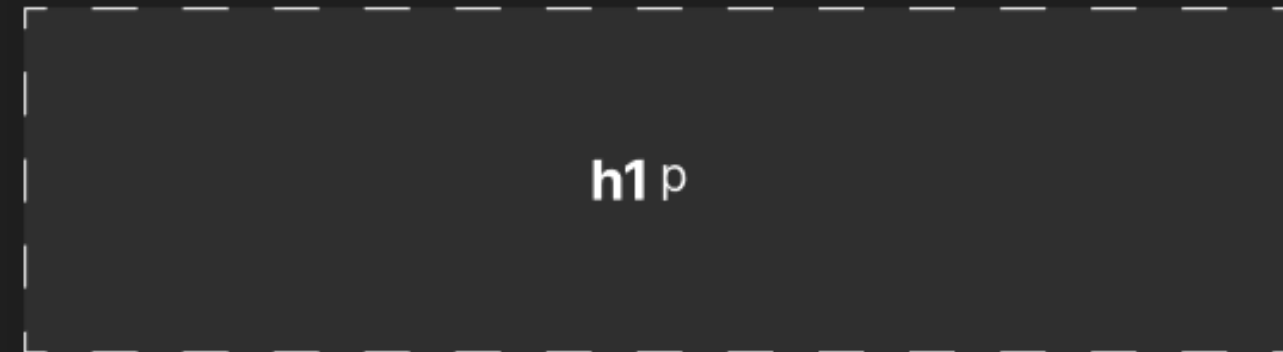
**flex-wrap: wrap**



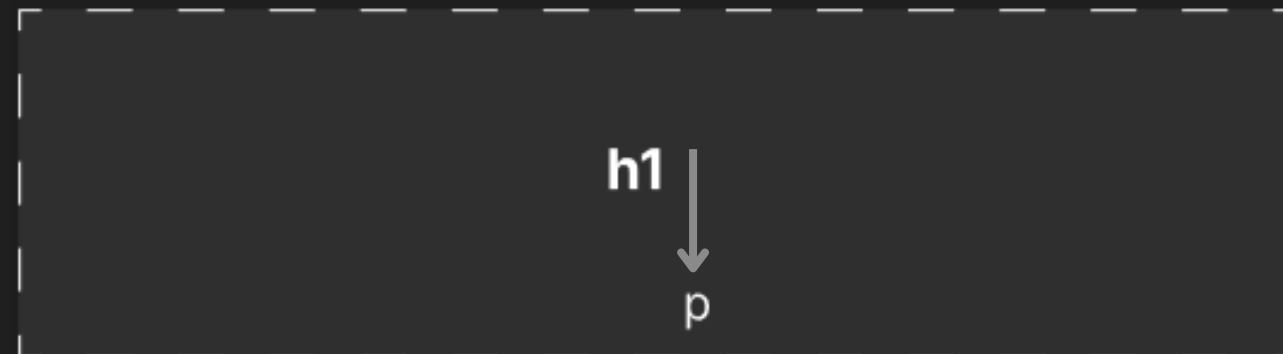
# ALIGN/JUSTIFY-SELF

control whether the flex container should wrap its items onto multiple lines or not

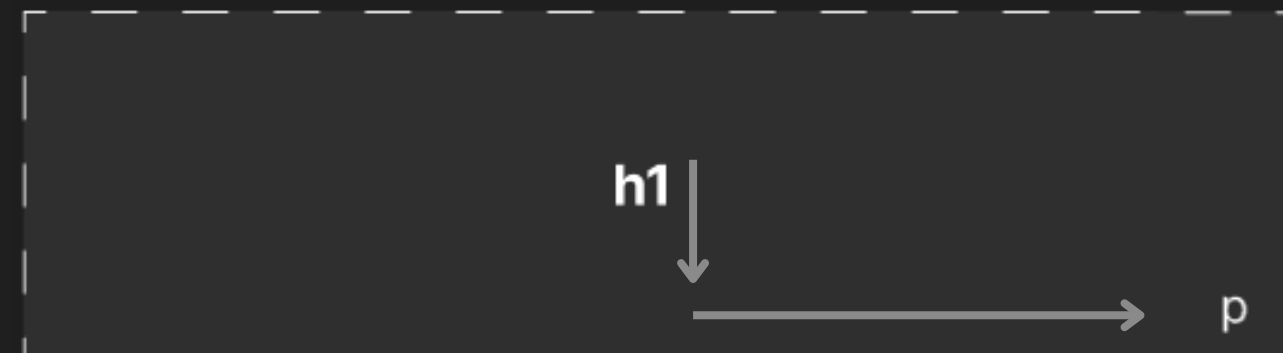
```
.container {  
  justify-content: center;  
  align-items: center;  
}
```



```
.container p {  
  align-self: end;  
}
```



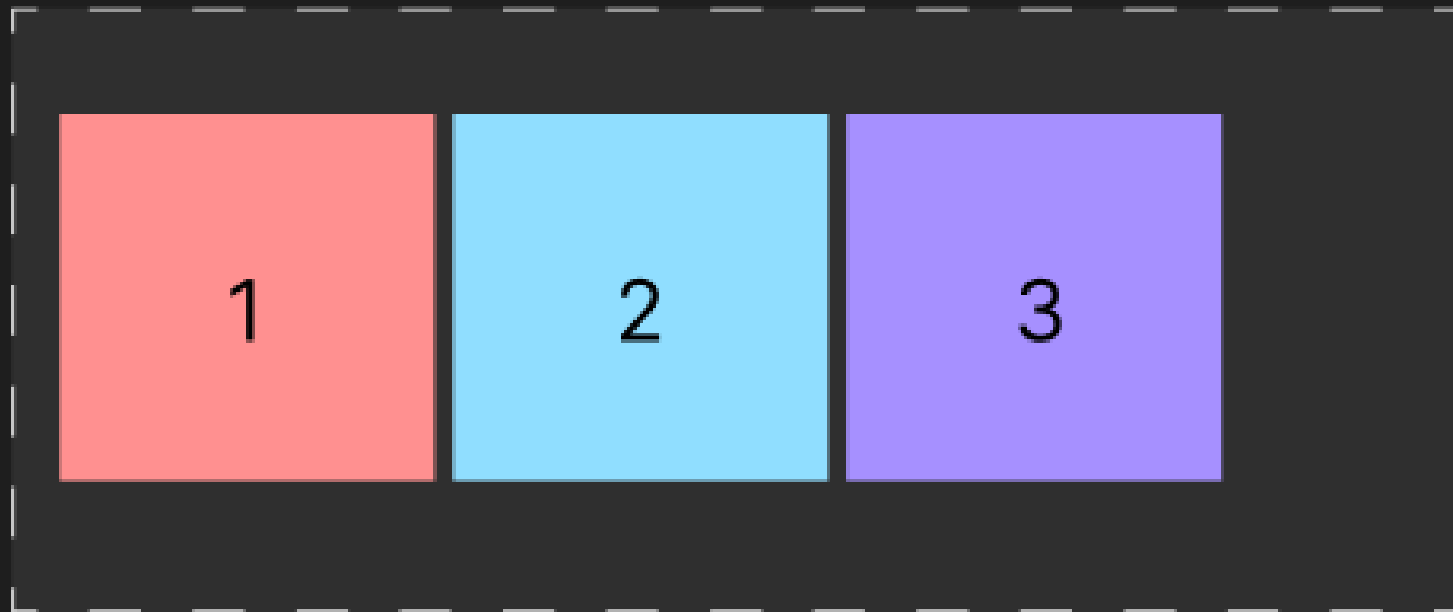
```
.container p {  
  align-self: end;  
  justify-self: end;  
}
```



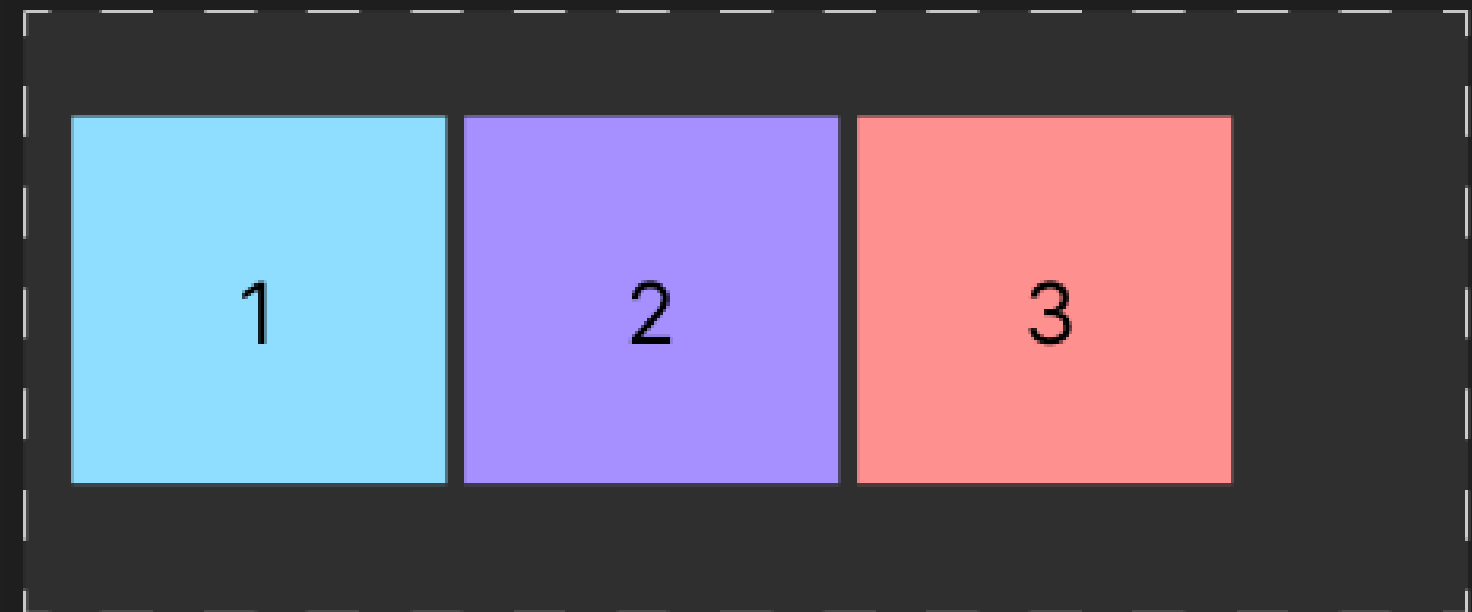
# ORDER

specify the order in which flex items are  
displayed within a flex container

```
red-box{ order: 1 };
```



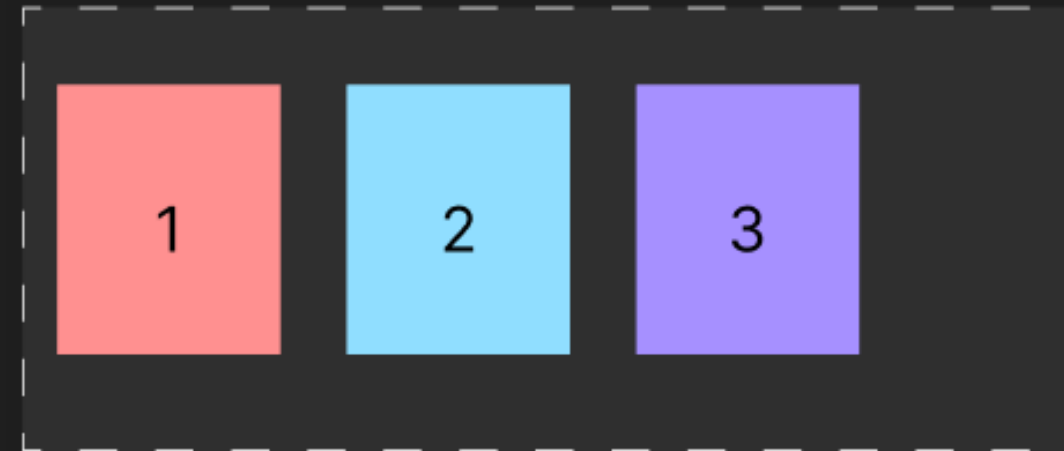
```
red-box{ order: 3 };
```



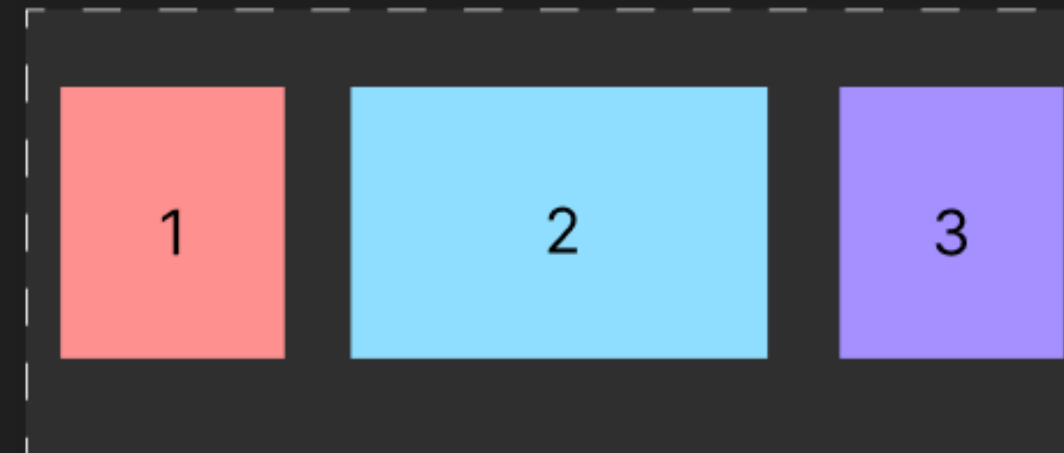
# FLEX-GROW

determines how a flex item grows relative to the other flex items in the same flex container

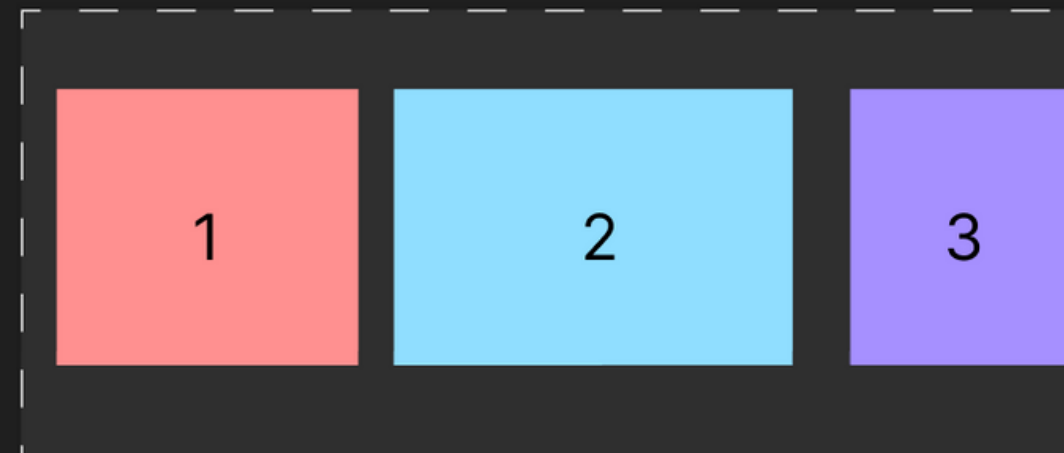
default



```
.box-2{  
  flex-grow: 1;  
}
```



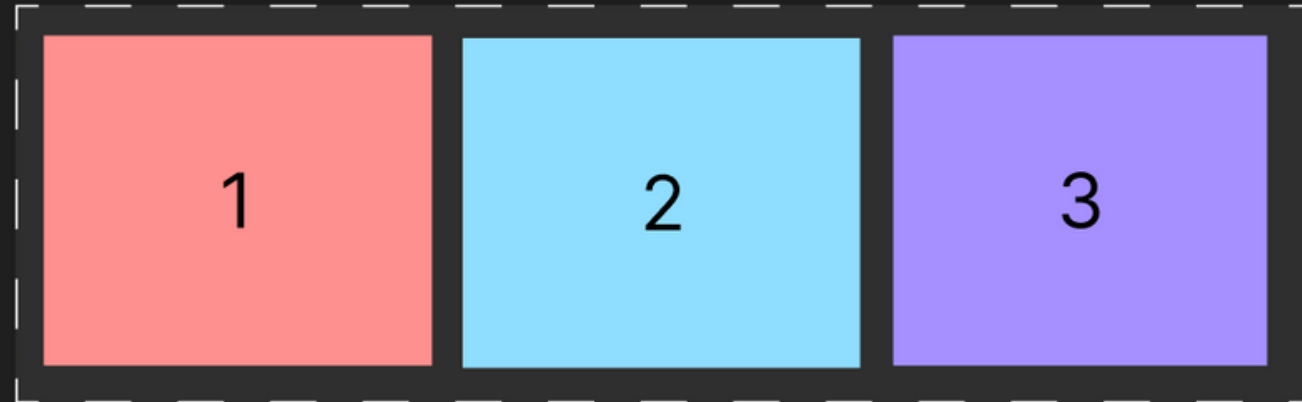
```
.box-1{  
  flex-grow: 1;  
}  
  
.box-2{  
  flex-grow: 2;  
}
```



# FLEX-SHRINK

define how a flex item should shrink relative to the other flex items in the flex container

default



```
.first-box {  
  flex-shrink: 1;  
}
```



```
.first-box {  
  flex-shrink: 1;  
}  
  
.second-box {  
  flex-shrink: 2;  
}
```

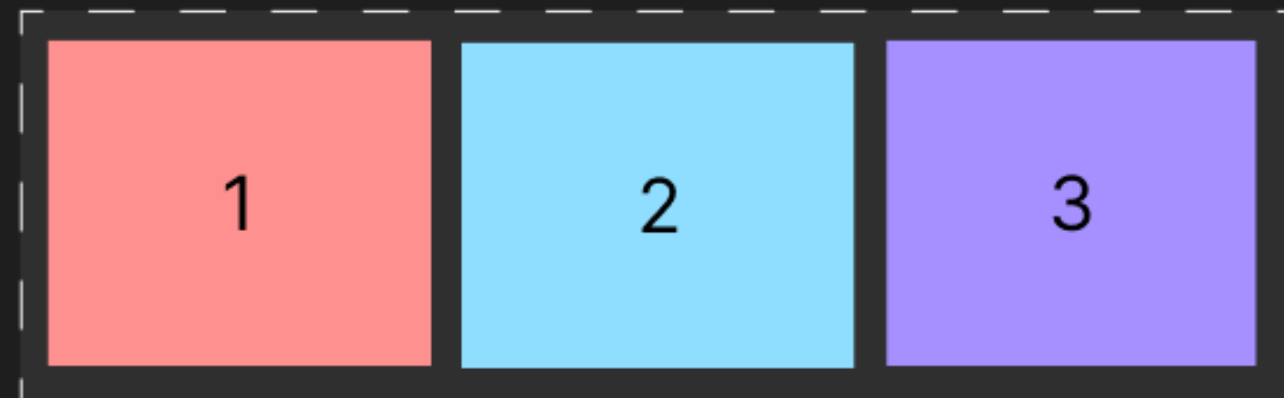


# FLEX-BASIS

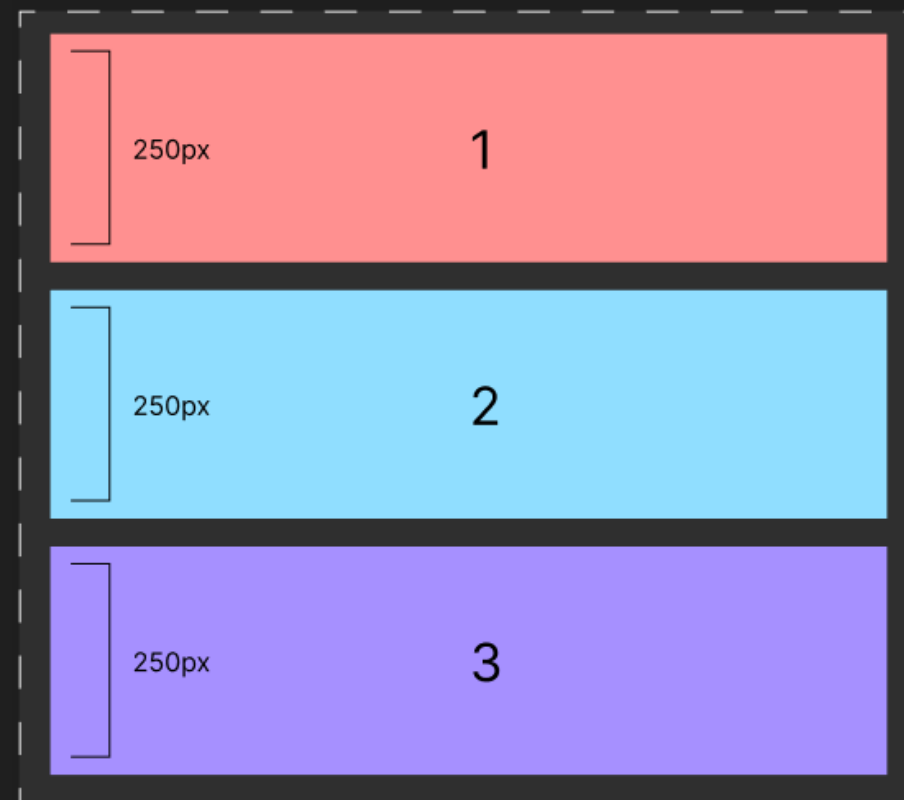
used in flexbox layouts to set the initial size of a flex item along the main axis before any remaining space is distributed.

**flex-basis: 250px**

ROW



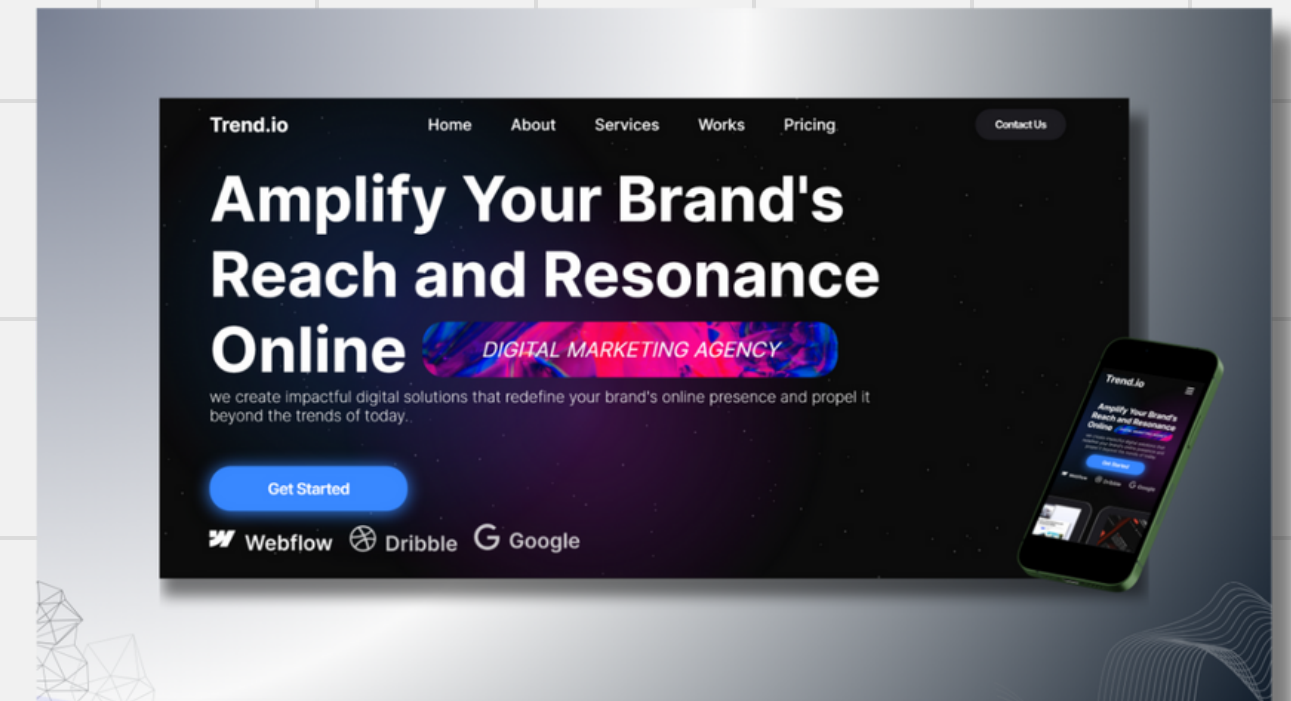
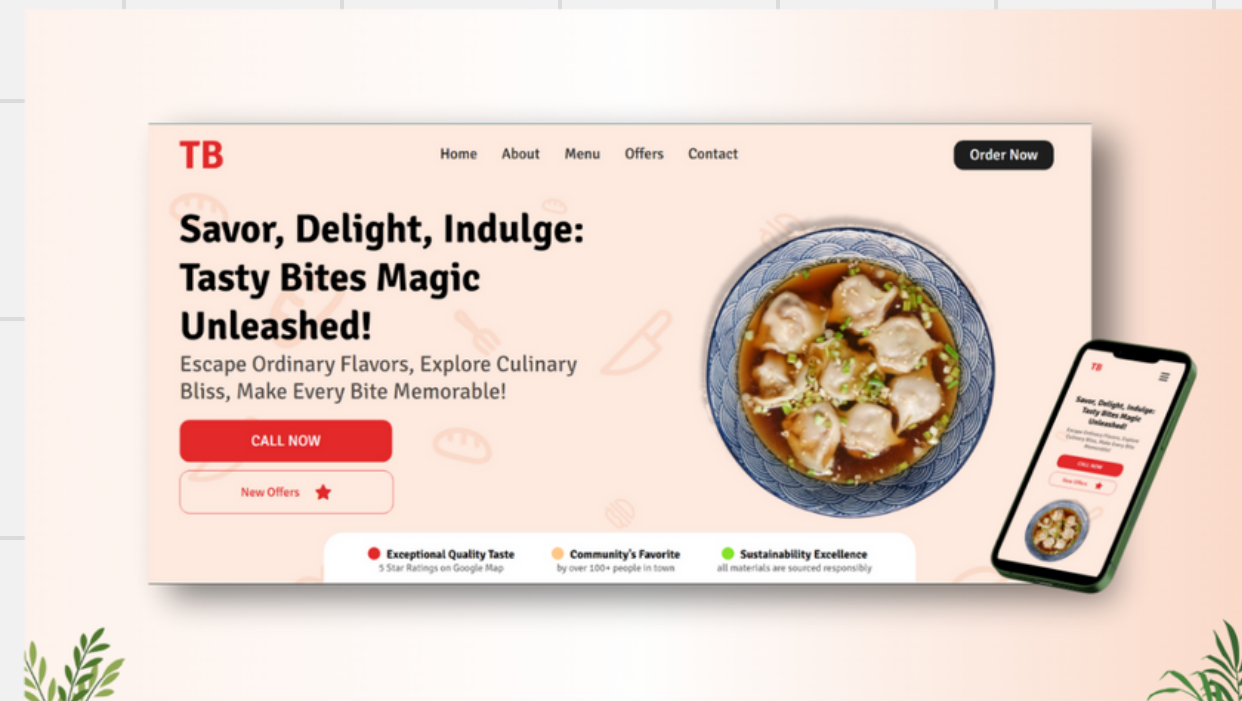
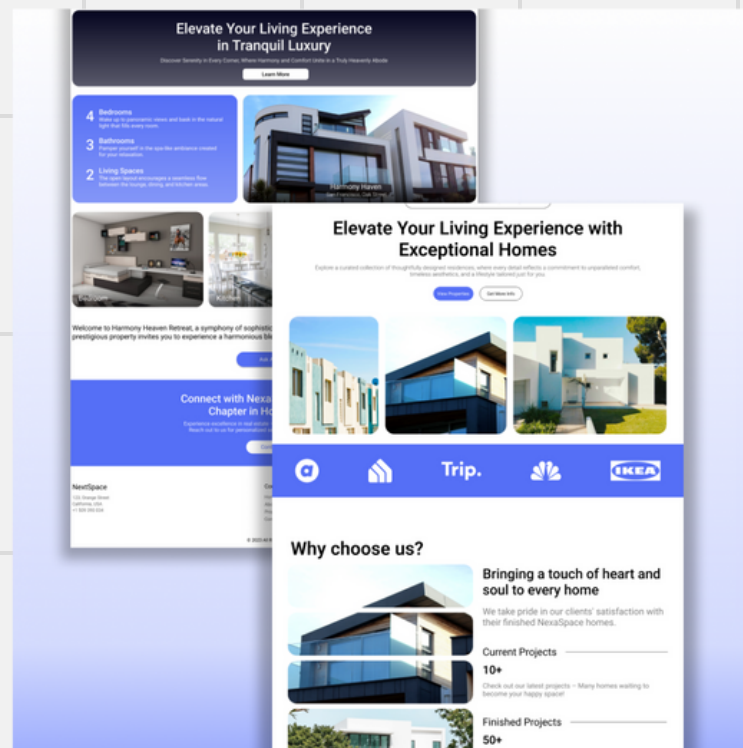
COLUMN



@NOVADESIGNS

# NEED MORE?

You can practice your FLEXPBOX skills  
in our YT Channel: **NOVA DESIGNS**



If you need a custom website, you can message us on:  
**novadesigns.creative@gmail.com**