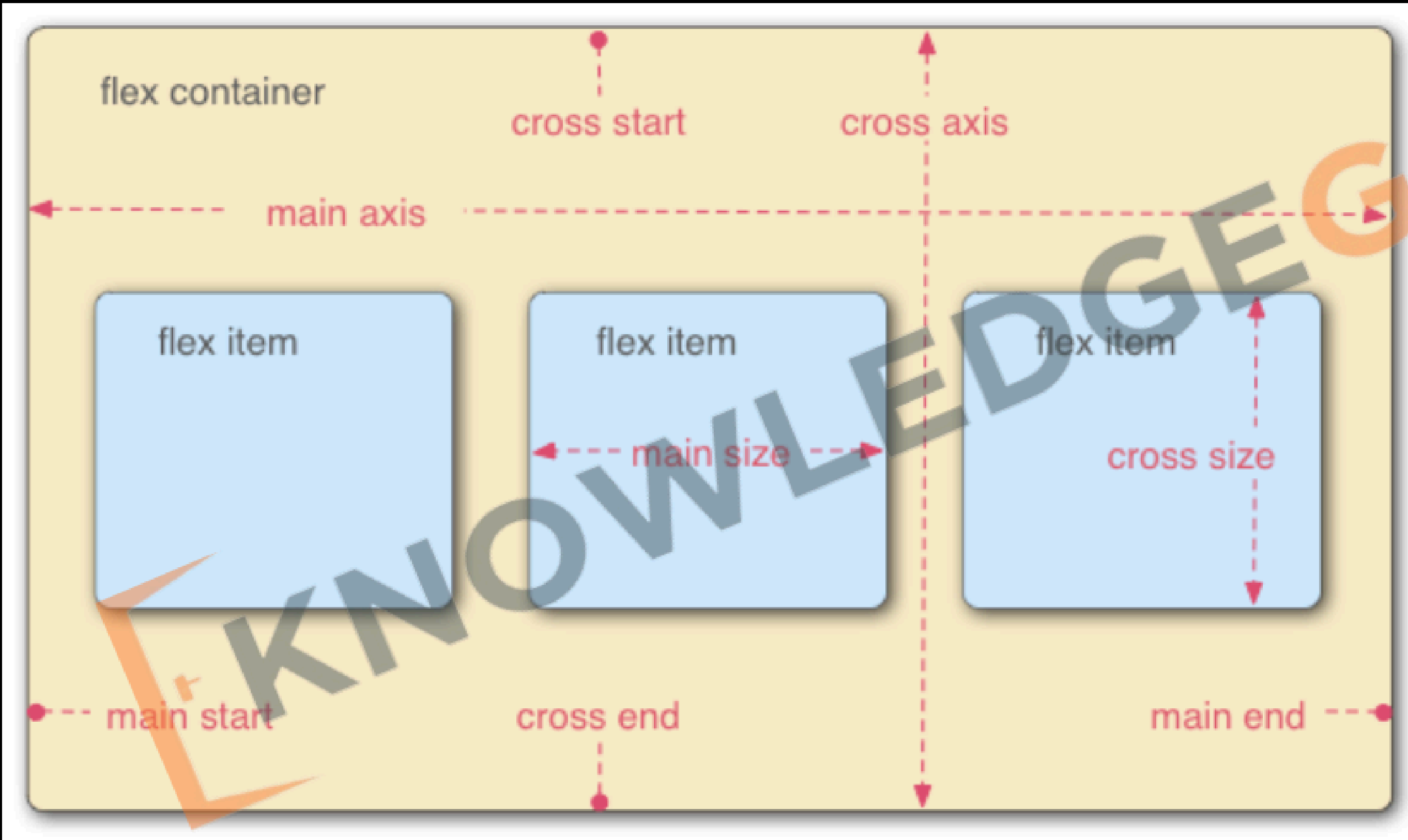


32. What is Flexbox?

Flexbox is a **one-dimensional layout** method for **arranging items in rows or columns**. Items *flex* (expand) to fill additional space or shrink to fit into smaller spaces.



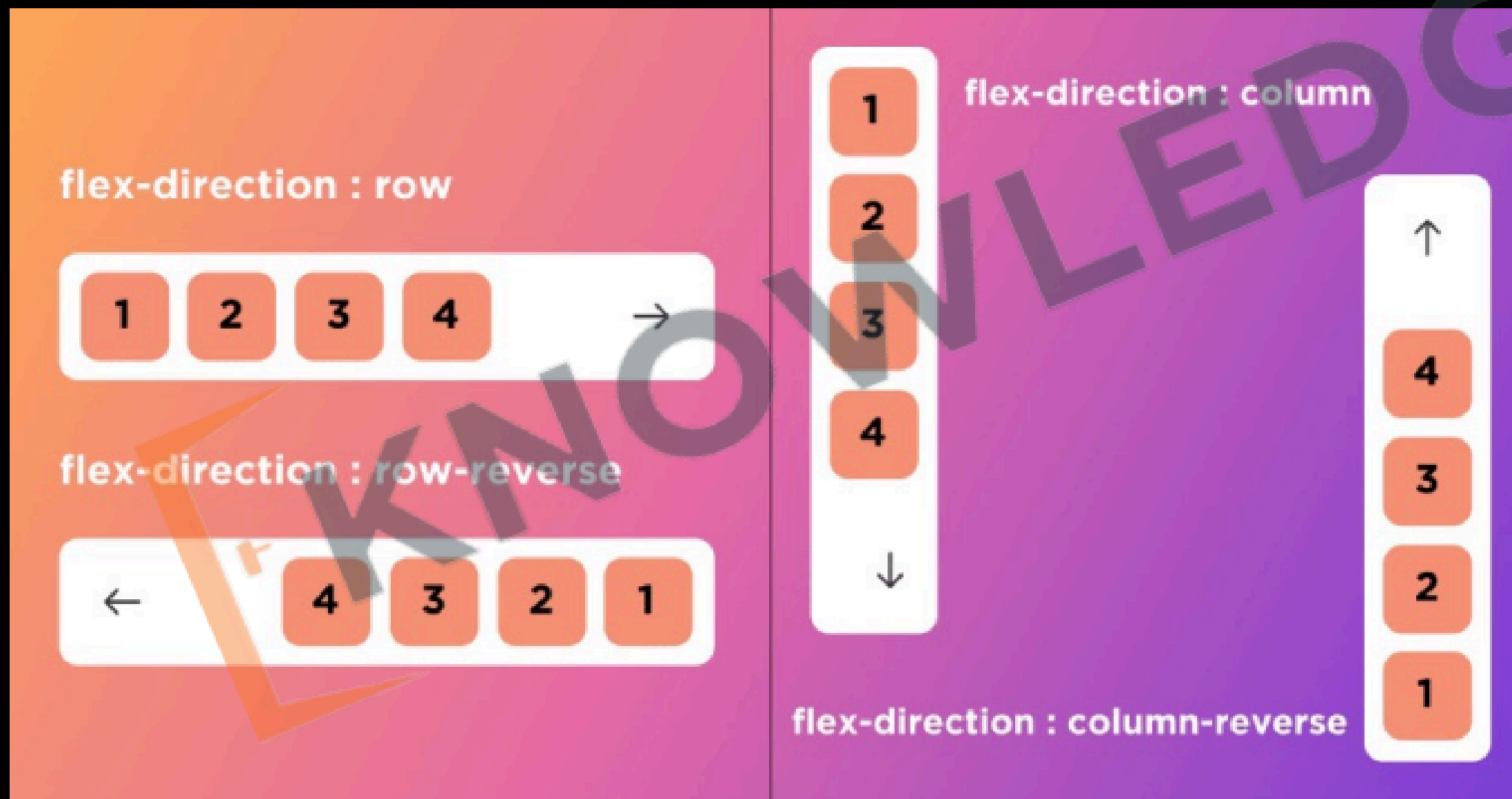
33. Flex Model



display: flex

34. Flexbox Direction

- **Property Name:** flex-direction is the property used to define the direction in a flex container.
- **Row Layout:** row value aligns the flex items horizontally, in a left-to-right fashion.
- **Column Layout:** column value stacks the flex items vertically, from top to bottom.
- **Reverse Direction:** Adding -reverse to row or column (as in row-reverse or column-reverse) reverses the order of the items.



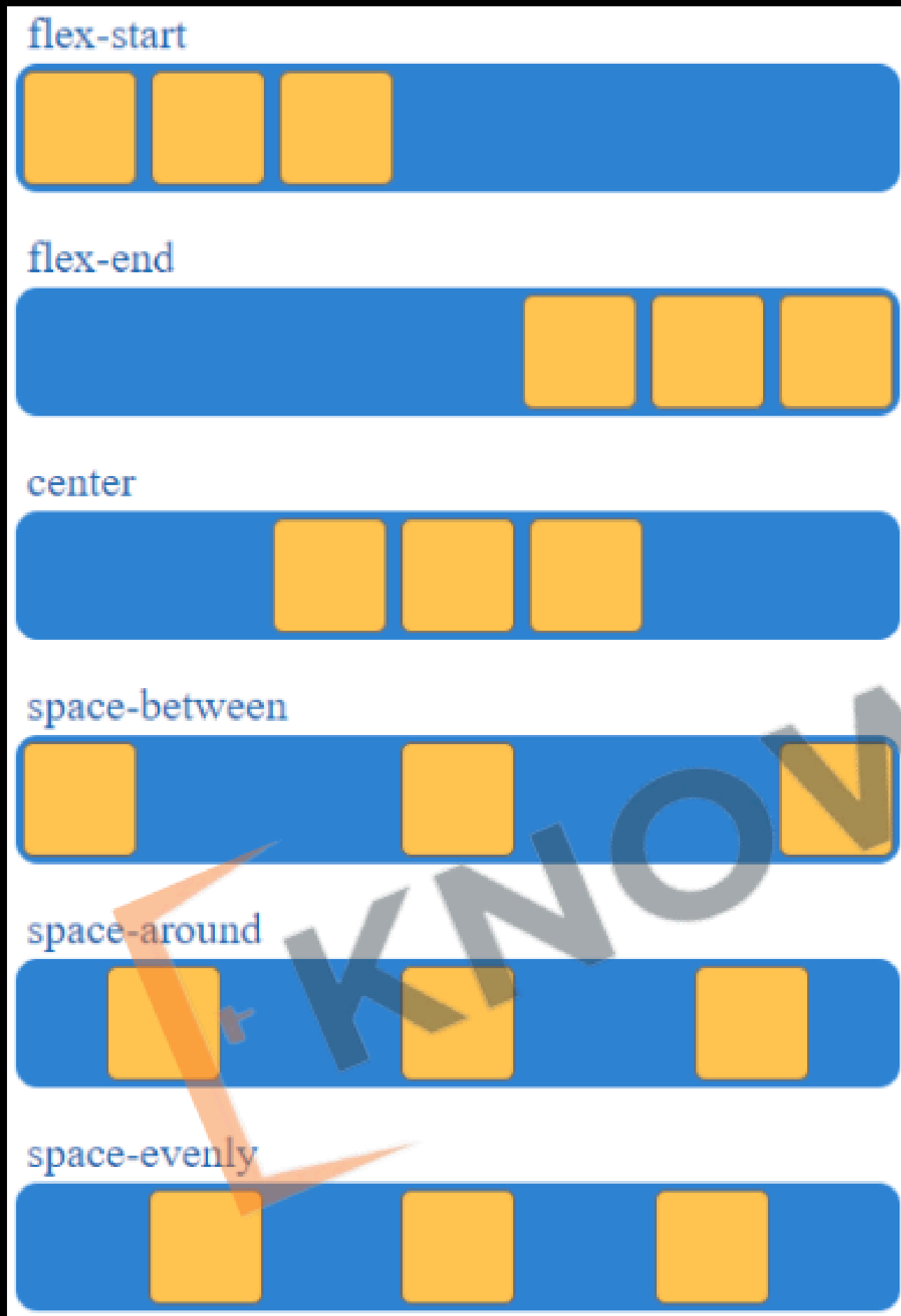
34. Flexbox Direction

```
.box {  
  height: 100px;  
  width: 100px;  
  border: 3px solid black;  
  margin-right: 5px;  
}  
#heading {margin-left: 50px;}  
#container {  
  height: 150px;  
  width: 200px;  
  padding: 10px;  
  margin: 20px;  
  border: 3px solid black;  
  display: flex;  
  flex-direction: row-reverse;  
}
```

Flex Container



35. Properties: Flexbox container (Justify Content)



- **Alignment:** Aligns flex items along the main axis.
- **flex-start:** Items align to the start of the flex container.
- **flex-end:** Items align to the end of the flex container.
- **Center:** Items are centered within the flex container.
- **space-between/space-around/space-evenly:** Distributes space between items evenly.

35. Properties: Flexbox container (Justify Content)

Flexbox Container



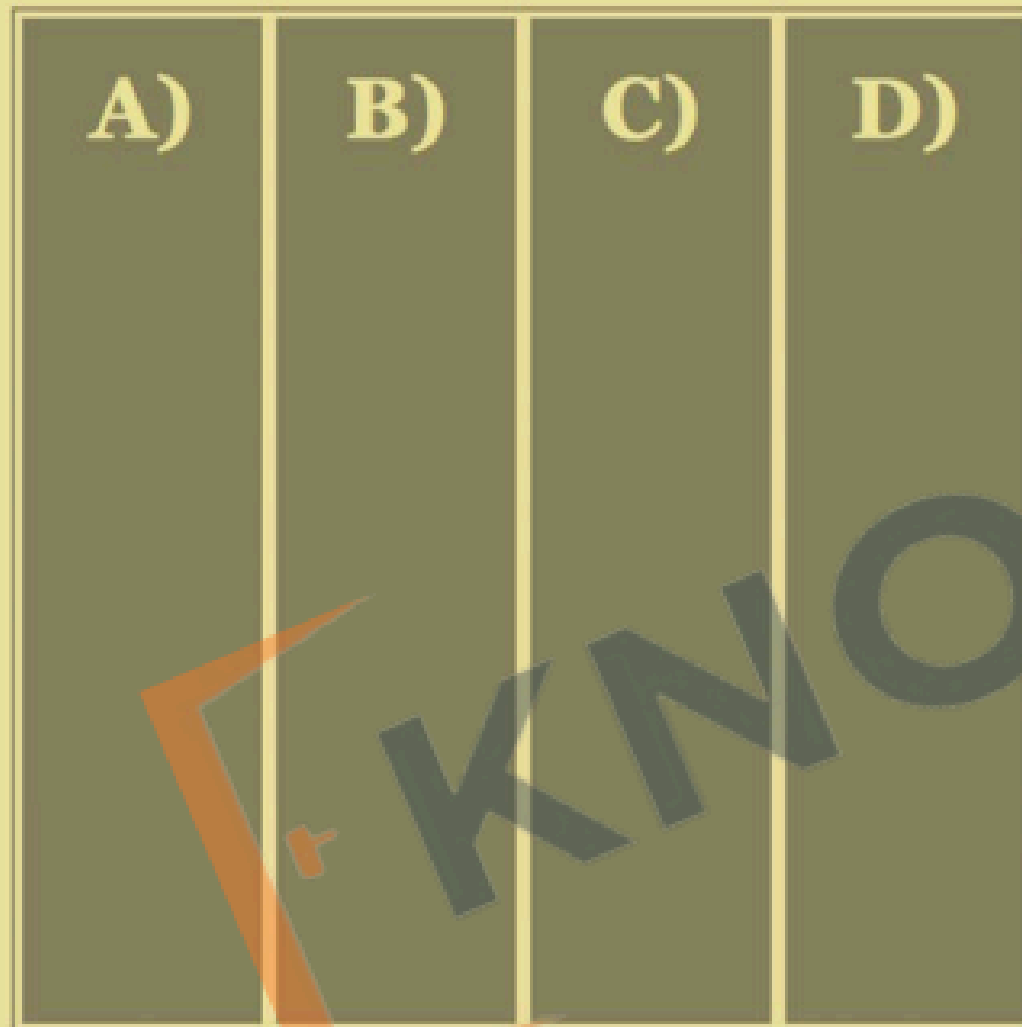
```
display: flex;
```

```
justify-content: space-between;
```

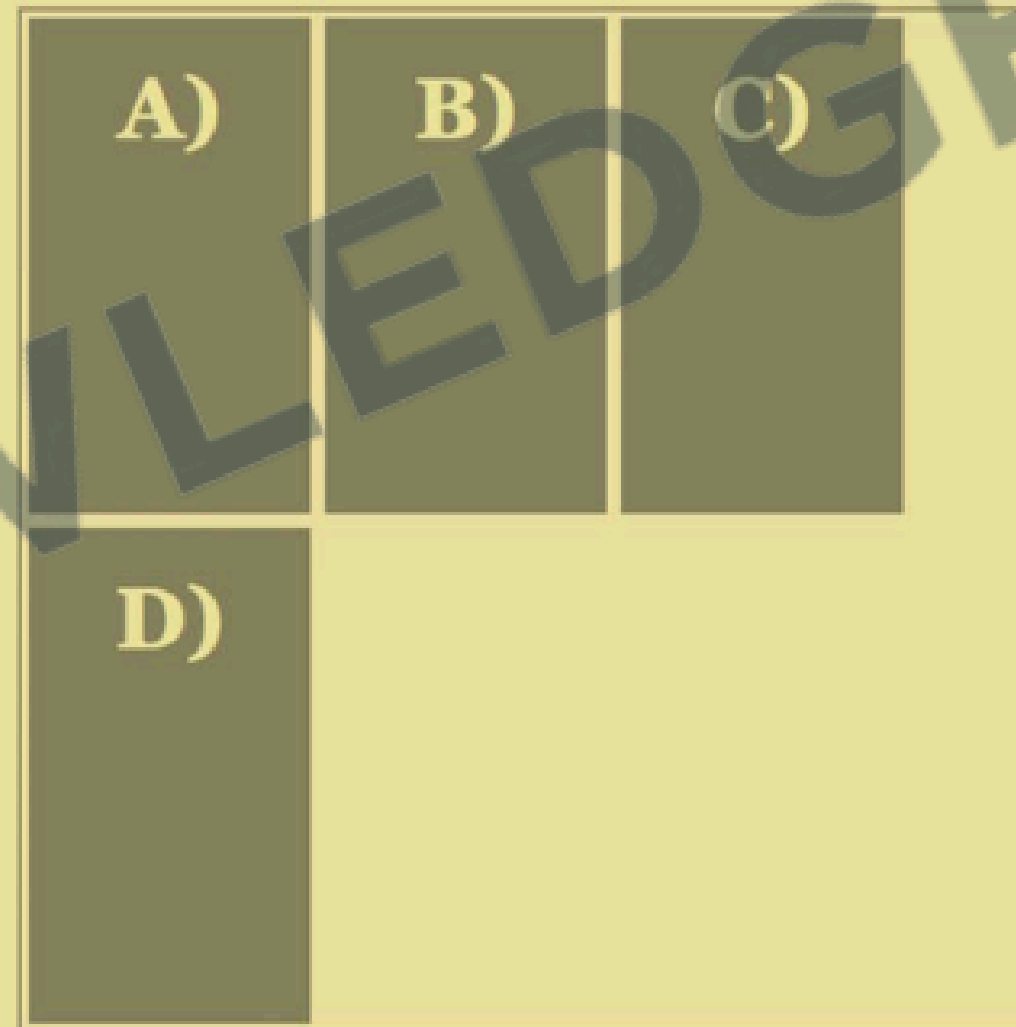
35. Properties: Flexbox container (Flex Wrap)

flex-wrap:

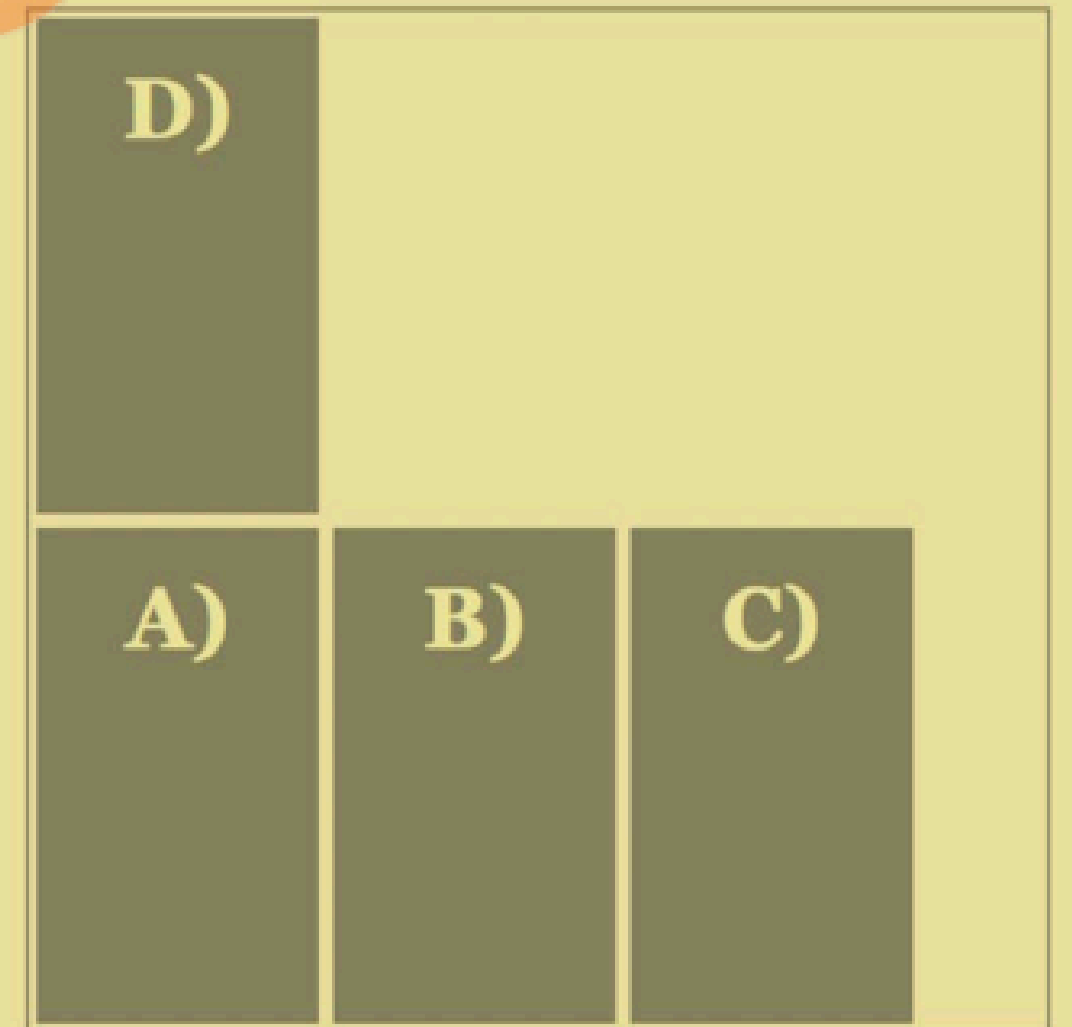
nowrap (default)



wrap



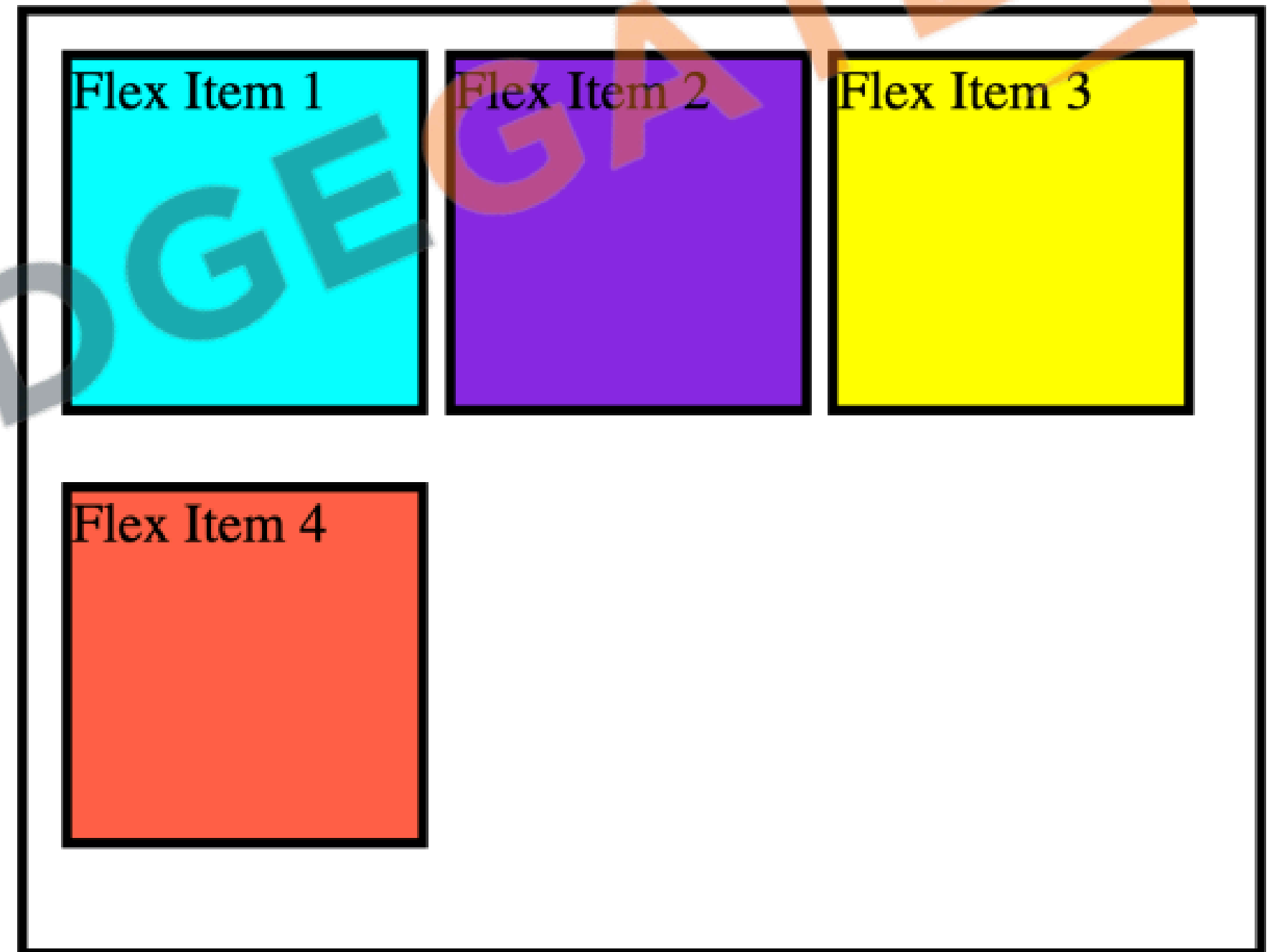
wrap-reverse



35. Properties: Flexbox container (Flex Wrap)

```
#container {  
  height: 250px;  
  width: 335px;  
  padding: 10px;  
  margin: 20px;  
  border: 3px solid black;  
  
  display: flex;  
  flex-wrap: wrap;  
}
```

Flex Container

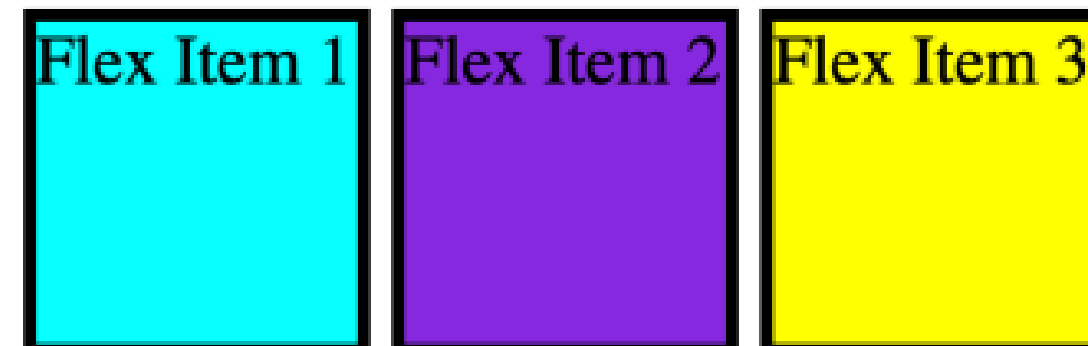


35. Properties: Flexbox container (Align Items)

This property is used to align the flex container's items along the cross-axis, which is perpendicular to the main axis.

```
display: flex;  
flex-direction: row;  
justify-content: center;  
align-items: center;
```

Flexbox Container



35. Properties: Flexbox container (Align Content)

It is utilized to adjust the spacing between flex lines within a flex container, particularly when there is extra space along the cross-axis.

```
display: flex;  
flex-direction: row;  
justify-content: center;  
flex-wrap: wrap;  
align-content: center;
```

Flexbox Container

Flex Item 1

Flex Item 2

Flex Item 3

Flex Item 4

Flex Item 5

Flex Item 6

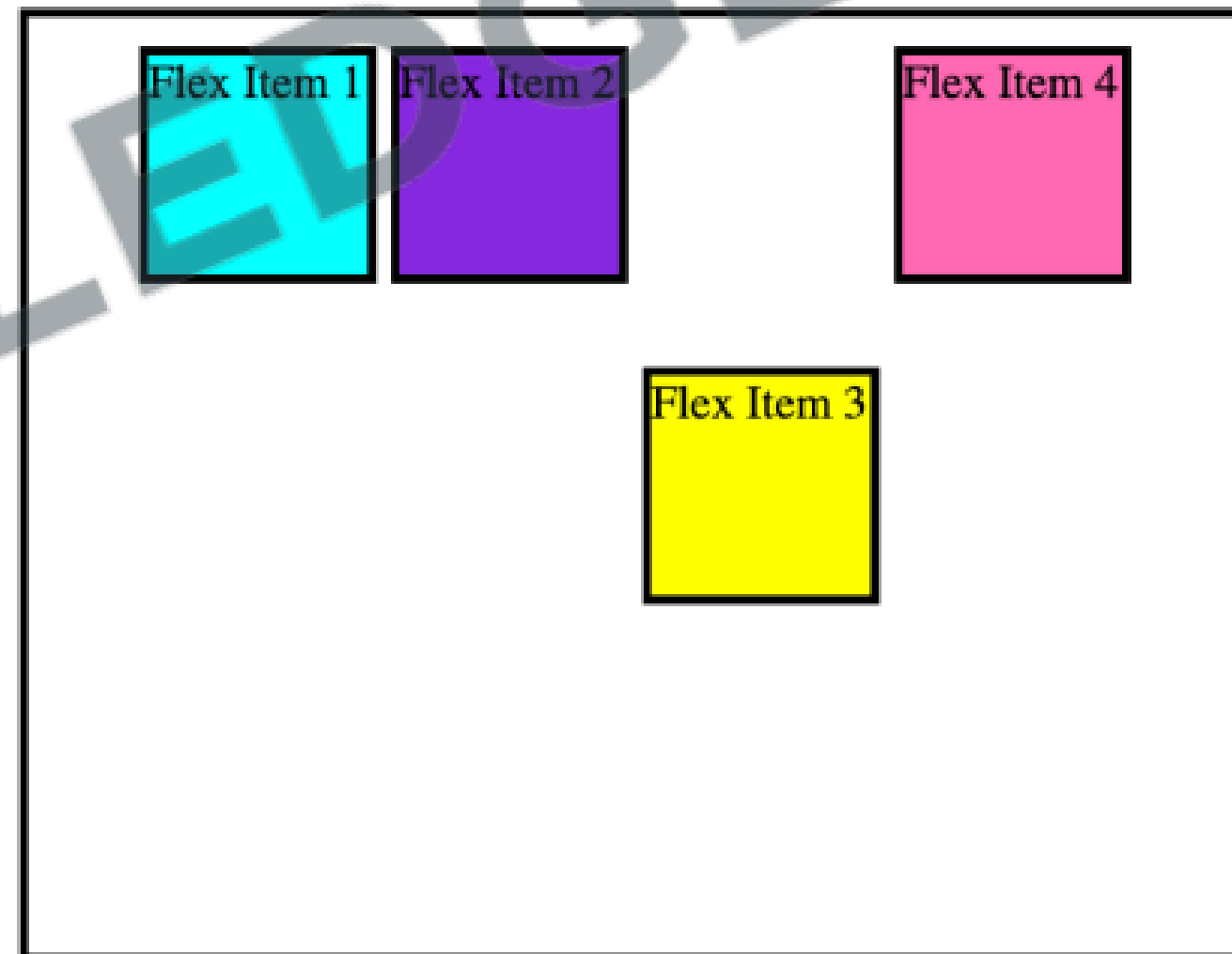
36. Properties: Flex Items (Align Self)

```
#container {  
  height: 300px;  
  width: 400px;  
  padding: 10px;  
  margin: 20px;  
  border: 3px solid black;  
  
  display: flex;  
  flex-direction: row;  
  justify-content: center;  
  flex-wrap: wrap;  
  align-items: start;  
}
```

```
#box1 { background-color: aqua;}  
#box2 { background-color: blueviolet;}  
#box4 { background-color: hotpink;}  
#box3 {  
  background-color: yellow;  
  align-self: center;  
}
```

Allows individual flex items to **override the container's align-items** property, aligning them differently along the cross-axis.

Flexbox Container

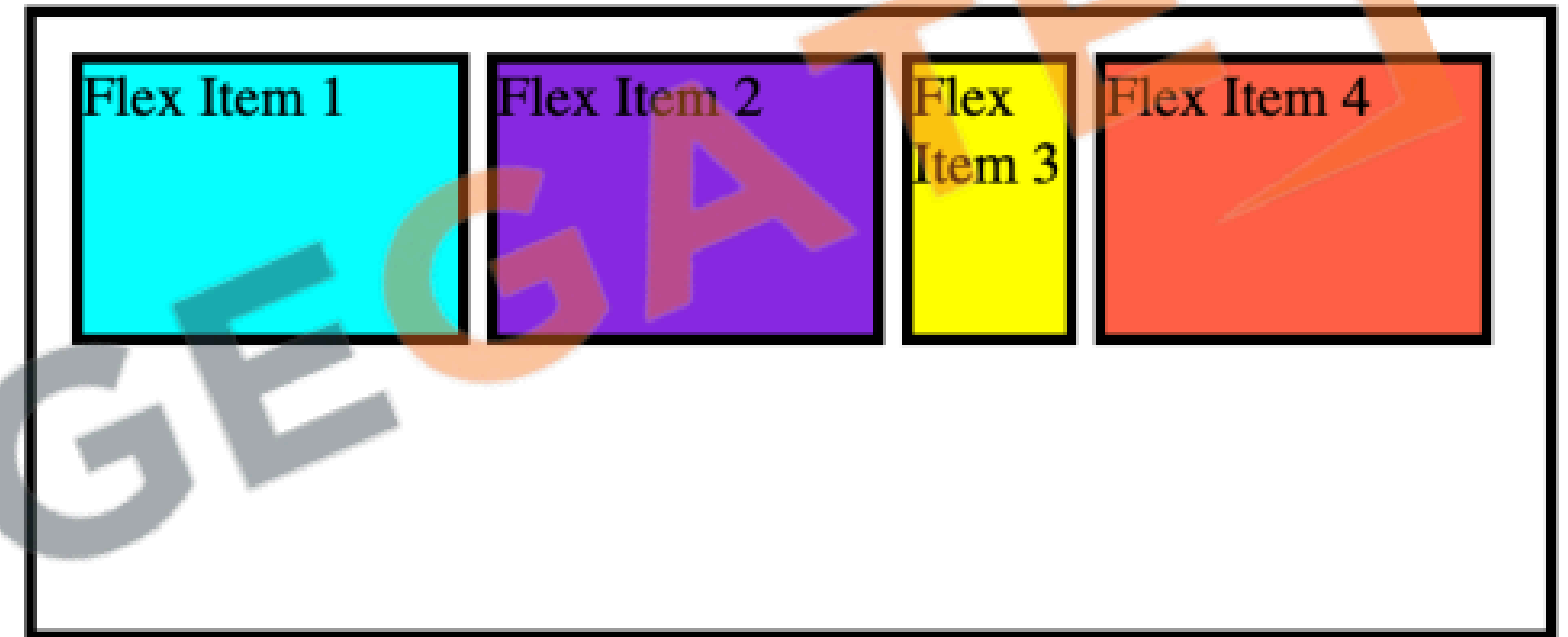


36. Properties: Flex Items (Flex Shrink)

```
#box3 {  
  background-color: yellow;  
  flex-shrink: 4;  
}
```

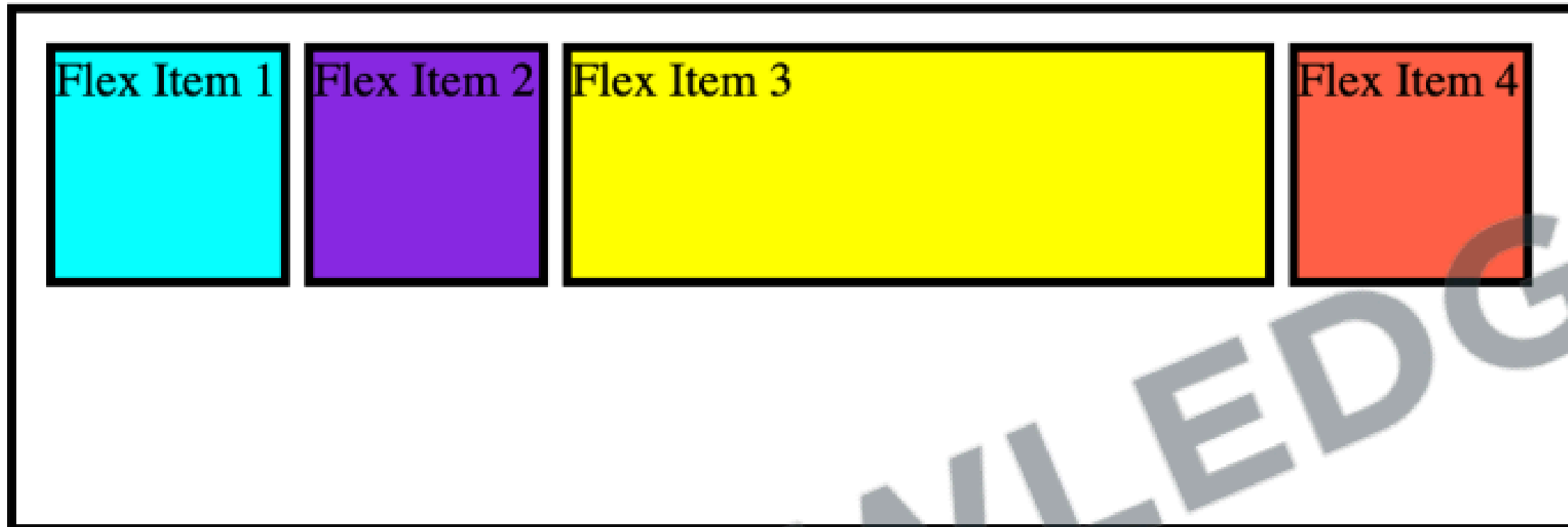
The "flex-shrink" property in CSS determines how much a flex item will shrink relative to other items in the flex container if there is insufficient space.

Flexbox Container



36. Properties: Flex Items (Flex Grow)

Flexbox Container



```
#box3 {  
  background-color: yellow;  
  flex-grow: 1;  
}
```

The "flex-grow" property in CSS specifies how much a **flex item** will grow relative to other items in the flex container when additional space is available.

36. Properties: Flex Items (Order)

```
#box1 {  
  background-color: ■ aqua;  
  order: 3  
}  
#box2 {  
  background-color: ■ blueviolet;  
  order: 1;  
}  
#box3 {  
  background-color: ■ yellow;  
  order: 4;  
}  
#box4 {  
  background-color: ■ tomato;  
  order: 2;  
}
```

The "order" property in CSS allows you to define the sequence in which flex items appear within the flex container, overriding their original order in the HTML.

Flexbox Container

