

# Treinamento Flexbox

Vicente Calfo

[vicentecalfo@gmail.com](mailto:vicentecalfo@gmail.com)

# Apresentação

O módulo de layout de caixa flexível torna mais fácil projetar uma estrutura de layout responsiva flexível sem usar flutuação ou posicionamento.

# Propriedades do Pai (flex container)

## display

```
.container {  
  display: flex; /* or inline-flex */  
}
```

# Propriedades do Pai (flex container)

## flex-direction

```
.container {  
  flex-direction: row; /*row | row-reverse | column | column-reverse*/  
}
```

- **row** (padrão - *default*): da esquerda para a direita
- **row-reverse**: da direita para a esquerda
- **column**: da esquerda para a direita, mas do topo para parte inferior
- **column-reverse**: da direita para a esquerda, mas da parte inferior para o topo

# Propriedades do Pai (flex container)

## flex-wrap

```
.container {  
  flex-direction: row; /*row | row-reverse | column | column-reverse*/  
}
```

- **nowrap** (padrão - *default*): todos os itens ficaram em uma linha
- **wrap**: os itens serão quebrados em múltiplas linhas do topo para parte inferior
- **wrap-reverse**: os itens serão quebrados em múltiplas linhas da parte inferior para topo

# Propriedades do Pai (flex container)

## flex-flow

Este é a declaração compacta das propriedades flex-direction e flex-wrap.

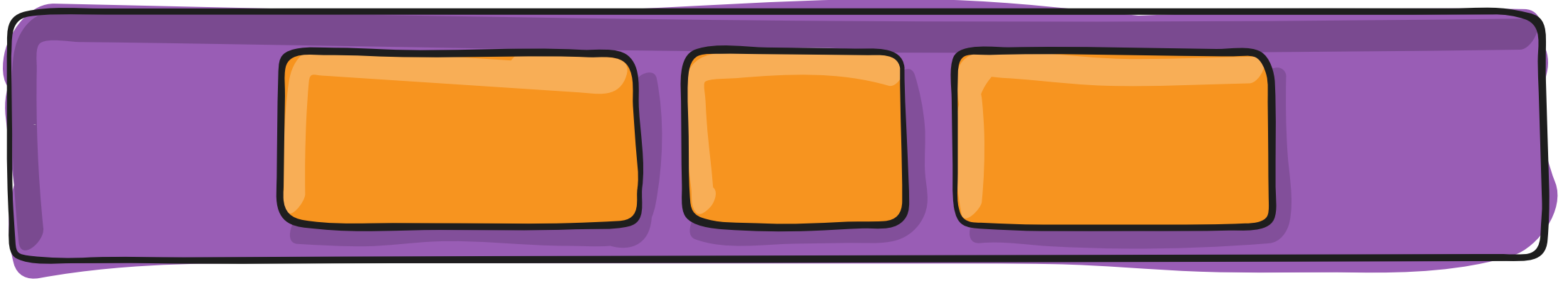
```
.container {  
  flex-flow: column wrap;  
}
```

# Propriedades do Pai (flex container)

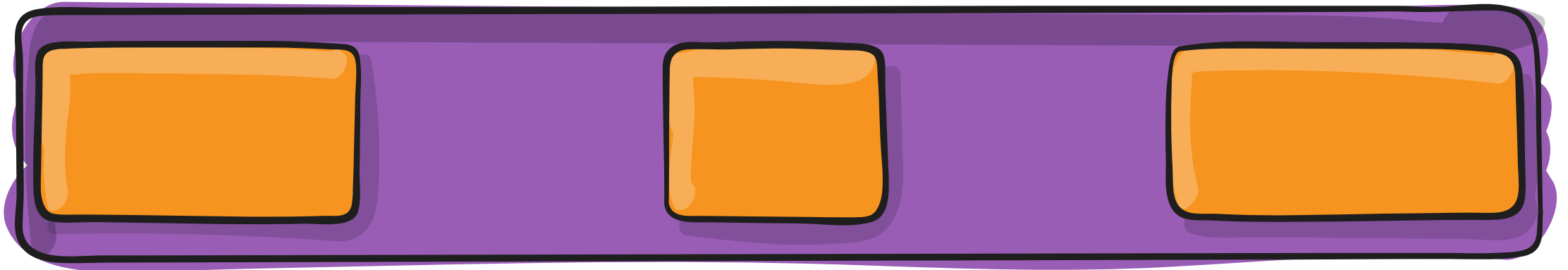
## justify-content

```
.container {  
  justify-content: flex-start; /*flex-start | flex-end | center |  
    space-between | space-around | space-evenly |  
    start | end | left | right*/  
}
```

center



space-between



space-around



# Propriedades do Pai (flex container)

## justify-content

- **flex-start** (padrão - *default*): os itens serão organizados pelo início respeitando a flex-direction
- **flex-end**: os itens serão organizados pelo final respeitando a flex-direction
- **start**: os itens serão organizados pelo início respeitando a writing-mode
- **end**: os itens serão organizados pelo final respeitando a writing-mode
- **left**: os itens serão organizados pelo início em direção a borda esquerda
- **right**: os itens serão organizados pelo início em direção a borda direita
- **center**: itens são centralizados ao longo da linha

# Propriedades do Pai (flex container)

## justify-content

- **space-between**: os itens serão distribuídos uniformemente na linha
- **space-around**: os itens são distribuídos uniformemente na linha com espaço igual ao redor deles
- **space-evenly**: os itens serão distribuídos de forma que o espaçamento entre os itens sejam iguais.

# Propriedades do Pai (flex container)

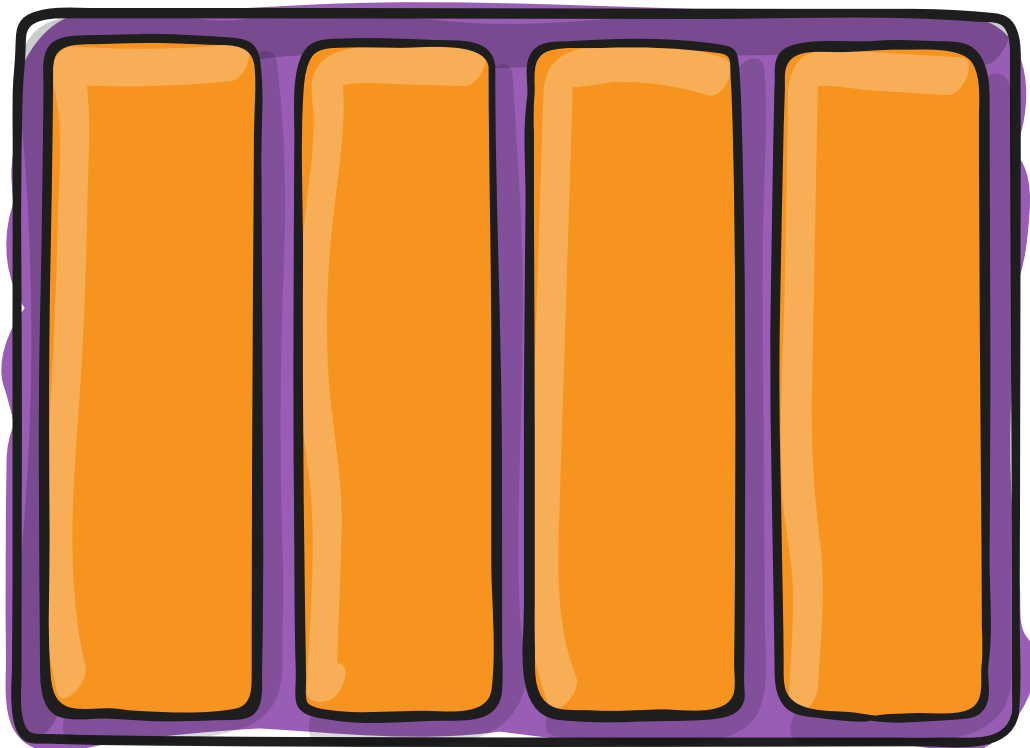
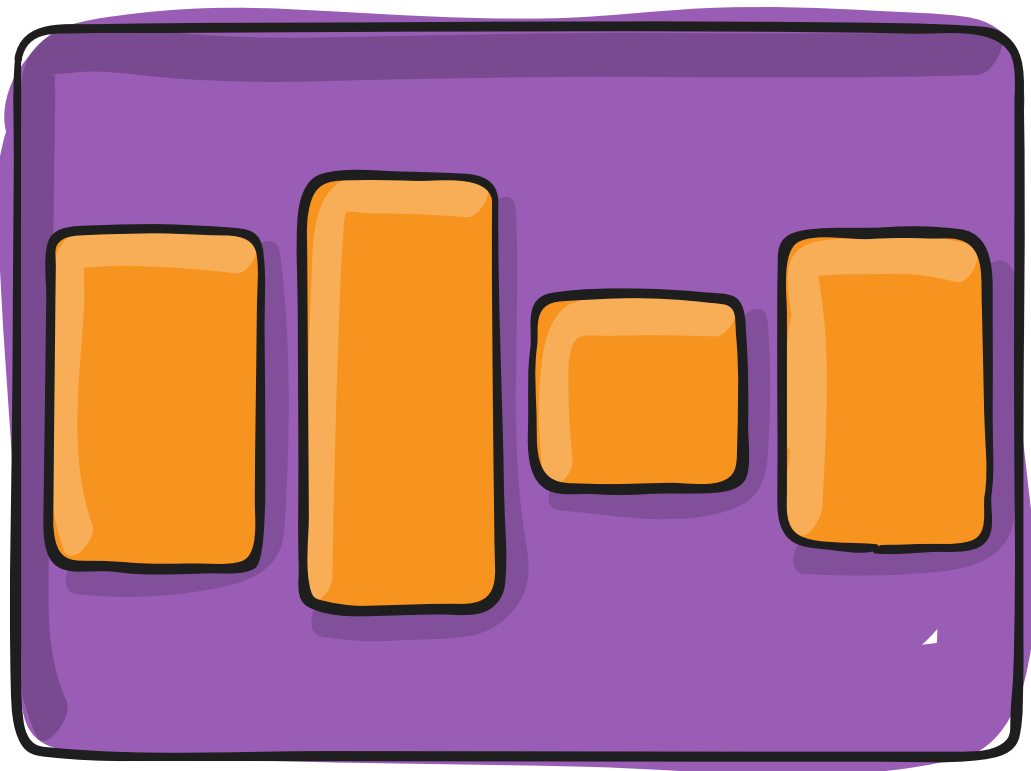
## align-items

```
.container {  
  align-items: stretch ; /*stretch | flex-start | flex-end | center |  
  baseline | first baseline | last baseline | start |  
  end | self-start | self-end;*/  
}
```



center

stretch



baseline

# Propriedades do Pai (flex container)

## align-items

- **stretch** (padrão - *default*): os itens irão preencher o container
- **flex-start / start / self-start**: os itens serão colocados no início
- **flex-end / end / self-end**: os itens serão colocados no final
- **center**: os itens serão centralizados
- **baseline**: os itens serão alinhados, como suas linhas de base se alinham

# Propriedades do Pai (flex container)

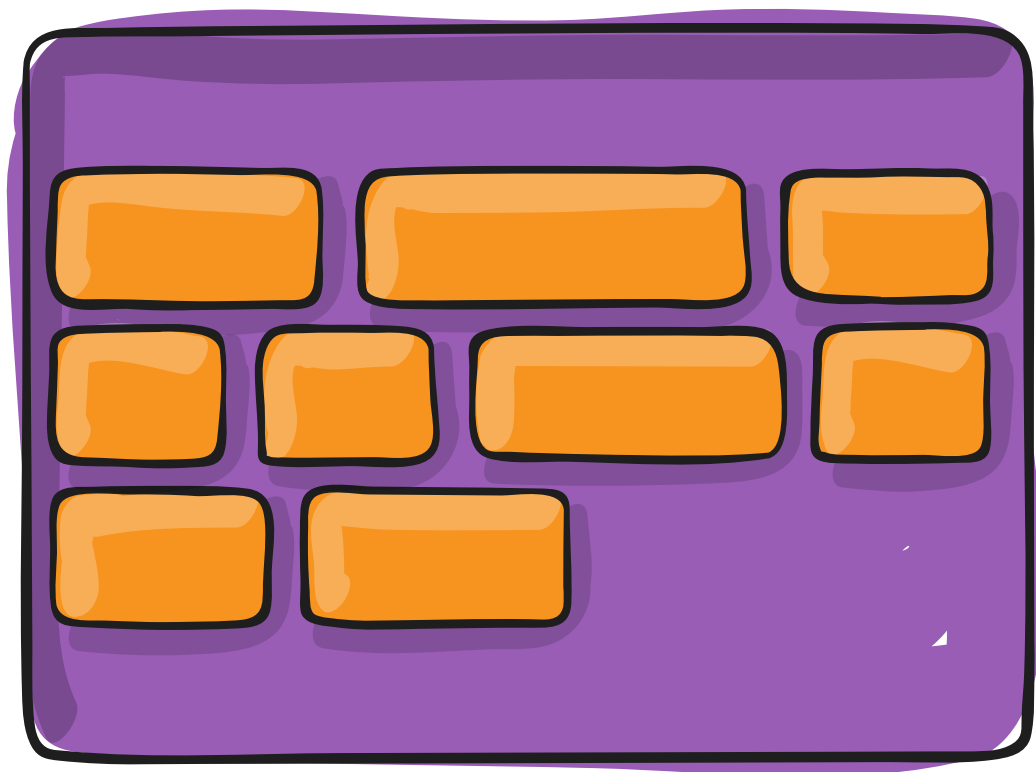
## align-content

Esta propriedade só tem efeito em contêineres de multilinha.

```
.container {  
  align-content: flex-start; /*flex-start | flex-end | center | space-between |  
    space-around | space-evenly | stretch | start | end | baseline*/  
}
```



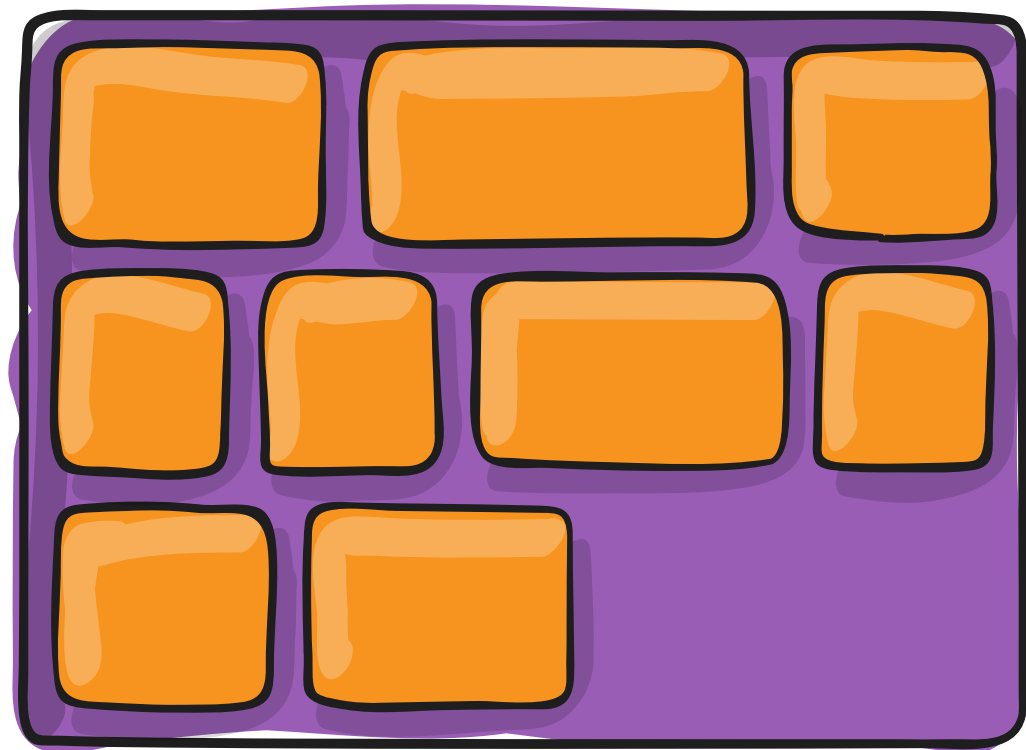
center



space-between



stretch



space-around



# Propriedades do Pai (flex container)

## align-content

- **normal** (padrão - *default*): os itens são compactados em sua posição padrão
- **flex-start / start / self-start**: itens embalados no início do contêiner
- **flex-end / end / self-end**: itens embalados no final do contêiner
- **center**: itens centralizados no contêiner
- **space-between**: itens serão distribuídos uniformemente
- **space-around**: itens serão distribuídos uniformemente com espaço igual ao redor de cada linha
- **space-evenly**: itens serão distribuídos uniformemente com espaço igual ao seu redor
- **stretch**: as linhas se estendem para ocupar o espaço restante

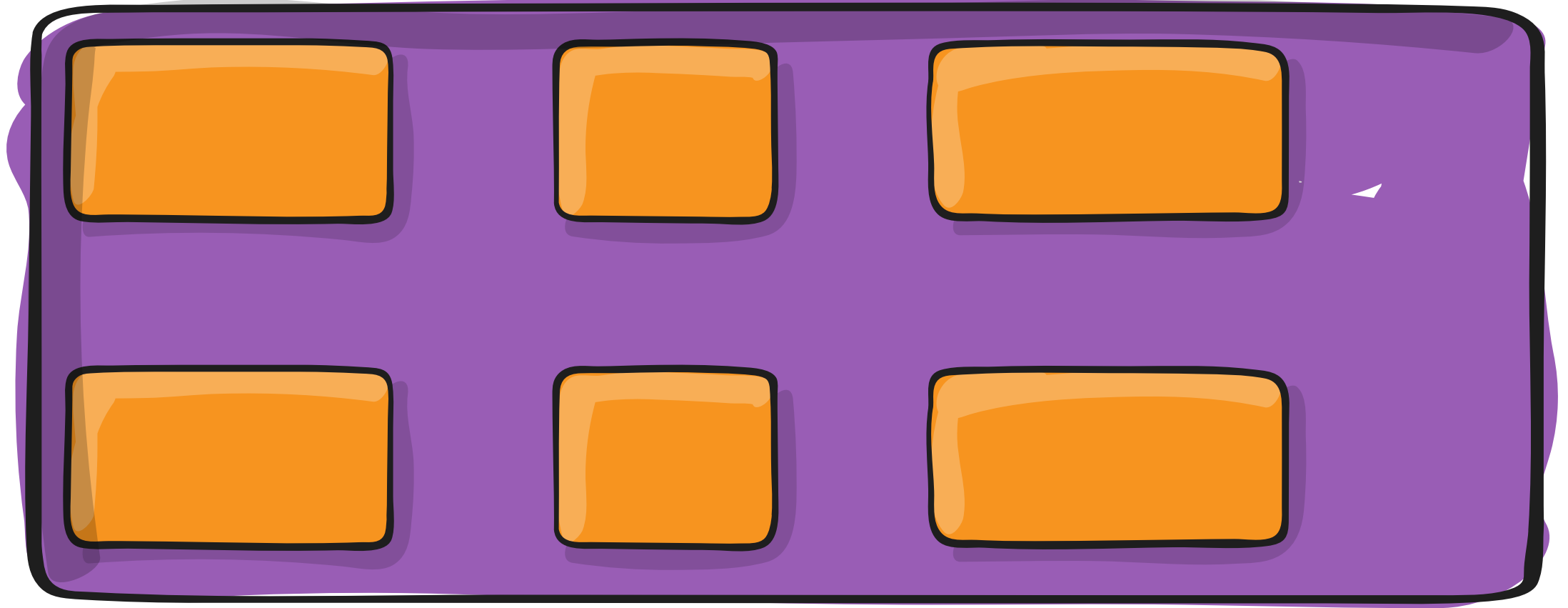


# Propriedades do Pai (flex container)

## gap, row-gap, column-gap

```
.container {  
  display: flex;  
  ...  
  gap: 10px;  
  /* gap: 10px 20px;   row-gap column gap */  
  /* row-gap: 10px;  
     column-gap: 20px; */  
}
```

gap: 30px



gap: 10px 30px

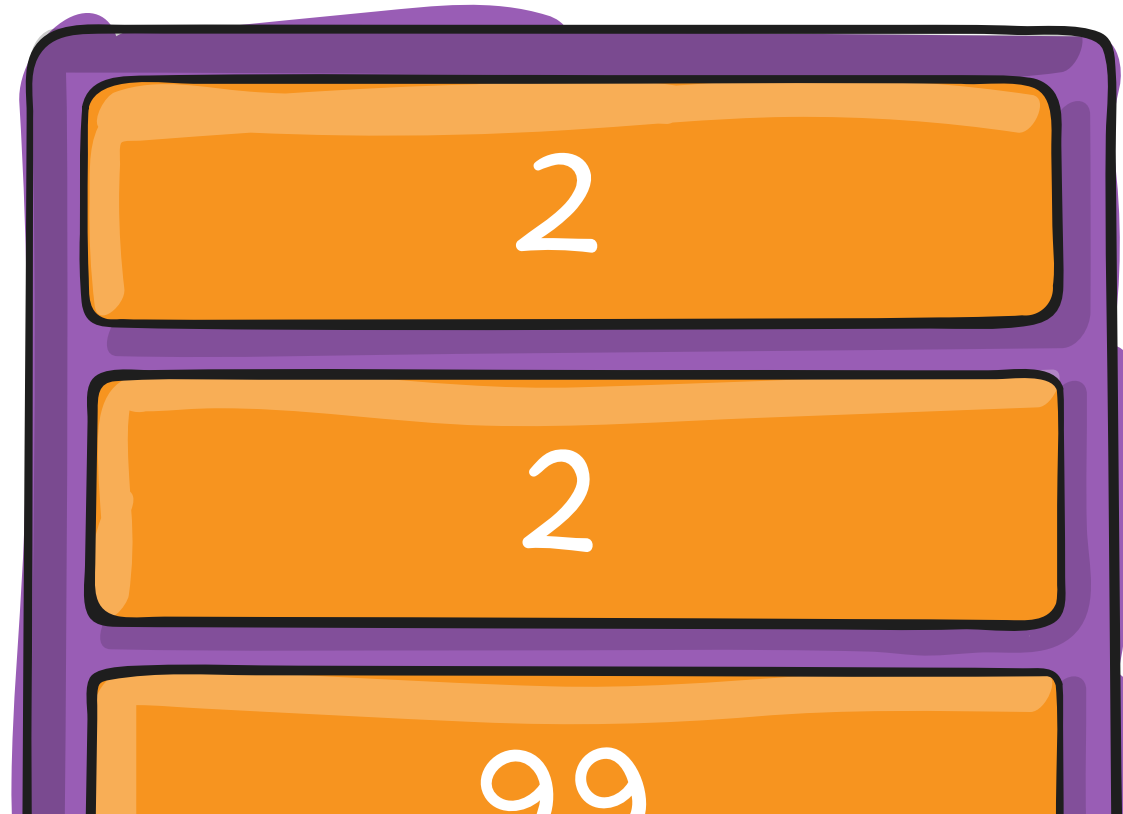


# Propriedades dos filhos (flex items)

## order

Por padrão, os *flex-item*s são dispostos na ordem de origem. No entanto, a propriedade `order` controla a ordem em que aparecem no contêiner.

```
.item {  
  order: 5; /*padrão (default) é 0 */  
}
```

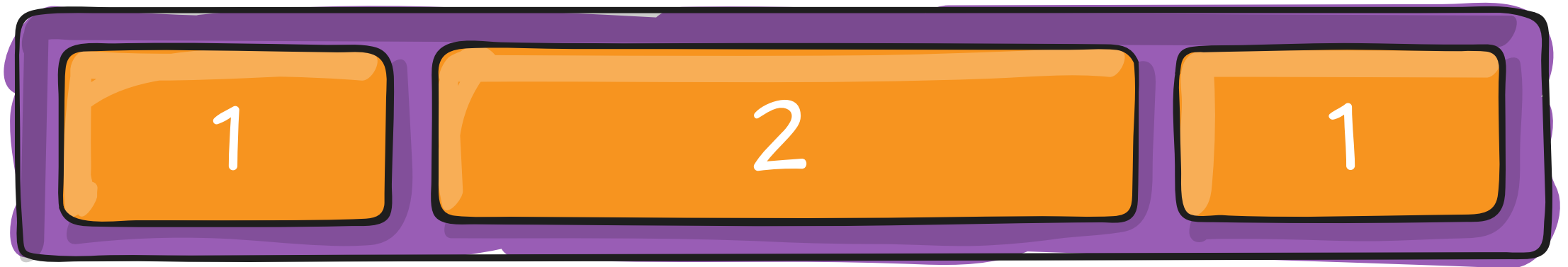
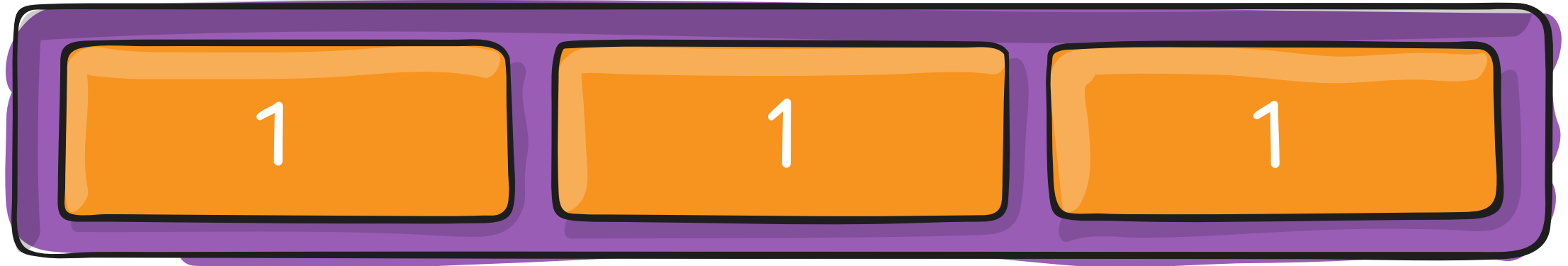


# Propriedades dos filhos (flex items)

## flex-grow

Isso define a capacidade de um item crescer, se necessário.

```
.item {  
  flex-grow: 4; /* default 0 */  
}
```



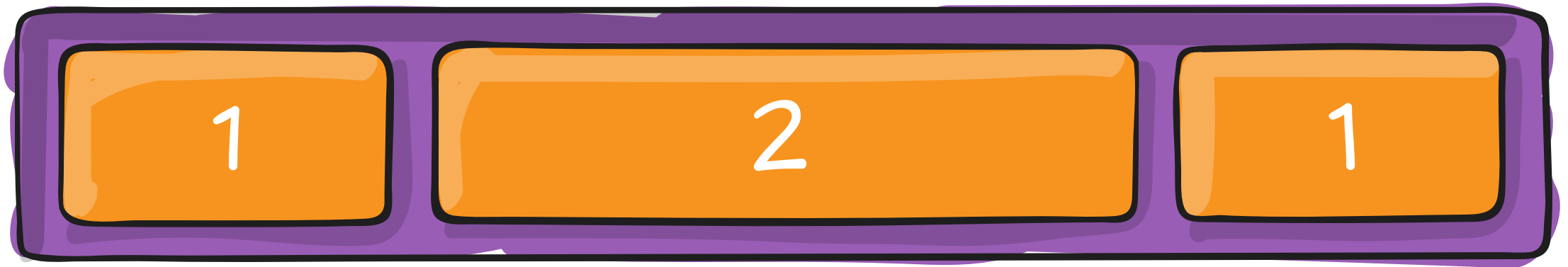
\* <https://css-tricks.com/snippets/css/a-guide-to-flexbox/>

# Propriedades dos filhos (flex items)

## align-self

Isso define a capacidade de um item crescer, se necessário.

```
.item {  
  align-self: auto; /*auto | flex-start | flex-end | center | baseline | stretch*/  
}
```



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



