



Módulo 2

HTML e CSS

< thefutureisblue.me />



Objetivos da aula

CSS
(Cascading Style Sheets)



O que é **CSS**?

CSS (*Cascading Style Sheets*), traduzido para o português, significa Folha de Estilo em Cascatas. O **CSS** é fácil de aprender e entender e é facilmente utilizado com as linguagens de marcação HTML.



Como usar?

```
<!DOCTYPE html>
<html>
  <head>
    <style type="text/css">
      body {
        background-color: blue;
      }

      .titulo {
        font-weight: bold;
        color: white;
      }

      #nome {
        font-weight: bold;
        color: yellow;
      }
    </style>
  </head>
  <body>
    <h1 class="titulo">Olá Mundo</h1>
    <h2 id="nome">Meu nome é Blue</h2>
  </body>
</html>
```

```
<!DOCTYPE html>
<html>
  <head>
    <link rel="stylesheet" type="text/css" href="estilo.css" />
  </head>
  <body>
    <h1 class="titulo">Olá Mundo</h1>
    <h2 id="nome">Meu nome é Blue</h2>
  </body>
</html>
```

```
body {
  background-color: blue;
}

.titulo {
  font-weight: bold;
  color: white;
}

#nome {
  font-weight: bold;
  color: yellow;
}
```

Em um arquivo .css separado

Dentro do arquivo .html





Seletores básicos

- Seletores de tipo
- Seletores de **class**
- Seletores de **id**

```
body {  
  background-color: blue;  
}  
  
.titulo {  
  font-weight: bold;  
  color: white;  
}  
  
#nome {  
  font-weight: bold;  
  color: yellow;  
}
```





Como ficará a nossa página?



Módulo 3 - Front-end

Agenda de hoje

- Introdução ao desenvolvimento web.
- Estrutura HTML.
- Layout de páginas com CSS.

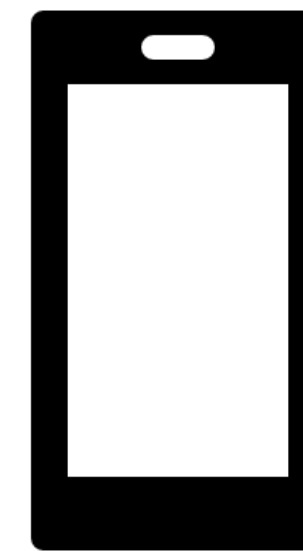
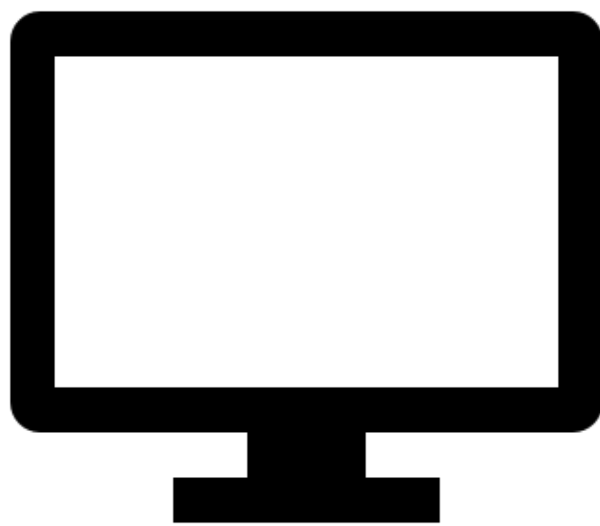
Lista de presença

Aluno	Presente?
Felipe Moraes	Sim
Gabriela Silva	Não

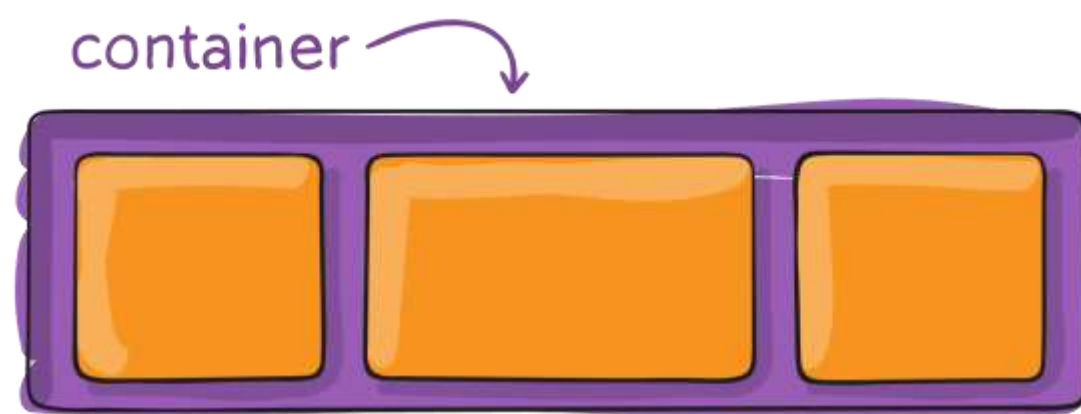


Layout Flex Box

- Alinhar e distribuir items dentro de um container
- Usamos o Flex Box quando o tamanho da página é desconhecido ou dinâmico

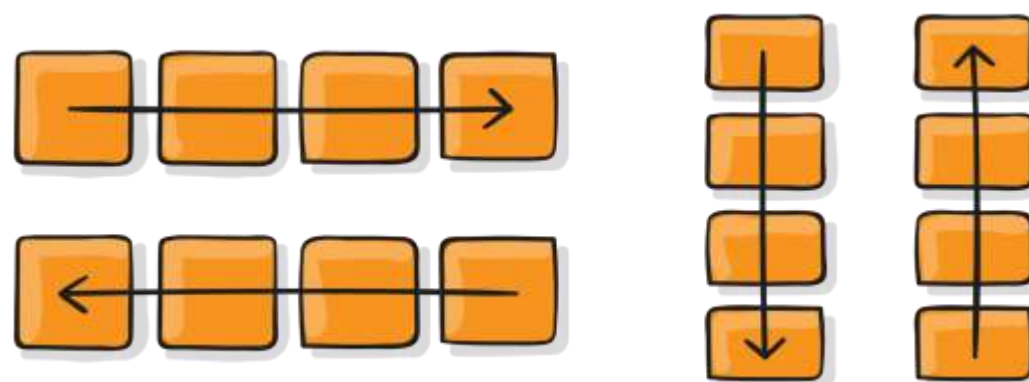


Layout Flex Box Container

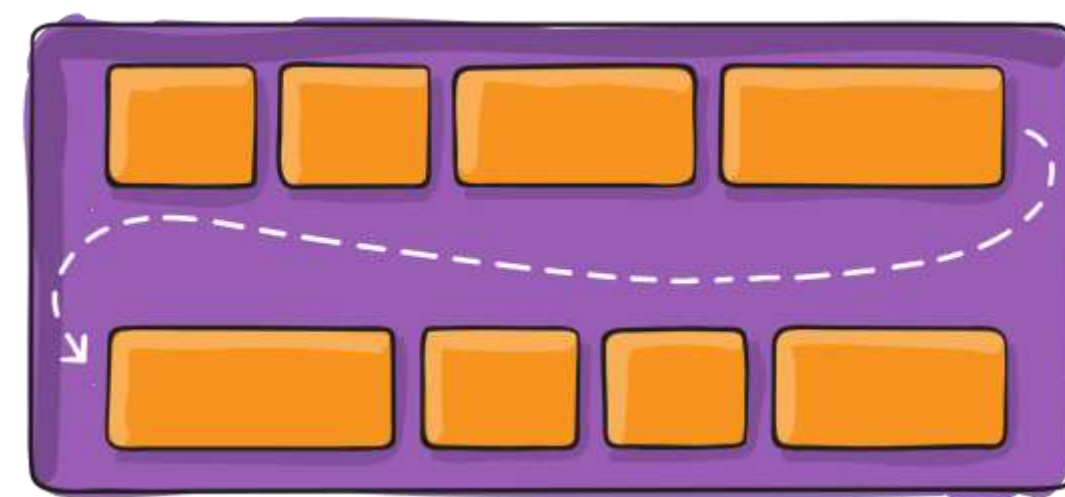


```
.container {  
  display: flex;  
}
```

`<div></div>`



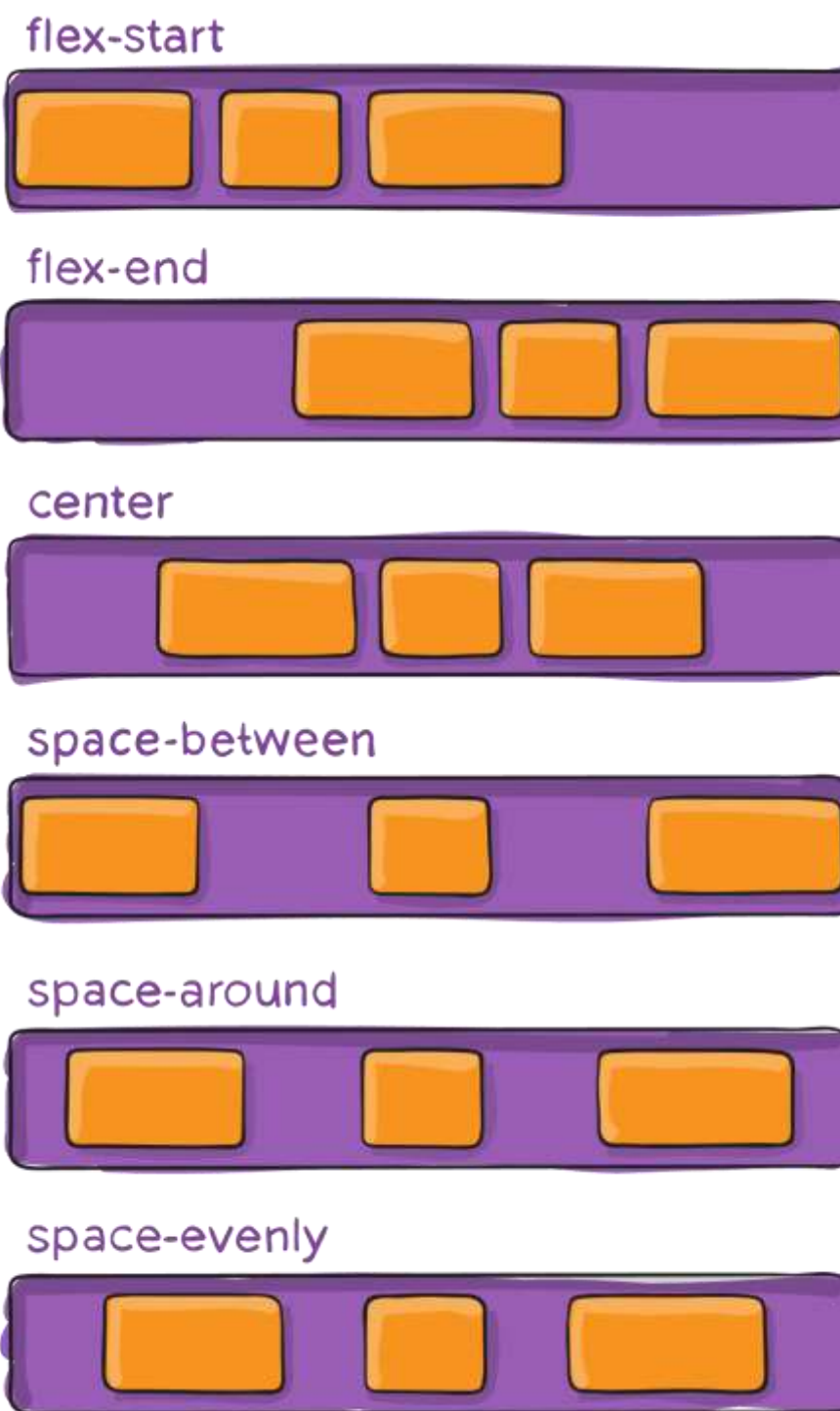
```
.container {  
  flex-direction: row | row-reverse | column | column-reverse;  
}
```



```
.container {  
  flex-wrap: wrap | nowrap | wrap-reverse;  
}
```



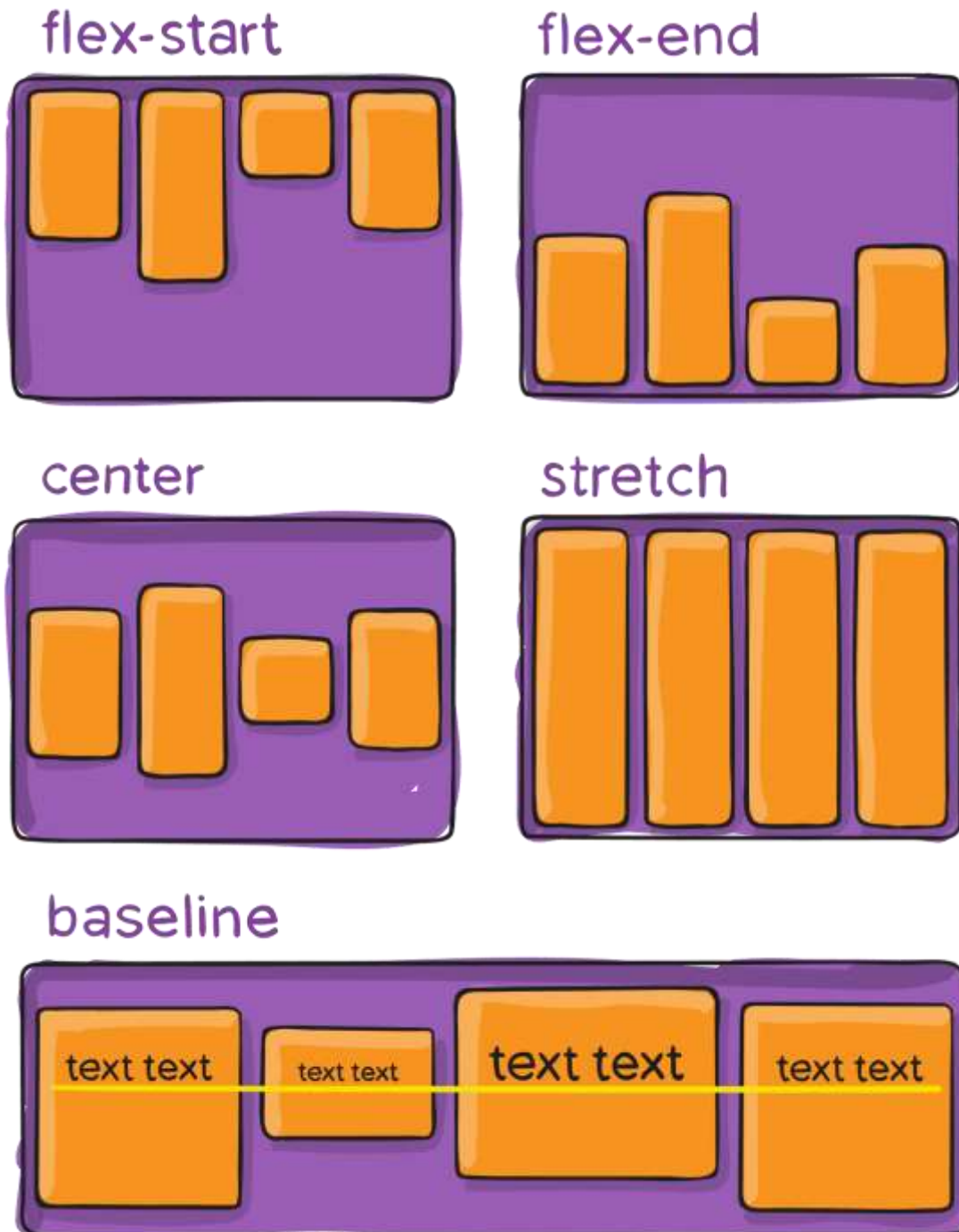
Layout Flex Box Container



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



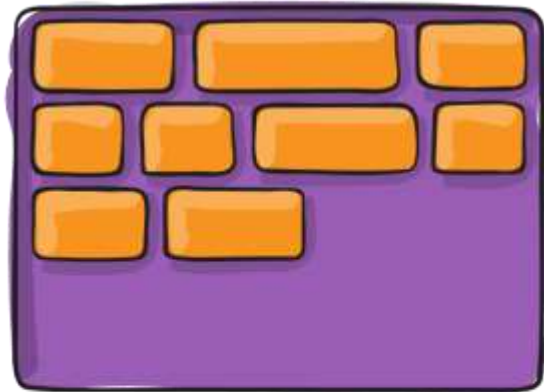
Layout Flex Box Container



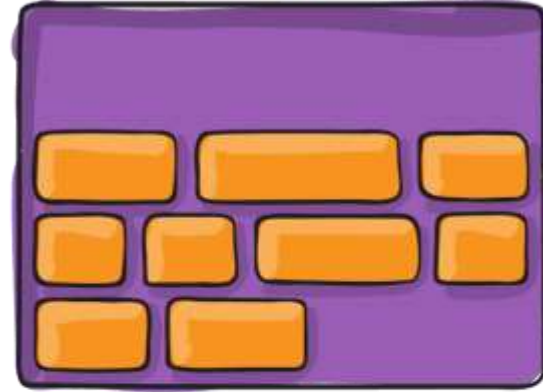
```
.container {  
  align-items: stretch | flex-start | flex-end | center | baseline ;  
}
```

Layout Flex Box Container

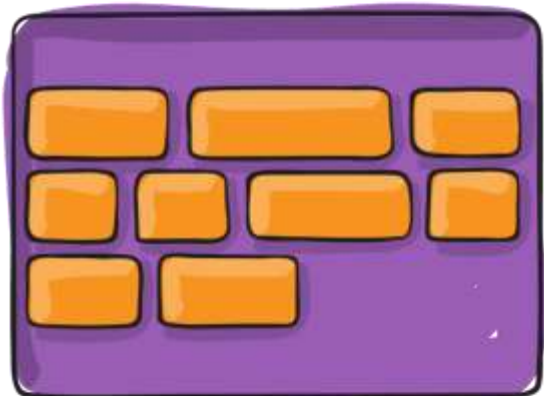
flex-start



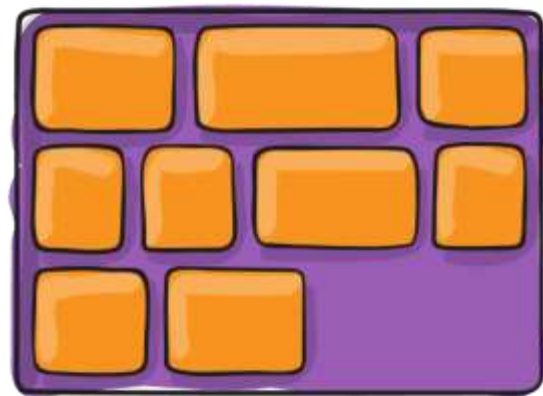
flex-end



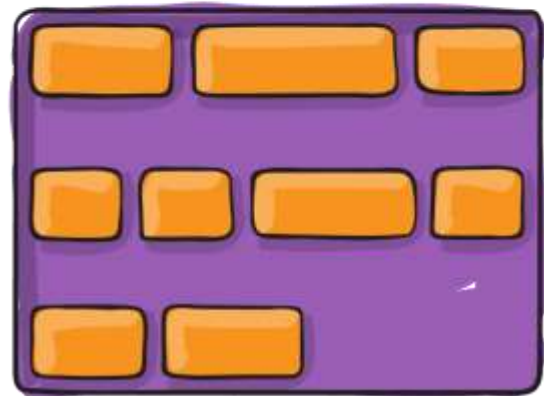
center



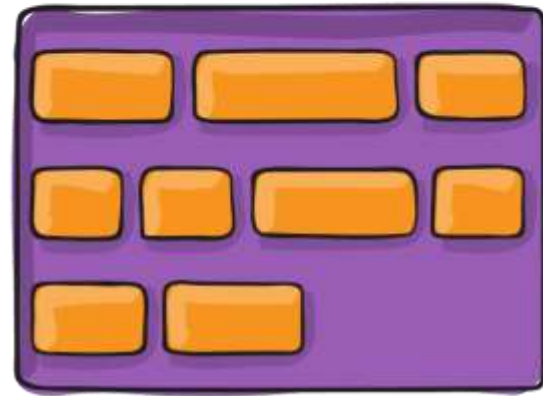
stretch



space-between



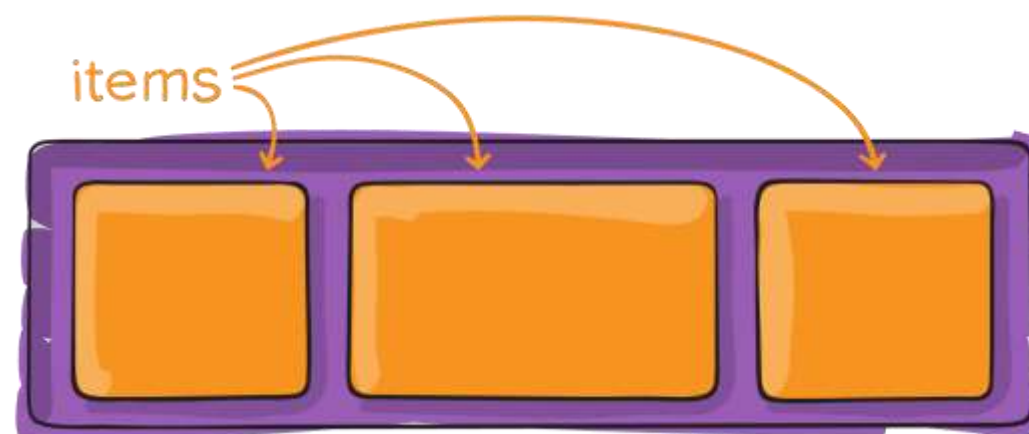
space-around



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

Layout Flex Box

Items

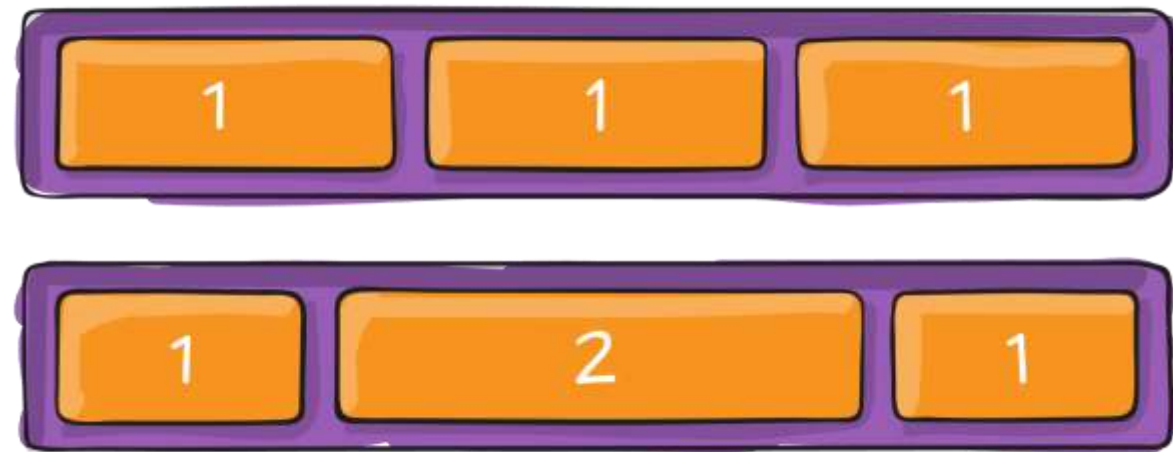


Elementos dentro de um container

```
.item {  
  order: 5;  
}
```

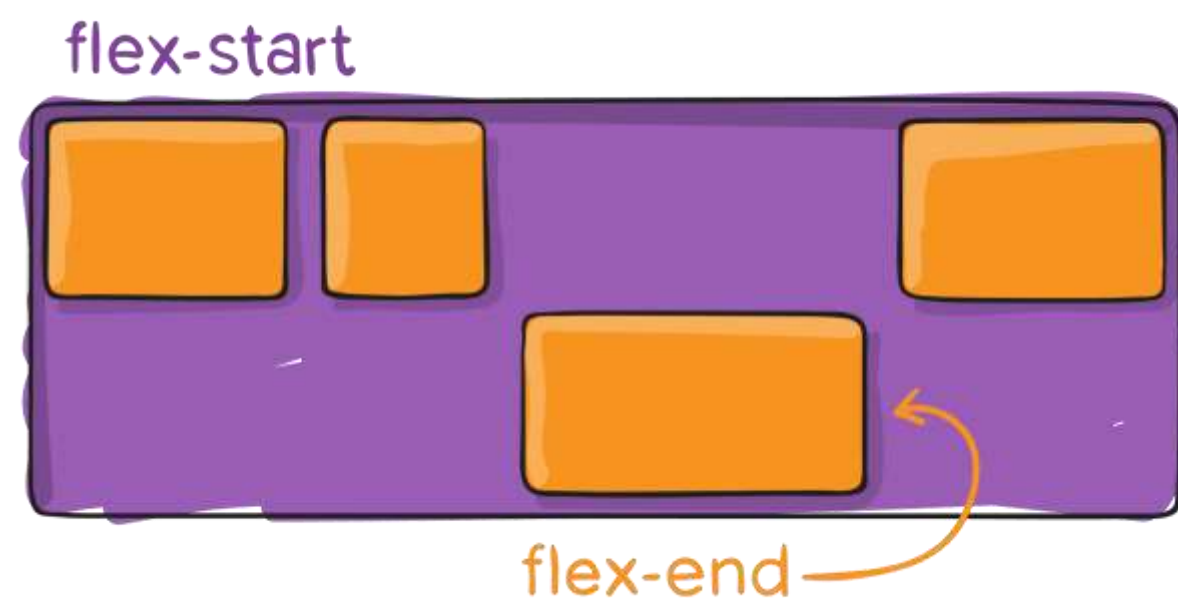
Layout Flex Box

Items



```
.item {  
  flex-grow: 4;  
}
```

```
.item {  
  flex-shrink: 3;  
}
```



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

Layout Flex Box

Referências

- Guia completo de Flex Box (em inglês): <https://css-tricks.com/snippets/css/a-guide-to-flexbox/>
- Flexbox Froggy: <https://flexboxfroggy.com/>

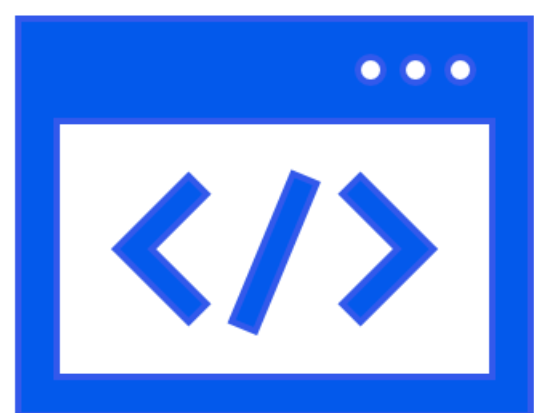
Layout Grid

Para saber mais!

- Guia completo de Grid (em inglês): <https://css-tricks.com/snippets/css/complete-guide-grid/>
- Grid Garden: <https://cssgridgarden.com/>



**Botando pra
rodar**



CSS

**REFERÊNCIAS PARA
CONSULTA**



CSS

Cores no css

Existem diferentes formas de definir a cor de um elemento

- código hexadecimal: #ffc6d9, #ffc5d8
- código rgb: rgb(255,235,0)
- código hsl: hsl(210,100%,40%)
- código hsla: hsla(155,80%,35%,0.4)
- palavra-chave: red, blue, green...etc
- transparente: transparent

O site auxilia na escolha de cores:

https://www.w3schools.com/colors/colors_picker.asp

```
body {  
  background-color: yellow;  
}  
  
#corpo{  
  width: 1000px;  
}  
  
.corpo{  
  background-color: #E6E6FA;  
}
```



Propriedades de Tamanho

- **px** Unidade de pixels. Muita usada em dimensões de blocos e fontes.

Exemplo: `font-size: 20px;`

- **em** Indicada para tamanhos de fontes. Ao usarmos o tamanho padrão do dispositivo do usuário podemos expressar o tamanho da fonte de outros elementos através dessa unidade. Assim um elemento com tamanho de fonte de 2em (`font-size: 2em`) tem duas vezes o tamanho padrão. Caso o tamanho padrão seja 8 pixels, uma fonte de 2 em equivalerá a 16 pixels.

Exemplo: `font-size: 2em;`

- **%** Unidade de porcentagem. Ela é relativa ao bloco onde o elemento está contido. Se um bloco tem 150 pixels e for atribuído à outro bloco contido nele uma largura de 50%, equivalerá a 75 pixels.

Exemplo: `font-size: 120%;`




CSS

Bordas

Border é a propriedade responsável por reunir em uma declaração única os valores das propriedades:

- border-style
- border-width
- border-color


```
#corpo{  
  width: 1000px;  
  margin: 0 auto;  
  border-width: 2px;  
  border-style: solid;  
  border-color:  #2980B9;  
}
```


CSS

Bordas

Border-style: especifica o estilo da borda.

- **none:** nenhuma borda
- **hidden:** equivalente a none
- **dotted:** borda pontilhada
- **dashed:** borda tracejada
- **solid:** borda contínua
- **double:** borda dupla
- **groove:** borda entalhada
- **ridge:** borda em ressaltado
- **inset:** borda em baixo relevo
- **outset:** borda em alto relevo

```
#corpo{  
    width: 1000px;  
    margin: 0 auto;  
    border-width: 2px;  
    border-style: solid;  
    border-color:  #2980B9;  
}
```

CSS

Bordas

Border-width: especifica a espessura.

- **thin:** borda fina
- **medium:** borda média
- **thick:** borda grossa
- **length:** unidade de medida CSS
(px, pt, em, cm, ...)

```
#corpo{  
  width: 1000px;  
  margin: 0 auto;  
  border-width: 2px;  
  border-style: solid;  
  border-color:  #2980B9;  
}
```


Por hoje é só!
Obrigado! =)