

17 de à 27 de Junho

# React Descomplicado: Fundamentos do front-end moderno

By: Marcos Renê  
&





Instrutor

**Marcos Renê**

Desenvolvedor web

## Apresentação simples



Desenvolvedor web com 3 anos de experiência em React. Atualmente, trabalho como desenvolvedor fullstack em projetos freelancers.

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

# CRONOGRAMA DAS AULAS

## Semana 1

Data	Assunto
17/06	Aula 01: Nivelamento da turma com projeto utilizando HTML, CSS e Javascript
18/06	Aula 02: Introdução ao React e início do projeto
19/06	Aula 03: Desenvolvimento do projeto
20/06	Aula 04: Aplicando os conhecimentos em React

# CRONOGRAMA DAS AULAS

## Semana 2

Data	Assunto
24/06	Aula 05: Como funciona uma API
25/06	Aula 06: Integração do projeto com API
26/06	Aula 07: Integração do projeto com API
27/06	Aula 08: Ferramentas de testes e encerramento do curso.

# Aula 01

**Nivelamento da turma - Fundamentos de  
HTML, CSS e Javascript**



# Aula 01 - Nivelamento da turma

O que vamos ver hoje?

- 👉 HTML - Entender a estrutura
- 👉 CSS - Como funciona a estilização
- 👉 JavaScript - Revisão básica e como manipular os elementos da página com JS

Ao final da aula teremos uma tela de login implementada apenas com HTML, CSS e JavaScript.

**Gerencie  
seus  
horários.**

[Cadastre-se aqui](#) 🖱️

ENTRAR





# O que é HTML?

HTML (Hypertext Markup Language), é a linguagem padrão para a criação e estruturação de páginas web.

É fundamental, pois define a estrutura básica e o conteúdo de uma página web, permitindo a organização e apresentação dos elementos visuais e textuais.

# Estrutura base do HTML

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Título da nossa página</title>
</head>
<body>

</body>
</html>
```

# Tags no HTML

Tags são elementos que especificam diferentes tipos de conteúdo em um documento HTML.

```
<!--# Paragrafo-->
<p>Lorem ipsum dolor sit amet consectetur adipisicing elit.</p>

# Título
<h1>Lorem ipsum</h1>

# Link
<a href="https://google.com" target="_blank">Lorem ipsum</a>
```

**Praticando...**



# O que é CSS?

CSS (Cascading Style Sheets) é uma linguagem de estilo usada para descrever a apresentação de um documento escrito em HTML ou XML.

A principal função do CSS é separar o conteúdo de um documento HTML da sua apresentação visual.

## Formas de incluir o CSS em um documento HTML

- ➡ Inline
- ➡ Incorporado
- ➡ Externo

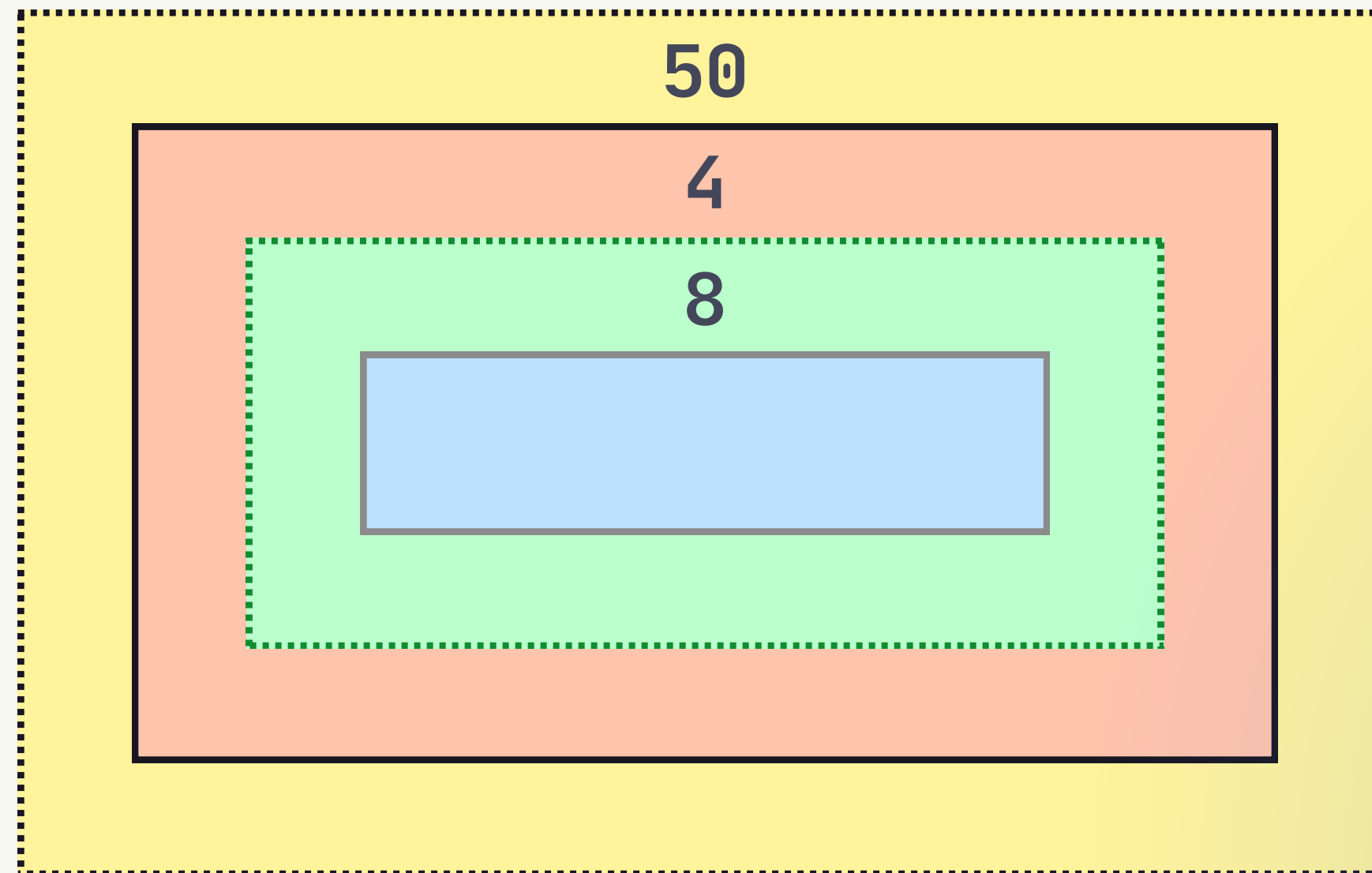
# Seletores CSS

Os seletores em CSS são padrões usados para selecionar os elementos HTML aos quais queremos aplicar estilos.

```
<1  /* Global */  
2  * {}  
3  
4  /* Elemento */  
5  h1 {}  
6  
7  /* Id */  
8  #button {}  
9  
10 /* Classe */  
11 .header {}
```

# Box Model

O Box Model que descreve como os elementos HTML são renderizados e ocupam espaço em uma página da web.

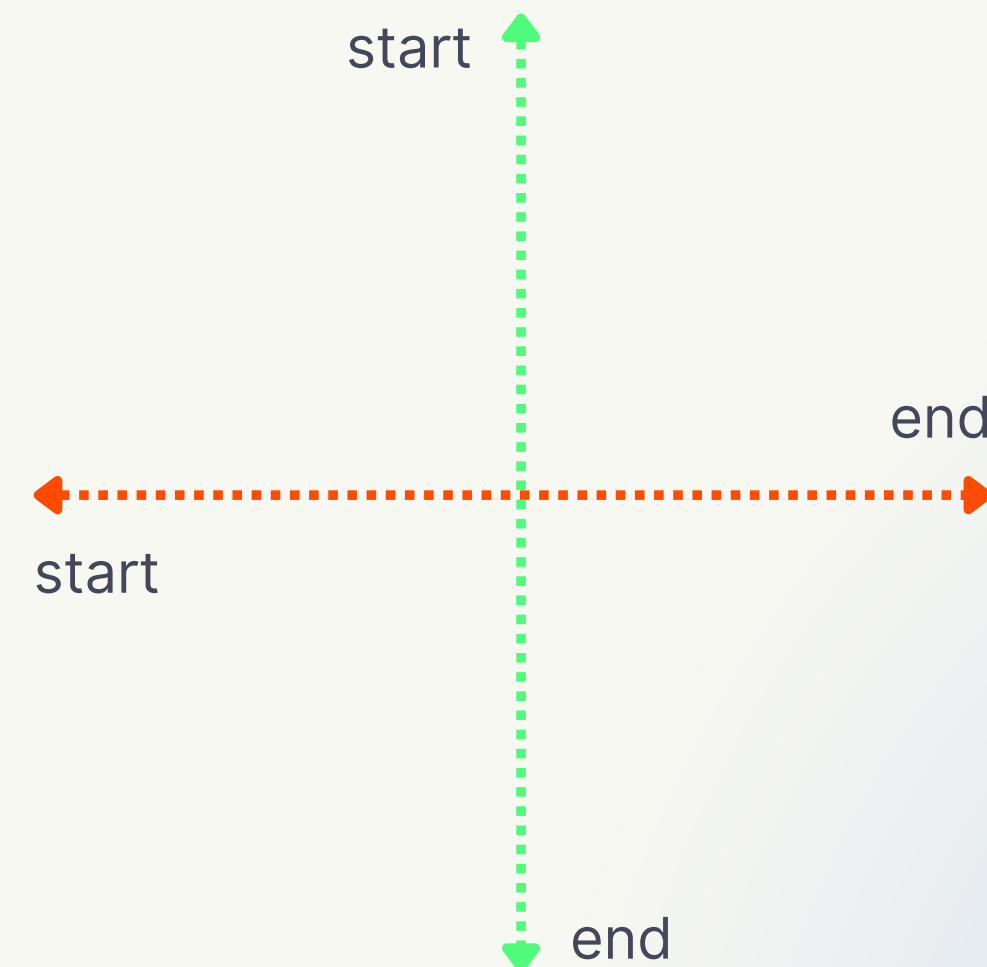
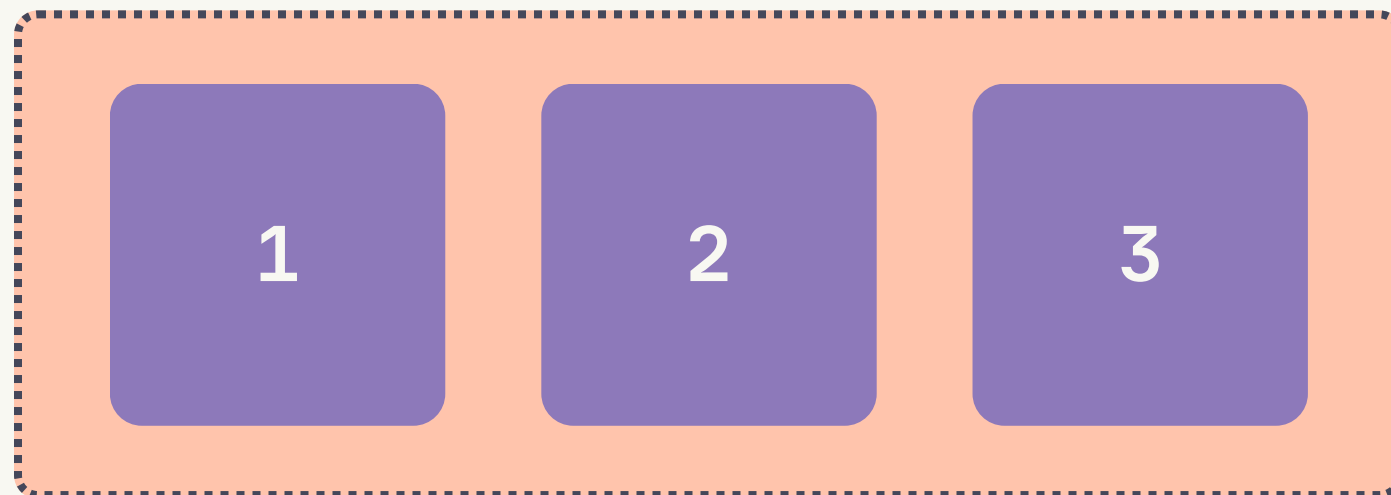




# Flexbox

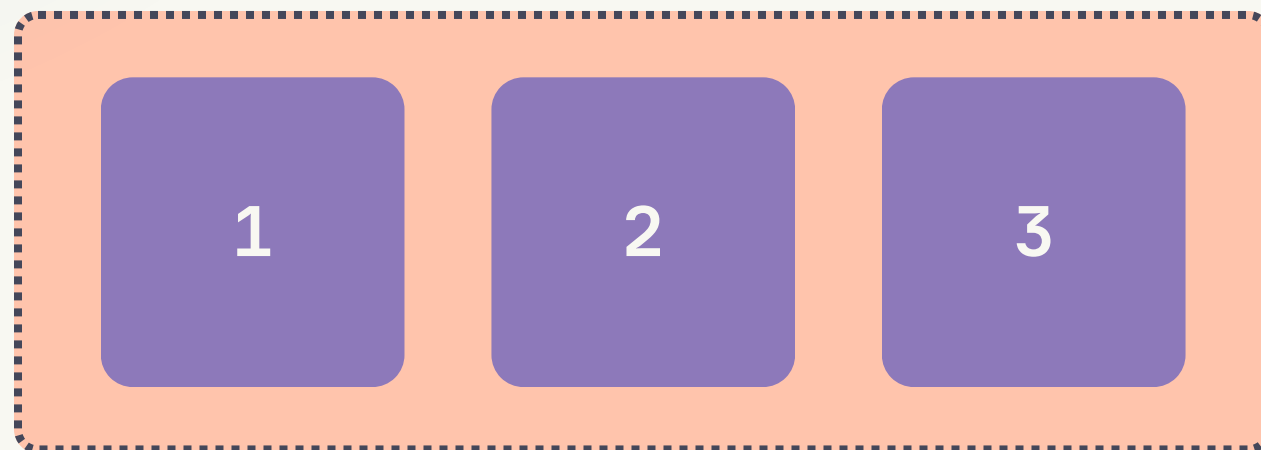
Flexible Box Layout, é uma técnica de layout em CSS que permite criar layouts mais flexíveis e dinâmicos em páginas HTML.

## Flex container / Flex item

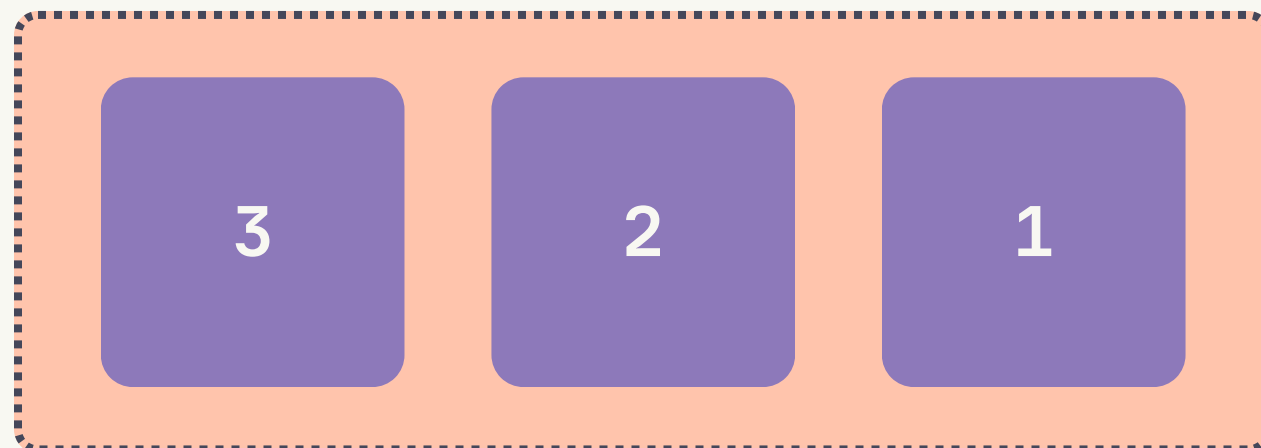


# Flexbox (Direcionamentos)

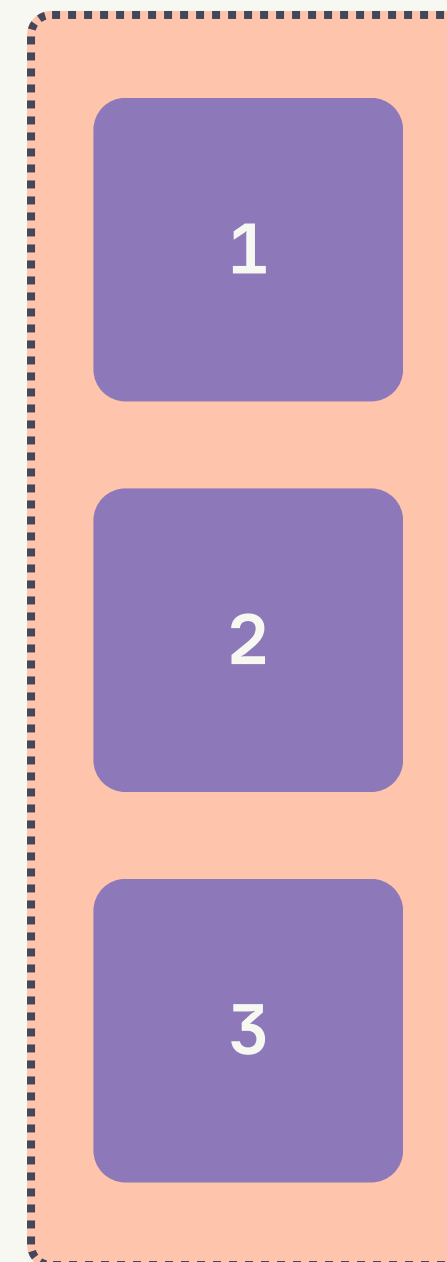
**row**



**row-reverse**



**column**

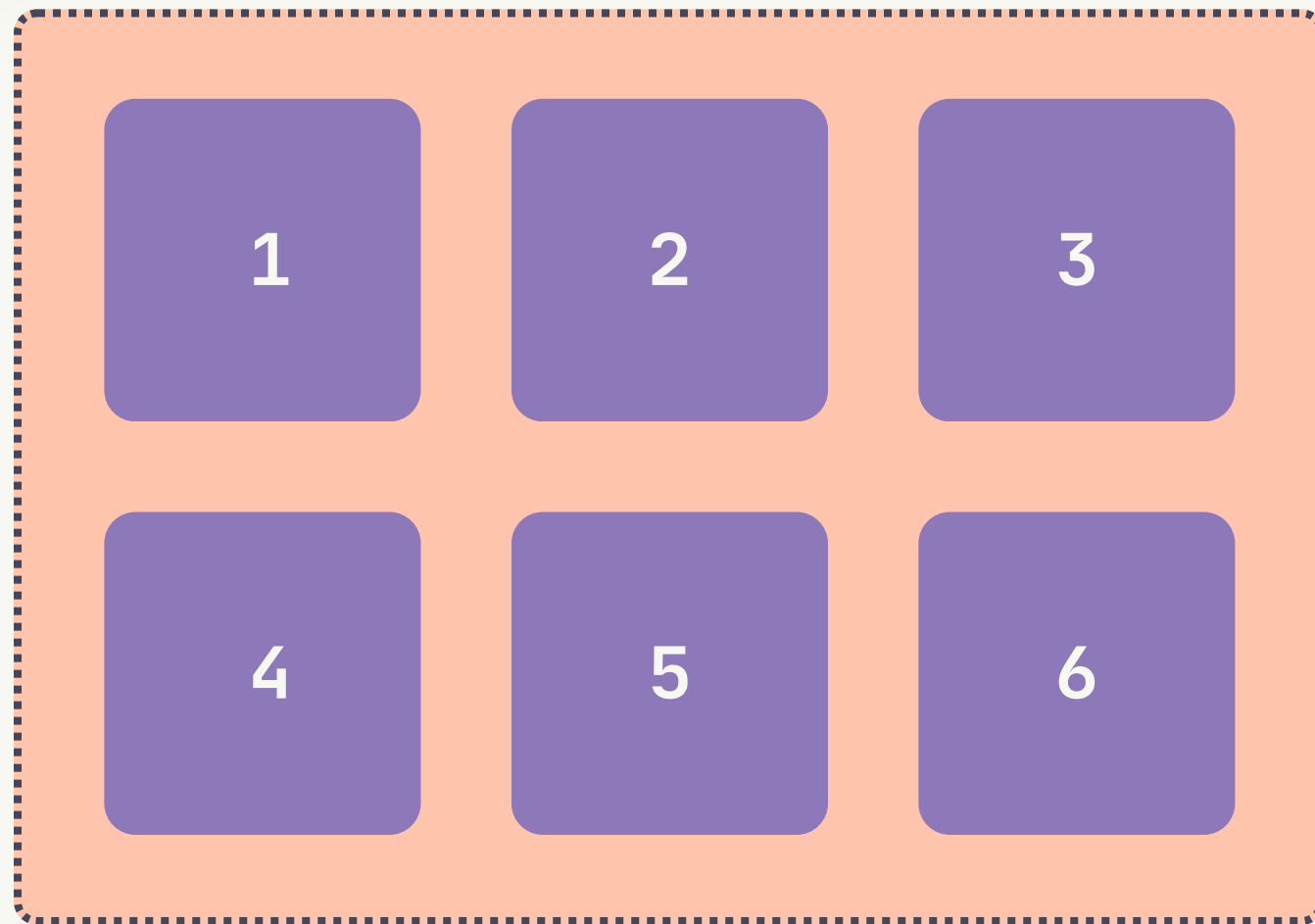


**column-reverse**

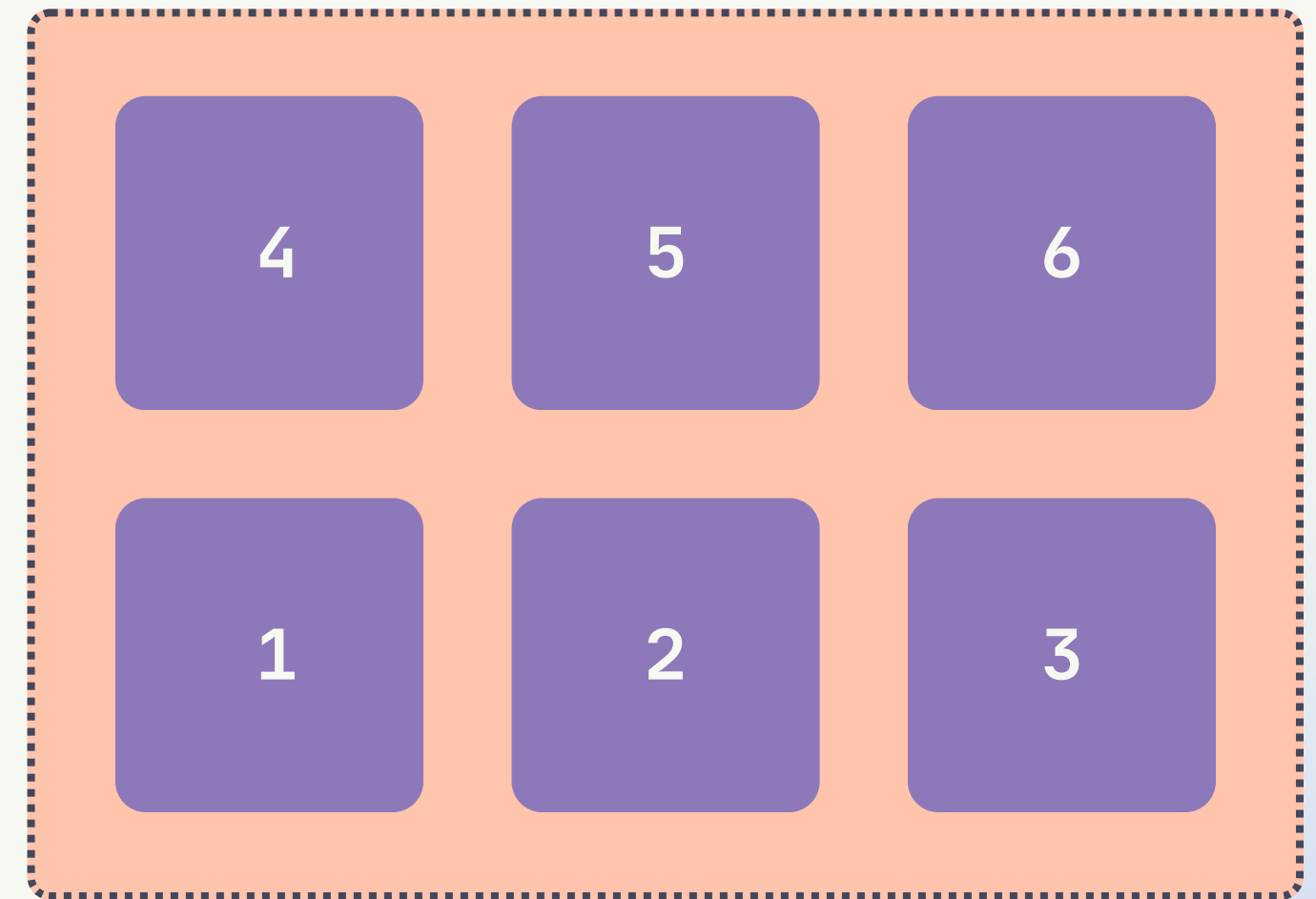


# Flexbox (Mult linhas)

**wrap**

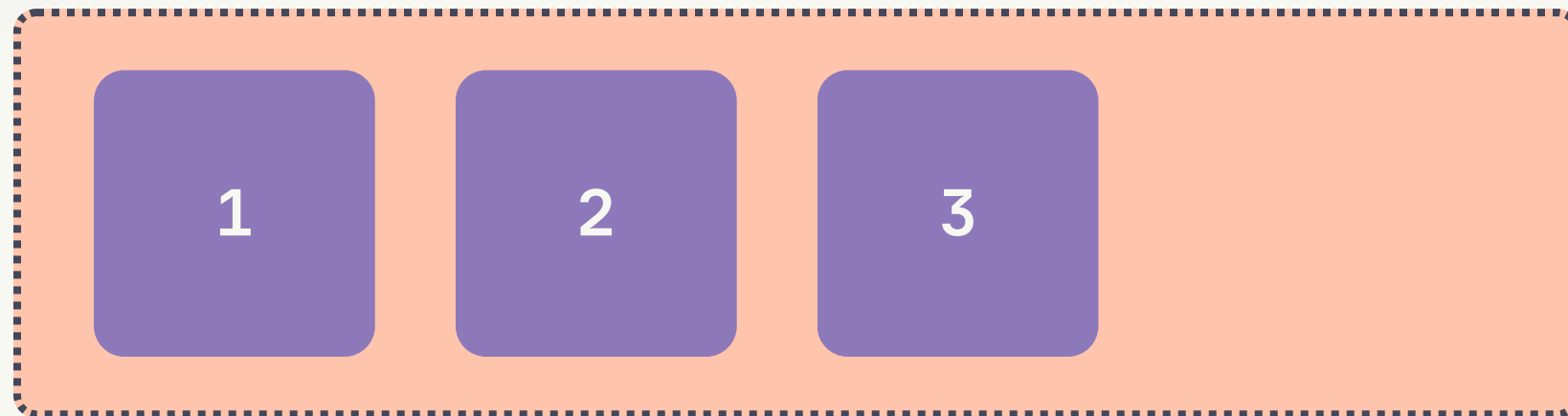


**wrap-reverse**

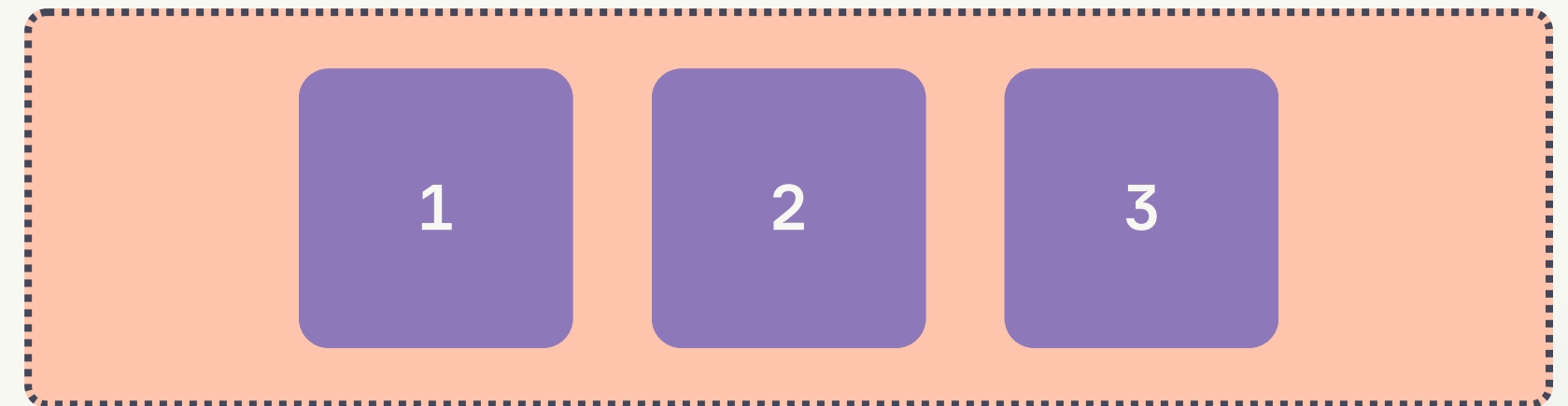


# Flexbox (Alinhamentos eixo x)

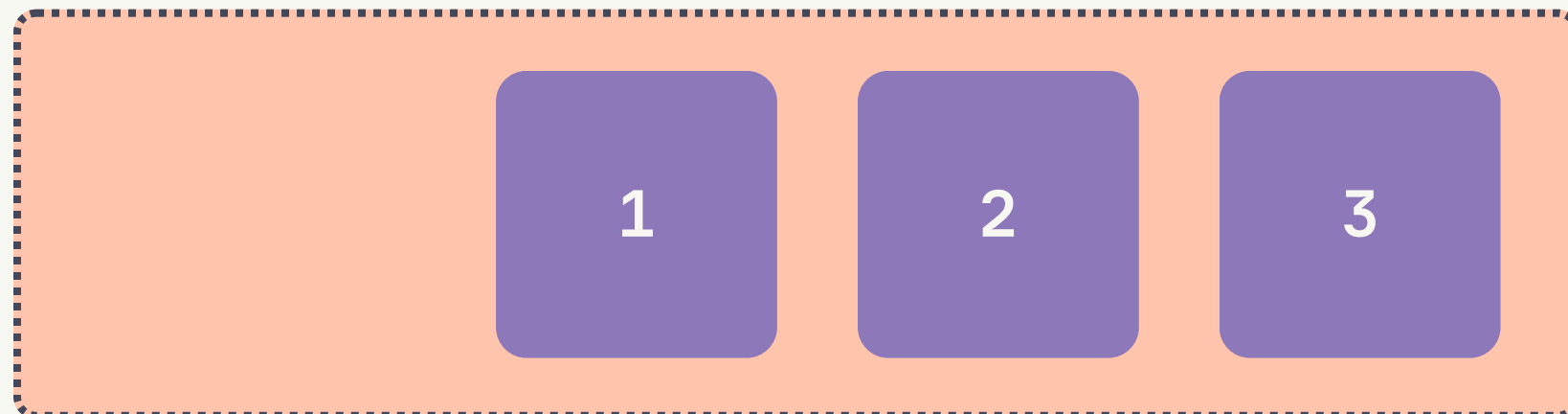
**flex-start**



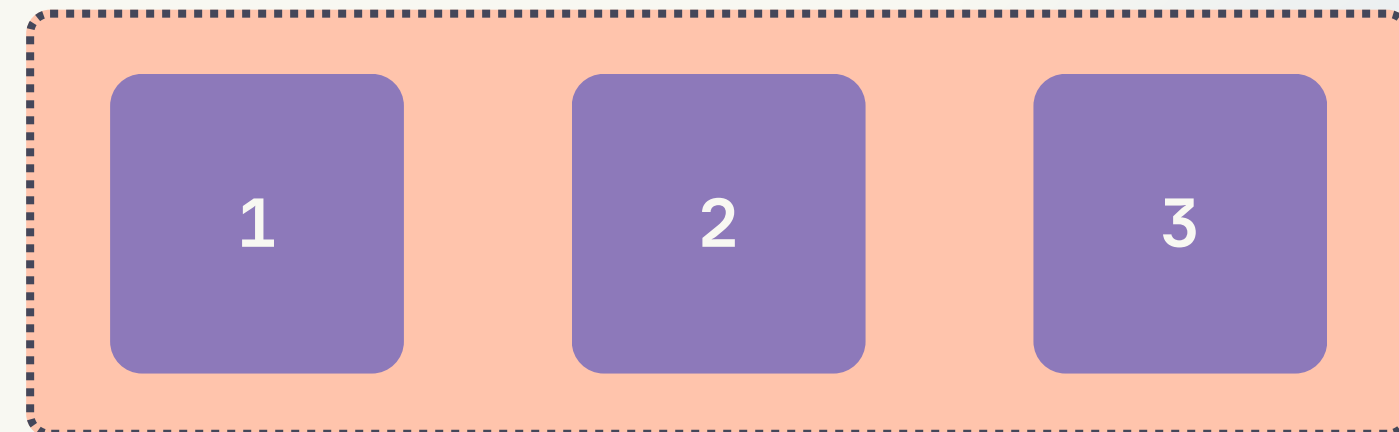
**center**



**flex-end**

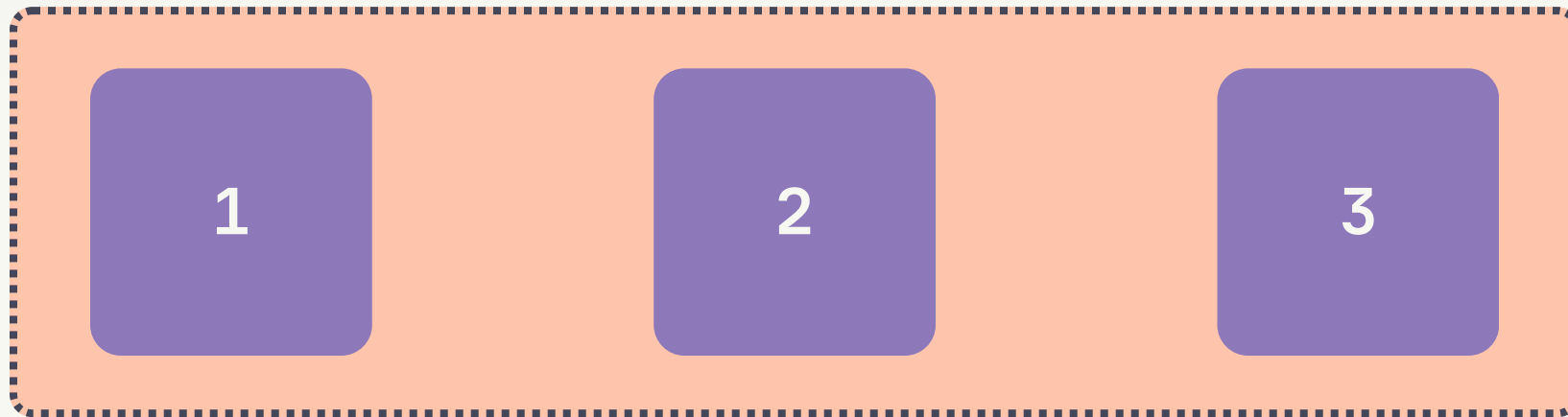


**space-around**

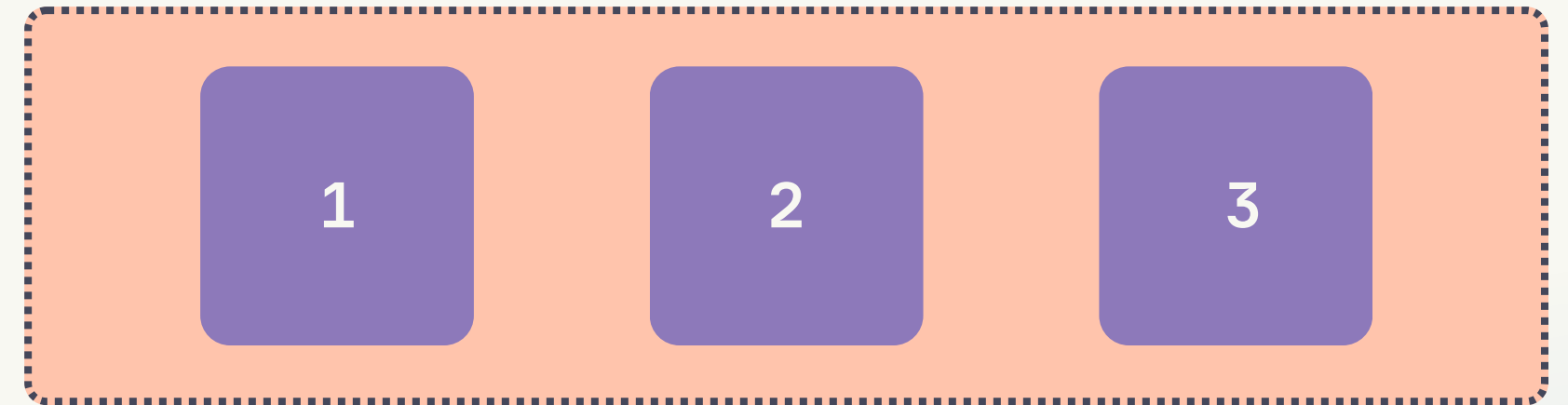


# Flexbox (Alinhamentos eixo x)

**space-between**

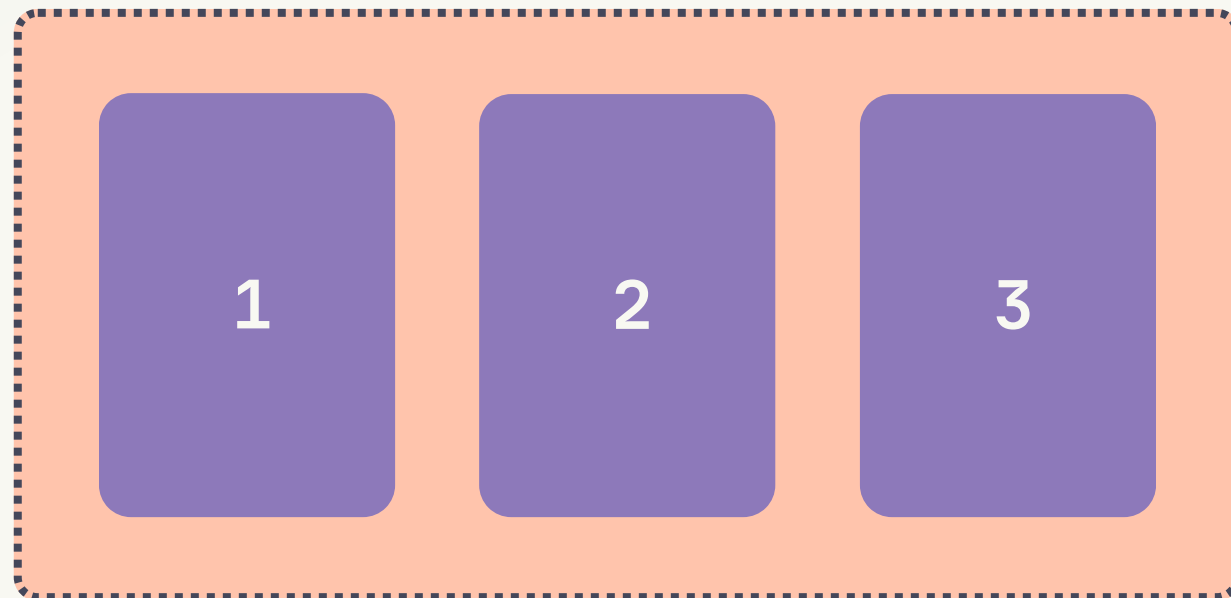


**space-evenly**

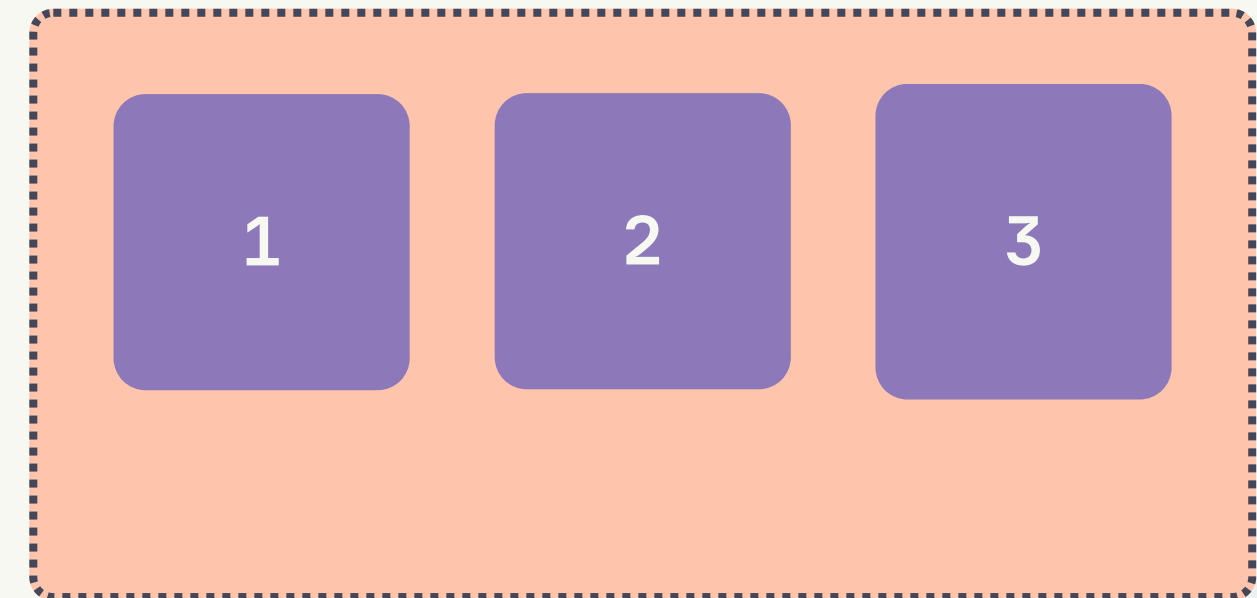


# Flexbox (Alinhamentos eixo y)

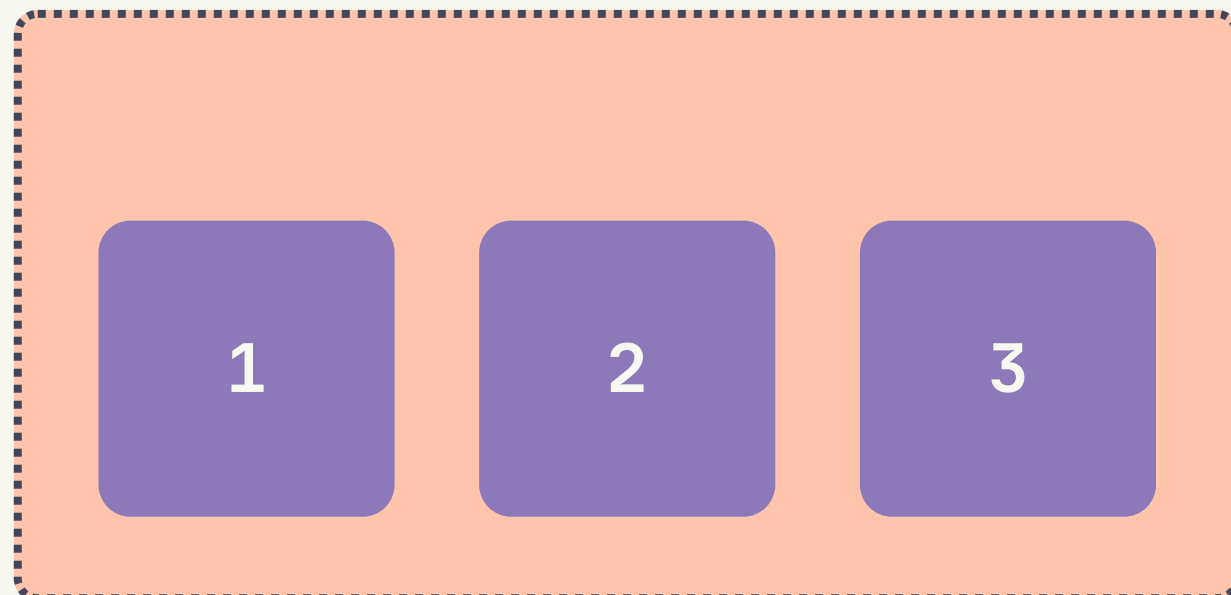
**stretch**



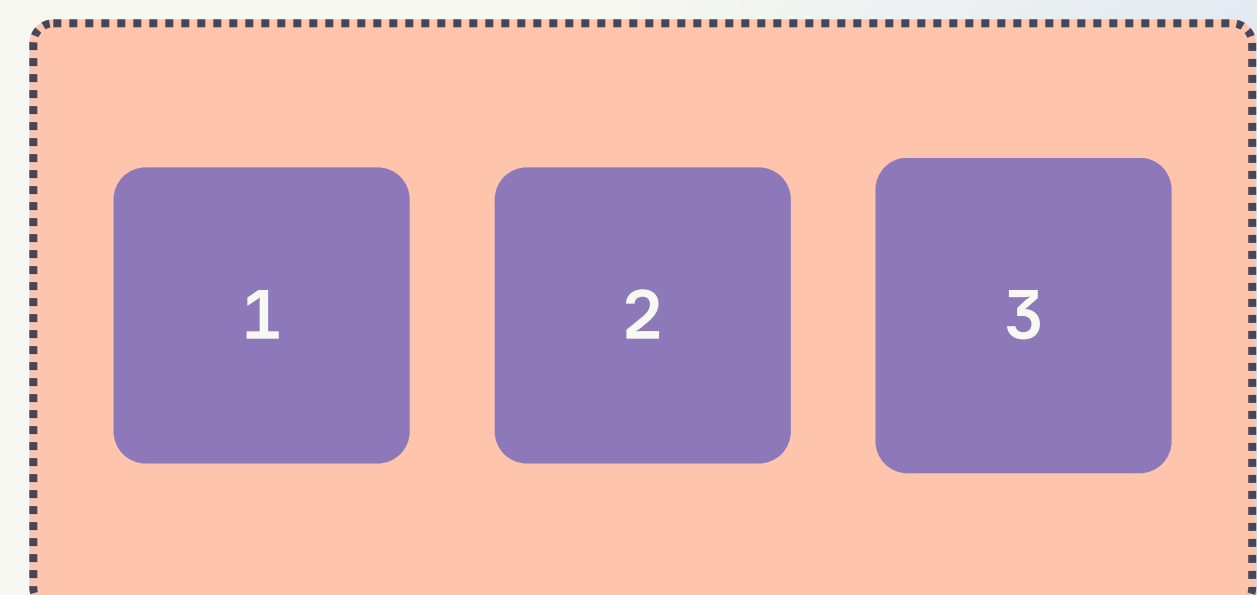
**flex-start**



**flex-end**

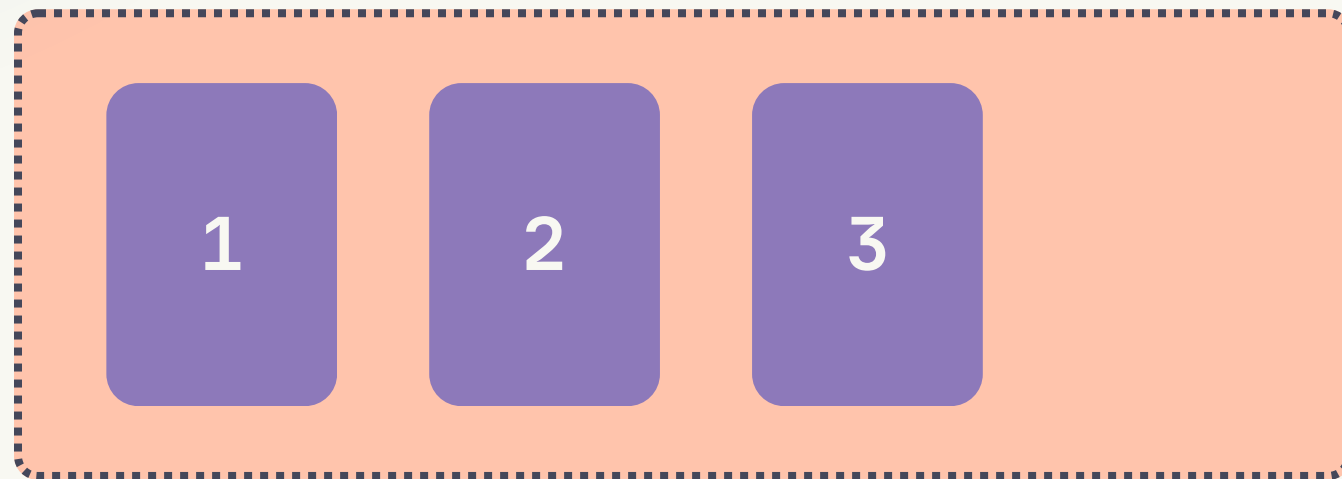


**center**

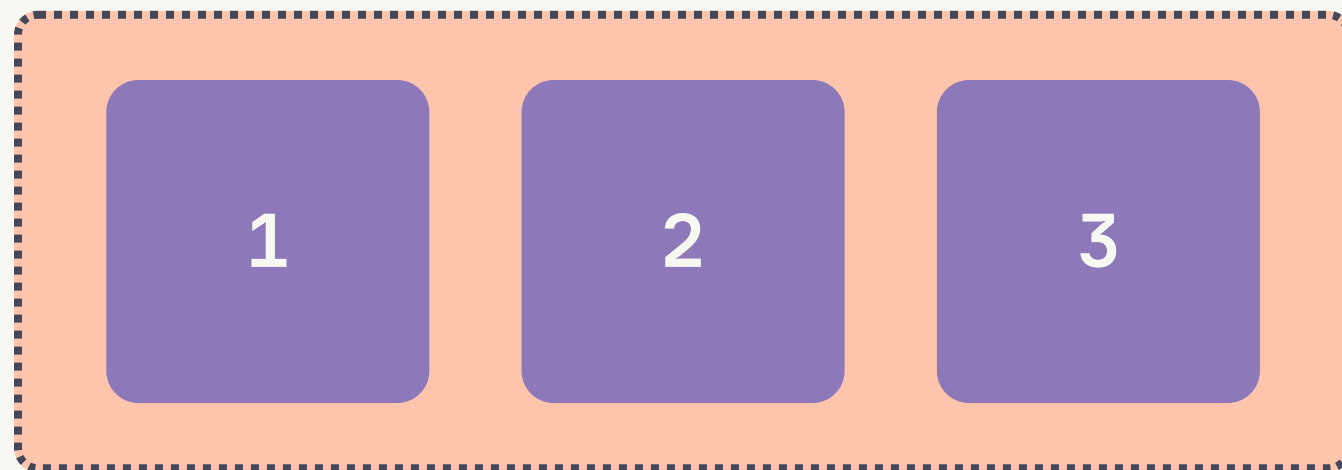


# Flexbox (Propriedades para os itens)

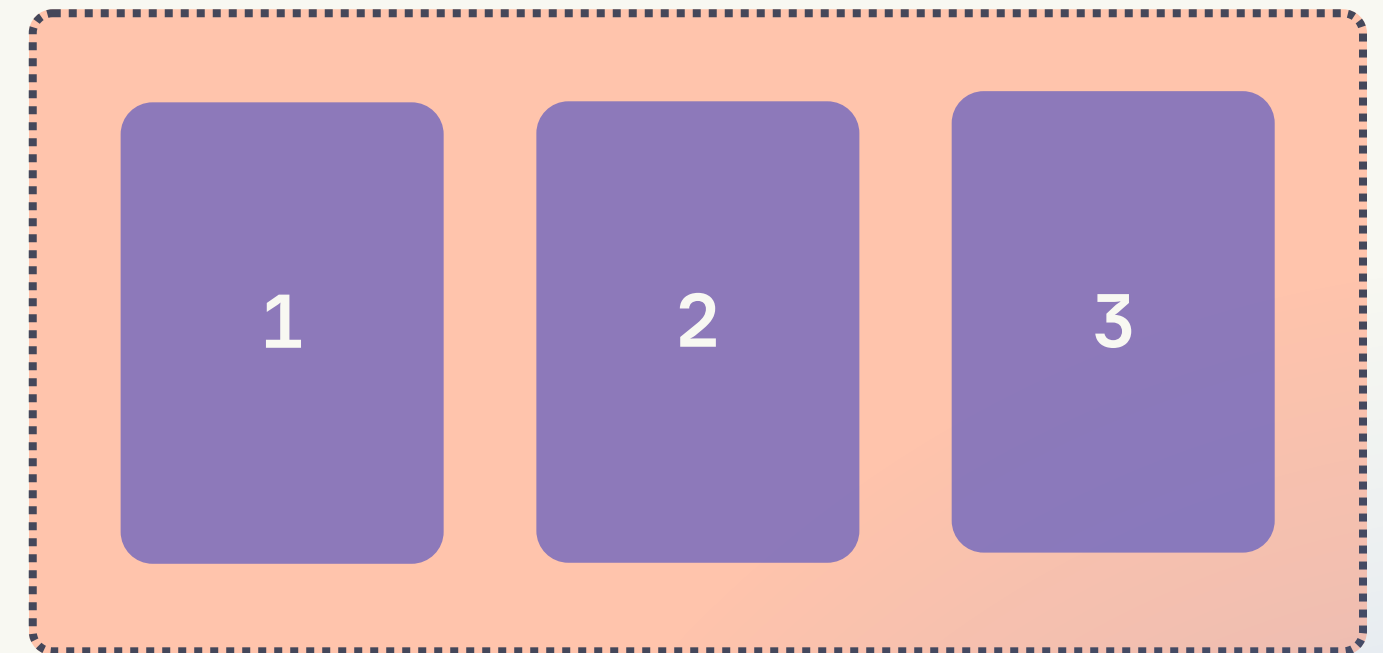
**flex-basis: 100px**



**flex-shrink: 0**



**flex-grow: 280px**



# Flexbox (Propriedades para os itens)

order



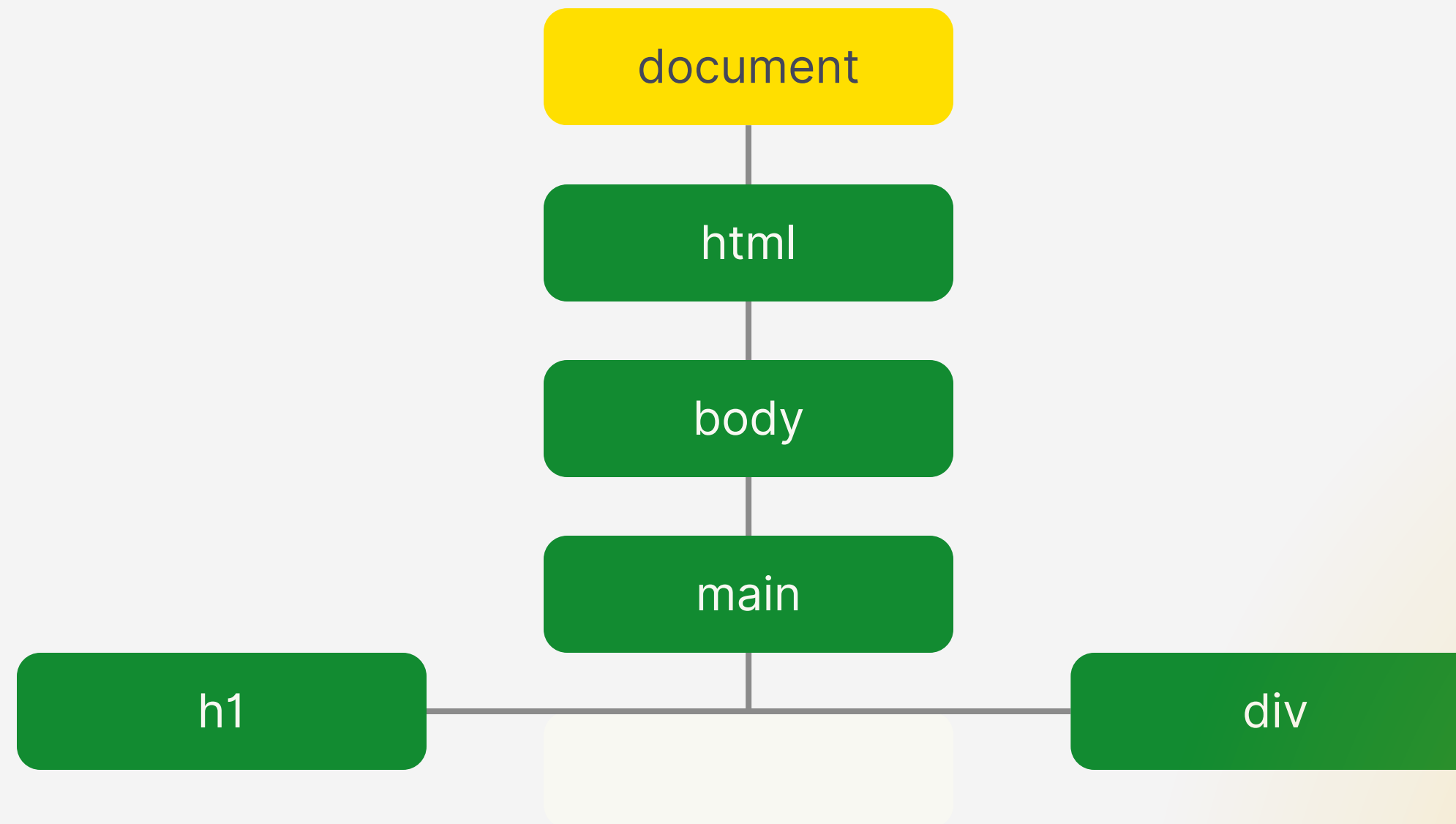


**Praticando...**



# DOM (Document Object Model)

É a representação de dados dos objetos que compõem a estrutura e o conteúdo de um documento na Web.



```
<!DOCTYPE html>
<html dir="ltr" lang="pt-BR" class="theme dark EHoceA">
  ><head> ... </head>
  ▼<body class="preload-pos-fixed">
    ><div class="NqU4Yg" style="z-index: 5;"> ... </div>
    ><script nonce> ... </script>
    ▼<div id="root">
      ▼<div class="yIDCqA">
        ▼<div class="yIDCqA">
          ▼<div class="SEkd2Q YRVZMA" tabindex="-1"> flex
            <section class="L6HKTQ"></section>
            ▼<div class="IYiBuQ"> flex
              ><div class="FmtbLQ ObBgBw"> ... </div>
              ><div class="FmtbLQ ObBgBw"> ... </div>
              <div class="A7Jzyg"></div>
            ▼<div class="NRPtWA"> flex
```

# O que é o JavaScript?

É uma **linguagem de programação** amplamente utilizada em desenvolvimento web.  
É uma linguagem de **script**, o que significa que é executada no browser do cliente.

```
1  function decrementQuantity() {  
2    let currentValue = parseInt(productsAmount.value);  
3    if (currentValue > 0) {  
4      productsAmount.value = currentValue - 1;  
5    }  
6  }
```

# Tipo de dados

Tipos de dados são classificações que definem o tipo de valor que uma variável pode conter.

**String** - "Olá"

**Number** - 1, 100, 1000, 1.0, NaN

**Boolean** - true, false

**Object** - {  
    name: "Fiat toro",  
    color: "black",  
    model: "volcano"  
}

**Array** - ["Olá", "Mundo"], [1, 2, 3], [{ }, { }]

**Undefined** - var name;

**Null** - { }

# Controle de fluxo

Capacidade de direcionar o fluxo de execução do código com base em certas condições.

```
<1  if(condição) { } else { }
2
3
4  switch(expression) { case 1: ... }
5
6
7  throw e try { } catch(error) { }
8
```

# Estruturas de repetição

É um recurso que permite executar um bloco de código várias vezes, enquanto uma condição específica for verdadeira.

```
1  for(i = 0; i < 10; i++) { }  
2  
3  
4  while(i < 10) { }  
5  
6  
7  for(let value of name) { }  
8  
9  
10 for(let key in name) { }  
11
```



# Funções

São blocos de código reutilizáveis que podem ser chamados para executar uma tarefa específica.

```
1 function helloWorld() {  
2   // código aqui  
3 }  
4  
5 helloWorld()
```

```
1 const helloWorld = function() {  
2   // código aqui  
3 }  
4  
5 helloWorld()
```

# Funções (Parâmetros e Argumentos)

**Parâmetros** - Nomes definidos na declaração da função para representar os valores que ela espera receber.

**Argumentos** - São os valores reais passados para uma função quando ela é chamada.

```
1  parâmetros
2  function sum(a, b) {
3    console.log(a, b)
5  }
6
7  argumentos
8  sum(1, 2)
```

**Praticando...**

**Isso não é tudo que podemos  
fazer com o Javascript**

# Funções (Arrow function)

Um forma reduzida/curta de escrever funções.

```
1  parâmetros
2  const sum = (a, b) => {
3    console.log(a, b)
5  }
6
7  argumentos
8  sum(1, 2)
```

# Funções (Callback function)

É uma função que é passada como argumento para outra função.

```
1  function sayMyName(functionName) {  
2    functionName();  
3  }  
4  
5  sayMyName(function() {  
6    console.log('Marcos')  
7  })
```

# Expressões e operadores

**Expressão** é qualquer linha de código que resolve alguma coisa no JavaScript.

**Operador** é um símbolo ou palavra-chave que executa uma operação em um ou mais operandos.

```
1 // Binário
2 15 + 5 = 20
3
4 // Unário
5 value = false
6 verdadeiro = !valor
```

```
1 // Ternário
2 idade = 20
3 status = (idade < 18) ? '👎' : '👍'
4
5 // 👍
6
```

# Javascript Assíncrono

No Javascript sistemas assíncronos são aqueles que são executados de forma independente.

## O que é uma Promise?

É um objeto usado para representar a eventual conclusão (ou falha) de uma operação assíncrona.

👉 Pending

👉 Fullfilled

👉 Rejected



# Javascript Assíncrono (then/catch)

```
<1 promise
2   .then((response) => console.log(response))
3   .catch((error) => console.log(error))
4
```

# Javascript Assíncrono (async/await)

```
<1  async function run() {  
2    const response = await promise  
3    console.log(response)  
4  }  
5  
6  run()  
7
```



**Até a próxima aula!**

