



SÃO  
PAULO  
TECH  
SCHOOL

```
import './index.css';  
import { ReactComponent as ArrowIcon } from '../assets/icons/arrow.svg';  
import { ReactComponent as BoltIcon } from '../assets/icons/bolt.svg';  
import { ReactComponent as RightArrowIcon } from '../assets/icons/right-arrow.svg';  
  
import React, { useState, useEffect, useRef } from 'react';  
import { CSSTransition } from 'react-transition-group';
```

```
"eslintConfig": {  
  "extends": [  
    "react-app",  
    "react-app" ]  
}
```

# Parte I



---

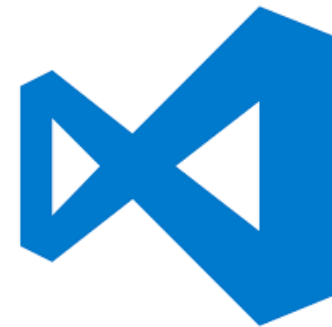
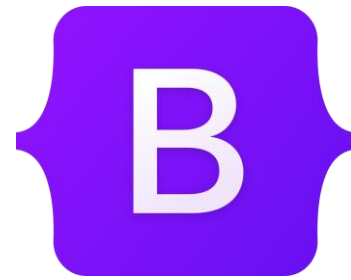
## O que faremos no Bootcamp

---

- Como integrar front-end e back-end usando apenas JS
- Alterando comportamento e estilização das telas de acordo com retorno de uma API
- Complementar nosso conhecimento em Programação

```
if ($(window).scrollTop() > 0) {  
    if (parseInt(header1.css('padding-top')) < 50) {  
        header1.css('padding-top', '50px');  
    }  
} else {  
    header1.css('padding-top', '0px');  
}  
  
if ($(window).scrollTop() > 0) {  
    if (parseInt(header2.css('padding-top')) < 50) {  
        header2.css('padding-top', '50px');  
    }  
}
```

# O que usaremos





# Agenda de Streaming de Game





## TÓPICOS DA AULA

- Conhecendo o projeto **Game Cast**
- Prática:
  - Clonando Template (Front-end)
  - Entender o Projeto
  - Criando **Mock API**
  - Testando requisições no **Insomnia**

# Parte II



## TÓPICOS DA AULA

- Conceitos Importantes (DOM)
- Prática:
  - Montando Mock API
  - Inserindo registros (POST)
  - Fetch API
  - Promise & Async/Await
  - Exibindo no Front



# Como o navegador entende o código que escrevemos?

Ele tem o DOM (Document Object Model)



## O que é o DOM ?

- O **DOM** (Document Object Model) é uma interface que representa como os documentos HTML e XML são lidos pelo browser
- Representado em formato de **árvore**
- O **JavaScript** usa o DOM para interagir com a página HTML e criar experiências web interativas e dinâmicas.

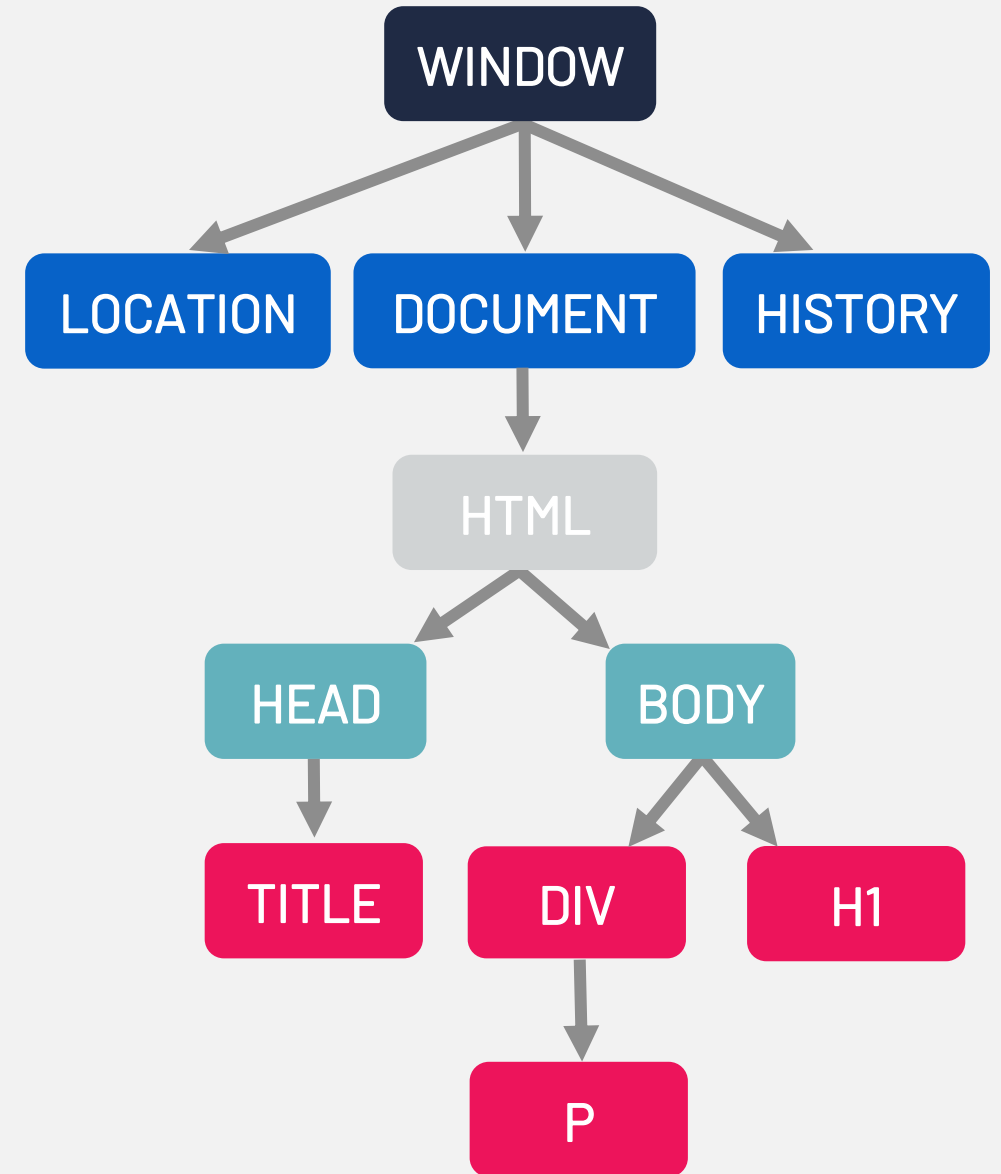


# HTML

```
<!DOCTYPE html>
<html>
  <head>
    <title>Document</title>
  </head>
  <body>
    <div id="hello">
      <p>Hello World</p>
    </div>
    <h1>Primeira Página</h1>
  </body>
</html>
```

# JavaScript

```
<script>
  document.getElementById("hello")
</script>
```





# Fetch API

- A Fetch API é uma interface do JavaScript para acessar e manipular recursos em redes, como por exemplo realizar requisições HTTP.
- É nativo do JS, então não precisamos instalar



A background image showing a business meeting. In the upper half, a woman in a grey blazer is gesturing with her hand while talking to a man in a dark suit and striped tie. In the lower half, several hands are gathered around a tablet displaying a dashboard with charts and graphs. A coffee cup is also visible on the table.

# CONSUMINDO UMA API

## Síncrono vs. Assíncrono

- Um código assíncrono não bloqueia a execução do programa enquanto uma tarefa está sendo realizada.
- Já o código síncrono é executado sequencialmente

**Analogia:** em um jogo de corrida, não precisamos esperar um carro terminar para que o outro inicie a corrida (**assíncrono**).





## Promessas no JavaScript

- As promises permitem trabalhar com operações **assíncronas**
- O objeto **Promise** representa uma operação que pode estar em um de três estados: **pendente**, **resolvida** ou **rejeitada**.

Isso significa que o valor de um objeto promise pode **não** estar disponível quando a variável é instanciada.



**Agradeço**  
a sua atenção!

giuliana.franca@sptech.school

SÃO  
PAULO  
TECH  
SCHOOL