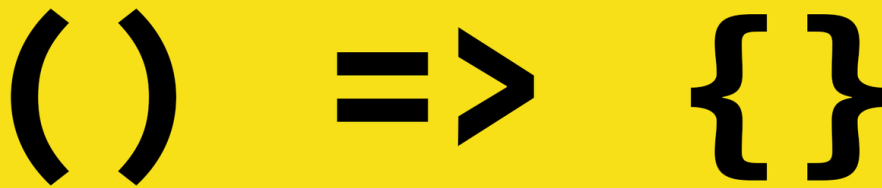


# Aula 05 - Oi Devs

- ✓ ~~GRA~~
- ✓ ~~Modelagem baseada em componentes~~
- ✓ ~~React Router~~
- ✓ ~~Props~~

## Arrow Function



A diagram on a yellow background showing the transformation of a function signature. It consists of a large opening parenthesis '(', followed by an equals sign and a greater-than sign '=>', followed by a large closing curly brace '}'.

```
1 // (parametro1, parametro2, ..., parametroN) => expressao;
2
3 // ES5: Sem arrow function
4 function obterResultado(usuario, pontos) {
5     return usuario + " marcou " + pontos + " pontos!";
6 }
7
8 // ES6: Com arrow function
9 const obterResultado = ({ usuario, pontos }) => {
10     return `${usuario} marcou ${pontos} pontos!`;
11 }
```



## Atribuição via desestruturação (destructuring assi...

A sintaxe de atribuição via desestruturação (destructuring assignment) é uma expressão JavaScript que possibilita extrair...

[developer.mozilla.org](https://developer.mozilla.org)



Atribuição via desestruturação (destructuring assignment) - JavaScript | MDN • [developer.mozilla.org](https://developer.mozilla.org)



## Template strings - JavaScript | MDN

Template Strings são `_string_s` que permitem expressões embutidas. Você pode utilizar string multi-linhas e interpolação de string com...

[developer.mozilla.org](https://developer.mozilla.org)



Template strings - JavaScript | MDN • [developer.mozilla.org](https://developer.mozilla.org)

```
1
2   const MyButton = ({ onClick }) => (
3     <button onClick={() => onClick({ someSpecialData })}>
4       Click Me
5     </button>
6   );
```

# Proptypes

```
npm install prop-types --save
```

```

src > JS About.js M × ...
1 import React from "react";
2 import Count from "./Count";
3 const About = () => {
4   return (
5     <>
6       <div className="app">
7         <Count name="Ateev"
8           age={25} />
9       </div>
10    </>
11  );
12 };
13 export default About;
14

src > JS Count.js M ×
1 import React from "react";
2 import PropTypes from "prop-types";
3 const Count = (props) => {
4   return (
5     <div className="bg-dark text-white">
6       <h1 className="px-5 pt-5">
7         My name is {props.name} of &nbsp;
8         {props.age}.
9       </h1>
10      <p className="fs-5 px-5 pb-5">I am a
11        front-end developer.</p>
12    </div>
13  );
14 };
15 Count.propTypes = {
16   name: PropTypes.string,
17   age: PropTypes.number,
18 };
19 export default Count;

```

```

1 Input.propTypes = {
2   label: PropTypes.string,
3 };
4 Input.defaultProps = {
5   label: "Usuário",
6 };

```

## Estado

O estado (state) nos permite gerenciar a mudança de dados em uma aplicação.

**const [state, setState] = useState(initialState)**



The name of  
your state



The function you'll  
eventually use to  
change the value of this  
state



The initial value  
of your state

```
1 import React, { useState } from 'react';  
2  
3 const App = () => {  
4  
5   const [mssg, setMssg] = useState('This is a functional component!');  
6  
7   return <h1>{mssg}</h1>  
8 }
```

- ☐ State Lifting
- ☐ Formulários e Binds
- ☐ Renderização Condicional
- ☐ Renderização de Listas
- ☐ Composição de Componentes (children)
- ☐ Eventos
- ☐ Métodos do Ciclo de Vida