



Imersão Desenvolvimento Web

Modulo II - Dia 2

Taylane Brandão

Renan Verissimo

Laion Luiz

Índice:

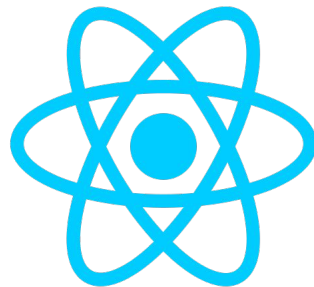
- **JSX**
- **Componentes**
- **E mais componentes**

JSX

Parte I

—

O que é **JSX**?



- Imagine que um código meio **HTML** + meio **JavaScript** == **JSX**
- Exemplo:

```
const nav = (  
  <nav>  
    <ul>  
      <li><a href="#">Início</a></li>  
      <li><a href="#">Sobre</a></li>  
      <li><a href="#">Contato</a></li>  
    </ul>  
  </nav>  
)
```

HTML modularizado ?

- Dessa forma podemos usar nossas funções para retornar um **HTML** inteiro.
- Aumenta a legibilidade do código.
- Facilita nossa vida na hora de criar **HTML**, sem precisar manipular o **DOM** diretamente.
- Além disso, nos dá de brinde o modelo de desenvolvimento orientado a **Componentes**.

Alguns exemplos

- Criar um **H1**:

```
const element = <h1>Hello, {name}</h1>;
```

- Usando o **retorno** de uma função:

```
const element = (  
  <h1>O retorno é: {funcao(parametro)}!</h1>  
)
```

- Usando **condicionais**:

```
function condicional(parametro) {  
  if (expressao) {  
    return <h1>Hello!</h1>;  
  }  
  return <h1>Bye!</h1>;  
}
```

Lidando com atributos no **JSX**

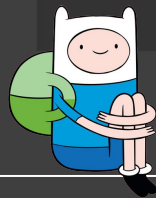
- Como **String**:

```
const element = <a href="https://akoni.global/">O APRENDIZADO TRANSFORMA</a>;
```

- Com **interpolação**:

```
const element = <img src={variavelCaminho}></img>;
```

É hora de codar!



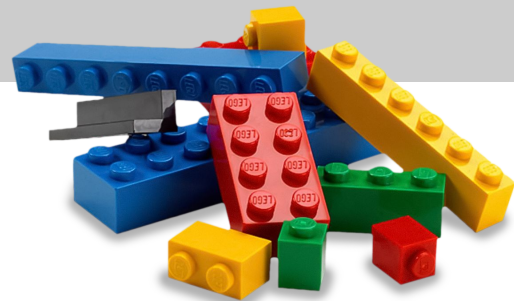
- Explore essa nova sintaxe, criando algumas `<div>`, `<p>` e outros elementos **HTML**.
- Obs: Não se esqueça do **Hello World** !!

Componentes

Parte II

—

Componentes



- Imagine pequenos blocos que nos permitem construir coisas maiores, esses são nossos **componentes**.



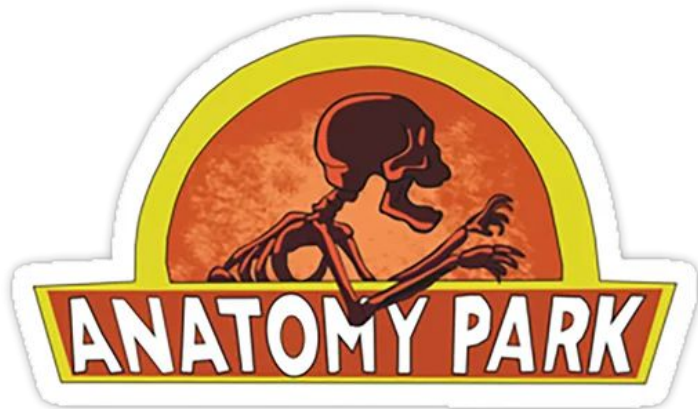
- Com **componentes** de **botão**, **input**, **texto** podemos reaproveitar esses itens agilizando o desenvolvimento.
- Lembrando sempre de criar **componentes** abstratos.

! Atenção!!

- Existem **DUAS** sintaxes de React:
 - **Funcional**;
 - **Orientada a objetos**;
- Nós vamos aprender a mais recente e recomendada, a **Funcional**.
- **Lembre-se** sempre de conferir a sintaxe quando for pesquisar um código na internet.

Anatomia de um Componente

- A estrutura de um componente é normalmente composta por:
 - Uma **função** JavaScript;
 - **Parametros** e **Props**;
 - Retornam uma **composição** visual;



Render



- O React renderiza o **HTML** da pagina usando a função `render()`.
- Neste método podemos ler as **Props/Estado** do componente.

```
render() {  
  return <h1>Olá, {nome}</h1>;  
}
```

??? Props ???

- Props são propriedades **transmitidas** entre os componentes.
- Podem ser de qualquer **tipo**.
- Permitem **enviar** dados do **pai para o filho**.

Componentes na pratica

```
//Aqui definimos o componente
function Componente01a(props) {
  return <h1>Olá, {props.nome}</h1>;
}

function App() {
  return (
    <div>
      <Componente01a nome="Laion" />
      <Componente01a nome="Katreque" />
      <Componente01a nome="Taylane" />
    </div>
  );
}
```

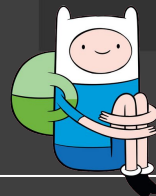
//Aqui utilizamos ele várias vezes no app

Importação



- Vamos precisar importar o que for ser utilizado naquele arquivo;
- A importação é feita no início de cada arquivo;
- Ela funciona de forma similar a tag `<link>` do **HTML**;

É hora de codar!



- Vamos transformar o **JSX** criado anteriormente em um componente;
- Agora **reutilize** ele varias vezes;

Componentes++

Parte III

—

Eventos

- Os eventos em **React** são um pouco diferentes do que nós estamos acostumados.

- **HTML** classico:

```
<button onclick="ligarLuzes()">  
  Liga as luzes  
</button>
```

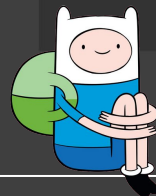
- **React:**

```
<button onClick={ligarLuzes}>  
  Liga as luzes  
</button>
```

Forms

- Podemos criar **Forms** da maneira que estamos acostumados.
- Mas e na hora de pegar o valor dos **inputs**?
- Podemos fazer de **duas** formas:
 - Usar **Eventos** e funções para capturar o valor que foi mudado;
 - Usar **Hooks** (Conceito que veremos no próximo módulo);
- **ATENÇÃO**: Devemos evitar interagir diretamente com o **DOM**.

É hora de codar!



- Criar um cadastro de **Nome**.
- Deve permitir digitar um nome.
- E possuir um botão de **Salvar**.

Mostrar e Esconder

- Nós já sabemos mudar a visibilidade dos elementos usando **CSS**.
- O **React** fornece mais uma forma de fazer isso:

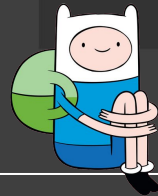
```
{expressão &&  
  <componente>  
    Assim eu mostro ou não o componente  
  </componente>  
}
```

Listas

- Lembra dos nosso **Arrays**?
- E se quisermos mostrar todos aqueles dados em tela?
- Podemos usar o **Map**, uma função que itera por todo o array, **item** à **item** usando uma **Arrow Function**.

```
const array= [conteudo do array];  
const elementosCriados = array.map((itemDoArray) =>  
  <elemento>{itemDoArray}</elemento>  
);
```

É hora de codar!



- Criar um array de numeros;
- Criar um p para cada elemento;
- Que não aparece se o número for ímpar;

