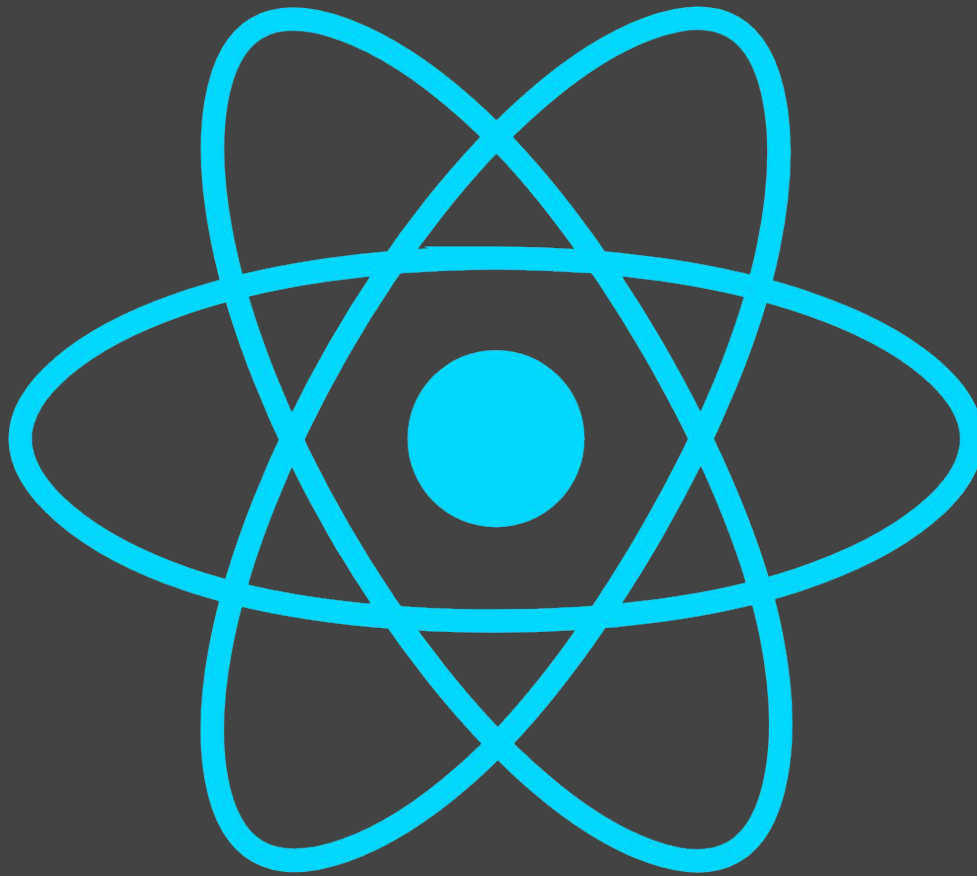


# React



**ANA CAROLINA N R GRACIOSO**  
carol.nrg@gmail.com

# O que é o React

“O React é uma biblioteca JavaScript declarativa, eficiente e flexível para criar interfaces com o usuário.

Ele permite compor UIs complexas a partir de pequenos e isolados códigos chamados componentes.”

Podemos criar Apps web ou mobile com o React. Na disciplina só trabalharemos com Apps web.

# Formas de Utilização

Podemos construir uma aplicação do zero com React ou inserir componentes react em aplicações que já existem.

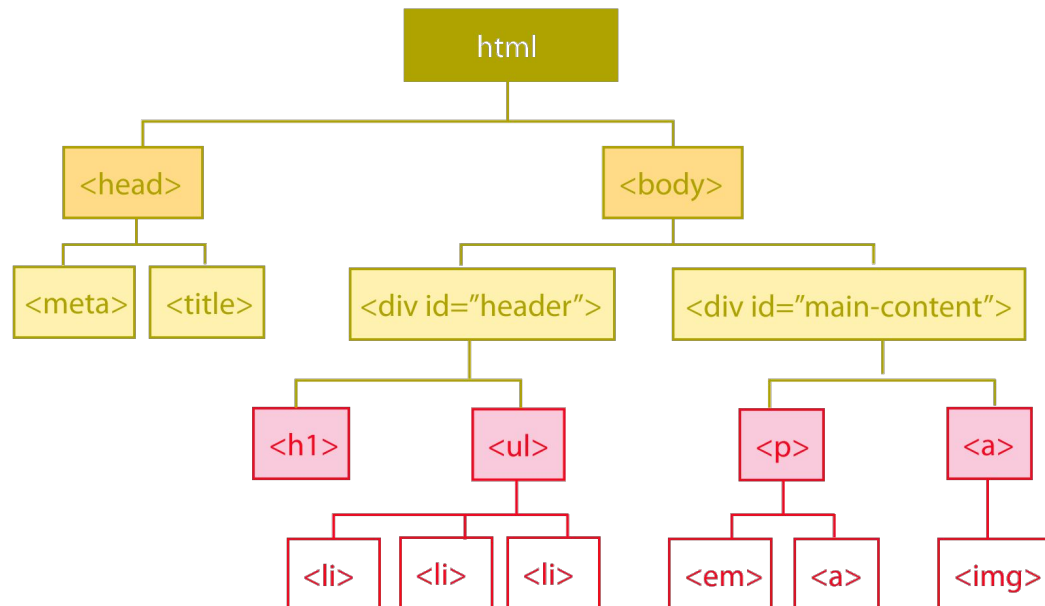
Na aula de hoje aprenderemos a inserir o react em aplicações web que já existem. Nas próximas aulas aprenderemos a criar uma aplicação totalmente em React.

Então... o **React não é uma nova linguagem**. É uma biblioteca JavaScript que permite colocar códigos HTML e CSS dentro do JS.

Pré-requisitos: HTML, CSS e JS.

# Virtual DOM x DOM

- No DOM, para alterar qualquer elemento da árvore, ela tem que ser percorrida hierarquicamente até que o elemento seja localizado;
- No Virtual DOM: cada elemento possui um ID associado no React. Assim, não precisa-se percorrer toda a árvore.



É uma especificação de linguagem.

Não estamos criando um novo HTML, nem um novo JavaScript ou CSS. Somente inserindo o HTML no JS de forma mais simples!

**JS:**

```
var paragrafo = document.createElement('p');  
var texto = document.createTextNode("Um texto qualquer");  
paragrafo.appendChild(texto);
```

**JSX:**

```
const paragrafo = () => {  
  return <p>Um texto qualquer</p>  
};
```

# Babel

Babel é um conjunto de ferramentas usadas principalmente para converter o código ECMAScript 2015+ em uma versão compatível com versões anteriores do JavaScript.

```
<script type="text/babel">  
  const element = <h1>Hello, world</h1>;  
  ReactDOM.render(element,document.getElementById('root'));  
</script>
```

# Exemplo 1 - Crie a página HTML

```
1 <!-- PAGINA BASICA HTML -->
2 <!DOCTYPE html>
3 <head>
4     <title>Introdução</title>
5     <meta charset="UTF-8">
6 </head>
7 <body>
8 </body>
9 </html>
```



# Exemplo 1 - Adicione o react

```
1  <!DOCTYPE html>
2  <head>
3      <title>Adicionando react à página html</title>
4      <meta charset="UTF-8">
5      <script src="https://unpkg.com/react@16/umd/
6          react.development.js" crossorigin></script>
7      <script src="https://unpkg.com/react-dom@16/umd/
8          react-dom.development.js" crossorigin></script>
9  </head>
10 <body>
11 </body>
12 </html>
```

# Exemplo 1 - Adicione seu script react no corpo da página

```
8 <body>
9   <div id="root"></div>
10  <script>
11    // Sintaxe JSX (não é strint nem html)
12    const element = <h1>Hello, world</h1>;
13    ReactDOM.render(element,document.getElementById('root'));
14  </script>
15 </body>
```

# Exemplo 1 - Adicione o Babel e teste

```
10 <body>
11   <div id="root"></div>
12   <!-- type define o uso do babel no script -->
13   <script type="text/babel">
14     const element = <h1>Hello, world</h1>;
15     ReactDOM.render(element,document.getElementById('root'));
16   </script>
17 </body>
```

# Exemplo 2 - Incorporando Expressões em JSX

```
<body>
  <div id="root"></div>
  <script type="text/babel">
    const name = 'Carol';
    const element = <h1>Hello, {name}</h1>;
    ReactDOM.render(element,document.getElementById('root'));
  </script>
</body>
```

# Exemplo 3 - Incorporando Expressões em JSX

```
<body>
  <div id="root"></div>
  <script type="text/babel">

    const user = {
      firstName: 'Ana Carolina',
      lastName: 'Gracioso'
    };

    function formatName(user) {
      return user.firstName + ' ' + user.lastName;
    }

    const element = (<h1>Hello, {formatName(user)}!</h1>);
    ReactDOM.render(element, document.getElementById('root'));
  </script>
</body>
```

# Exemplo 4 - Especificando Elementos Filhos com JSX

```
<body>
  <div id="root"></div>
  <script type="text/babel">
    const element = (
      <div>
        <h1>Hello!</h1>
        <h2>Good to see you here.</h2>
      </div>);
    ReactDOM.render(element,document.getElementById('root'));
  </script>
</body>
```

# Exemplo 5 - Representando objetos com JSX

```
<body>
  <div id="root"></div>
  <script type="text/babel">
    const element = (
      <h1 className="greeting">
        Hello, world!
      </h1>
    );
    /*
    const element = React.createElement(
      'h1',
      {className: 'greeting'},
      'Hello, world!'
    );
    */
    ReactDOM.render(element, document.getElementById('root'));
  </script>
</body>
```

# Exemplo 6 - Atualizando elementos renderizados

```
<body>
  <div id="root"></div>
  <script type="text/babel">
    function tick() {
      const element = (
        <div>
          <h1>Hello, world!</h1>
          <h2>It is {new Date().toLocaleTimeString()}.</h2>
        </div>
      );
      ReactDOM.render(element,
        document.getElementById('root'));
    }
    setInterval(tick, 1000);
  </script>
</body>
```



# Atividade

Implemente uma página HTML com um componente React que simule um letreiro com o texto "Venha estudar na Fatec". O texto deve aparecer letra a letra na página como se estivesse sendo digitado.

# Referências

- <https://pt-br.reactjs.org/tutorial/tutorial.html>