UNINASSAU
Campus Graças

Código de Alta Performance -WEB
AULA 05- Descrevendo a IU

PROFESSOR:

Dr. Diogo Rodrigues

CURSO (2024.2)

Blocos de construção da UI

UNINASSAU
Campus Graças

- Na Web, o HTML nos permite criar documentos ricamente estruturados com seu conjunto integrado de tags como <h1>e <li>.
- O React permite que você combine sua *marcação, CSS e JavaScript em "componentes" personalizados,* elementos de UI reutilizáveis para seu aplicativo.

```
<article>
  <h1>My First Component</h1>
  <ol>
    <li>Components: UI Building Blocks</li>
    <li>Defining a Component</li>
    <li>Using a Component</li>
  </ol>
</article>
```

Blocos de construção da UI

UNINASSAU
Campus Graças

- Assim como com tags HTML, você pode compor, ordenar e aninhar componentes para projetar páginas inteiras. Por exemplo o código ao lado composto de componentes react.
- Conforme seu projeto cresce, você notará que muitos de seus designs podem ser compostos reutilizando componentes que você já escreveu, acelerando seu desenvolvimento.

```
<PageLayout>
 <NavigationHeader>
  <SearchBar />
  <Link to="/docs">Docs</Link>
 </NavigationHeader>
 <Sidebar />
 <PageContent>
  <TableOfContents />
  <DocumentationText />
 </PageContent>
</PageLayout>
```

Definindo uma componente

UNINASSAU
Campus Graças

```
export default function Profile() {
  return (
    <img
      src="https://i.imgur.com/MK3eW3Am.jpg"
      alt="Katherine Johnson"
    />
  )
```

Definindo uma componente

UNINASSAU
Campus Graças

1. **Exporta componentes:** O prefixo **export default** é uma sintaxe JavaScript padrão (não específica para React). Ele permite que você marque a função principal em um arquivo para que você possa importá-la posteriormente de outros arquivos.

2. **Defina a função**: Com function Profile() { }você você define uma função JavaScript com o nome Profile. *Os componentes React são funções JavaScript comuns, mas seus nomes devem começar com letra maiúscula ou não funcionarão!*

3. **Adicionar marcação HTML:** O componente retorna uma <img /> tag com atributos src e é escrito como HTML, mas na verdade é JavaScript por baixo dos panos! **Essa sintaxe é chamada JSX** , e *permite que você incorpore marcação dentro do JavaScript.alt<img />*

Usando uma componente

UNINASSAU
Campus Graças

Agora que você definiu seu Profile componente, você pode aninhá-lo dentro de outros componentes. Por exemplo, você pode exportar um Gallery componente que usa múltiplos Profile componentes:

```
import Profile from "../Profile";

export default function Gallery() {
  return (
    <section>
      <h1>Amazing scientists</h1>
      <Profile />
      <Profile />
      <Profile />
    </section>
  );
}
```

Escrevendo marcação com JSX

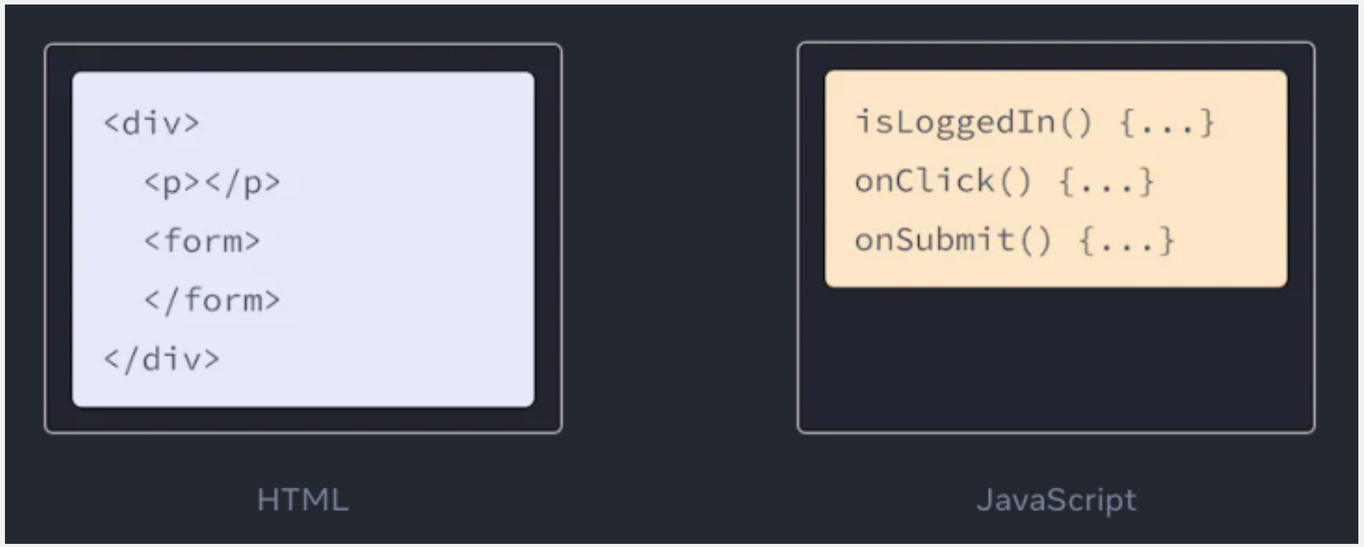
UNINASSAU
Campus Graças

JSX é uma extensão de sintaxe para JavaScript que **permite que você escreva marcação semelhante a HTML dentro de um arquivo JavaScript.** Embora existam outras maneiras de escrever componentes, a maioria dos desenvolvedores React prefere a concisão do JSX.

Escrevendo marcação com JSX

UNINASSAU
Campus Graças

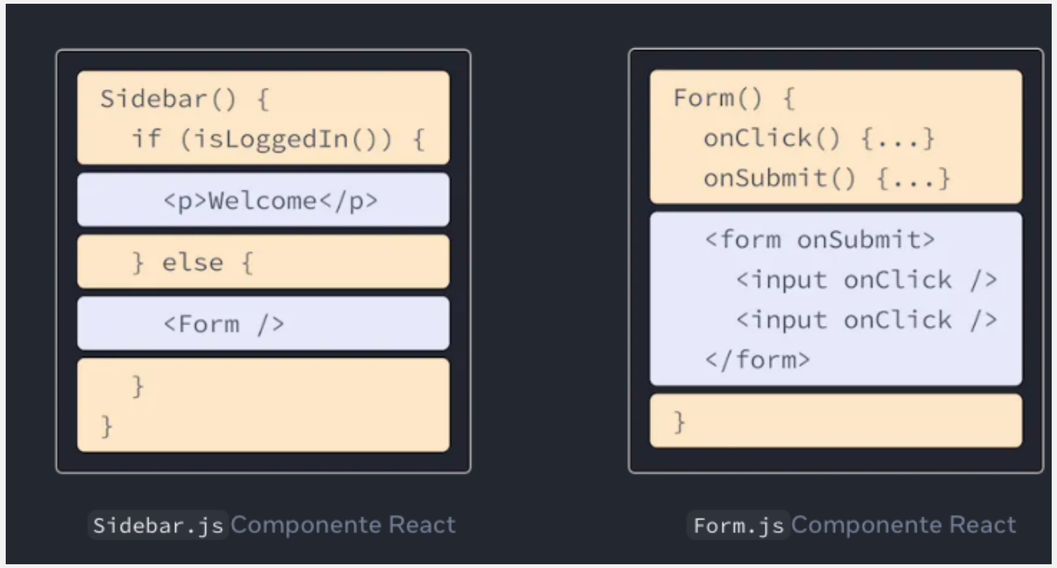
A Web foi construída em HTML, CSS e JavaScript. Por muitos anos, os desenvolvedores web mantiveram o conteúdo em HTML, o design em CSS e a lógica em JavaScript — **geralmente em arquivos separados!** O conteúdo era marcado dentro do HTML, enquanto a lógica da página vivia separadamente em JavaScript:



Escrevendo marcação com JSX

UNINASSAU
Campus Graças

Mas, à medida que a Web se tornou mais interativa, a lógica determinou cada vez mais o conteúdo. O JavaScript estava no comando do HTML! É por isso que **no React, a lógica de renderização e a marcação vivem juntas no mesmo lugar — componentes.**



As regras do JSX

UNINASSAU
Campus Gracas

1- Retornar um único elemento raiz

Para retornar vários elementos de um componente, envolva-os com uma única tag pai.

Por exemplo, você pode usar um .

Se você não quiser adicionar nada extra à sua marcação, você pode escrever <>e </> (Fragmento) permitem que você agrupe coisas sem deixar rastros na árvore HTML do navegador.

```
<div>
 <h1>Hedy Lamarr's Todos</h1>
 <img
   src="https://i.imgur.com/yXOvdOSs.jpg"
   alt="Hedy Lamarr"
   class="photo"
 >
 <ul>
   ...
 </ul>
</div>
```

As regras do JSX

UNINASSAU
Campus Graças

2- Feche todas as TAGS

O JSX exige que as tags sejam fechadas explicitamente: tags de fechamento automático como <img>devem se tornar <img />, e tags de encapsulamento como orangesdevem ser escritas como oranges.

As regras do JSX

UNINASSAU
Campus Graças

3- camelCase

JSX se transforma em JavaScript e atributos escritos em JSX se tornam chaves de objetos JavaScript. Em seus próprios componentes, você frequentemente desejará ler esses atributos em variáveis. Mas JavaScript tem limitações em nomes de variáveis. Por exemplo, seus nomes não podem conter traços ou ser palavras reservadas como class.

É por isso que, no React, muitos atributos HTML e SVG são escritos em camelCase. Por exemplo, em vez de stroke-widthvocê usa strokeWidth. Como class é uma palavra reservada, no React você escreve className, nomeado após a propriedade DOM correspondente

JavaScript em JSX

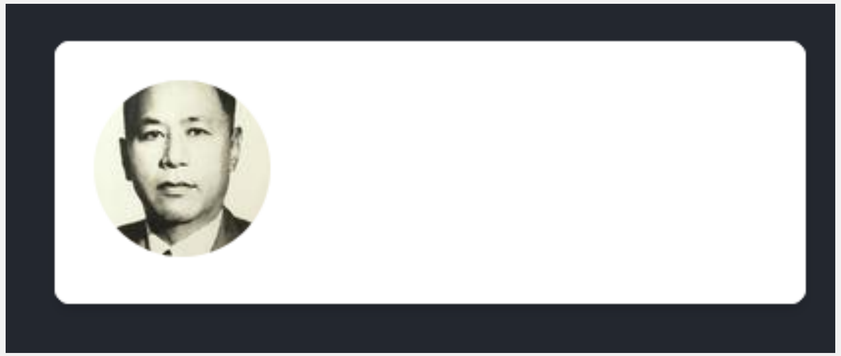
UNINASSAU
Campus Graças

O JSX permite que você escreva marcação semelhante a HTML dentro de um arquivo JavaScript, mantendo a lógica de renderização e o conteúdo no mesmo lugar. Às vezes, você vai querer adicionar um pouco de lógica JavaScript ou referenciar uma propriedade dinâmica dentro dessa marcação. **Nessa situação, você pode usar chaves no seu JSX para abrir uma janela para JavaScript.**

JavaScript em JSX

UNINASSAU
Campus Graças

Quando você deseja passar um atributo de string para o JSX, você o coloca entre aspas simples ou duplas:

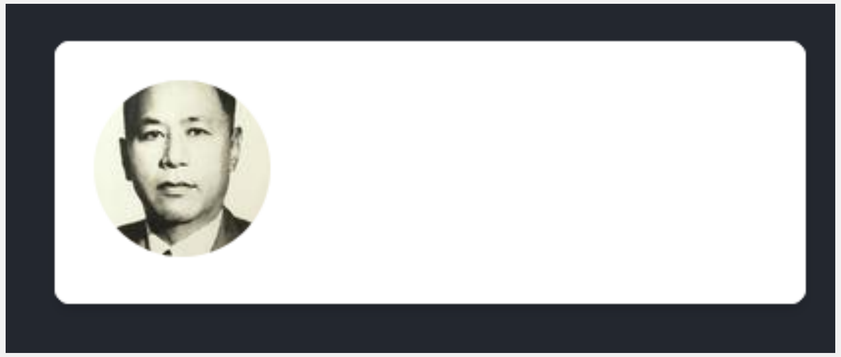


```
export default function Avatar() {
  return (
    <img
      className="avatar"
      src="https://i.imgur.com/7vQD0fPs.jpg"
      alt="Gregorio Y. Zara"
    />
  );
```

JavaScript em JSX

UNINASSAU
Campus Graças

Mas e se você quiser especificar dinamicamente o texto src or alt? Você pode usar um valor do JavaScript substituindo "and "por {and} :



```
export default function Avatar() {
  const avatar = 'https://i.imgur.com/7vQD0fPs.jpg';
  const description = 'Gregorio Y. Zara';
  return (
    <img
      className="avatar"
      src={avatar}
      alt={description}
```

Usando chaves {}

UNINASSAU
Campus Graças

JSX é uma maneira especial de escrever JavaScript. Isso significa que é possível usar JavaScript dentro dele — com chaves { }. O exemplo ao lado primeiro declara um nome para o cientista, name, então o incorpora com chaves dentro de <h1>:

```
export default function TodoList() {
  const name = 'Gregorio Y. Zara';
  return (
    <h1>{name}'s To Do List</h1>
  );
}
```

Usando chaves {}

UNINASSAU
Campus Graças

Qualquer expressão JavaScript funcionará entre chaves, incluindo chamadas de função como formatDate():

```
const today = new Date();
function formatDate(date) {
  return new Intl.DateTimeFormat(
    'en-US',
    { weekday: 'long' }
  ).format(date);
}

export default function TodoList() {
  return (
    <h1>To Do List for {formatDate(today)}</h1>
  );
}
```

Onde utilizar a chaves { }?

UNINASSAU
Campus Graças

Você só pode usar chaves de duas maneiras dentro do JSX:

- **Como texto diretamente dentro de uma tag JSX:** <h1>{name}'s To Do List</h1>
- **Como atributos imediatamente após o =:** src={avatar}lerá a variável avatar, mas src="{avatar}"passará a string “{avatar}".

Chave dupla para css e objetos

UNINASSAU
Campus Graças

- Além de strings, números e outras expressões JavaScript, você pode até mesmo passar objetos em JSX. Objetos também são denotados com chaves, como { name: "Hedy Lamarr", inventions: 5 }. **Portanto, para passar um objeto JS em JSX, você deve envolver o objeto em outro par de chaves: person={{ name: "Hedy Lamarr", inventions: 5 }}.**

- Você pode ver isso com estilos CSS inline em JSX. O React não exige que você use estilos inline (classes CSS funcionam muito bem para a maioria dos casos).

Chaves duplas em objetos

UNINASSAU
Campus Graças

```
export default function TodoList() {
  return (
    <ul style={{
      backgroundColor: 'black',
      color: 'pink'
    }}>
      <li>Improve the videophone</li>
      <li>Prepare aeronautics lectures</li>
      <li>Work on the alcohol-fuelled engine</li>
    </ul>
```

Chaves duplas em objetos

UNINASSAU
Campus Graças

```
const person = {
  name: 'Gregorio Y. Zara',
  theme: {
    backgroundColor: 'black',
    color: 'pink'
  }
};
```

```
export default function TodoList() {
  return (
    <div style={person.theme}>
      <h1>{person.name}'s Todos</h1>
      <img
        className="avatar"
        src="https://i.imgur.com/7vQD0fPs.jpg"
        alt="Gregorio Y. Zara"
      />
      <ul>
        <li>Improve the videophone</li>
        <li>Prepare aeronautics lectures</li>
        <li>Work on the alcohol-fuelled engine</li>
      </ul>
    </div>
  );
```

Passando props para um componente

UNINASSAU
Campus Graças

Os componentes React usam *props* para se comunicarem entre si. Cada componente pai pode passar algumas informações para seus componentes filhos dando a eles props.

Props podem lembrar atributos HTML, mas você pode passar qualquer valor JavaScript por eles, incluindo objetos, arrays e funções.

Props

UNINASSAU
Campus Graças

Props são as informações que você passa para uma tag JSX. Por exemplo, className, src, alt, width, e height são alguns dos props que você pode passar para um <img>:

```
function Avatar() {
  return (
    <img
      className="avatar"
      src="https://i.imgur.com/1bX5QH6.jpg"
      alt="Lin Lanying"
      width={100}
      height={100}
    />
  );
}

export default function Profile() {
  return (
    <Avatar />
  );
```

Passando props para componentes

UNINASSAU
Campus Graças

1- passe alguns props para Avatar. Por exemplo, vamos passar dois props: person(um objeto) e size(um número):

```
export default function Profile() {

  return (

    <Avatar

      person={{ name: 'Lin Lanying',
imageId: '1bX5QH6' }}

      size={100}

    />

  );
```

Passando props para componentes

UNINASSAU
Campus Graças

2- leia os props dentro do componente filho :

Você pode ler essas props listando seus nomes person, size separados por vírgulas dentro ({e })diretamente depois de function Avatar. Isso permite que você as use dentro do Avatarcódigo, como faria com uma variável.

```
function Avatar({ person, size }) {

  // person and size are available here

}
```

Passando props para componentes

UNINASSAU
Campus Graças

2- leia os props dentro do componente filho :

Agora podemos configurar Avatar para renderizar de muitas maneiras diferentes com diferentes props.

```
function Avatar({ person, size }) {
  return (
    <img
      className="avatar"
      src={getImageUrl(person)}
      alt={person.name}
      width={size}
      height={size}
    />
  );
}

export default function Profile() {
  return (
    <div>
      <Avatar
        size={100}
        person={{
```

Passando props para componentes

UNINASSAU
Campus Graças

2- leia os props dentro do componente filho :

Agora podemos configurar Avatar para renderizar de muitas maneiras diferentes com diferentes props.

```
    name: 'Katsuko Saruhashi',
    imageId: 'YfeOqp2'
  }}
/>
<Avatar
  size={80}
  person={{
    name: 'Aklilu Lemma',
    imageId: 'OKS67lh'
  }}
/>
<Avatar
  size={50}
  person={{
    name: 'Lin Lanying',
    imageId: '1bX5QH6'
  }}
/>
```

Passando JSX como filhos

UNINASSAU
Campus Graças

É comum aninhar tags internas do navegador:

```
<div>
  <img />
</div>
```

Passando JSX como filhos

UNINASSAU
Campus Graças

Às vezes, você desejará aninhar seus próprios componentes da mesma maneira:

```
<Card>
  <Avatar />
</Card>
```

Passando JSX como filhos

UNINASSAU
Campus Graças

Quando você aninha conteúdo dentro de uma tag JSX, o componente pai receberá esse conteúdo em uma prop chamada children. Por exemplo, o componente Card ao lado receberá uma prop children definida como <Avatar />e a renderizará em uma div wrapper:

```jsx
import Avatar from './Avatar.js';

function Card({ children }) {
  return (
    <div className="card">
      {children}
    </div>
  );
}

export default function Profile() {
  return (
    <Card>
      <Avatar
        size={100}
        person={{
          name: 'Katsuko Saruhashi',
          imageId: 'YfeOqp2'
        }}
      />
    </Card>
  );
}
```

Desafios

UNINASSAU
Campus Graças

https://react.dev/learn/passing-props-to-a-component

UNINASSAU
Campus Graças

Obrigado

E-mail: 010117368@prof.uninassau.edu.br