

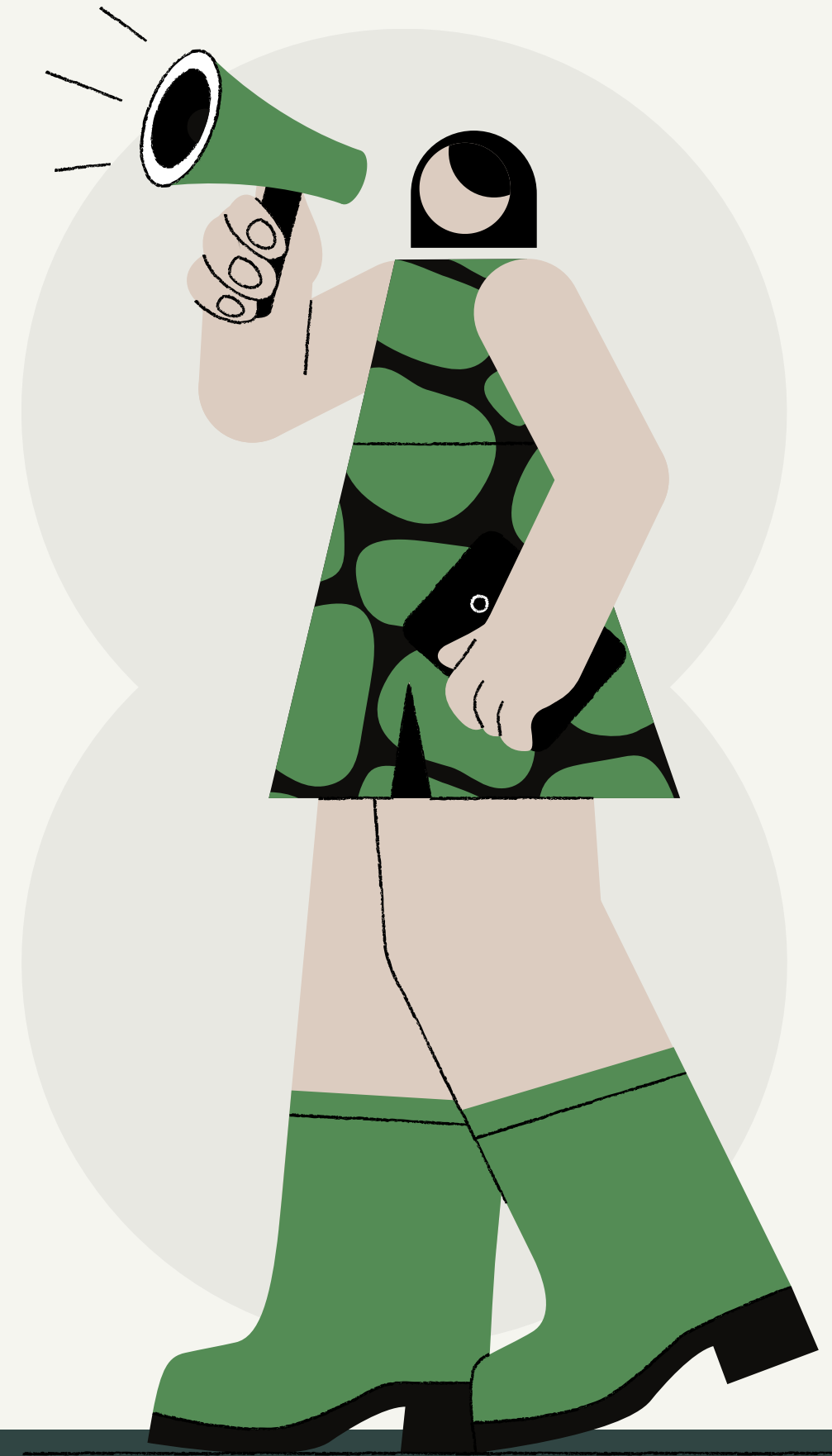
Kaian Novais: Excellence in Education

# Aprenda React



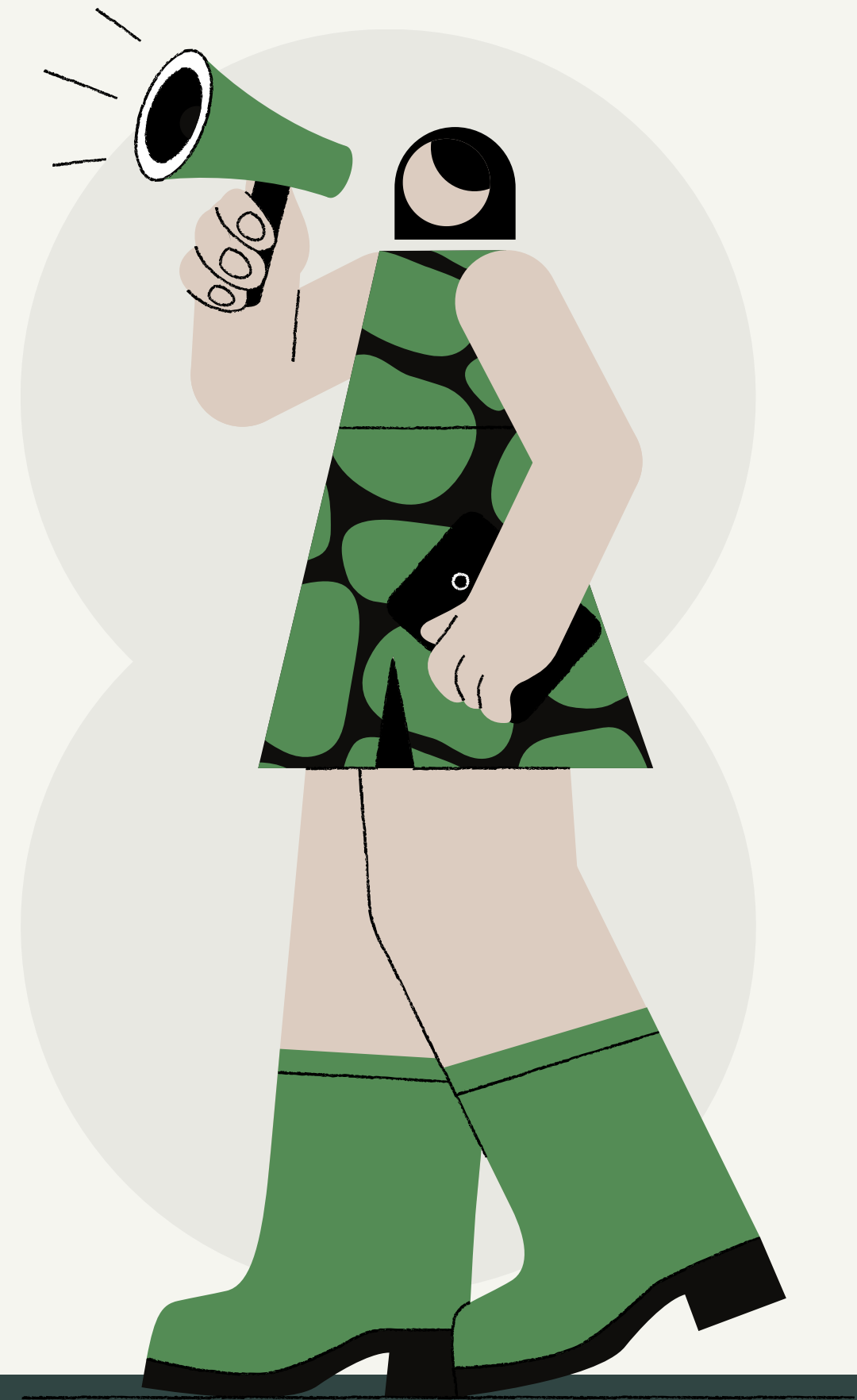
# O que é?

React.js é uma ferramenta criada por engenheiros do Facebook que ajuda os programadores a construir sites e aplicativos. Pense nela como uma caixa de ferramentas que torna mais fácil montar as diferentes partes de um site, assim como você usaria diferentes ferramentas para construir algo em casa.



# O que é?

- React is fast;
- React is popular;
- React is modular;
- React is scalable.



# O que é esse código?

```
const h1 = <h1>Hello world</h1>;
```

The background features four decorative geometric patterns in the corners. The top-left corner has a series of parallel diagonal lines. The top-right corner contains a cluster of semi-circles in red, teal, and blue. The bottom-left corner features a cluster of semi-circles in red, teal, blue, and yellow. The bottom-right corner has a large, faint circular outline and a few diagonal lines.

# JSX

# O QUE É?

JSX é uma **extensão de sintaxe** para JavaScript. Ele foi escrito para ser usado com o React. O código JSX se parece muito com HTML.

## *O que significa “extensão de sintaxe”?*


Nesse caso, significa que o JSX não é um JavaScript válido. Os navegadores da Web não podem lê-lo!

Se um arquivo JavaScript contiver código JSX, esse arquivo terá que ser compilado . Isso significa que, antes que o arquivo chegue a um navegador da Web, um compilador JSX converterá qualquer JSX em JavaScript normal.



```
const navBar = <nav>I am a nav bar</nav>;
```

```
const myTeam = {  
  center: <li>Benzo Walli</li>,  
  powerForward: <li>Rasha Loa</li>,  
  smallForward: <li>Tayshaun Dasmoto</li>,  
  shootingGuard: <li>Colmar Cumberbatch</li>,  
  pointGuard: <li>Femi Billon</li>  
};
```



Se uma expressão JSX ocupar mais de uma linha, você deverá agrupar a expressão JSX de várias linhas entre parênteses

```
(  
  <a href="https://www.example.com">  
    <h1>  
      Click me!  
    </h1>  
  </a>  
)
```



Uma expressão JSX deve ter exatamente um elemento externo.



```
const paragraphs = (  
  <div id="i-am-the-outermost-element">  
    <p>I am a paragraph.</p>  
    <p>I, too, am a paragraph.</p>  
  </div>  
>);
```



```
const paragraphs = (  
  <p>I am a paragraph.</p>  
  <p>I, too, am a paragraph.</p>  
>);
```

# FUNCIONAMENTO

É essencial entender que o React depende de duas coisas para renderizar: qual conteúdo renderizar e onde colocar o conteúdo.

```
1  import React from 'react';
2  import { createRoot } from 'react-dom/client';
3
4  //Pegamos o elemento do nosso html cujo id seja
5  //container
6  const container = document.getElementById('container');
7
8  //aqui definimos onde nosso conteúdo será renderizado
9  //neste caso, dentro do container lá no html
10 const root = createRoot(container);
11
12
13 //aqui definimos qual conteúdo será renderizado
14
15 root.render(<h1>Hello World</h1>);
```

```
1  import React from 'react';
2  import { createRoot } from 'react-dom/client';
3
4  //Pegamos o elemento do nosso html cujo id seja
5  //container
6  const container = document.getElementById('container');
7
8  //aqui definimos onde nosso conteúdo será renderizado
9  //neste caso, dentro do container lá no html
10 const root = createRoot(container);
11
12 const myText = <p>Hello World</p>
13
14 //aqui definimos qual conteúdo será renderizado
15 //nesse caso o conteúdo da variável myText
16 root.render(myText);
```

# FUNCIONAMENTO

```
const toDoList = (  
  <ol>  
    <li>Learn React</li>  
    <li>Become a Developer</li>  
  </ol>  
>;  
  
const container  
= document.getElementById('app');  
const root = createRoot(container);  
root.render(toDoList);
```

**class vs**  
**className**



# Definição



A gramática em JSX é basicamente a mesma que em HTML, mas há diferenças sutis a serem observadas. O mais frequente deles envolve a palavra class.

```
<h1 className="big">Title</h1>
```

No JSX, você não pode usar a palavra class! Você tem que usar className em vez disso.

Isso ocorre porque JSX é traduzido para JavaScript e class é uma palavra reservada em JavaScript.

---

# Tag de fechamento



No JSX, você precisa incluir a barra. Se você escrever uma tag de fechamento automático em JSX e esquecer a barra, gerará um erro:

```
// Fine in JSX:  
<br />  
  
// NOT FINE AT ALL in JSX:  
<br>
```

---

# Chaves { }

Qualquer código entre as tags de um elemento JSX será lido como JSX, não como JavaScript normal! Para fazer com que o código seja lido como JS Você pode fazer isso envolvendo seu código entre chaves .

```
1  import React from 'react';
2  import { createRoot } from 'react-dom/client';
3
4  const container = document.getElementById('app');
5  const root = createRoot(container);
6
7  root.render(<h1>2 + 3</h1>);
8  //saída 2 + 3
9
```

```
1  import React from 'react';
2  import { createRoot } from 'react-dom/client';
3
4  const container = document.getElementById('app');
5  const root = createRoot(container);
6
7  root.render(<h1>{2 + 3}</h1>);
8  //saída 5
9
```



# Variables in JSX



# Funcionamento

Quando você injeta **JavaScript no JSX**, esse JavaScript faz parte do mesmo ambiente que o resto do JavaScript em seu arquivo. Isso significa que você **pode acessar variáveis dentro de uma expressão JSX**, mesmo que essas variáveis tenham sido declaradas fora do bloco de código JSX.

```
// Declare a variable:  
const name = 'Gerdo';  
  
// Access your variable inside of a JSX  
expression:  
const greeting = <p>Hello, {name}!</p>;
```

**THANK'S  
FOR  
WATCHING**