



Projeto 1 - Site de Informações ReactJS

O que vamos aprender

- Por que usar o React
- JSX
- Criar componentes React
- Fazer estilização CSS

1. Faça você mesmo

Crie uma pasta chamada `react`. Recomendamos que esta pasta seja criada dentro da pasta que foi criada para o curso.

Execute `npx create-react-app siteinfo-reactfacts`, dê o comando `cd` em seu projeto e abra-o. No componente `App.js` Você pode excluir tudo na instrução `return` de forma que vai retornar apenas um elemento `div` vazio. Você também pode excluir todos os templates create-react-app fornecidos e apenas deixar `index.js` e `App.js` no diretório `src`.

2. JSX

`index.js`

Desafio 1:

Crie uma barra de navegação (navbar) no JSX:

- Use o elemento ``nav`` semântico como o wrapper pai;
- Tenha um elemento `h1` com a marca do seu "site";
- Insira uma lista não ordenada para os outros elementos de navegação;
- Dentro do ``ul``, tem três ``li``s para "Preço", "Quem Somos" e "Contato";
- Não se preocupe com o estilo ainda - será apenas HTML de aparência simples por enquanto.



Inicial:

index.js

```
const navbar = (  
  <nav>  
    <h1></h1>  
  </nav>  
)
```

final:

index.js

```
const navbar = (  
  <nav>  
    <h1>Bistro da 3way</h1>  
    <ul>  
      <li>Preço</li>  
      <li>Quem Somos</li>  
      <li>Contato</li>  
    </ul>  
  </nav>  
)
```

3. Marcação

Desafio 2:

- Começar do zero, construir e renderizar o HTML para nosso projeto de seção.
- Verifique o resultado desejado abaixo para o que você está tentando construir.
- Nós estaremos adicionando estilo a ele mais tarde.

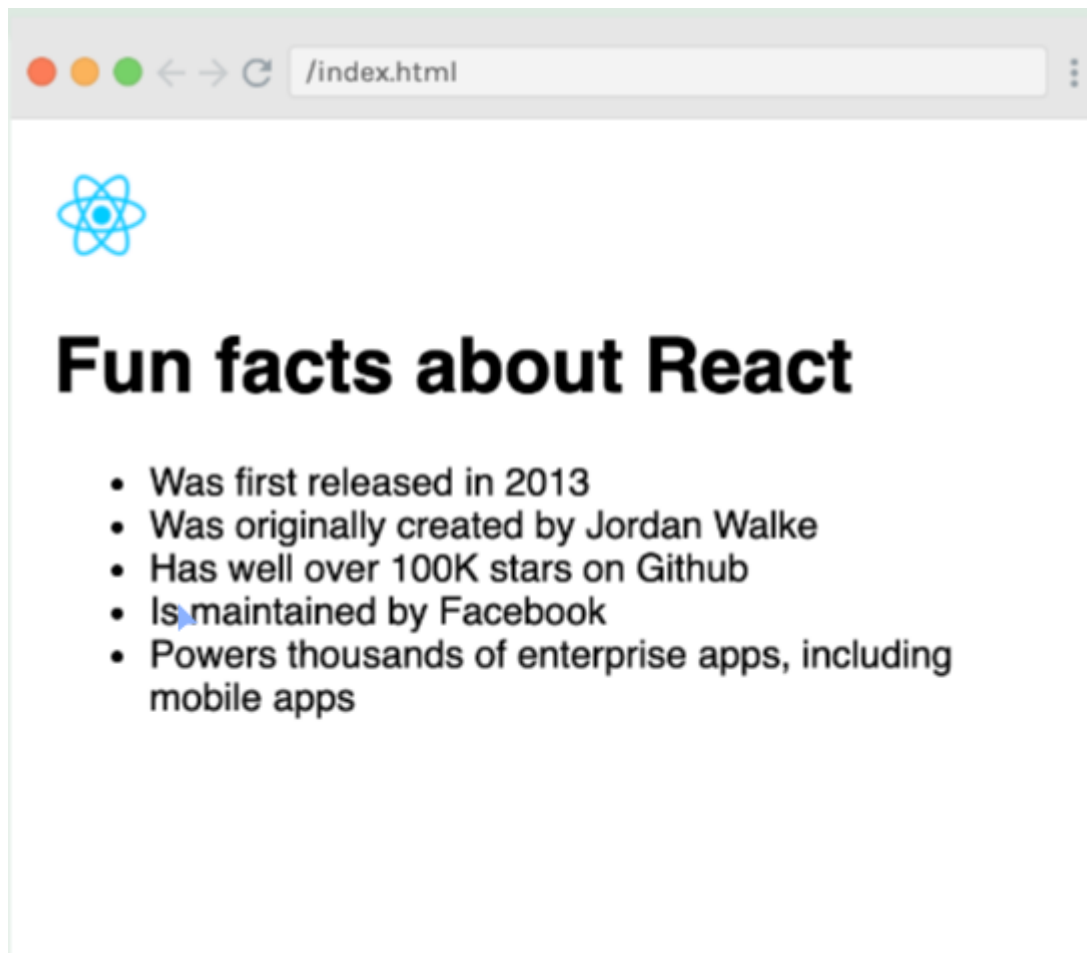
Dicas:

* O [logotipo do React](#) é um arquivo na árvore do projeto, então você pode acessá-lo usando `src="./react-logo.png"` em sua imagem elemento.



* Você também pode definir o atributo `width` do elemento de imagem assim como em HTML. No slide, eu configurei para 40px.

Resultado desejado:



index.js

```
const page = (  
  <div>  
    <img src={require("../react-logo.png")}  
width="40px" />  
    <h1>Curiosidades sobre o React</h1>  
    <ul>  
      <li>Foi lançado pela primeira vez em 2013</li>  
      <li>Foi originalmente criado por Jordan Walke</li>  
      <li>Tem mais de 100 mil estrelas no GitHub</li>  
      <li>É mantido pelo Facebook</li>  
    </ul>  
  </div>  
)
```



```
        <li>Capacita milhares de aplicativos corporativos,  
incluindo aplicativos móveis</li>  
    </ul>  
</div>  
)
```

4. Customizar componentes

1. **Mude** o código acima **para função**, com nome em notação Pascal Case.
2. Na renderização ReactDOM, insira o nome da função em notação de componente React. < NomeDaFuncao />

index.js

```
const root =  
ReactDOM.createRoot(document.getElementById('root'));  
root.render(  
  <React.StrictMode>  
    <NomeDaFuncao />  
  </React.StrictMode>  
) ;
```

Desafio 3:

Parte 1:

- Crie sua própria página usando um componente de página personalizado (Page).
- Ele deve retornar uma lista ordenada com os motivos pelos quais você está animado para aprender React.
- Renderize sua lista para a página.

```
function Page() {  
  return (  
    <ol>  
      <li>É uma biblioteca popular, com grande  
comunidade!</li>  
      <li>É mais provável que eu consiga um emprego
```



```
        como desenvolvedor se souber React</li>
    </ol>
  )
}

const root =
ReactDOM.createRoot(document.getElementById('root'));
root.render(
  <React.StrictMode>
    <Page />
  </React.StrictMode>
);
```

Parte 2:

- Adicione um elemento `header` com um elemento `nav` aninhado. Dentro do `nav`, inclua um elemento `img` com a imagem do [logotipo do React](#) dentro (src="./react-logo.png") e certifique-se de definir a largura para algo mais gerenciável para não ocupar a tela inteira
- Adicione um `h1` com algum texto descrevendo a página. (Ex.: "Motivos Estou animado para aprender React"). Coloque-o acima da lista ordenada.
- Adicione um `footer` após a lista que diz:
"© 20xx <sobrenome aqui> desenvolvimento. Todos os direitos reservados."

index.js

```
function Page() {
  return (
    <div>
      <header>
        <nav>
          <img src={require("./react-logo.png")} width="40px"
/>
        </nav>
      </header>
      <h1>Razões pelas quais estou animado para aprender
React</h1>
```



```
<ol>
  <li>É uma biblioteca popular, com grande
comunidade!</li>
  <li>
    É mais provável que eu consiga um emprego como
desenvolvedor se souber
    React
  </li>
</ol>
<footer>
  <small>© 2022 desenvolvedor <seu sobrenome>. Todos
direitos reservados.</small>
</footer>
</div>
);
}
```

5. Componentes pais/filhos

Minidesafio:

- Mova o elemento `header` da página para seu próprio componente chamado "Header"

```
function Header() {
  return (
    <header>
      <nav>
        <img src={require("../react-logo.png")}
width="40px" />
      </nav>
    </header>
  )
}

function Page() {
  return (
```



```
<div>
  <Header />
  <h1>Reasons I'm excited to learn React</h1>
  <ol>
    <li>It's a popular library, so I'll be
      able to fit in with the cool kids!</li>
    <li>I'm more likely to get a job as a developer
      if I know React</li>
  </ol>
  <footer>
    <small>© 2021 Ziroll development. All rights
reserved.</small>
  </footer>
</div>
)
}
```

Desafio 4:

- Mova o `footer` para seu próprio componente chamado "Footer" e renderize esse componente dentro do componente Page.
- Mova `h1` e `ol` juntos em outro componente chamado "MainContent" e renderizar dentro do componente Page também.

index.js

```
function Header() {
  return (
    <header>
      <nav>
        <img src={require("../react-logo.png")}
width="40px" />
      </nav>
    </header>
  )
}

function Footer() {
```



```
return (
  <footer>
    <small>© 2021 Ziroll development. All rights
reserved.</small>
  </footer>
)
}

function MainContent() {
  return (
    <div>
      <h1>Reasons I'm excited to learn React</h1>
      <ol>
        <li>It's a popular library, so I'll be
able to fit in with the cool kids!</li>
        <li>I'm more likely to get a job as a developer
if I know React</li>
      </ol>
    </div>
  )
}

function Page() {
  return (
    <div>
      <Header />
      <MainContent />
      <Footer />
    </div>
  )
}
```




6. Estilizando com classes

Desafio 5:

- Adicione um `ul` dentro do `nav` do cabeçalho e crie os seguintes `li`s: "Preço", "Quem Somos" e "Contato";
- Usando o flexbox, alinhe os itens de navegação horizontalmente e coloque-os alinhados com o logotipo do React;
- Altere o estilo da imagem para acontecer em CSS em vez de in-line para praticar, adicione uma nova classe à imagem para estilizá-la.

index.js

```
function Header() {  
  return (  
    <header>  
      <nav>  
        <img src={require("../react-logo.png")}  
width="40px" />  
        <ul className="nav-items">  
          <li>Preço</li>  
          <li>Quem Somos</li>  
          <li>Contato</li>  
        </ul>  
      </nav>  
    </header>  
  )  
}
```

styles.css

```
.nav {  
  display: flex;  
  justify-content: space-between;  
  align-items: center;  
}  
  
.nav-logo {  
  width: 60px;
```



```
}  
  
.nav-items {  
  list-style: none;  
  display: flex;  
}  
  
.nav-items > li {  
  padding: 10px;  
}
```

7. Organizando componentes

- Mova a função Header para um arquivo Header.js;
- Faça a importação acrescentando na primeira linha:

```
import React from "react"
```

- No arquivo index.js faça a importação:

```
import Header from "../Header"
```

Header.js

```
import React from "react"  
  
export default function Header() {  
  return (  
    <header>  
      <nav className="nav">  
        <img src={require("../react-logo.png")}  
className="nav-logo" />  
        <ul className="nav-items">  
          <li>Preço</li>  
          <li>Quem Somos</li>  
          <li>Contato</li>  
        </ul>  
      </nav>  
    </header>  
  )  
}
```



```
        </ul>
      </nav>
    </header>
  )
}
```

Desafio 6:

- Mover os componentes Footer e MainContent em seus próprios arquivos.

8. Configurando o projeto - ReactFacts

Desafio 7: configuração do projeto

- Crie uma pasta `components`;
- Crie os seguintes componentes em arquivos separados dentro da pasta componentes. Em cada um, basta renderizar um `h1` com o nome do componente (por exemplo, retorne `<h1>Navbar vai aqui</h1>`):
 - Navbar
 - Main
- Crie um componente App fora da pasta de componentes (irmão do arquivo index.js);
- Faça com que App renderize a barra de navegação e os componentes principais;
- Importe e renderize o componente App dentro de index.js usando ReactDOM;
- Neste ponto, você deve ter o seu "Navbar vai aqui" etc. aparecendo no mini-navegador;
- Acesse as fontes do Google e obtenha a fonte "Inter" com pesos 400, 600 e 700. Coloque os links para essas fontes ACIMA do link styles.css em index.html (use os elementos `` em vez das opções @import ou npm para obter as fontes. Você pode precisar fazer alguma pesquisa extra para descobrir como isso funciona se você não tiver feito isso antes).

Navbar.js

```
import React from "react"

export default function Navbar() {
  return <h1>Navbar component</h1>
}
```



Main.js

```
import React from "react"

export default function Main() {
  return <h1>Main component</h1>
}
```

App.js

```
import React from "react"
import Navbar from "../components/Navbar"
import Main from "../components/Main"

export default function App() {
  return (
    <div className="container">
      <Navbar />
      <Main />
    </div>
  )
}
```

index.js

```
import React from "react"
import ReactDOM from "react-dom"
import App from "./App"

ReactDOM.render(<App />, document.getElementById("root"))
```

Navbar e estilização

Desafio 8:

- Complete a Navbar para combinar com o design.



- Dica: use o arquivo Figma para obter uma visão mais precisa do design (cores, tamanhos, fontes, etc.)

App.js

```
import React from "react"
import Navbar from "../components/Navbar"
import Main from "../components/Main"

export default function App() {
  return (
    <div className="container">
      <Navbar />
      <Main />
    </div>
  )
}
```

index.js

```
import React from "react"
import ReactDOM from "react-dom"
import App from "./App"

const root =
ReactDOM.createRoot(document.getElementById('root'));
root.render(
  <React.StrictMode>
    <App />
  </React.StrictMode>
);
```

styles.css

```
* {
  box-sizing: border-box;
```



```
}

body {
  margin: 0;
  font-family: Inter, sans-serif;
}

nav {
  display: flex;
  align-items: center;
  background-color: #21222A;
  height: 90px;
  padding: 30px 25px;
}

.nav--logo_text, .nav--title {
  margin: 0;
}

.nav--logo_text {
  margin-right: auto;
  color: #61DAFB;
  font-weight: 700;
  font-size: 22px;
}

.nav--title {
  color: #DEEBF8;
  font-weight: 600;
}

.nav--icon {
  height: 30px;
  margin-right: 7px;
}
```



Seção Main

Desafio 9:

- Construir a seção principal! Ignore 2 aspectos do design por enquanto:
 - Os marcadores coloridos na lista
 - O logotipo maior do React na lateral

styles.css

```
* {
  box-sizing: border-box;
}

body {
  margin: 0;
  font-family: Inter, sans-serif;
  height: 100vh;
  background-color: #282D35;
}

nav {
  display: flex;
  align-items: center;
  background-color: #21222A;
  height: 90px;
  padding: 30px 25px;
}

.nav--logo_text, .nav--title {
  margin: 0;
}

.nav--logo_text {
  margin-right: auto;
  color: #61DAFB;
  font-weight: 700;
  font-size: 22px;
}
```



```
}

.nav--title {
  color: #DEEBF8;
  font-weight: 600;
}

.nav--icon {
  height: 30px;
  margin-right: 7px;
}

main {
  padding: 57px 27px;
  color: white;
}

.main--title {
  margin: 0;
  font-size: 39px;
  letter-spacing: -0.05em;
}

.main--facts {
  margin-top: 46px;
  max-width: 400px;
}

.main--facts > li {
  line-height: 19px;
  padding-block: 10px;
}
```

Colorindo os botões



styles.css

```
.main--facts > li::marker {  
  font-size: 1.4rem;  
  color: #61DAFB;  
}
```

Adicionando o logo no fundo

Desafio 10:

- Colocar o logo cinza do react ao fundo Não use um elemento `img`, mas defina-o como o `background-image` do elemento `main`.
 - Dica: você precisará usar as seguintes propriedades:
 - `background-image`
 - `background-repeat`
 - `background-position`(Ou a propriedade abreviada `background` com valores para todas essas outras propriedades)

styles.css

```
main {  
  padding: 57px 27px;  
  color: white;  
  background-image: url(./images/react-icon-large.png);  
  background-repeat: no-repeat;  
  background-position: right 75%;  
}  
  
.main--title {  
  margin: 0;  
  font-size: 39px;  
  letter-spacing: -0.05em;  
}  
  
.main--facts {
```



```
margin-top: 46px;
max-width: 400px;
}

.main--facts > li {
  line-height: 19px;
  padding-block: 10px;
}

.main--facts > li::marker {
  font-size: 1.4rem;
  color: #61DAFB;
}
```