

Principais elementos do HTML

Essa lista não contém todos elementos do HTML, apenas os principais.
Para uma lista completa acesse: <https://developer.mozilla.org/pt-BR/docs/Web/HTML/Element>

Elemento raiz

`<html> ... </html>`

Define um documento HTML, **deve ficar por volta de todo conteúdo da nossa página**. Apenas um por página.

Elementos não-visuais

`<head> ... </head>`

Define o cabeçalho da aplicação, **não deve conter nenhum conteúdo visual**, apenas informações de configuração da página como importações de estilos, scripts, título e metadados. Apenas um por página.

`<style> ... </style>`

Adiciona uma estilização CSS no documento, **prefira sempre adicionar estilizações em arquivos externos** e importando com a tag <link>. Utilizado apenas dentro do <head>.

`<link rel="stylesheet" href="styles.css" />`

Adiciona uma estilização CSS no documento, **prefira sempre adicionar estilizações em arquivos externos** e importando com a tag <link>. Utilizado apenas dentro do <head>.

`<title> ... </title>`

Define o título do documento. Apenas um por página. Utilizado apenas dentro do <head>.

`<meta name="description" content="Um site muito legal" />`

Define informações de metadados do documento, por exemplo, descrição, palavras-chave, etc. Utilizado apenas dentro do <head>.

`<script> ... </script> ou <script src="scripts.js"></script>`

Define um script JavaScript para ser executado na página. O conteúdo do script pode estar dentro do elemento <script> ou ser importado de um arquivo externo usando o atributo "src". Geralmente utilizado **no fim do documento** antes da tag </body>.

`<body> ... </body>`

Define o corpo do documento, onde ficará **toda parte visual da nossa página**. Apenas um por página.

Elementos não-visuais

`<div> ... </div>`

Define um **bloco genérico sem significado**, serve apenas para agrupamento e estilização.

`<section> ... </section>`

Define um **bloco de conteúdo** no HTML que está isolado do restante do documento.

`<article> ... </article>`

Elemento usado para identificar um **conteúdo repetível** na página como um artigo, post, comentário, etc.

`<aside> ... </aside>`

Elemento usado para identificar uma sidebar ou **barra lateral** ao conteúdo principal.

`<header> ... </header>`

Define um **cabeçalho**.

`<footer> ... </footer>`

Define um **rodapé**.

`<main> ... </main>`

Define o **principal conteúdo** dentro da tag <body>, o que possui mais **relevância**.

`<nav> ... </nav>`

Elemento usado para identificar um **bloco de navegação** com múltiplos links na página, por exemplo, nosso **menu**.

Elemento visuais de lista

` ... `

Cria uma lista **sem ordenação**, apenas com • no início de cada item.

` ... `

Cria uma lista **com ordenação**, adiciona automaticamente um contador antes de cada item (1, 2, 3, ...).

` ... `

Elemento usado para identificar um item de uma lista, deve ser o **único elemento** dentro de uma ou .

Elemento visuais de multimídia

``

Exibe uma imagem dentro do documento. Pode ser uma imagem local ou uma URL na web.

`<video src="imagem.jpg" />`

Exibe um vídeo dentro do documento. Pode ser um vídeo local ou uma URL na web.

Elemento visuais de texto

`<h1> ... </h1>` (ou h2, h3, h4, h5 e h6)

Define um **título da página por ordem de importância**, um título h2 é menos importante que um título h1.

`<p> ... </p>`

Define um parágrafo de texto na página.

` ... `

Define um texto **em negrito com maior importância** do restante do texto.

` ... `

Define um texto **em itálico com maior ênfase** do restante do texto.

` ... `

Define um **texto genérico**, utilizado apenas para estilização.

` ... `

Define um **link** para um outro documento ou para uma URL externa.

` ... `

Define um texto **menor e com menos importância** do restante.

```
const [date, setDate] = useState(new Date());
```

```
const [date, setDate] = useState(new Date());
const [time, setTime] = useState(0);
```

```
const provider = navigation.getParam('provider');
```

```
useEffect(() => {
  async function loadProviders() {
    const response = await api.get('providers',{providerId:provider});
    const {
      data: data.getTime(),
    },
    {}
  };
  setTime(response.data);
}, [provider]);
```

```
loadProviders();
}, [date]);
```

```
function handleSelectDateTime() {
  navigation.navigate('Confirm', {
    provider,
    time,
  });
}
```

```
keyExtractor(item => item.time)
renderItem({ item }) => {
  time
  renderItem(item.available)
  renderItem(item.available)
  renderItem(item.available)
```