

O que é o React

Pablo Henrique

Desenvolvedor Full Stack na Fleye
Educador e Sócio fundador do Catapulta

@pablohdev

Objetivo Geral

Entender o que é o React e seus conceitos e criar a sua primeira aplicação.

Pré-requisitos

- Editor de código (Visual Studio Code)

Percurso

Etapa 1

O que é o React

Etapa 2

Framework x Biblioteca

Etapa 3

Criando Primeiro Projeto.

Percurso

Etapa 4

Componentes de Classes x Funcionais

Etapa 5

Conceito de components e Props

Etapa 6

Entendendo o ciclo de vida de um componente

Etapa 1

O que é o React

A história do React

React JS é uma biblioteca JavaScript para a criação de interfaces de usuário. Criado em 2011 pelo time do Facebook, o React surgiu com o objetivo de otimizar a atualização e a sincronização de atividades simultâneas no feed de notícias da rede social, entre eles chat, status, listagem de contatos e outros.

A história do React

A princípio, todas essas atividades, chamadas de estados, tinham uma descrição muito complexa. Com o React, esta descrição torna-se mais simples, bem como também é simplificada a conexão entre HTML, CSS e JavaScript e todos os componentes de uma página.

A história do React

Por ter demonstrado grande eficiência, nos anos que se seguiram o React foi incorporado à interface de outras redes sociais do grupo, como o Instagram e, em 2013, seu código foi aberto à comunidade, dando início a sua popularização

Percurso

Etapa 1

~~O que é o React~~

Etapa 2

Framework x Biblioteca

Etapa 3

Criando Primeiro Projeto.

Etapa 2

Framework x Biblioteca

Biblioteca

Esse é o recurso mais utilizado no mundo da programação e muitas pessoas nem se dão conta do quanto utilizam. A ideia da biblioteca é compartilhar soluções por meio de funções ou métodos. Por Exemplo: Se você tiver que fazer um trabalho de matemática, por exemplo, poderá ir até uma biblioteca física, pegar um livro e utilizar equações desenvolvidas no livro. Então, não será preciso desenvolver as equações desde o início

Exemplo de Biblioteca

- Moment.js
- Chart.js
- Voca
- mo.js
- React

Frameworks

O framework nada mais é do que uma ferramenta que vai te ajudar a ter como único objetivo focar em desenvolver o projeto, não em detalhes de configurações.

Exemplos de Frameworks

- Angular
- Vuex
- Ionic
- Next
- Express
- LoopBack

Percurso

~~Etapa 1~~

~~O que é o React~~

~~Etapa 2~~

~~Framework x Biblioteca~~

Etapa 3

Criando Primeiro Projeto.

Etapa 3

Criando Primeiro Projeto.

Percurso

~~Etapa 1~~

~~O que é o React~~

~~Etapa 2~~

~~Framework x Biblioteca~~

~~Etapa 3~~

~~Criando Primeiro Projeto.~~

Percurso

Etapa 4

Componentes de Classes x Funcionais

Etapa 5

Conceito de components e Props

Etapa 6

Entendendo o ciclo de vida de um componente

Etapa 4

Componentes de Classes x Funcionais

Percurso

~~Etapa 4~~

~~Componentes de Classes x Funcionais~~

Etapa 5

Conceito de components e Props

Etapa 6

Entendendo o ciclo de vida de um componente

Etapa 5

Conceito de components e Props

Percurso

~~Etapa 4~~

~~Componentes de Classes x Funcionais~~

~~Etapa 5~~

~~Conceito de components e Props~~

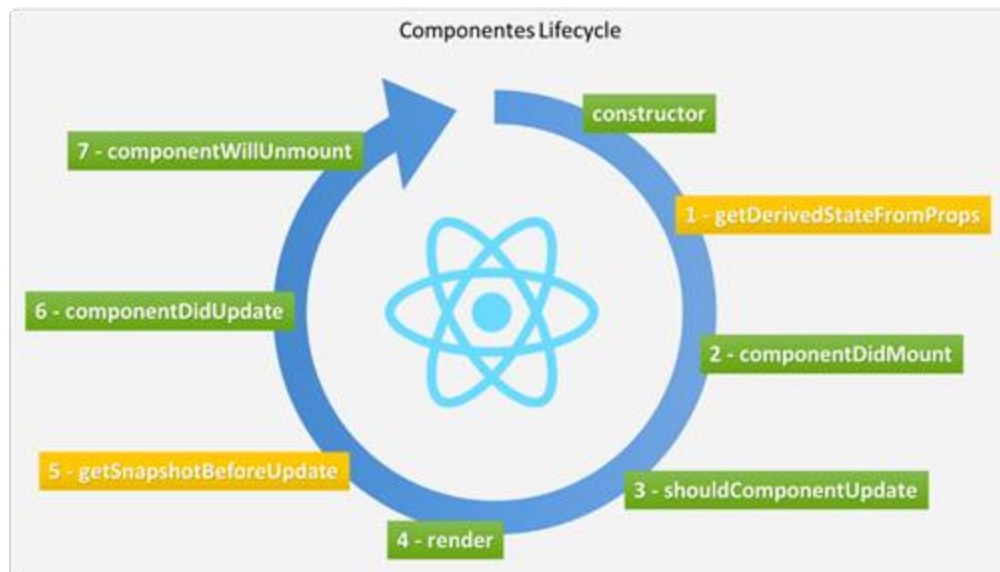
Etapa 6

Entendendo o ciclo de vida de um componente

Etapa 6

Entendendo o ciclo de vida de um componente

Entendendo o ciclo de vida de um componente



Percurso

~~Etapa 4~~

~~Componentes de Classes x Funcionais~~

~~Etapa 5~~

~~Conceito de components e Props~~

~~Etapa 6~~

~~Entendendo o ciclo de vida de um componente~~

Links Úteis

- Referências:

- <https://pt-br.reactjs.org/docs/getting-started.html>
- <https://www.treinaweb.com.br/blog/qual-a-diferenca-entre-framework-e-biblioteca>

Dúvidas?

- > Contem comigo.
- > Comunidade Online (Discord)



Entendendo a DOM (Document Object Model)

Pablo Henrique

Desenvolvedor Full Stack na Fleye
Educador e Sócio fundador do Catapulta

@pablohdev

Objetivo Geral

Entender o que é a DOM (Document Object Model) e como manipulada usando javascript.

Pré-requisitos

- Editor de código (Visual Studio Code)

Percurso

Etapa 1

O que é DOM

Etapa 2

Manipulando a DOM com javascript

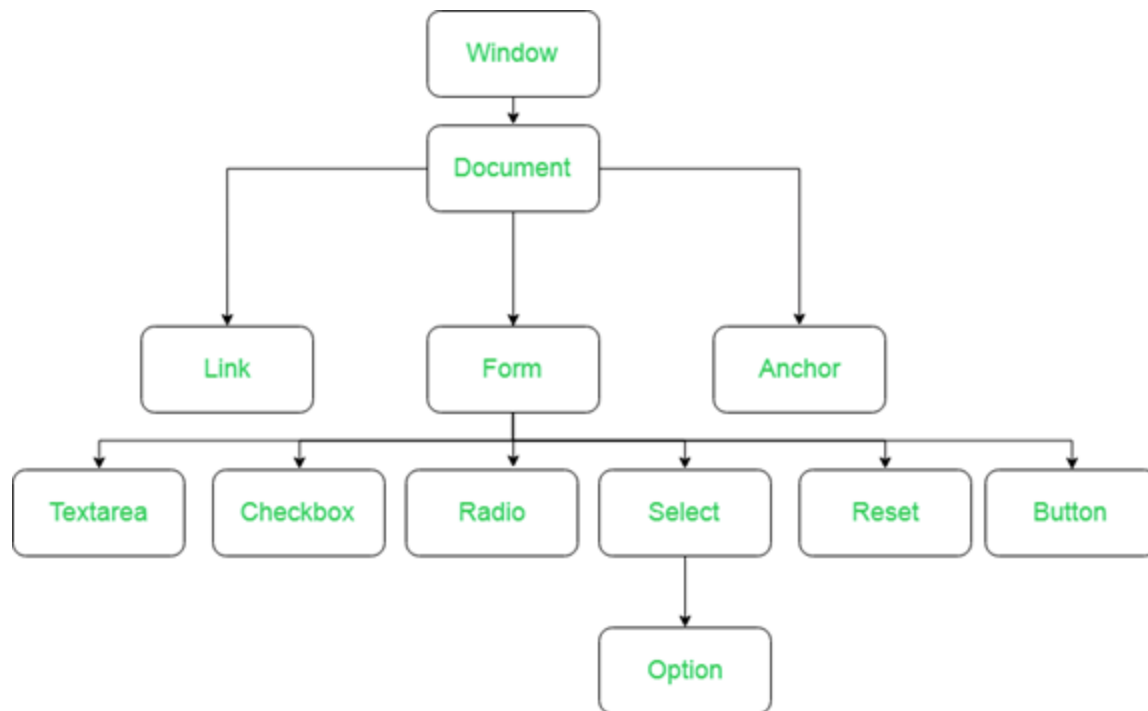
Etapa 1

O que é DOM

O que é DOM

O DOM (Document Object Model) é a representação de dados dos objetos que compõem a estrutura e o conteúdo de um documento na Web. O Document Object Model (DOM) é uma interface de programação para os documentos HTML e XML. Representa a página de forma que os programas possam alterar a estrutura do documento, alterar o estilo e conteúdo. O DOM representa o documento com nós e objetos.

O que é DOM



Percurso

Etapa 1

~~O que é DOM~~

Etapa 2

Manipulando a DOM com javascript

Etapa 2

Manipulando a DOM com javascript

Manipulando a DOM com javascript

O DOM não é uma linguagem de programação, mas sem ele, a linguagem JavaScript não teria nenhum modelo ou noção de páginas web, páginas XML e elementos com os quais ela geralmente lida. Cada elemento de um documento faz parte do Document Object Model, para que possam ser acessados e manipulados usando o DOM e uma linguagem de script, como JavaScript.

Operações Básicas

- `getElementsByTagName`
- `getElementsByClassName`
- `getElementById`
- `querySelector`

Percurso

~~Etapa 1~~

~~O que é DOM~~

~~Etapa 2~~

~~Manipulando a DOM com javascript~~

Links Úteis

- Referências:
 - [https://developer.mozilla.org/pt-BR/docs/Web/API/Document Object Model/Introduction](https://developer.mozilla.org/pt-BR/docs/Web/API/Document_Object_Model/Introduction)

Dúvidas?

- > Contem comigo.
- > Comunidade Online (Discord)



Javascript

Pablo Henrique

Desenvolvedor Full Stack na Fleye
Educador e Sócio fundador do Catapulta

@pablohdev

Objetivo Geral

Entender o que é o Javascript e fazer interações nas páginas web tornando-as mais dinâmicas.

Pré-requisitos

- Editor de código (Visual Studio Code)

Percurso

Etapa 1

O que é Javascript

Etapa 2

Operações Básicas

Etapa 3

Condicionais

Percurso

Etapa 4

Tipos de variáveis

Etapa 5

Laços de repetições

Etapa 6

Arrow Functions

Percurso

Etapa 7

Arrays

Etapa 8

Funcionalidades do ES6+

Etapa 1

O que é Javascript

A história do Javascript

JavaScript é uma Linguagem de Programação, criada por Brendan Eich, a pedido da empresa Netscape, em meados de 1995. No início, o JavaScript foi batizado com outro nome: LiveScript. No entanto, a Netscape não ficou sozinha com o desenvolvimento do JavaScript. A empresa SUN Microsystems interessou-se por ela e entrou de cabeça no desenvolvimento desta nova linguagem, uma vez que acreditava na ideia inovadora que era o JavaScript.

JAVA x JavaScript

Com o sucesso inicial do JavaScript, a mudança do nome de LiveScript para JavaScript foi inevitável, e, com certeza, veio por influência da própria SUN, que mantém uma Linguagem de Programação chamada JAVA. É claro que as Linguagens de Programação JAVA e JavaScript são parecidas somente no nome, já que se diferem no conceito e no uso.

ECMAScript e JavaScript

A ideia da web é funcionar igual para todos. A fim de manter um padrão, em novembro de 1996 a Netscape enviou o JavaScript à ECMA International, uma associação fundada em 1961 que padroniza sistemas de informação. Já que o nome JavaScript pertencia à Sun, foi preciso registrar com um novo nome. Assim surgiu o nome ECMAScript. Porém, como o nome JavaScript já estava famoso, continuamos chamando a linguagem assim até hoje em todos os lugares.

O que é Javascript

JavaScript é uma linguagem de programação de alto nível criada, a princípio, para ser executada em navegadores e manipular comportamentos de páginas web. Com seus scripts é possível incluir, em uma página estática, elementos dinâmicos como mapas, formulários, operações numéricas, animações, infográficos interativos e muito mais.

Percurso

~~Etapa 1~~

~~O que é Javascript~~

Etapa 2

Operações Básicas

Etapa 3

Condicionais

Etapa 2

Operações Básicas

Operações Básicas

Os operadores aritméticos são utilizados para a realização de cálculos simples em JavaScript. Representam as operações matemáticas básicas de soma, subtração, divisão e multiplicação, além de algumas operações especiais, como o módulo e incremento / decremento.

Operações Básicas

Operador	Significado
+	Soma
-	Subtração
*	Multiplicação
/	Divisão
%	Módulo
++	Incremento
--	Decremento

Hands On!

***“Falar é fácil.
Mostre-me o código!”***

Linus Torvalds

Percurso

~~Etapa 1~~

~~O que é Javascript~~

~~Etapa 2~~

~~Operações Básicas~~

Etapa 3

Condicionais

Etapa 3

Condicionalis

Condicionais

As estruturas condicionais estão ligadas à tomada de decisão de um algoritmo. Ao utilizar expressões que retornam verdadeiro ou falso, o algoritmo executa o bloco de comandos relativos a este resultado.

Estruturas condicionais básicas

- if
- else
- switch
- ternario

Operadores lógicos

Operador	Significado
&&	AND (E)
	OR (OU)
== ===	IGUAL
!= ou !==	DIFERENTE
>	Maior que
<	Menor que

Hands On!

***“Falar é fácil.
Mostre-me o código!”***

Linus Torvalds

Percurso

~~Etapa 1~~

~~O que é Javascript~~

~~Etapa 2~~

~~Operações Básicas~~

~~Etapa 3~~

Condicionais

Percurso

Etapa 4

Tipos de variáveis

Etapa 5

Laços de repetições

Etapa 6

Arrow Functions

Etapa 4

Tipos de variáveis

O que é variáveis.

Você usa variáveis como nomes simbólicos para os valores em sua aplicação. O nome das variáveis, chamados de identificadores, obedecem determinadas regras.

Uma variável em JavaScript deve começar com uma letra, underline (`_`), os caracteres subsequentes podem também ser números (0-9). Devido JavaScript ser case-sensitive, letras incluem caracteres de "A" a "Z" (maiúsculos) e caracteres de "a" a "z" (minúsculos).

Declarações de variáveis

Existem três tipos de declarações em JavaScript.

```
var nome = 'Pablo'
```

```
let idade = 25
```

```
const aprovado = true
```

Tipos de variáveis

Boolean = true e false.

null = Nulo

undefined = Não definido

Number = 42 ou 3.14159

String = "Howdy"

Hands On!

***“Falar é fácil.
Mostre-me o código!”***

Linus Torvalds

Percurso

~~Etapa 4~~

~~Tipos de variáveis~~

Etapa 5

Laços de repetições

Etapa 6

Arrow Functions

Etapa 5

Laços de repetições

Laços de repetições

Laços em Javascript (ou loops) são uma maneira eficaz de se trabalhar com estruturas de repetições baseadas em estruturas condicionais dentro da linguagem, ao invés de repetir milhares de vezes determinado código, você pode utilizar os laços para repetir qualquer coisa até que qualquer outra coisa aconteça

Laços de repetições

- for
- while
- foreach
- map

Hands On!

***“Falar é fácil.
Mostre-me o código!”***

Linus Torvalds

Percurso

~~Etapa 4~~

~~Tipos de variáveis~~

~~Etapa 5~~

~~Laços de repetições~~

Etapa 6

Arrow Functions

Etapa 6

Arrow Functions

Functions

De modo geral, função é um "subprograma" que pode ser chamado por código externo (ou interno no caso de recursão) à função. Assim como o programa em si, uma função é composta por uma sequência de instruções chamada corpo da função. Valores podem ser passados para uma função e ela vai retornar um valor.

Functions

```
function      Soma(num1,      num2)      {  
    return    num1+num2  
}
```


Arrow Functions

A versão do ECMA Script 2015 do JavaScript, trouxe uma nova forma mais sucinta de trabalhar com funções chamada de Arrow Functions, por causa da sintaxe que lembra uma flecha:

`() =>.`

Arrow Functions

```
const soma = (num1, num2) => {  
    return num1+num2  
}
```

Hands On!

*“Falar é fácil.
Mostre-me o código!”*

Linus Torvalds

Percurso

~~Etapa 4~~

~~Tipos de variáveis~~

~~Etapa 5~~

~~Laços de repetições~~

~~Etapa 6~~

~~Arrow Functions~~

Percurso

Etapa 7

Arrays

Etapa 8

Funcionalidades do ES6+

Etapa 7

Arrays

O que são Arrays

Arrays são geralmente descritas como "lista de objetos"; elas são basicamente objetos que contem múltiplos valores armazenados em uma lista. Um objeto array pode ser armazenado em variáveis e ser tratado de forma muito similar a qualquer outro tipo de valor, a diferença está em podermos acessar cada valor dentro da lista individualmente, e fazer super úteis e eficientes coisas com a lista, como laço através da lista e fazer a mesma coisa para cada valor

Exemplo de arrays

```
var shopping = ['bread', 'milk', 'cheese', 'hummus', 'noodles'];  
shopping;
```


Exemplo de arrays

```
let cities = [  
  {name: 'Los Angeles', population: 3792621},  
  {name: 'New York', population: 8175133},  
  {name: 'Chicago', population: 2695598},  
  {name: 'Houston', population: 2099451},  
  {name: 'Philadelphia', population: 1526006}  
];
```

Hands On!

*“Falar é fácil.
Mostre-me o código!”*

Linus Torvalds

Percurso

~~Etapa 7~~

Arrays

Etapa 8

Funcionalidades do ES6+

Etapa 8

Funcionalidades do ES6+

Funcionalidades do ES6+

- filter
- find
- findIndex
- reduce
- some
- every

Hands On!

***“Falar é fácil.
Mostre-me o código!”***

Linus Torvalds

Percurso

~~Etapa 7~~

~~Arrays~~

~~Etapa 8~~

~~Funcionalidades do ES6+~~

Links Úteis

- **Referências:**

- <https://www.w3schools.com/js/default.asp>
- <https://developer.mozilla.org/pt-BR/docs/Learn/JavaScript>

Dúvidas?

- > Contem comigo.
- > Comunidade Online (Discord)



CSS

Pablo Henrique

Desenvolvedor Full Stack na Fleye
Educador e Sócio fundador do Catapulta

@pablohdev

Objetivo Geral

Entender o que é o CSS e como fazer estilizações na página web.

Pré-requisitos

- Editor de código (Visual Studio Code)

Percurso

Etapa 1

O que é CSS

Etapa 2

O que é seletores

Etapa 3

Caixas (Box Models)

Percurso

Etapa 4

Flexbox

Etapa 5

Pseudo-classes e elementos

Etapa 6

Media Queries

Etapa 1

O que é CSS

O que é CSS

O CSS é chamado de linguagem Cascading Style Sheet e é usado para estilizar elementos nas páginas web. O CSS separa o conteúdo da representação visual do site. Pense na decoração da sua página. Utilizando o CSS é possível alterar a cor do texto e do fundo, fonte e espaçamento entre parágrafos. Também pode criar tabelas, usar variações de layouts, ajustar imagens para suas respectivas telas e assim por diante.

Primeiro contato com css

```
* {  
  margin: 0;  
  padding: 0;  
}  
  
html, body {  
  height: 100%;  
}  
  
ul {  
  list-style-position: inside;  
}  
  
ul li:nth-of-type(1) { /* vai selecionar o primeiro elemento da ul que seja do tipo li */  
  color: red;  
}  
  
ul li:nth-child(3) { /* vai selecionar o terceiro elemento da ul sem precisar ser de algum tipo */  
  color: blue;  
}  
  
ul li:last-of-type {  
  color: orange;  
}  
  
ul:last-child {  
  color: yellow;  
}
```

Percurso

Etapa 1

~~O que é CSS~~

Etapa 2

O que é seletores

Etapa 3

Caixas (Box Models)

Etapa 2

O que é seletores

O que é seletores

Os seletores são usados para direcionar os elementos HTML em nossas páginas da web que queremos estilizar. Há uma grande variedade de seletores CSS disponíveis, permitindo uma precisão refinada ao selecionar os elementos a serem estilizados.

Exemplo de seletores

```
h1 {  
  color: blue;  
  background-color: yellow;  
}  
  
p {  
  color: red;  
}
```

Exemplo de seletores

Tipo

```
h1 { }
```



Classe

```
.box { }
```



ID

```
#unique { }
```



Exemplo de seletores

Seletores de atributos

```
a[title] { }
```



```
a[href="https://example.com"] { }
```



Hands On!

***“Falar é fácil.
Mostre-me o código!”***

Linus Torvalds

Percurso

Etapa 1

~~O que é CSS~~

Etapa 2

~~O que é seletores~~

Etapa 3

Caixas (Box Models)

Etapa 3

Caixas (Box Models)

Caixas (Box Models)

Em uma página WEB, cada elemento é representado como um box retangular. Determinar o tamanho, propriedades - como sua cor, fundo, estilo das bordas - e a posição desses boxes é o objetivo do mecanismo de renderização.

Caixas (Box Models)

No CSS, cada um desses boxes retangulares é descrita usando o box model padrão. Este modelo descreve o conteúdo do espaço ocupado por um elemento. Cada box possui 4 edges: margin edge, border edge, padding edge e content edge.



Percurso

~~Etapa 1~~

~~O que é CSS~~

~~Etapa 2~~

~~O que é seletores~~

~~Etapa 3~~

~~Caixas (Box Models)~~

Percurso

Etapa 4

Flexbox

Etapa 5

Pseudo-classes e elementos

Etapa 6

Media Queries

Etapa 4

Flexbox

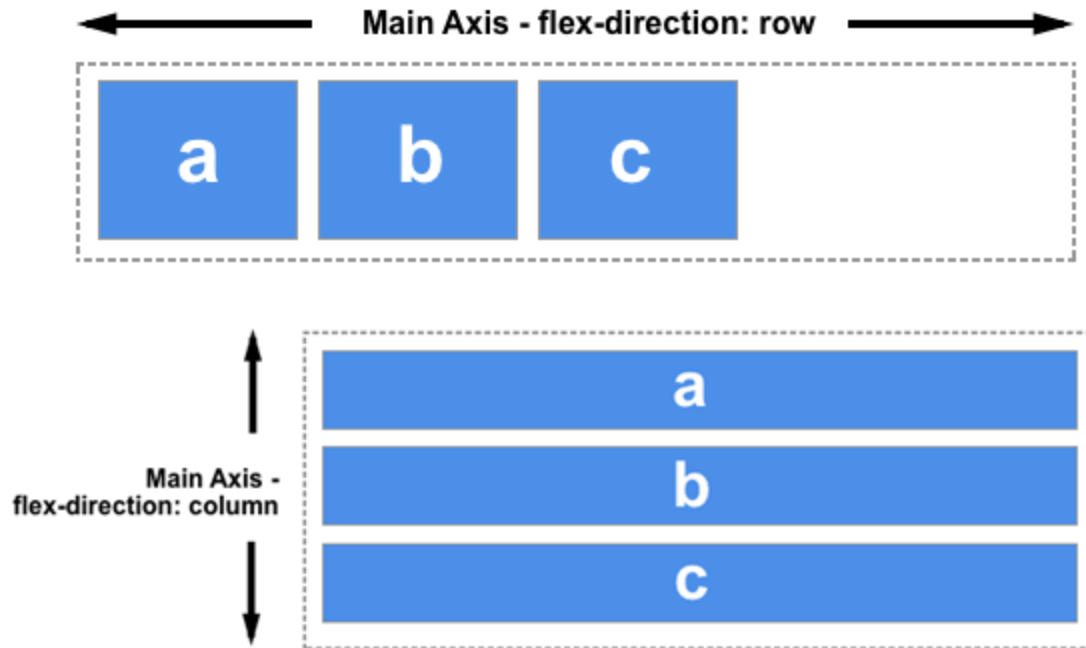
Flexbox

O flexbox foi projetado tanto como um modelo de layout unidimensional quanto como um método capaz de organizar espacialmente os elementos em uma interface, além de possuir capacidades avançadas de alinhamento. Quando se descreve o flexbox como sendo unidimensional, enfatiza-se o fato de que ele lida com o layout em uma dimensão de cada vez - seja uma linha ou uma coluna.

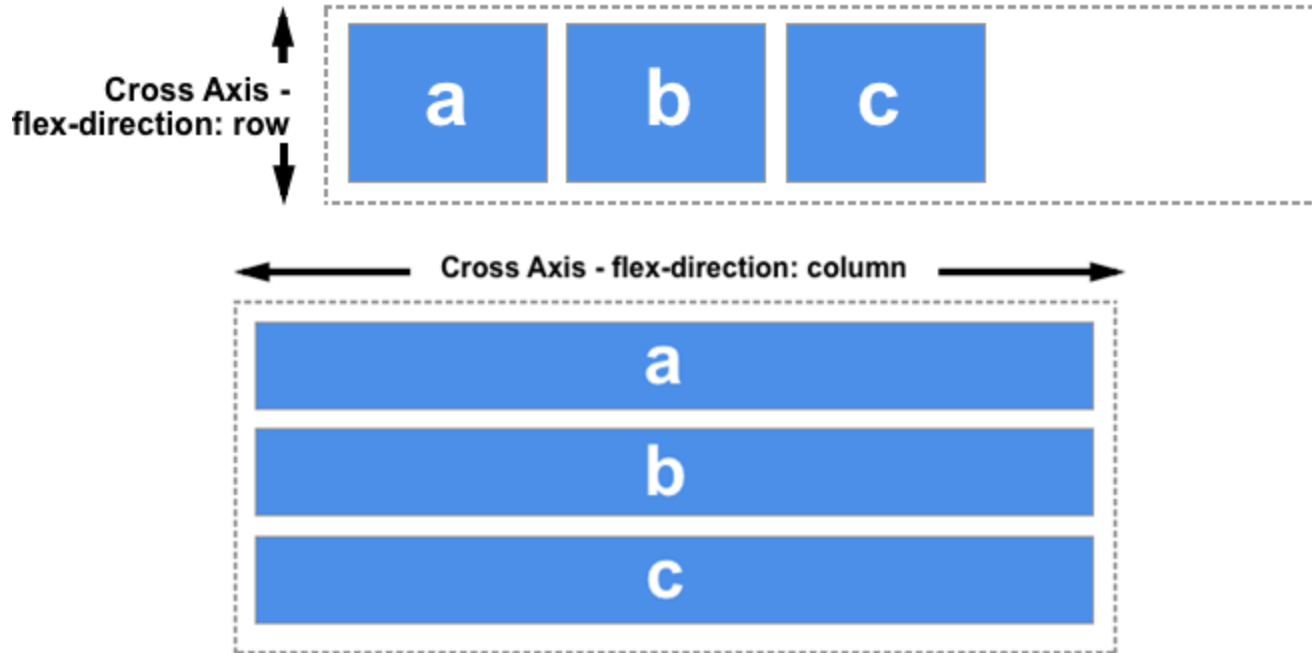
Os eixos do flexbox

Para se utilizar o flexbox, é preciso ter em mente que todas as operações realizadas relacionam-se a dois eixos: o eixo principal e o eixo transversal. O eixo principal é definido através da propriedade **flex-direction** e o eixo transversal encontra-se na direção perpendicular a ele.

O eixo principal (Main Axis)



O eixo transversal (Cross Axis)



Hands On!

***“Falar é fácil.
Mostre-me o código!”***

Linus Torvalds

Percurso

~~Etapa 4~~

~~Flexbox~~

Etapa 5

Pseudo-classes e elementos

Etapa 6

Media Queries

Etapa 5

Pseudo-classes e elementos

Pseudo-classes e elementos

Uma pseudo-classe CSS é uma palavra-chave adicionada a seletores que especifica um estado especial do elemento selecionado. Por exemplo, **:hover** pode ser usado para alterar a cor de um botão quando o usuário passar o cursor sobre ele.

Pseudo-classes e elementos

```
/*Qualquer botão sobre o qual o ponteiro do usuário esteja passando  
sobre*/  
button:hover {  
    color: blue;  
}
```



Hands On!

***“Falar é fácil.
Mostre-me o código!”***

Linus Torvalds

Percurso

~~Etapa 4~~

~~Flexbox~~

~~Etapa 5~~

~~Pseudo-classes e elementos~~

Etapa 6

Media Queries

Etapa 6

Media Queries

Media Queries

é a utilização de Media Types com uma ou mais expressões envolvendo características de uma media para definir formatações para diversos dispositivos. Os browsers ou as aplicações lêem as expressões definidas na Query, caso o dispositivo em questão se encaixe nessas requisições o CSS será aplicado.

Media types

Os media types definem para que tipo de media um certo código CSS é direcionado. O HTML foi gerado para ser portátil, ou seja, lido e interpretado por qualquer tipo de dispositivo. Só que cada dispositivo exibe o HTML de uma maneira, devido a fatores como, por exemplo, as dimensões da tela. Supondo que visitamos um site através de um desktop, ele não terá a mesma reação de que se visitarmos por um dispositivo móvel.

Utilizando Media Queries

```
<style>
@media (max-width: 600px)
{
  .facet_sidebar
  {
    display: none;
  }
}
</style>
```

Hands On!

*“Falar é fácil.
Mostre-me o código!”*

Linus Torvalds

Percurso

~~Etapa 4~~

~~Flexbox~~

~~Etapa 5~~

~~Pseudo-classes e elementos~~

~~Etapa 6~~

~~Media Queries~~

Links Úteis

- **Referências:**

- <https://www.w3schools.com/css/default.asp>
- <https://developer.mozilla.org/pt-BR/docs/Web/CSS>

Dúvidas?

- > Contem comigo.
- > Comunidade Online (Discord)



HTML

Pablo Henrique

Desenvolvedor Full Stack na Fleye
Educador e Sócio fundador do Catapulta

@pablohdev

Objetivo Geral

Entender o que é o HTML e construir estruturas básicas de um site.

Pré-requisitos

- Editor de código (Visual Studio Code)

Percurso

Etapa 1

Como funciona a Web

Etapa 2

O que é HTML

Etapa 3

Tags básicas

Percurso

Etapa 4

Metas tags

Etapa 5

Listas

Etapa 6

Tabelas

Percurso

Etapa 7

Formulários

Etapa 8

HTML Semântico

Etapa 1

Como funciona a Web

Como funciona a Web

A Web é constituída por 3 partes.



Percurso

Etapa 1 ~~Como funciona a Web~~

Etapa 2 **O que é HTML**

Etapa 3 Tags básicas

Etapa 2

O que é HTML

O que é HTML

O acrônimo HTML significa HyperText Markup Language, traduzindo ao português: Linguagem de Marcação de Hipertexto. O HTML é o componente básico da web, ele permite inserir o conteúdo e estabelecer a estrutura básica de um website. Portanto, ele serve para dar significado e organizar as informações de uma página na web.

O que é HTML

Através de um documento HTML, ou seja, um documento com a extensão .html ou .htm., o navegador faz a leitura do arquivo e renderiza o seu conteúdo para que o usuário final possa visualizá-lo. Os arquivos .html podem ser visualizados em qualquer navegador (Google Chrome, Safari, ou Mozilla Firefox).

O que é HTML

Geralmente um site é composto por diversas páginas HTML, sendo uma para cada página do website. Cada página consiste em uma série de tags , que podem ser considerados os blocos de construção das páginas. Portanto, esses blocos são a maneira com a qual o HTML faz a marcação dos conteúdos, criando a hierarquia e a estrutura do mesmo, dividido entre seções, parágrafos, cabeçalhos, e outros.

Hands On!

*“Falar é fácil.
Mostre-me o código!”*

Linus Torvalds

Percurso

Etapa 1 ~~Como funciona a Web~~

Etapa 2 ~~O que é HTML~~

Etapa 3 **Tags básicas**

Etapa 3

Tags básicas

Tags básicas

`<h1>,<h2>,<h3>,<h4>,<h5> e <h6>`

Tags para definir um título e subtítulos, variando de 1 a 6, sendo h1 o título mais importante e h6 o de menor importância.

Tags básicas

`<p>` - Tags para definir um parágrafo;

`<a>` – Tag de link, junto ao atributo href;

`<header>` – define um cabeçalho;

`<section>` – define uma seção;

`<article>` – define um artigo;

Tags básicas

`<div>` – define uma divisão;

`<footer>` – define um rodapé;

`<nav>` – define uma área de navegação (como menus);

`<table>` – define uma tabela;

Tags básicas

`` – define uma lista ordenada;

`` – define uma lista não ordenada;

`` – define o item de uma lista;

`<form>` – define um formulário;

`<input>` – define os campos do formulário;

`<textarea>` – define uma área para o usuário digitar um texto;

Tags básicas

`<button>` – define um botão;

`` – permite inserir uma imagem no seu documento.

Hands On!

*“Falar é fácil.
Mostre-me o código!”*

Linus Torvalds

Percurso

~~Etapa 1~~ ~~Como funciona a Web~~

~~Etapa 2~~ ~~O que é HTML~~

~~Etapa 3~~ ~~Tags básicas~~

Percorso

Etapa 4

Metas tags

Etapa 5

Listas

Etapa 6

Tabelas

Etapa 4

Meta Tags

Meta Tags

Meta tags são linhas de código HTML ou "etiquetas" que, entre outras coisas, descrevem o conteúdo do seu site para os buscadores. É nelas que você vai inserir as palavras-chave que facilitarão a vida do usuário na hora de te encontrar

Meta Tags básicas

```
<meta      name="keywords"      content="sites,      web,  
desenvolvimento, html, design">
```

Meta Tags básicas

```
<meta name="description" content=" Meta Tags - O que são e  
como utilizá-las">
```

Meta Tags básicas

```
<meta http-equiv="content-language" content="pt-br">
```

Meta Tags básicas

```
<meta http-equiv="content-type" content="text/html;  
charset=iso-8859-1">
```


Meta Tags básicas

```
<meta name="author" content="Pablo Henrique">
```

Meta Tags básicas

```
<meta http-equiv="refresh" content="5  
;url=http://www.novosite.com/">
```

Percurso

~~Etapa 4~~

~~Metas tags~~

Etapa 5

Listas

Etapa 6

Tabelas

Etapa 5

Listas

Listas

As listas são muito importantes quando queremos listar alguns itens no site e também para a criação de menu de navegação.

Tipos de Listas

- Listas ordenadas
- Listas desordenadas
- Listas por definição

Hands On!

***“Falar é fácil.
Mostre-me o código!”***

Linus Torvalds

Percurso

~~Etapa 4~~

~~Metas tags~~

~~Etapa 5~~

~~Listas~~

Etapa 6

Tabelas

Etapa 6

Tabelas

Tabelas

As tabelas são listas de dados em duas dimensões e são compostas por linhas e colunas. Portanto, são muito utilizadas para apresentar dados de uma forma organizada.

Exemplo de tabela

Marca:	Modelo:	Ano:	Valor:
Volkswagen	Golf	2009	RS\$38.500
Fiat	Palio	2010	RS\$19.700
Honda	Civic	2012	RS\$61.500
O melhor preço da região			

Nº	Nome	Sobrenome	Usuário
1	Joao	Mateus	johny
2	Jose	Silva	jsilva
Nº	Nome	Sobrenome	Usuário
1	Joao	Mateus	johny
2	Jose	Silva	jsilva

Hands On!

*“Falar é fácil.
Mostre-me o código!”*

Linus Torvalds

Percurso

~~Etapa 4~~

~~Metas tags~~

~~Etapa 5~~

~~Listas~~

~~Etapa 6~~

~~Tabelas~~

Percurso

Etapa 7

Formulários

Etapa 8

HTML Semântico

Etapa 7

Formulários

Formulários

O formulário HTML é um formulário de preenchimento de dados ou que resulta em uma ação desejada utilizando a linguagem de marcação HTML. É formado por um ou mais campos. Esses campos podem ser de textos, caixas de seleção, botões, radio buttons e checkboxes utilizando tags do próprio HTML. Dessa forma, o usuário pode interagir com a página ao executar ações através desses formulários.

Hands On!

***“Falar é fácil.
Mostre-me o código!”***

Linus Torvalds

Percurso

~~Etapa 7~~

~~Formulários~~

Etapa 8

HTML Semântico

Etapa 8

HTML Semântico

HTML Semântico

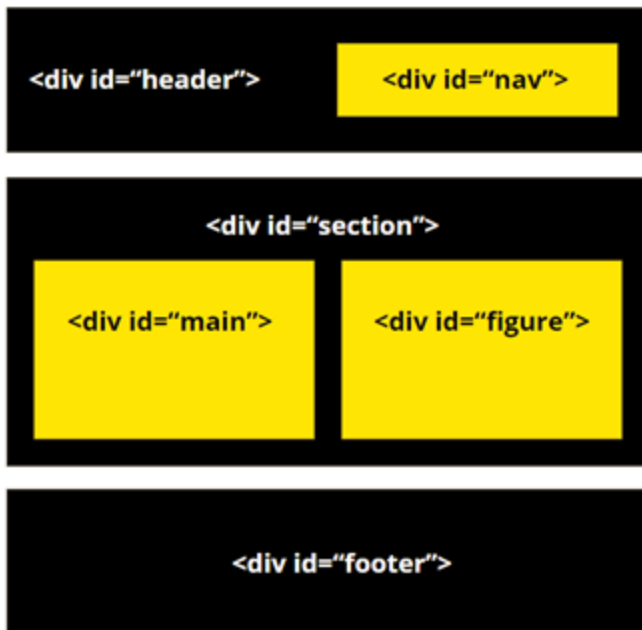
O HTML semântico é a forma de deixar o site com suas informações bem explicadas e compreensíveis para o computador, ajudando até mesmo em sua busca no Google e facilitando o entendimento de leitores de acessibilidade.

Porque utilizar HTML Semântico

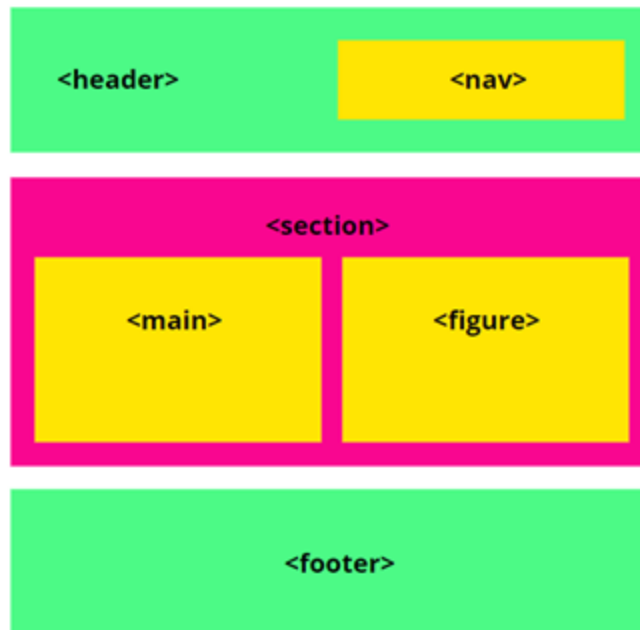
Além de facilitar a busca de forma orgânica e rankear sua página em mecanismos de busca, o HTML semântico ajuda os leitores de tela para deficientes visuais, facilitando seu entendimento. Ele, também, deixa seu código mais limpo e mais compreensível, tanto em organização como em facilidade de visualizar uma tag específica só de passar o olho pela tela.

Porque utilizar HTML Semântico

Antes



Hoje



Hands On!

*“Falar é fácil.
Mostre-me o código!”*

Linus Torvalds

Percurso

~~Etapa 4~~

~~Metas tags~~

~~Etapa 5~~

~~Listas~~

~~Etapa 6~~

~~Tabelas~~

Percurso

~~Etapa 7~~

~~Formulários~~

~~Etapa 8~~

~~HTML Semântico~~

Links Úteis

- **Referências:**

- <https://www.w3schools.com/html/default.asp>
- <https://developer.mozilla.org/pt-BR/docs/Web/HTML>

Dúvidas?

- > Contem comigo.
- > Comunidade Online (Discord)



Setup inicial e configuração de ambiente no React.

Pablo Henrique

Desenvolvedor Full Stack na Fleye
Educador e Sócio fundador do Catapulta

@pablohdev

Sobre Mim

Pablo Henrique

Desenvolvedor Full Stack na Fleye
Educador e Sócio fundador do Catapulta

@pablohdev



Objetivo Geral

Configurar todo o ambiente de desenvolvimento e estar apto a iniciar os estudos de React.

Pré-requisitos

- Computador com Windows, Linux ou Mac.

Percurso

Etapa 1

Instalando o visual studio code.

Etapa 2

Node e NPM

Etapa 3

Yarn

Percurso

Etapa 4

React DevTools

Etapa 5

Extensões para Visual Studio Code

Etapa 1

Instalando o visual studio code.

O que é o Visual Studio Code

O Visual Studio Code é um editor de código-fonte desenvolvido pela Microsoft para Windows, Linux e macOS. Ele inclui suporte para controle de versionamento, realce de sintaxe, snippets e refatoração de código.



Alternativa para VSCode



Como instalar o VSCode

<https://code.visualstudio.com>

Percurso

~~Etapa 1~~

~~Instalando o visual studio code~~

Etapa 2

Node e NPM

Etapa 3

Yarn

Percurso

Etapa 4

React DevTools

Etapa 5

Extensões para Visual Studio Code

Etapa 2

Node e NPM

O que é o Node

Node.js é um runtime de JavaScript, criado pelo desenvolvedor Ryan Dahl em 2009. o Node.js, foi desenvolvido em cima do motor JavaScript V8 – engine criada pelo Google e utilizada nos navegadores Chrome.



O que é o NPM

O NPM (Node Package Manager), é o gerenciador de pacote padrão do node. Ele é utilizado como gerenciamento de pacotes, fluxo de trabalho em linguagens de programação e ferramenta para construção de front-ends em aplicativos e websites.



<https://www.npmjs.com/>

Instalar o Node e NPM

<https://nodejs.org/en/>

Percurso

~~Etapa 1~~

~~Instalando o visual studio code~~

~~Etapa 2~~

~~Node e NPM~~

Etapa 3

Yarn

Percurso

Etapa 4

React DevTools

Etapa 5

Extensões para Visual Studio Code

Etapa 3

Yarn

O que é o Yarn

O Yarn é um gerenciador de pacotes para aplicar comandos prontos ao código de uma aplicação.



Instalar o Yarn

<https://yarnpkg.com/getting-started/install>

Percurso

~~Etapa 1~~

~~Instalando o visual studio code~~

~~Etapa 2~~

~~Node e NPM~~

~~Etapa 3~~

~~Yarn~~

Percurso

Etapa 4

React DevTools

Etapa 5

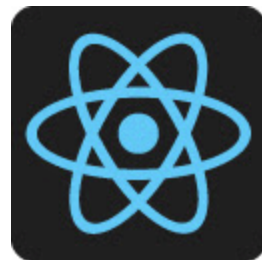
Extensões para Visual Studio Code

Etapa 4

React DevTools

O que é o React DevTools

É uma extensão disponível para o Chrome, Firefox e também como um aplicativo independente que permite inspeccionar a hierarquia de componentes do React nas Ferramentas do desenvolvedor do navegador.



Instalar o React DevTools

<https://github.com/facebook/react/tree/main/packages/react-devtools-extensions>

Percurso

~~Etapa 1~~

~~Instalando o visual studio code~~

~~Etapa 2~~

~~Node e NPM~~

~~Etapa 3~~

~~Yarn~~

Percurso

~~Etapa 4~~

~~React DevTools~~

Etapa 5

Extensões para Visual Studio Code

Etapa 5

Extensões para Visual Studio Code

O porque utilizar extensões?

Costumam ser leves, não comprometem o funcionamento do software e são de fácil instalação. Cada dev tem a sua maneira de programar e é por isso que as extensões podem ser ótimas aliadas! Com elas, conseguimos adaptar o VSCode às nossas necessidades, criando na rotina de trabalho e estudo um ambiente mais funcional.

Percurso

~~Etapa 1~~

~~Instalando o visual studio code~~

~~Etapa 2~~

~~Node e NPM~~

~~Etapa 3~~

~~Yarn~~

Percurso

~~Etapa 4~~

~~React DevTools~~

~~Etapa 5~~

~~Extensões para Visual Studio Code~~

Estamos prontos!

“A jornada é sua e somente sua. Outros podem caminhar com você, mas ninguém pode caminhar por você”

Links Úteis

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

Para saber mais



CURSO
Instalando e
Configurando seu
Ambiente Node.js

**Instalando e Configurando seu
Ambiente Node.js**

Aprenda de forma simplificada como
fazer a instalação e configuração das
principais ferramentas para iniciar a
programação com Node.js.

Básico 1 hrs

COMEÇAR AGORA



CURSO
Instalando
e Preparando o
Ambiente Angular

**Instalando e Preparando o Ambiente
Angular**

Está preparado para iniciar a seus
estudos sobre desenvolvimento Web com
Angular!? Neste curso, você conhecerá
todas as ferramentas necessárias para
iniciar a sua jornada como um Dev Front-
end.

Básico 2 hrs

COMEÇAR AGORA

Dúvidas?

- > Contem comigo.
- > Comunidade Online (Discord)

