Desenvolvimento Frontend II 09/05/2024

* Javascript 🡪 Vida da aplicação WEB
* Java =/= Javascript (Diferentes)

Ideia principal JS: Dar Dinamismo na aplicacao WEB em diferentes servidores.

* Fullstack 🡪 JS-BackEnd & JS-FrontEnd

Checar:

* Reacj JS : <https://react.dev/>

É uma biblioteca não um framework realiza a criação de tags próprias no HTML

* Anglular <https://angular.dev/>
* NextJS <https://nextjs.org/>

Framework para React Estrutura mais consolidada

* Vuejs <https://vuejs.org/>

Biblioteca

Não tão usada

Javascript progressivo

Framework nuxtjs : <https://nuxt.com/>

* three js <https://threejs.org/>

Bibliotea para desenvolvimento 3D (animação) com java SCRIPT

Testar

* 3 Principais frameworks:
  1. React
  2. Angular
  3. Vue
* Backend JavaScript

NodeJS – Javascript Lado Servidor

Express web framework(NODEJS)

Mongoose (NODEJS WEBSERVER)

Mongodb banco de dados

* Documentacao:

<https://developer.mozilla.org/pt-BR/docs/Learn>

<https://developer.mozilla.org/en-US/>

Documentação técnica nao muito didática mas bem completa.

<https://www.w3schools.com/>

Documentação mais amigável para iniciantes.

* TypeScript

Nada mais é que um JavaScript Tipado (Com tipos de variáveis fixos)

React pode trabalhar com Javascript ou com typescript

* Boas Praticas

Link para o CSS deve estar no HEAD <linkrel>

Link para o Javascript deve estar no Body <scriptsrc>

A Tag <scriptsrc> no head funciona mas não e boa pratica e sintaticamente não faz sentido.

Alert no navegador não deve mais ser usado codificação antiga.

* Flutter (Pesquisar sobre)

<https://flutter.dev/>

Kit de desenvolvimento de software que possibilita 1 codigo ser compilado para diferentes O.S. (sistemas operacionais). Muito utilizado para codar apps em diferentes linguagens

Desenvolvimento HIBRIDO

Linguagem DART

Codificacao:

* <https://www.w3schools.com/jsref/met_document_createelement.asp>

Document createElement()

Cria um elemento no DOM

* <https://www.w3schools.com/jsref/met_node_appendchild.asp>

HTML DOM Element appendChild()

Pendura um filho em um objeto

* DOM – Document Object Model

<https://www.w3schools.com/js/js_htmldom.asp>

The HTML DOM (Document Object Model)

When a web page is loaded, the browser creates a Document Object Model of the page.



* <https://www.w3schools.com/jsref/met_element_addeventlistener.asp>

HTML DOM Element addEventListener()

Adiciona um captador de eventos

* Let Var e Const
* **var** (**escopo global**) - escopo fora do bloco.
* **let** (**escopo local**) - escopo restrito ao bloco.
* **const** (não permitindo reatribuição e nem redeclaração) - constante.

ariáveis de **var** podem ser atualizadas e declaradas novamente dentro de seu escopo (região do código). Esse tipo de variável existe desde o começo na linguagem JavaScript.

As variáveis de **let** podem ser atualizadas, mas não podem ser declaradas novamente.

As variáveis de **const** não podem ser atualizadas nem declaradas novamente (são utilizadas para valores constantes, como o valor de pi, por exemplo).

Enquanto **var** e **let** podem ser declarados sem serem inicializadas, variáveis declaradas com **const** devem ser inicializadas no momento da declaração.

Atualmente, **let** tem sido preferida pois em um código muito extenso, criado por um grupo de pessoas, por exemplo, evita que ocorra de duas variáveis distintas serem declaradas com o mesmo nome, o que causaria conflito e um retorno totalmente diferente do esperado, entre outros possíveis problemas. Ela foi criada posteriormente, na linguagem JavaScript (e também **const**), justamente para evitar confusão.

**var**

* a variável **var** é declarada e iniciada no escopo da função
* **não** respeitando bloco e permitindo a redeclaração e reatribuição.

**let**

* a variável **let** é declarada no escopo da função, mas só é inicializada posteriormente
* respeitando bloco e permitindo reatribuição, mas **não** permite a redeclaração.

**const**

* a variável **const** é declarada no escopo da função, mas só é inicializada posteriormente
* respeitando bloco e **não** permitindo reatribuição **nem** redeclaração.

**OBS 1**:

* Ao declarar uma variável sem o var, sem o let ou sem o const, ela é criada no escopo global.

**OBS 2**:

* Nunca declare variáveis sem **var, let e const** e evite a utilização de **var**.

Usar const sempre que possível

Quando não possível usar const usar LET

Nunca usar VAR

Pesquisar mais sobre

* <https://www.w3schools.com/jsref/jsref_parseint.asp>

JavaScript parseInt()

Converte texto em string

EM Js

== (compara ao valor)

=== (Compara tipo e valor)

* Plugin VSCODE

TOGGLE SCREENCAST MODE

Mostra atalhos e commandos

* Spa (Single Page Application) MPA (Multi-Page Applications)

<https://www.orientsoftware.com/blog/spa-vs-mpa/>

* Antes de estudar Frameworks estudar:
  1. HTML

CSS

POSICIONAMENTO

FLEXBOX

Teste final pegar um figma e implementar um layout

Tester recriar paginas famosas

* 1. Java Scritpt

Manipular Tela

Criar elementos

* 1. Frameworks

Angluar

React

* Json (Java Script Object Notation)
* Fetch :Funcao JS para acessar endereço e deceber a resposta de uma requisição pipeline http

<https://www.w3schools.com/jsref/api_fetch.asp>

JavaScript Fetch API

* Promise : Eventual conclusão (ou falha0 de uma operação assíncrona e seu valor resultante

<https://www.w3schools.com/js/js_promise.asp>

quando retorno for promise e quisermos os dados temos que usar o await

Porem somente podemos usar await em funções async

<https://www.w3schools.com/js/js_async.asp>

* .Json \(transforma retorno em JSON)

<https://developer.mozilla.org/en-US/docs/Web/API/Response/json>

* Console.table em um JSON transforma o JSON em tabela.

<https://www.w3schools.com/jsref/met_console_table.asp>

* Template String: Foenece uma maneira fácil de interpolar variáveis e expressões em STRINGS

<https://www.w3schools.com/js/js_string_templates.asp>

* Section Java script pesquisar
* Hoisting (Içar)

Ele lê (precompila) todas as funções em memoria antes de usar.

<https://www.w3schools.com/js/js_hoisting.asp>

Usar para teste de loja própria

<https://makeup-api.herokuapp.com/api/v1/products.json>

<https://makeup-api.herokuapp.com/api/v1/products.json>

<https://www.macoratti.net/19/07/net_apipub1.htm>

<https://medium.com/reactbrasil/10-apis-gr%C3%A1tis-e-legais-para-voc%C3%AA-consumir-69141988ea0b>

<https://www.google.com/search?sca_esv=e7e7bfd0bcfc3035&rlz=1C1GCEU_enBR1009BR1009&q=API+de+teste+online&sa=X&ved=2ahUKEwiW6t3m0eKGAxVjrZUCHeuEALAQ1QJ6BAg9EAE&biw=1920&bih=953&dpr=1>