Guia Rápido de Comandos

Atalhos dentro do VS Code

https://code.visualstudio.com/shortcuts/keyboard-shortcuts-windows.pdf

HTML Tags

https://developer.mozilla.org/pt-BR/docs/Web/HTML/Element

Declaração de variáveis, constantes e seus escopos

var

Declara uma variável em escopo global.
https://developer.mozilla.org/pt-BR/docs/Web/JavaScript/Reference/Statements/var

let

Declara uma variável local no escopo do bloco atual, opcionalmente iniciando-a com um valor.

const

https://developer.mozilla.org/pt-BR/docs/Web/JavaScript/Reference/Statements/constt

Declara uma constante com escopo de bloco.
O valor de uma constante não pode ser alterado por uma atribuição, e ela não pode ser redeclarada.

Estruturas da linguagem Javascript

if...else

https://developer.mozilla.org/pt-BR/docs/Web/JavaScript/Reference/Statements/if...else

A condicional if é uma **estrutura condicional** que executa a afirmação, dentro do bloco, se determinada condição for verdadeira. Se for falsa, executa as afirmações dentro de else.

while

https://developer.mozilla.org/pt-BR/docs/Web/JavaScript/Reference/Statements/while

A declaração while cria um **laço (loop)** que executa uma rotina especifica enquanto a condição de teste for avaliada como verdadeira. A condição é avaliada antes da execução da rotina.

for

https://developer.mozilla.org/pt-BR/docs/Web/JavaScript/Reference/Statements/for

A instrução for cria um **laço (loop)** que consiste em três expressões opcionais, dentro de parênteses e separadas por ponto e vírgula, seguidas por uma declaração ou uma sequência de declarações executadas em sequência.

Funções

Declaração de função

https://developer.mozilla.org/pt-BR/docs/Web/JavaScript/Reference/Statements/function#declara%C3%A7%C3%B5es_de_fun%C3%A7%C3%B5es_hoisting

Arrow functions

https://developer.mozilla.org/pt-BR/docs/Web/JavaScript/Reference/Functions/Arrow_functions

DOM

document.getElementById()

https://developer.mozilla.org/pt-BR/docs/Web/API/Document/getElementById

Retorna a referência do elemento através do seu ID.

document.getElementsByClassName()

https://developer.mozilla.org/pt-BR/docs/Web/API/Document/getElementsByClassName

Retorna um **vetor de objetos** com todos os **elementos filhos que possuem o nome da classe dada**. Quando invocado no objeto document, o documento é examinado por completo, incluindo o nó raiz. Você também pode invocar getElementsByClassName() em qualquer elemento; isso retornaria somente elementos que são descendentes do nó raiz especificado com o nome da classe.

Propriedades de objetos de elementos HTML

.textContent

https://developer.mozilla.org/pt-BR/docs/Web/API/Node/textContent

A propriedade textContent da interface Node representa o conteúdo de texto de um nó e dos seus descendentes.

.innerText

https://developer.mozilla.org/pt-BR/docs/Web/API/HTMLElement/innerText

innerText é uma propriedade que representa o conteúdo textual "renderizado" de um nó e seus descendentes. Usada como getter, retorna de maneira aproximada o texto que o usuário obteria caso tivesse selecionado o conteúdo e copiado para a área de transferência.

.innerHTML

https://developer.mozilla.org/pt-BR/docs/Web/API/Element/innerHTML

A propriedade Element.innerHTML define ou obtém a sintaxe HTML ou XML descrevendo os elementos descendentes.

Math

https://developer.mozilla.org/pt-BR/docs/Web/JavaScript/Reference/Global_Objects/Math

Math é um objeto embutido que tem propriedades e métodos para constantes e funções matemáticas. Não é um objeto de função.

Arrays

.length

https://developer.mozilla.org/pt-BR/docs/Web/JavaScript/Reference/Global_Objects/Function/length

A propriedade length especifica o número de argumentos esperados pela função.

.push()

https://developer.mozilla.org/pt-BR/docs/Web/JavaScript/Reference/Global_Objects/Array/push

O método push() adiciona um ou mais elementos ao **final** de um array e retorna o novo comprimento desse array.

.pop()

https://developer.mozilla.org/pt-BR/docs/Web/JavaScript/Reference/Global_Objects/Array/pop

O método pop() **remove o último** elemento de um array e retorna aquele elemento.

.shift()

https://developer.mozilla.org/pt-BR/docs/Web/JavaScript/Reference/Global_Objects/Array/shift

O método shift() **remove o primeiro** elemento de um array e retorna esse elemento. Este método muda o tamanho do array.

.unshift()

https://developer.mozilla.org/pt-BR/docs/Web/JavaScript/Reference/Global_Objects/Array/unshift

O método unshift() **adiciona** um ou mais elementos no **início** de um array e retorna o número de elementos (propriedade length) atualizado.

.splice()

https://developer.mozilla.org/pt-BR/docs/Web/JavaScript/Reference/Global_Objects/Array/splice

O método splice() **altera o conteúdo de uma lista**, **adicionando** novos elementos enquanto **remove** elementos antigos.

Element.classList

https://developer.mozilla.org/pt-BR/docs/Web/API/Element/classList

O Element.classList é uma propriedade somente leitura que retorna uma coleção DOMTokenList (en-US) ativa dos atributos de classe do elemento.

Usar classList é uma alternativa conveniente para acessar a lista de classes de um elemento como uma seqüência delimitada por espaço através de element.className.

Métodos:

classList.remove();
classList.add();
classList. contains();
classList.toggle();

Window.localStorage

https://developer.mozilla.org/pt-BR/docs/Web/API/Window/localStorage

A propriedade localStorage permite acessar um objeto Storage local

localStorage.getItem()

https://developer.mozilla.org/pt-BR/docs/Web/API/Storage/getItem

Retorna itens armazenados no **local storage**.

localStorage.setItem()

https://developer.mozilla.org/pt-BR/docs/Web/API/Storage/setItem

O método setItem() da interface Storage, quando passado 'chave' e 'valor', irá adicionar esta chave ao storage, ou atualizar o valor caso a chave já exista.

localStorage.removeItem()

https://developer.mozilla.org/pt-BR/docs/Web/API/Storage/removeItem

Remove um item armazenado no local storage. Quando passado um nome de chave, irá remover essa chave do armazenamento.

Storage.clear()

https://developer.mozilla.org/pt-BR/docs/Web/API/Storage/clear

O método clear() limpa todas as chaves e dados armazenados no local storage.

Extensões do VS Code

vscode-icons

https://marketplace.visualstudio.com/items?itemName=vscode-icons-team.vscode-icons

Live Preview

https://marketplace.visualstudio.com/items?itemName=ms-vscode.live-server

Live Server

https://marketplace.visualstudio.com/items?itemName=ritwickdey.LiveServer

Error Lens

https://marketplace.visualstudio.com/items?itemName=usernamehw.errorlens

Tema - Dracula Official

https://marketplace.visualstudio.com/items?itemName=dracula-theme.theme-dracula

HTML CSS Support

https://marketplace.visualstudio.com/items?itemName=ecmel.vscode-html-css