**O que é o JS?**

* O JavaScript é uma linguagem de programação de alto nível, que integra o desenvolvimento de apps e páginas web. É uma das linguagens mais usadas no mundo do desenvolvimento.

**Função**

* Cria scripts dinâmicos que realizam a interação de apps ou páginas web. Imagine que o HTML é a estrutura, o CSS é o design e o JavaScript é a parte interativa do desenvolvimento.

**Tipagem**

* A tipagem funciona como uma regra de uso de dados, quanto mais forte for, mais obrigatório é a declaração do tipo de dado. A tipagem em JavaScript é fraca, a declaração dos dados acontece de modo dinâmico;
* Ex: ao criar uma variável com valor entre aspas (“valor”) o JavaScript já converte o dado para o tipo string.

**Tipos Primitivos**

* As variáveis em JavaScript podem guardar tipos de dados que chamamos de tipos primitivos. Eles são: boolean, null, undefined, number, string, array, object, function.

**Declaração de Variáveis**

* Existem 3 modos de declarar as variáveis em JavaScript:

1. var – escopo global e local, pode ter seu valor alterado, se não tiver um valor inicial será tratada como null;
2. let – escopo local de bloco, pode ter seu valor alterado, se não tiver um valor inicial será tratada como null;
3. const – escopo local de bloco, somente leitura, o valor inicial é obrigatório e não pode ser alterado.

**Escopo**

* O escopo em JavaScript define a limitação e visibilidade de um bloco de código.

**-** Escopo global: quando a variável é declarada fora de qualquer bloco, sua visibilidade fica disponível em todo o código;

**-** Escopo local: quando a variável é declarada dentro de um bloco, sua visibilidade pode ficar disponível ou não.

**Regras para Criação de Variáveis**

* Inicie o nome da variável com letra, underscore ou cifrão. Não inicie com número;
* Não utilize espaços. Use camelCase, PascalCase ou snake\_case;
* Não utilize palavras reservadas;
* Declare as variáveis no topo do bloco de código.

**Atribuição**

* O sinal de igualdade “=” em JavaScript, significa atribuição.

**Comparação**

* Para fazermos uma comparação de valores em JavaScript usamos “==”.

**Comparação Idêntica**

* Para fazermos uma comparação de valores e tipos em JavaScript usamos “===”.

**Operadores Aritméticos**

* Adição (+), subtração (-), multiplicação (\*), divisão real (/), resto da divisão (%), potenciação (\*\*).

**Operadores Relacionais**

* Maior que (>), menor que (<), maior ou igual a (>=), menor ou igual a (<=).

**Operadores Lógicos**

* E (&&) --> Considera que todos os valores sejam true;
* Ou (||) --> Considera que qualquer um dos valores seja true;
* Não (!) --> Inverte o valor de true para false ou vice-versa.

**Vetores ou Arrays**

* Arrays são um tipo de lista, ou matriz de variáveis, onde cada variável possui um índice. Os valores podem ser de vários tipos;
* O array deve ser declarado entre colchetes “[]”, e podem guardar qualquer valor dentro de seus índices, inclusive outros arrays;
* O índice só é acessado por um número inteiro, onde 0 é o primeiro índice e cada índice é separado por uma vírgula;
* Ao ser declarado, o Array traz consigo uma sério de métodos para manipulá-lo (Exemplos: forEach(), push (), pop (), etc).

**Objetos**

* Dados que possuem propriedades e valores que definem suas características. Deve ser declarado entre chaves “{}”;
* As propriedades dos objetos podem ser atribuídas as variáveis, facilitando a manipulação do objeto. Chamamos isso de desestruturação;
* Outra forma de fazer a desestruturação é utilizando chaves ao declarar a variável.

**Estruturas Condicionais**

* São instruções para realizar determinadas tarefas a partir de uma condição, seja de decisão ou repetição.

**Estruturas de Decisão**

* Podemos usar as palavras reservadas “if”, “else” e “else if” para estabelecer condições;
* Podemos usar a palavra “if” para estabelecer uma condição, “else” no caso de a condição não ser atendida e “else if” caso haja mais de uma condição;
* Podemos usar o “if” dentro de um outro “if”, chamamos isso de aninhamento de if’s ou ninho de if’s;
* Podemos também fazer uma verificação em uma única linha usando o “if” ternário;
* O “switch/case” também funciona como uma estrutura condicional.

**Estruturas de Repetição**

* Laços de repetição são estruturas condicionais que repetem uma instrução até atingir determinada condição;
* For: funciona como uma repetição de instrução até que a condição seja falsa;
* For/in: funciona como uma repetição a partir de uma propriedade;
* For/of: funciona como uma repetição a partir de um valor. O For/of não funciona com objetos, pois as propriedades variam, diferentes do índice em um array que sempre serão números inteiros;
* While: executa uma instrução “enquanto” determinada condição for verdadeira, a verificação é feita antes da execução.

**Comandos / Funções**

* alert (‘Texto’); -->
* var / let / const nome da variável = atribuição; -->
* console.log (nome da variável); -->
* console.log (typeof (nome da variável)); -->
* // ou /\* \*/ -->

**Links Úteis:**

**Glossário:**

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

**O que é o JavaScript:**

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

**Repositório do Curso “Sintaxe Básica em JavaScript”:**

github.com/DianaMartine/curso-dio-sintaxe-basica-javascript

**Tipagem:**

https://danvitoriano.medium.com/tipagem-din%C3%A2mica-no-javascript-e3551a445b38

**Sintaxe e Tipos de Variáveis:**

<https://developer.mozilla.org/pt-BR/docs/Web/JavaScript/Guide/Grammar_and_types>

**Escopos:**

<https://imasters.com.br/desenvolvimento/escopos-em-javascript>

**Expressões e Operadores:**

<https://developer.mozilla.org/pt-BR/docs/Web/JavaScript/Guide/Expressions_and_Operators#operador_atribuicao>

**Arrays:**

https://developer.mozilla.org/pt-BR/docs/Web/JavaScript/Reference/Global\_Objects/Array

**Array.protoype.indexOf():**

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

**Trabalhando com Objetos:**

<https://developer.mozilla.org/pt-BR/docs/Web/JavaScript/Guide/Working_with_Objects>

**Desestruturação, parâmetros rest e espalhamento:**

<https://www.digitalocean.com/community/tutorials/understanding-destructuring-rest-parameters-and-spread-syntax-in-javascript-pt>

**Estruturas Condicionais**

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

**Laços de Repetição**

https://developer.mozilla.org/pt-BR/docs/Web/JavaScript/Guide/Loops\_and\_iteration