**Java Script**

Java Script e Java são duas linguagens diferentes, a razão pelos nomes serem parecidos é que o JS se inspirou no nome Java para repercutir e ganhar fama, foi uma jogada de marketing. Foi criada em 1995.

Várias ferramentas surgiram com a evolução do JS. Por isso que é bom SABER SOBRE JS ANTES DE APRENDER SOBRE ESSAS FERRAMENTAS.

Ecma Script: É o JS padronizado. O Ecma define padrões para uma linguagem, e o JS é a implementação desses padrões.

Node.JS: Software de código aberto, baseado no interpretador V8 do google que permite a execução de códigos JS fora de um navegador.

TypeScript: Linguagem de programação de código aberto desenvolvida pela Microsoft. É um superconjunto sintático estrito de JavaScript e adiciona tipagem estática opcional à linguagem.

jQuery: Conjunto de bibliotecas facilitando interatividade em JS;

Angular: Framework criado pelo Google em 2009, facilita aplicações web, bem declarativa, semelhante ao SQL para banco de dados, usa TypeScript que por sua vez usa JS;

React: Biblioteca criada pelo facebook, semelhante ao Angular, porém mais flexível em algumas áreas;

Vue: Framework Js de código aberto, focado no desenvolvimento de interfaces de usuário e aplicativos de página única;

Electron: Framework de código aberto agora desenvolvido pelo GitHub. Permite desenvolver aplicações para desktop GUI usando componentes front end e back end originalmente criados para aplicações web;

Ionic: SDK (ou kit de desenvolvimento de software) para criar aplicações para dispositivos mobile;

Cordova: Estrutura de desenvolvimentos de aplicativos mobile criada pela Nitobi.

Comandos básicos JS:

Document: Representa o conteúdo HTML na tela;

Write: Vai escrever alguma coisa;

Window: Abre uma janela ao entrar na página (Geralmente transforma tudo em string...);

Alert: Soltará um alerta assim que se entrar na página;

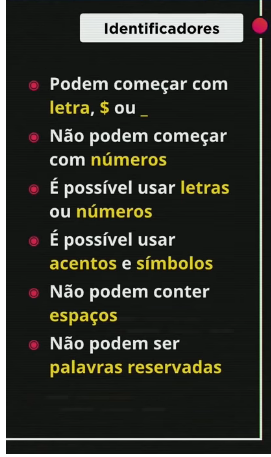
Confirm: Pedirá uma confirmação do usuário;

Prompt: Pedirá uma informação específica para o usuário digitar.

Armazenamento de Dados: Sempre serão guardados em espaços pré-definidos na memória do pc (Como em pastas). O mesmo se dá aos dados no código, para armazená-los precisamos definir espaços para eles.

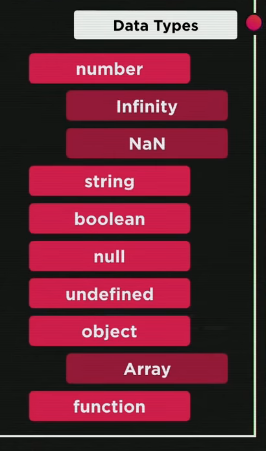
Variáveis em JS

As variáveis são os espaços que pré-determinamos para guardar dados, para identifica-las precisamos nomeá-las (identifica-las). Quando um valor é colocado dentro da variável chamamos esse processo de “atribuição”.



Tipos em JS

Os tipos mostram de que tipo é o dado informado na variável (number, string, boolean, etc).



Primitivos

Number: Representa números inteiros ou de ponto flutuante, exemplo: 42 ou 3.14.

Boolean: Representa valores lógicos, verdadeiro ou falso (True or False);

Undefined: Representa a ausência de valor;

Symbol: Tipo único e imutável, utilizado como identificadores únicos para objetos;

Typeof: Mostra o tipo da variável.

String: Representa uma cadeia de caracteres, exemplo: “Olá mundo”;

Null: Nulo.

Não primitivos (tipos de objetos)

Object: representa uma coleção de propriedades, que podem ser acessadas por chave-valor;

Array: representa uma coleção ordenada de valores, que podem ser acessados por índice;

Function: representa uma função que pode ser executada para realizar uma tarefa específica;

Date: representa uma data e hora específicas;

Map: representa uma coleção de pares chave-valor;

Set: representa uma coleção única de valores;

RegExp: representa uma expressão regular, usada para fazer correspondências em cadeias de caracteres;

Promise: representa uma operação assíncrona que eventualmente retorna um valor.

Conversão

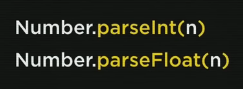
String > número

Usamos parse + o formato do valor;

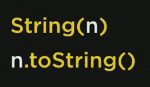
Int: Para inteiros;

Float: Para flutuantes;

Number(n): O código se vira, muito utilizado hoje em dia para conversão.

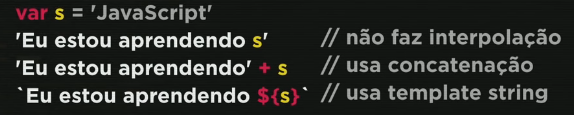


Número > String



Formatando

Strings



Template: é a forma mais recente e fácil de formatar Strings;

Length: Mostra quantos caracteres a string tem;

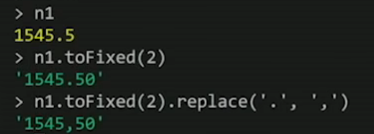
toUpperCase: Deixa tudo em letras maiúsculas;

toUpperCase: Deixa tudo em letras minúsculas.

Números

ToFixed(): Pode determinar quantas em casa queremos o número;

.replace(): Substitui;



N1.toLocaleString( ‘language’ ({style: ‘currency’, currency: ‘...’}): Muito utilizado para conversão de números em moedas.



Operadores

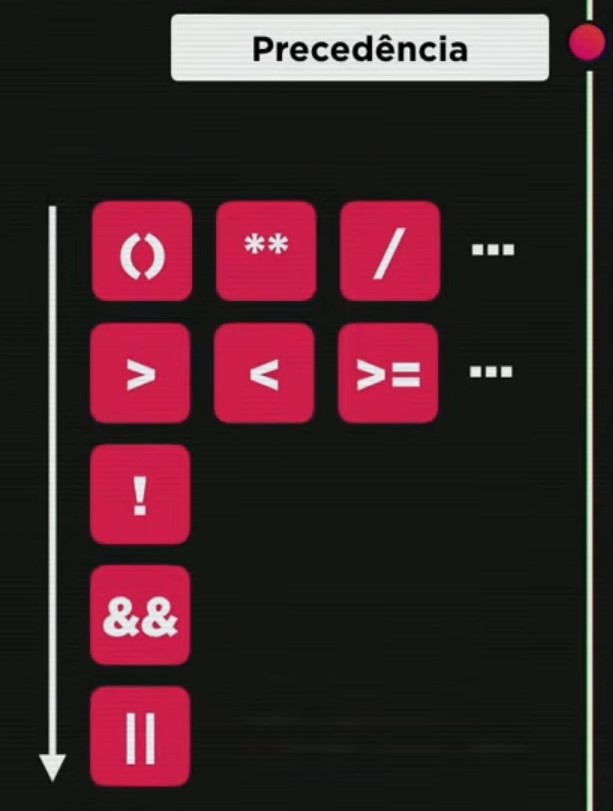
Primitivos: Manipulam tipos de dados primitivos, que são os tipos de dados mais básicos da linguagem.

* Aritméticos: +, -, \*, /, %: Determina o resto da divisão inteira, ++, -- e \*\*;
* Atribuição: =, +=, -=, \*=, /= e %=;
* Comparação: ==, !=, ===, !==, >, <, >= e <=;
* Relacionais: >, <, >=, ==, !=;
* Lógicos: ! negação, && conjução, e || disjunção;
* Ternários: ? e :; Exemplo: média>=7.0 **?** “Aprovado” **:** “Reprovado”;
* Concatenação de Strings: String + String para concatenação;
* Bitwise: &, |, ^, ~, <<, >>, e >>>;

Não primitivos: Operam em objetos em vez de valores primitivos. Não são símbolos especiais, mas funções que são definidas como propriedade dos objetos.

* New: Usado para criar uma nova instancia de um objeto. Ex: `var objeto = new Object();`.
* Delete: Remove uma propriedade de um objeto. Ex: `delete objeto.propriedade;`.
* In: Usado para verificar se uma propriedade existe em um objeto. Ex: ìf (“propriedade” in objeto) {...}`.
* Instanceof: Usado para verificar se um objeto é uma instancia de uma determinada classe. Ex: `if (objeto instanceof Classe) {...}`.
* Typeof: Usado para obter o tipo de um valor. Ex: `typeof objeto`.
* This: Usado para se referir ao objeto atual em que o código está sendo executado. Ex: `this.propriedade`.
* New.target: Usado para verificar se uma função foi chamada com o operador “new”. Ex: `if (new.target === undefined) {...}`.
* Identidade: “===” servem para comparar igualdade estrita entre dois valores, tanto seu tipo como seu valor de grandeza.

Precedência dos operadores:

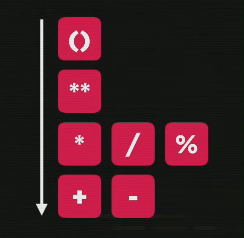


CUIDADO COM PRECEDENCIA DE OPERADORES!

Ex: Na programação no exemplo de 5 + 3 / 2, fazemos primeiro a DIVISÃO e não a ADIÇÃO como é comum. Sendo assim o resultado não é 4, e sim 6,5.

Mas se no caso você quiser fazer a adição primeiro basta colocar parênteses: (5 + 3) / 2.

Ordem de precedência seguida na programação

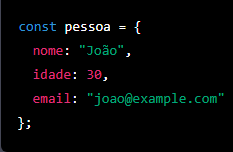


Métodos

São uma das características dos objetos, que por não serem primitivos, os métodos também não são. Isso significa que não é possível adicionar um método a uma String ou a um número em Java Script. Existem vários, procure na internet se quiser saber mais...

Propriedades

Pares de nome/valor que descrevem as características ou informações relacionadas a um objeto. Podem ser definidas na criação do objeto, após a criação ou usando o método Object.defineProperty, existem outras formas de fazer isso, veja na internet. Veja um exemplo da primeira forma citada:



Continuar praticando operadores e iniciar modulo c