Variáveis e Tipos com JavaScript

Atribuindo Valores

|  |  |
| --- | --- |
| Case Type | Exemplos |
| Original Variable as String | Some awesome var |
| Camel Case | someAwesomeVar |
| Snake Case | some\_awesome\_var |
| Kebab Case | some-awesome-var |
| Pascal Case | SomeAwesomeVar |
| Upper Case Snake Case | SOME\_AWESOME\_VAR |

Variáveis

Hoisting

O código JavaScript é executado em duas fases. Primeiro, é feito o parsing, em que são vasculhadas declarações de variáveis, funções e parâmetros. Só depois é feito a execução de fato.

O efeito disso é que é como se as declarações de variáveis fossem levantadas para o topo da função ou arquivo em que estão definidas. O nome chique para esse levantamento é variable hoisting. Coisa parecida é feita com funções...

num = 6; Inicializa

var num; Declara

var num = 6; Inicializa e Declara

var = Scope global

let = scope de bloco

Escopos diferentes

Hoisting

Redeclaração

Reatribuição

Declaradas em camelCase

const = Escopo de bloco

Declarada em SNAKE\_UPPER\_CASE

Não faz hoisting

|  |  |  |  |
| --- | --- | --- | --- |
|  | var | const | let |
| escopo | Global ou local | bloco | bloco |
| redeclarar | sim | não | não |
| reatribuir | sim | não | sim |
| hoisting | sim | não | não |

Estrutura de Dados

JavaScript é uma linguagem de tipagem dinâmica.

Ou seja, antes de declarar um valor, você não especifica o **tipo** dele.

Typeof variável | serve para Mostar qual o tipo da sua variável.

Tipos de Dados

Tipos Primitivos

* Números
* Strings
* Booleanos
* Null
* Undefined

Tipos Não Primitivos

* Objetos
* Arrays

Tipos

Srings

Concatenação

nome.concat(sobrenome)

Ou usar o mais + nome + “ ” + sobrenome

para imprimir aspas é necessário uso de barra “\””

quebra de linha \n

variável.split(“”) separa cada espaço e virgula se colocar um espaço entre as aspas ele separa só as palavras variave.split(“ ”)

variável.includes(“palavra”) procura a palavra especifica entre aspas dentro da sua frase e retorna um true ou false

variável.startswith(“X”) procura na variável se a palavra começa com a letra selecionada e retorna true ou false

variável.endswith(“X”) procura na variável se a palavra termina com a letra selecionada e retorna true ou false

variável.replace(“,”, “!”) substitui um sinal, palavra ou letra

Interpolação: templete string `${nome} ${sobrenome}` colocar variáveis dentro de strings

Número

Inteiros ou Decimais

Biblioteca Math

Math.PI valor do PI

Math.floor() arredonda para baixo

Math.ceil() arredonda para cima

variavel.toString() converte um dado para string

Booleanos

True ou False

Pode ser usado para validar condicionais

Arrays

Lista iteráveis de elementos

Index começa no 0

Let array = []

Array.push() adiciona um elemento no array

Array.pop() tira o elemento no final do array

Array.shift() tira o primeiro elemento do array

Array.unshift() adiciona um elemento no início do array

Array.includes(elemento) verifica se possui esse elemento no array e retorna um valor booleano

Array.every(item => item === 5) verifica se todos os valores no array são 5 e retorna um valor booleano

Array.some(item => item === 5) verifica se um dos valores no array é 5 e retorna um valor booleano

Array.reverse() reverte os elementos do array

Objetos

Let obj = {

Nome: Bruno

Idade: 24

}

Chaves = nome e idade

Valores = Bruno, idade

Object.values(x) mostra os valores do objeto

Object.keys(x) mostra as chaves do objeto

Variável.name retorna o valor da chave name

Variável[“name”] retorna o valor da chave name

Variável[“numberOfSiblings”] = 2 isso adiciona ao objeto essa chave e valor

Empty, undefined e null

Empty não possui valor dentro

Null quando você propositalmente define que o valor não existe

Undefined o valor não existe

Falsy = quando algo é tratado como falso, mas não o falso do booleano

Palíndromo