**Conceitos Revisão JS**

* **Fundamentos**
* **Array**

**Sumário**

O objeto Array do JavaScript é um objeto global usado na construção de 'arrays': objetos de alto nível semelhantes a listas.

Está listado dentro de colchetes [x, y]

**Criando um Array:**

Let frutas = [‘banana’ , ‘maça’, ‘morango’, ‘uva’]

Console.log(frutas.lenght)

// 4

**Acessando os items de um Array:**

Let primeiro\_Item = frutas[1]

Console.log(primeiro\_item)

// banana

let compras = ['maçã', 'banana', 'morango', 'abacaxi', 'batata doce', 'ovos']

console.log(compras.length)

//6

console.log(compras)

[‘maça’,’banana’,’morango’,’abacaxi’,’batata doce’,’ovos’]

console.log(compras[1])

// banana

compras[3] = 'carne' // trocando o abacaxi por carne

console.log(compras)

[‘maça’,’banana’,’morango’,’carne’,’batata doce’,’ovos’]

var ultimo = compras[compras.length - 1]

compras.push('Creatina', 'whey')

// adicionando items

console.log(compras)

[‘maça’,’banana’,’morango’,’carne’,’batata doce’,’ovos’,’Creatina’,’Whey’]

compras.pop()

// n tem dinheiro pra whey kk

console.log(compras)

[‘maça’,’banana’,’morango’,’carne’,’batata doce’,’ovos’,’Creatina’]

console.log(compras.indexOf('morango'))

// saber o índice do item // 2

* **Booleanos**

O valor passado como primeiro parâmetro é convertido para um valor boleano, se necessário. Se o valor é omitido ou é 0, -0, null, false, NaN, undefined ou é uma string vazia(""), o objeto terá um valor inicial de false. Todos outros valores, incluindo qualquer objeto ou string "false", criam um objeto com valor inicial true.

* **Função**

A função é um verbo , uma ação que executa um processo baseado em uma sentença de código , é um trecho de código com entradas , comandos e resultado

Funções são blocos de construção fundamentais em JavaScript. Uma função é um procedimento de JavaScript - um conjunto de instruções que executa uma tarefa ou calcula um valor. Para usar uma função, você deve defini-la em algum lugar no escopo do qual você quiser chamá-la.

A definição da função (também chamada de declaração de função) consiste no uso da palavra chave function (en-US), seguida por:

* Nome da Função.
* Lista de argumentos para a função, entre parênteses e separados por vírgulas.
* Declarações JavaScript que definem a função, entre chaves { }.

Comparando funções com e sem retorno:

A grande diferença entre a função com retorno e a função sem retorno, é que na função com retorno o console.log precisa ser executado fora da função, diretamente na chamada dela. Já na função sem retorno o console.log é executado dentro da função.