**Arrays**

Arrays são dados que guardam uma lista de valores em uma única variável.

* **Array x NodeList:**

Apesar de parecidos NodeList é uma coleção de nós do DOM. Ele não aceita métodos de Array.

// Apesar de aceitar o método: **.*forEach()***

* **Convertendo um NodeList em Array:**

let itensArray = ***Array.from*** ***(***document.querySelectorAll(“li”)***)***

Agora itensArray recebeu a coleção de nós <li> como valores de seu Array.

* **Criando Arrays:**

let numeros = [1,2,3,4] //*Números*  
let nomes = [“Ana”, “João”, “Pedro”] //*Strings*

let misto = [“Texto”, 42, true] //*Misto: Texto, Número, Booleano, etc.*

let arrayObjeto = [

{nome: “Ana”, idade: 23, votou: true}, //*Objetos*

{nome: “João”, idade: 24, votou: false}

]

* **Índice ou Index dos Elementos na Array:**

Os itens na Array possuem uma posição (índice) começando do 0.

* Acessando um elemento do Array pelo índice:

**nomes[0]** // “Ana”  
**nomes[1]** // “Bruno”

* E se for um valor do objeto que quisermos acessar?

nomes[***0***]***.idade*** *// retorna o valor da chave “idade” do primeiro elemento da array que é* ***23.***

nomes[***1***]***.votou*** *// retorna o valor da chave “votou” do segundo elemento da array que é* ***false.***

* Métodos mais utilizados trabalhando com Arrays:
* Adicionando itens:

nomes***.push***(“Roberval”) *//adiciona no* ***final****.*

nomes***.unshift***(“Roberval”) *//adiciona no* ***início****.*

* Removendo Itens:

nomes***.pop***() *//remove o* ***último*** *item.*

nomes***.shift***() *//remove o* ***primeiro*** *item.*

* Modificar Itens:

nomes[0] = “Natália” *//****Substituiu*** *o primeiro item do Array de “Ana” para “Natália”*

* Percorrendo uma Array:

For (let i = 0; i < nomes.length; i++){

console.log(nomes[i])

}

*//Usa a estrutura de repetição* ***for*** *para imprimir o item do índice correspondente e passar para o próximo item (let i = 0 & i++) até que o índice atinja o tamanho da array (i < nomes.length). Recomendável dar uma estudada em* ***estruturas de repetição*** *e seus parâmetros.*

Ou podemos usar o ***.forEach():***

nomes***.forEach(***function(nome) {

console.log(nome)

}***)***

*//Esse método tem uma estrutura mais complexa. Vamos entendê-lo melhor:* ***.forEach()*** *pede uma função de call-back em sua estrutura e passa três argumentos para ela: valor, índice, arrayOriginal. Nessa ordem, os argumentos ocupam os parâmetros da função de call-back, veja:*

***.forEach(****função (valor, índice, arrayOriginal){}****)***

*1º Valor = O* ***elemento******atual*** *do array;*

*2º Indice = O* ***índice numérico*** *desse elemento;*

*3º ArrayOriginal = O* ***array*** *original* ***inteiro****.*

*\*São argumentos passados pelo método que se tornam parâmetros para a função (Entender essas nomenclaturas é importante para um bom programador).*

***.forEach()*** *é um método nativo dos arrays mas pode ser usado para estruturas parecidas como NodeList. Apesar de parecido com uma lista os objetos* ***não*** *podem fazer uso de .forEach().*

*Exemplo de* ***.forEach()*** *sendo usado por uma NodeList:*

*document.querySelectorAll(“li”)****.forEach(****function(item) {*

*console.log (item)*

*}****)***

* Verificar se um valor está no Array:

nomes***.includes(***“Ana”***)*** *// Retorna* ***true ou false****.*

* Retornando o valor da posição de um elemento no Array(índice):

nomes***.indexOf(***“João”***)*** *// Retorna o* ***valor do índice*** *de “João” que é 1.*

* Retornando o tamanho do Array:

nomes***.length*** *// Retorna o tamanho do array, ou seja,* ***quantos elementos ele tem****, no caso 3.*

*// Percebesse pela falta dos “( )” que* ***.length*** *não é um método e sim uma propriedade.*

* Removendo um item do Array:

nomes***.splice(***2,1***)*** *//A partir do item de índice 2, remove 1 item. (Nesse caso removeu “Pedro”).*

Parâmetros: ***.splice(***índice, quantidadeParaRemover***).***

* Gerando uma String a partir do Array:

nomes***.join(***“, “***)*** *//Retorna: “Ana, João, Pedro”*

* Percorre o Array e retorna um novo Array com as transformações:  
  let novoArray = numeros***.map(***n => n\*\*2***)*** *// novoArray = [1,4,9,16]*

*// n => n\*\*2 é uma função de call-back e uma* ***Arrow Function****, ou seja (n=>n\*\*2) é a mesma coisa que: function (n){ n = n\*\*2}*

* Novo Array apenas com os itens que passarem na condição:

let novoArray = numeros***.filter(***n => n%2 === 0***)*** *//novoArray = [2,4]*