CAFÉ, DESENVOLVIMENTO & JAVASCRIPT.

STRING'S & ARRAY'S

CHEATSHEET

www.amigoni.com.br
paulo.amigoni@gmail.com
linkedin.com/in/pauloamigoni/
github.com/pauloamigoni/

CONHECENDO

STRINGS

STRING MÉTODOS

JS
JavaScript

charAt()

O método charAt() retorna o caractere no índice especificado em uma string. O índice do primeiro caractere é 0, o segundo caractere é 1 e assim por diante

```
var str="HELLO WORLD";
var res=str.charAt(0);

console.log(res);
// output H
```

concat()

O método concat() é usado para juntar dois ou mais array. Este método não altera o existente strings, mas retorna uma nova string contendo o texto das cordas unidas.

```
var strl="Hello";
var str2="world!";
var res=strl.concat(str2);

console.log(res)
// output Hello world!
```

charCodeAt()

O método charCodeAt() retorna o Unicode do caractere no índice especificado em uma string.

```
var str="HELLO WORLD";
var res=str.charCodeAt(0);

console.log(res);
// output 72
```

endsWith()

O método endsWith() determina se um string termina com os caracteres de uma string especificada. Este método retorna true se a string terminar com o caracteres, ou false se não.

```
var str="Hello world,welcome to the universe.";
var res=str.endsWith("universe.");

console.log(res)
// output true
```

fromCharCode()

O método fromCharCode() converte Unicode valores em caracteres. Este é um método estático de o objeto String e a sintaxe é sempre String.

```
var res = String.fromCharCode(65);

console.log(res)
// outputA
```

includes()

O método includes() determina se uma string contém os caracteres de uma string especificada.

```
var str = "Hello world, welcome to the universe.";
var res = str.includes("world");

console.log(res)
// output true
```

CONHECENDO

STRINGS

STRING MÉTODOS

JS

JavaScript

fromCharCode()

O método fromCharCode() converte Unicode valores em caracteres. Este é um método estático de o objeto String e a sintaxe é sempre String.

```
var res = String.fromCharCode(65);

console.log(res)
// outputA
```

indexOf()

O método indexOf() retorna a posição do primeira ocorrência de um valor especificado em uma string. este método retorna -1 se o valor a ser pesquisado nunca ocorre

```
var str = "Hello world,welcome to the universe.";
var res = str.indexOf("welcome");

console.log(res)
// output 13
```

charCodeAt()

O método charCodeAt() retorna o Unicode do caractere no índice especificado em uma string.

```
var str="HELLO WORLD";
var res=str.charCodeAt(0);

console.log(res);
// output 72
```

localeCompare()

O método localeCompare() compara duas strings na localidade atual. A localidade é baseada na configurações de idioma do navegador.
Retorna -1 se str1 for classificado antes de str2
Retorna 0 se as duas strings forem iguais
Retorna 1 se str1 for classificado após str2

```
var str1 = "ab";
var str2 = "cd";

var res = str1.localeCompare(str2);

console.log(res)
// output -1
```

lastIndexOf()

O método lastIndexOf() retorna a posição do última ocorrência de um valor especificado em uma string. a string é pesquisada do fim ao início, mas retorna o índice começando no início, em posição 0.

```
var str = "Hello planet earth, you are a great planet.";
var res = str.lastIndexOf("planet");

console.log(res)
// output 36
```

CONHECENDO

STRINGS

STRING MÉTODOS

JS
JavaScript

match()

O método match() procura uma string por uma correspondência contra uma expressão regular e retorna o corresponde, como um objeto Array.

```
var str = "The rain in SPAIN stays mainly in the plain";
var res = str.match(/ain/g);

console. log(res)
// output [“ain”, "ain" , "ain"]
```

repeat()

O método repeat() retorna uma nova string com um número especificado de cópias da string que foi chamado.

```
var str = "Hello world!";
str.repeat(2);

console.log(str)
// output Hello world!Hello world!
```

search()

O método search() procura uma string por um valor especificado e retorna a posição do pesquisada.

```
var str = "Visit W3Schools!";
var res = str.search("W3schools");

console. log( res)
// output 6
```

split()

O método split() é usado para dividir uma string em um array de substrings e retorna o novo array.

```
var str = "How are you doing today?";
str.split(" ");

console. log(res)
// output ["How", “are”, "you*, "doing", "today"]
```

replace()

O método replace() procura uma string por um valor especificado, ou uma expressão regular, e retorna um nova string onde os valores especificados são substituídos.

```
var str = "Visit Microsoft!";
var res = str.replace("Microsoft","W3Schools");
console.log(res)

// output Visit W3Schools!
```

slice()

O método slice() extrai partes de uma string e retorna as partes extraídas em uma nova string.

```
var str = "Hello world!";
var res = str.slice(0, 5);

console. log(res)
// output Hello
```

CONHECENDO

STRINGS

STRING MÉTODOS

JS

JavaScript

startsWith()

O método startsWith() determina se uma string começa com os caracteres de uma string especificada.

```
var str = "Hello world, welcome to the universe.";
var res = str.startswith("Hello");

console.log(res)
// output true
```

substr()

O método substr() extrai partes de uma string, começando no caractere na posição especificada, e retorna o número especificado de caracteres.

```
var str = "Hello world!";
var res = str.substr(1, 4);

console.log(res)
// output ello
```

toLocaleLowerCase()

O método toLocaleLowerCase() converte uma string para letras minúsculas.

```
var str = "Hello World!";
var res = str.tolocaleLowerCase();

console.log(res)
// output hello word
```

toUpperCase()

O método toUpperCase() converte uma string para letras maiúsculas.

```
var str = "Hello World!";
var res = str.toUpperCase();

console.log(res)
// output HELLO WORLD!
```

trim()

O método trim() remove espaços em branco de ambos lados de uma string.

```
let str = " Hello World  ";
console.log(str);
// Output: " Hello world "
let res = str.trim();
console.log(res);
//Output: Hello world
```

CONHECENDO

ARRAY

ARRAY PROPRIEDADES

JS

JavaScript

constructor

Em JavaScript, a propriedade do construtor retorna o função construtora para um objeto. Para JavaScript arrays a propriedade do construtor retorna a função Array() { [código nativo] }

```
var fruits = ['Banana', 'Orange', 'Apple'];
fruits.constructor;
// Return
// function Array() { [native code] }
```

prototype

O prototype constructor permite adicionar novas propriedades e métodos para o objeto Array()

```
Array.prototype.myUcase = function() {
    for (1 = 0; i < this.length; i++) {
        this[i] = this[i].toUpperCase();
    }
};

var fruits = ["Banana", "Orange", "Apple"];
fruits.myUcase();
```

toUpperCase()

O método toUpperCase() converte uma string para letras maiúsculas.

```
var str = "Hello World!";
var res = str.toUpperCase();

console.log(res)
// output HELLO WORLD!
```

length

A propriedade length define ou retorna o número de elementos em uma matriz.

```
var fruits = ["Banana", "Orange", "Apple"];
fruits.length;
// Return 3
var vegetables = [];
vegetables.length = 3;
// sets array length to 3
```

CONHECENDO

ARRAY

ARRAY MÉTODOS

JS
JavaScript

concat()

O método concat() é usado para juntar dois ou mais matrizes. Este método não altera o existente arrays, mas retorna um novo array, contendo o valores das matrizes unidas.

```
var hege = ["Cecilie", "Lone"];
var stale = ["Emil", "Tobias", "Linus"];
var children = hege.concat(stale);
// Returns a new array
```

entries()

O método entries() retorna um Array Iterator objeto com pares chave/valor.

```
var fruits = ["Banana", "Orange", "Apple"];
var f = fruits.entries();
/* returns new object
{
    [0, "Banana"],
    [1, "Orange"],
    [2, "Apple"]
}
*/
```

every()

O método every() verifica se todos os elementos em um array passar por um teste (fornecido como uma função).

```
var ages = [32, 33, 16, 40];
    function checkAdult(age) {
        return age >= 18;
    }
ages.every(checkAdult);
/* checks every element returns ->
false */
```

copyWithin()

copia os elementos da matriz para outra posição no array, sobrescrevendo os valores existentes. Este método nunca adicionará mais itens ao array. Nota: este método substitui o array original

```
var fruits = ["Banana", "Orange", "Apple"];
fruits.copyWithin(2, 0);
// overwrites the original array
```

CONHECENDO

ARRAY

ARRAY MÉTODOS

JS

JavaScript

fill()

O método fill() preenche os elementos especificados em uma matriz com um valor estático. Você pode especificar o posição de onde começar e terminar o enchimento. Se não especificado, todos os elementos serão preenchidos

```
var fruits = ["Banana", "Orange", "Apple"];
fruits. Fill("Kiwi");
/* returns new array ->
[Kiwi, Kiwi ,Kiwi,Kiwi] */
```

find()

O método find() retorna o valor do primeiro elemento em uma matriz que passa em um teste (fornecido como uma função).

```
var ages = [3, 10, 18, 20];
function checkAdult(age) {
    return age >= 18;
}
ages.find(checkAdult)
/* returns fist accurance of value -> 18 */
```

includes()

O método includes() determina se um array contém um elemento especificado.

```
var fruits = ["Banana", "Orange", "Apple", "Mango"];
var n = fruits.includes("Banana", 3);
console.log(n);
// Output false
```

filter()

O método filter() cria um array preenchido com todos elementos de matriz que passam em um teste (fornecidos como uma função).

```
var ages = [32, 33, 16, 40];
function checkAdult(age) {
    return age >= 18;
}
ages.filter(checkAdult)
/* returns new array ->
[32,33,40] */
```

findIndex()

O método findIndex() retorna o índice do primeiro elemento em uma matriz que passa em um teste (fornecido como uma função).

```
var ages = [3, 10, 18, 20];
function checkAdult(age) {
    return age >= 18;
}
ages. findIndex(checkAdult)
/* returns fist index of value -> 2 */
```

forEach()

O método forEach() chama uma função uma vez para cada elemento em uma matriz, em ordem

```
let numbers = [65, 44, 12, 4];
numbers.forEach(myFunction)
function myFunction(item, index, arr) {
    arr[index] = item * 10;
}
console. log( numbers);
// output => [650,440,120,40]
```

CONHECENDO

ARRAY

ARRAY MÉTODOS

JS
JavaScript

indexOf()

O método indexOf() pesquisa o array para o item especificado e retorna sua posição.

```
var fruits = ["Banana", "Orange", "Apple",
"Mango"];
var a = fruits.indexOf("Apple");

console.log(a);

// Output 2
```

join()

Converta os elementos de um array em uma string. O método join() retorna o array como uma string

```
var fruits = ["Banana", "Orange", "Apple",
"Mango"];

var energy = fruits.join(" and ");

console.log(energy)

// Output Banana and Orange and Apple and Mango
```

map()

O método map() cria um novo array com o resultados de chamar uma função para cada elemento da matriz.

```
var numbers = [4, 9, 16, 25];
var x = numbers.map(Math.sqrt)

console.log(x)

// Output [2,3,4,5]
```

isArray()

O método isArray() determina se um objeto é um array. Função retorna true se o objeto for um array e false se não.

```
var fruits = ["Banana", "Orange", "Apple",
"Mango"];

var result = Array.isArray(fruits);

console.log(result)
// Output true
```

lastIndexOf()

O método lastIndexOf() pesquisa o array para o item especificado e retorna sua posição.

```
var fruits = ["Banana", "Orange", "Apple",
"Mango"];
var a = fruits.lastIndexOf("Apple");

console.log(a)

// Output 2
```

pop()

O método pop() remove o último elemento de um array e retorna esse elemento.

```
var fruits = ["Banana", "Orange", "Apple",
"Mango"];
fruits.pop();

console.log(fruits)

// Output ["Banana", "Orange", "Apple"]
```

CONHECENDO

ARRAY

ARRAY MÉTODOS

JS

JavaScript

push()

O método push() adiciona novos itens ao final de um array e retorna o novo comprimento.

```
var fruits = ["Banana", "Orange", "Apple",
"Mango"];
var length = fruits.push("Kiwi");

console.log(fruits)
console.log(length)

// Output
// ["Banana", "Orange", "Apple", "Mango",
"Kiwi"]
// 5
```

reduceRight()

O método reduceRight() executa um
função para cada valor da matriz (da direita para a esquerda) e reduz a matriz a um único valor.

```
var numbers = [175, 50, 25];
function myFunc(total, num) {
  return total - num;
}

var result = numbers.reduceRight(myFunc);
console.log(result)

// Output -200
```

reduce()

O método reduce() executa uma função fornecida para cada valor da matriz (da esquerda para a direita) e reduz a matriz a um único valor.

```
var numbers = [15.5, 2.3, 1.1, 4.7];

function getSum(total, num) {
  return total + Math.round(num);
}

var result = numbers.reduce(getSum, 0);
console.log(result)

// Output 24
```

reverse()

O método reverse() inverte a ordem do elementos em uma matriz.

```
var fruits = ["Banana", "Orange", "Apple",
"Mango"];
fruits.reverse();

console.log(fruits)

// Output
["Mango","Apple","Orange","Banana"]
```

CONHECENDO

ARRAY

ARRAY MÉTODOS

JS
JavaScript

some()

O método some() verifica se algum dos elementos em uma matriz passam por um teste (fornecido como uma função). Executa a função uma vez para cada elemento presente na matriz.

```
var ages = [3, 10, 18, 20];

function checkAdult(age) {
  return age >= 18;
}

console.log(ages.some(checkAdult))
// Output true;
```

shift()

O método shift() remove o primeiro item de um variedade.

```
var fruits = ["Banana", "Orange", "Apple",
"Mango"];
fruits.shift();

console.log(fruits)

// Output ["Orange", "Apple", "Mango"];
```

splice()

O método splice() adiciona/remove itens de/para uma matriz e retorna o(s) item(ns) removido(s).

```
var fruits = ["Banana", "Orange", "Apple",
"Mango"];
fruits.splice(2, 0, "Lemon", "Kiwi");

console.log(fruits)
// Output
["Banana","Orange","Lemon","Kiwi","Apple","M
ango"];
```

slice()

O método slice() seleciona os elementos começando no argumento inicial dado, e termina em, mas não não inclui, o argumento final dado. Ele retorna o elementos selecionados em uma matriz, como um novo objeto de matriz.

```
var fruits = ["Banana", "Orange", "Lemon",
"Apple", "Mango"];
var citrus = fruits.slice(1, 3);

console.log(citrus)

// Output ["Orange", "Lemon"];
```

sort()

O método sort() classifica os itens de um array.

```
var fruits = ["Banana", "Orange", "Apple",
"Mango"];
fruits.sort();

console.log(fruits)
// Output
["Apple","Banana","Mango","Orange"];
```

toString()

O método toString() retorna uma string com todos os valores de matriz, separados por vírgulas.

```
var fruits = ["Banana", "Orange", "Apple",
"Mango"];
var x = fruits.toString();

console.log(fruits)
// Output Banana,Orange,Apple,Mango
```

CONHECENDO

ARRAY

ARRAY MÉTODOS

unshift()

O método unshift() adiciona novos itens ao início de uma matriz e retorna o novo comprimento.

```
var fruits = ["Banana", "Orange", "Apple",
"Mango"];
fruits.unshift("Lemon","Pineapple");

console.log(fruits)
// output
// ["Lemon", "Pineapple", "Banana", "Orange",
"Apple", "Mango"]
```

valueOf()

O método valueOf() retorna a matriz. Este method é o método padrão do objeto array. Array.valueOf() retornará o mesmo Array

```
var fruits = ["Banana", "Orange", "Apple",
"Mango"];
var v = fruits.valueOf();

console.log(v)
// output
// ["Banana", "Orange", "Apple", "Mango"]
```