

IOS – Instituto de  
Oportunidade Social

## Aula JS 3 - String



- > Mais sobre strings
- > Concatenando strings
- > Método filter()
- > Método find()
- > Método find()
- > Método find()

IOS – Instituto de  
Oportunidade Social

Mais sobre strings



String é um tipo especial em toda linguagem de programação e no JavaScript não seria diferente. O tipo de dados String tem propriedades e métodos, que permitem manipulá-las de diversas formas. Vamos aprender alguns desses métodos e dessas propriedades.

Strings é um conjunto de caracteres muito utilizado em diferentes linguagens de programação de diversas formas possíveis. Uma string pode conter letras, números e caracteres especiais e, dessa forma, possibilita enviar bloco de dados e não somente palavras ou frases

IOS – Instituto de  
Oportunidade Social

Concatenando strings



# Concatenando strings



## Concatenando strings

A concatenação de strings pode ser realizada com o operador + de forma semelhante a muito linguagem de programação, tais como: Java, C++, C#, etc.

### Concatenando strings

```
"Instituto" + ' ' + 'da' + " " + "Oportunidade" + ' ' + 'Social'
```

### Resultado

```
Instituto da Oportunidade Social
```

> Vamos Praticar

## Concatenando strings

Desse modo, a maneira de concatenar mostrada na segunda instrução de soma, conhecida como template strings, pois elas permitem embutir expressões de JavaScript dentro da string através do símbolo de \$ e a abertura e fechamento das chaves { }

```
${expressão_do_JavaScript}
```

> Vamos Praticar

IOS – Instituto de  
Oportunidade Social

## Métodos de strings





Em JavaScript praticamente qualquer tipo de dado é um objeto. Cada item dessa "coleção de valores", é chamado de propriedade. Cada propriedade é composta por um par de "nome: valor". Quando uma propriedade armazena uma função, ela se torna o que chamamos de método. Vamos aprender sobre os métodos de strings continuando a nossa programação no arquivo main.js.

> Vamos Praticar

IOS – Instituto de  
Oportunidade Social

Acesso a caracteres



Vamos começar pelo acesso à caracteres em uma string. Podemos fazer o acesso de duas formas, a primeiro é pelo método `charAt(n)`, onde `n` é o valor da posição do caractere na string, e a segunda é através de colchetes `[n]`, onde `n` é o valor da posição do caractere na string. Vamos ver isso na prática, siga os passos para continuar a implementação:

```
// Métodos de strings
console.clear();
const string01 = 'Instituto da Oportunidade Social';

// Acessando o primeiro caractere
console.log(string01.charAt(0));
console.log(string01[0]);
// Acessando o décimo primeiro caractere
console.log(string01.charAt(10));
console.log(string01[10]);
```

Observe também que para acessar um método em JavaScript colocamos o nome\_do\_objeto depois ponto final e depois o nome do método:

<b>string01</b>	.	<b>charAt(10)</b>
nome_do_objeto	ponto final	nome do método

IOS – Instituto de  
Oportunidade Social

Tamanho da string



A propriedade `length` retorna o tamanho da string, ou seja, o número de caracteres que a string possui. Esse método será muito útil em laços de repetição, quando iremos implementar um código para percorrer os caracteres de uma string. Vamos ver isso na prática, siga os passos para continuar a implementação:

```
// Tamanho de strings  
console.clear();  
let texto = 'abcdefghijklmnopqrstuvxyz';  
  
console.log(texto.length);
```

IOS – Instituto de  
Oportunidade Social

## Maiúsculas e minúsculas



Você pode transformar todos os caracteres de uma string para maiúsculo ou para minúsculo, isso facilita, por exemplo, comparar strings para verificar se são iguais. O método `toUpperCase` transforma todos os caracteres de uma string para maiúsculos e o método `toLowerCase` transforma para minúsculo. Vamos ver isso na prática, siga os passos para continuar a implementação:

```
// Maiusculo e minúsculo  
console.clear();  
const s = 'Hello World!';  
  
console.log(s.toUpperCase());  
console.log(s.toLowerCase());
```



IOS – Instituto de  
Oportunidade Social

Exercício



e um Array unidimensional com 8 posições contendo os