

IOS – Instituto de
Oportunidade Social

Aula JS 05 - Strings



- > Mais sobre strings
- > Concatenando strings
- > Métodos de strings
 - > Acesso a caracteres
 - > Tamanho da string
 - > Maiúsculas e minúsculas
 - > Substring e Slipt
 - > Substituir uma string dentro da string
 - > Remover espaços da string
 - > Métodos de buscar em strings

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.

```
const string1 = "Uma string";
```

```
const string2 = 'Também 123 [] é uma strings 231 -1';
```

Ela também pode ser colocada **entre crases**:

```
const string3 = `Essa também é uma string entre crases`;
```

IOS – Instituto de
Oportunidade Social

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

IOS – Instituto de
Oportunidade Social

Métodos de strings



> 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:

string01	.	charAt(10)
nome_do_objeto	ponto final	nome do método

> Vamos Praticar

> 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:

> Vamos Praticar

> Maiúsculas e minúsculas

Você pode transformar todos os caracteres de uma string para maiúsculo ou para minúsculo, isso facilitam, 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:

> Vamos Praticar

> Substring e Split

O método substring corta a string de acordo com os índices indicados entre os parênteses, por exemplo:

O comando `str.substring(1, 3)` corta a string a partir do índice 1 até o índice 3 e o comando `str.substring(2)` corta a string a partir do índice 2 até o seu final. O método `split` divide a string em substrings de acordo com a regra colocada entre parênteses. O objeto retornado por esse método é um array. Por exemplo:

> Vamos Praticar

> Substituir uma string dentro da string

O método `replace` substitui um pedaço específico da string por outra coisa que você desejar. Vamos ver isso na prática, siga os passos para continuar a implementação:

> Vamos Praticar

> Remover espaços da string

O método `trim` remove os espaços em branco do início e do final de uma string. Isso é útil por exemplo, quando estamos lendo um texto digitado pelo usuário em um formulário e esse texto vem com espaços no início e/ou no final. Vamos ver isso na prática, siga os passos para continuar a implementação:

> Vamos Praticar

> Métodos de buscar em strings

Existem alguns métodos usados para buscar algo dentro da string, eles são úteis para programas que precisem encontrar algum valor ou padrão na sequência de caracteres enviada. Os métodos são:

`String.indexOf()`: esse método retorna o índice (posição) da primeira vez que um texto especificado entre os parênteses foi encontrado na string.

`String.lastIndexOf()`: esse método retorna o índice da última ocorrência de um texto especificado entre os parênteses foi encontrado na string.

> Métodos de buscar em strings

`String.search()`: esse método busca um valor específico e retorna a posição inicial desse valor na primeira vez que ele é encontrado. Semelhante ao `indexOf()`.

`String.startsWith()`: esse método verifica se a string inicia com um valor específico.

`String.endsWith()`: esse método verifica se a string termina com um valor específico.

> Vamos Praticar

IOS – Instituto de
Oportunidade Social

Exercício



A partir desse ponto você irá seguir com o passo a passo na apostila esse será seu exercício de casa.

Suba o código no GitHub e disponibilize o link da atividade no Moodle.