

Métodos úteis de strings

Agora que nós vimos o básico de strings, vamos engatar a próxima marcha e começar a pensar sobre as operações úteis que podemos fazer em strings com métodos embutidos, como encontrar o comprimento de uma string, unir e dividir sequências de caracteres, substituindo um caracter em uma string por outro, e muito mais.

Quando criamos uma string, como por exemplo:

```
var string = 'Essa é a minha string'
```

a variável torna-se uma instância do objeto string e, como resultado, tem um grande número de propriedades e métodos disponíveis para ela.

Você não precisa saber sobre a maioria deles no início da sua jornada de aprendizado. Mas há alguns que você potencialmente usará com bastante frequência que veremos aqui.

Descobrimo o comprimento de uma string

Para descobrir o comprimento de uma string, ou o tamanho dela, usamos a propriedade `length`. Exemplo:

```
var string = 'Reprograme-se';  
console.log(string.length);
```

O resultado desse código vai ser a exibição do número 13 na tela, porque a string possui 13 caracteres.

Isso é útil por vários motivos; por exemplo, você pode querer encontrar os comprimentos de uma série de nomes para que você possa exibi-los em ordem de comprimento ou deixar um usuário saber que um nome de usuário que ele informou em um campo de formulário é muito longo se este for maior do que um certo comprimento.

Retornando um caractere específico de uma string

É possível retornar qualquer caractere presente em uma string usando colchetes `[]`, e informando dentro do colchetes o número referente à posição em que o caractere se encontra na string. Por exemplo, para retornar o primeiro caractere da string utilizamos:

```
console.log(string[0]);
```

O resultado do código acima será a exibição na tela da letra R, que é a primeira letra do conteúdo da string "Reprograme-se". Usamos o número 0 porque computadores começam a contar a partir do 0, e não do 1.

Vamos agora a um exemplo de como recuperar o último caractere de uma string. Para fazer isso, podemos combinar com a propriedade `length` que vimos anteriormente.

Sabendo o tamanho da string, nós podemos simplesmente subtrair 1 do tamanho que teremos a posição do último caractere da string. Exemplo:

```
console.log(string[string.length - 1]);
```

O resultado do código acima será a exibição na tela da letra "e", que é o último caractere da string "Reprograme-se".

Alterando entre maiúsculas e minúsculas

O método `string.toLowerCase()` e `string.toUpperCase()` converte todos os caracteres para minúsculo ou maiúsculo, respectivamente. Isso pode ser útil, por exemplo, se você quer normalizar todas as entradas de dados do usuário antes de armazenar em um banco de dados. Exemplo:

```
var string = 'Reprograme-se';  
console.log(string.toLowerCase());  
console.log(string.toUpperCase());
```

Execute o código acima e veja os retornos.

Substituir partes de uma string

É possível alterar partes de uma string com o método `replace()`. Nesse método, temos que passar por parâmetro a parte da string que vai ser substituída e o novo valor. Por exemplo:

```
var string = 'Reprograme-ME';  
var string2 = string.replace('ME', 'se');  
  
console.log(string);  
console.log(string2);
```

O resultado do código acima será o retorno da tela da palavra "Reprograme-ME" e logo depois a string com o final substituído, "Reprograme-se".

Saber se uma string termina ou começa com um conjunto de caracteres específicos

Para saber se uma string termina ou começa com um caractere específico ou um conjunto de caracteres, existem os métodos `startsWith()` e `endsWith()`. Vamos ao exemplo no código:

```
if(string.endsWith('e')){
    console.log('Frase termina com a letra e');
}

if(string.startsWith('e')){
    console.log('Frase começa com a letra E');
}

//ATENÇÃO AQUI, POIS O JAVASCRIPT FAZ DIFERENÇA ENTRE LETRAS MAIÚSCULAS E MI
```

Separar uma string por meio de um caractere separador

É possível separar uma string por meio de um caractere ou conjunto de caracteres separadores. Por exemplo, digamos que uma string tenha o seguinte valor:

```
var string = 'Objetivo 1 - Aprender JavaScript';
```

Nós podemos separar essa string usando como caractere separador o `"-"`. O método que faz essa função é o `split()`. Usando esse método e passando o caractere `"-"` como parâmetro, o retorno será um array com duas strings. Na primeira posição do array ficará uma string contendo `"Objetivo 1"` e na segunda posição ficará a string contendo `"Aprender JavaScript"`.

Exemplo:

```
var string = 'Objetivo 1 - Aprender JavaScript';

var strSeparada = string.split('-');

console.log(strSeparada);
```

Remover espaços em branco

É possível remover espaços em branco do início e do final de uma string. Para isso, usamos o método `trim()`. Existem também as opções de remover espaços em branco somente do início ou somente do final de uma string. Os métodos para isso são, respectivamente, `trimStart()` e `trimEnd()`.

Verificar se um caractere está incluído em uma string

É possível também verificar se um caractere ou um conjunto de caracteres está presente em uma string. Para fazer isso, usamos o método `includes()`.

Exemplo:

```
var string = 'Esta é uma string';

if(string.includes('uma')){
    console.log('Frase possui a palavra "uma"');
}
```