

Objetivo:

- I. Tipo de dado string;
- II. Acessar os caracteres da string;
- III. Percorrer os caracteres da string.

Observação: Para fazer os exercícios e exemplos recomenda-se o uso da interface de programação online <https://replit.com/>.

I. Tipo de dado string

Uma string é um conjunto de caracteres envolvidos por aspas simples ou dupla. A seguir temos alguns exemplos de string:

- "" string com zero caracteres;
- "a" string com um caractere;
- "12" string com dois caracteres;
- "\$-12.5" string com vários caracteres. Uma string pode conter qualquer caractere, inclusive dígitos numéricos, pontuação etc.

Os pares de aspas são obrigatórios na string. A seguir tem-se alguns exemplos de erros:

- "oi" errado pelo fato de ter iniciado com aspas duplas e fechado com aspas simples;
- "oi" errado pelo fato de ter iniciado com aspas simples e fechado com aspas duplas;
- oi errado pelo fato de ter colocado apenas as aspas de fechamento;
- "oi errado pelo fato de ter colocado apenas as aspas de abertura.

Uma variável recebe apenas um conteúdo por vez, mas alguns conteúdos são compostos, ou seja, eles são tipos de dados objetos. O tipo de dado string é um tipo de dado objeto no JavaScript, por este motivo ele possui métodos e propriedades. Os métodos e propriedades são acessados colocando um ponto (.) seguido pelo nome do método ou propriedade, no exemplo a seguir é mostrado como usar métodos e propriedades da string que está na variável `nome`:

```
nome = "ANA maria da silva"
// o método toUpperCase retorna todos os caracteres capitalizados
maiuscula = nome.toUpperCase()
// o método toLowerCase retorna todos os caracteres em minúsculo
minuscula = nome.toLowerCase()
console.log("Nome:", nome)
console.log("Maiúscula:", maiuscula)
console.log("Minúscula:", minuscula)
// a propriedade length possui a quantidade de caracteres da string
console.log("Quantidade:", nome.length)
```

```
Nome: ANA maria da silva
Maiúscula: ANA MARIA DA SILVA
Minúscula: ana maria da silva
Quantidade: 18
```

A propriedade `length` possui a quantidade de caracteres da string.

Um método é como uma função, mas ele está associado a string. Os métodos não alteram a própria string, no exemplo anterior o conteúdo da variável `nome` não foi alterado, ou seja, o método `toUpperCase` criou outra string e retornou ela para ser atribuída na variável `maiuscula`.

II. Acessar os caracteres da string

A string é formada por um conjunto de caracteres. Cada caractere possui uma posição na string. Como exemplo, observe a posição dos caracteres na string “Ana Souza”:

Ana Souza
↑ ↑ ↑ ↑ ↑ ↑ ↑ ↑
0 1 2 3 4 5 6 7 8

O 1º caractere está na posição zero.

Para acessar os caracteres de uma string precisamos colocar o índice de posição entre colchetes:

```
nome = "Ana Souza"
console.log("1o", nome[0])
console.log("2o", nome[1])
console.log("3o", nome[2])
console.log("4o", nome[3])
console.log("5o", nome[4])
console.log("6o", nome[5])
console.log("7o", nome[6])
console.log("8o", nome[7])
console.log("9o", nome[8])
```

1o	A
2o	n
3o	a
4o	S
5o	S
6o	o
7o	u
8o	z
9o	a

Observe que, apesar da string ter 9 caracteres, o último caractere está na posição 8 pelo fato do 1º caractere estar na posição 0.

III. Percorrer os caracteres da string

Cada caractere da string pode ser acessado individualmente através de um índice de posição, então podemos utilizar uma estrutura de repetição `while` para iterar sobre os caracteres da string. No exemplo ao lado a variável `pos` assume os índices de 0 a 8. Desta forma teremos `nome[0]`, `nome[1]`, ..., `nome[8]`.

```
nome = "Ana Souza"
pos = 0
while( pos < nome.length ){
    console.log(nome[pos])
    pos = pos + 1
}
```

A
n
a

S
o
u
z
a

Exercícios

Exercício 1: Fazer um programa que pede para o usuário entrar com um nome e, na sequência, o programa imprime na tela a quantidade de caracteres do nome.

Dica: use a propriedade `length` da string.

Exemplo de saída:

```
Entre com o nome:> Ana Maria
Quantidade: 9
```

Exercício 2: Fazer um programa que pede para o usuário entrar com um nome e, na sequência, o programa imprime na tela o 1º caractere desse nome.

Dica: use os colchetes com a posição do caractere. Lembre-se que o 1º caractere está na posição zero.

Exemplo de saída:

```
Entre com o nome:> Pedro Augusto
Primeiro: P
```

Exercício 3: Fazer um programa que pede para o usuário entrar com um nome e, na sequência, o programa imprime na tela o último caractere desse nome.

Dica: use a propriedade `length` para obter a quantidade de caracteres.

Exemplo de saída:

```
Entre com o nome:> Wesley
Último: y
```

Exercício 4: Fazer um programa que pede para o usuário entrar com um nome e, na sequência, o programa imprime na tela os caracteres na ordem invertida.

Dica: use a estrutura de repetição `while`.

Exemplo de saída:

```
Entre com o nome:> Bruna
a
n
u
r
B
```

Exercício 5: Fazer um programa que pede para o usuário entrar com um nome e, na sequência, o programa imprime na tela os caracteres que estão em posições pares.

Dica: use a estrutura de repetição `while`.

Exemplo de saída:

```
Entre com o nome:> Lucas Moura
L
c
s
M
u
a
```