

## Introdução

Uma *string* é o tipo de dado em JavaScript usado para armazenar texto.

---

## Criando uma String

Você cria uma string colocando o texto entre aspas simples ou aspas duplas. No Exercism, são usadas aspas simples.

```
'Hello, World!'  
"Hello, World!"
```

---

## Strings como listas de caracteres

Uma string pode ser tratada como uma lista de caracteres, onde o primeiro caractere tem índice 0.

Você pode acessar um caractere individual usando colchetes e o índice da letra desejada.

```
'cat'[1];  
// => 'a'
```

Você pode determinar o número de caracteres em uma string acessando a propriedade **length**.

```
'cat'.length;  
// => 3
```

---

## Concatenação e Métodos

A forma mais simples de concatenar strings é usando o operador +.

```
'I like' + ' ' + 'cats.';  
// => "I like cats."
```

As strings fornecem muitos métodos auxiliares. Veja a documentação da MDN sobre *String Methods* para a lista completa. Alguns métodos comuns são:

- **toUpperCase** e **toLowerCase** – mudam o texto para maiúsculas ou minúsculas
- **trim** – remove espaços em branco no início e no fim
- **includes**, **startsWith** e **endsWith** – verificam se uma string contém outra
- **slice** – extrai uma parte da string

As strings são **imutáveis** em JavaScript. Ou seja, qualquer “modificação” cria uma nova string.

---

## Instruções

Um novo clube de poesia abriu na cidade, e você está pensando em participar. Porém, devido a incidentes anteriores, o clube tem uma política rigorosa na porta que você precisa dominar antes de tentar entrar.

Há duas portas no clube de poesia: a porta da frente e a porta dos fundos, ambas com guardas. Para entrar, você precisa descobrir a **senha do dia**.

A senha é sempre baseada em um poema e pode ser derivada em dois passos:

1. O guarda recita o poema, uma linha por vez, e você deve responder com a letra apropriada daquela linha.
2. O guarda então diz todas as letras que você respondeu de uma vez, e você precisa escrevê-las em um formato específico.

Os detalhes do processo dependem de qual porta você está tentando entrar.

---

### 1. Obter a primeira letra de uma frase

Para descobrir as letras da senha da **porta da frente**, você deve responder com a **primeira letra** da linha do poema que o guarda recitar.

Os membros do clube gostam de usar poemas acrósticos, onde a primeira letra de cada verso forma uma palavra. Veja este exemplo de Michael Lockwood:

Stands so high  
Huge hooves too  
Impatiently waits for  
Reins and harness  
Eager to leave

Quando o guarda recitar *Stands so high*, você responde **S**.

Quando recitar *Huge hooves too*, você responde **H**.

Implemente a função `frontDoorResponse` que recebe uma linha do poema e retorna a primeira letra dessa linha.

```
frontDoorResponse('Stands so high');  
// => "S"
```

---

### 2. Capitalizar uma palavra

Agora que você tem todas as letras corretas, basta capitalizar a palavra corretamente para obter a senha da porta da frente.

Implemente a função `frontDoorPassword` que recebe uma string (as letras combinadas da etapa 1) e retorna a palavra capitalizada corretamente.

```
frontDoorPassword('SHIRE');  
// => "Shire"
```

```
frontDoorPassword('shire');  
// => "Shire"
```

---

### 3. Obter a última letra de uma frase

Para a senha da **porta dos fundos**, você deve responder com a **última letra** da linha recitada pelo guarda.

O poema também é um *teléstico*, ou seja, a última letra de cada verso também forma uma palavra:

```
Stands so high  
Huge hooves too  
Impatiently waits for  
Reins and harness  
Eager to leave
```

Quando o guarda recitar *Stands so high*, você responde **h**.

Quando recitar *Huge hooves too*, você responde **o**.

Observe que às vezes o guarda faz pausas estilísticas (espaços em branco) no início ou no fim da linha. Esses espaços devem ser ignorados.

Implemente a função `backDoorResponse` que recebe uma linha do poema e retorna a última letra que **não** seja um espaço em branco.

```
backDoorResponse('Stands so high');  
// => "h"
```

```
backDoorResponse('Stands so high      ');  
// => "h"
```

---

### 4. Seja educado

Para entrar pelo fundo, você precisa ser educado. A senha deve ser capitalizada corretamente e terminar com , **please**.

Implemente a função `backDoorPassword` que recebe uma string (as letras combinadas da etapa 3) e retorna a versão educada da senha.

```
backDoorPassword('horse');  
// => "Horse, please"
```