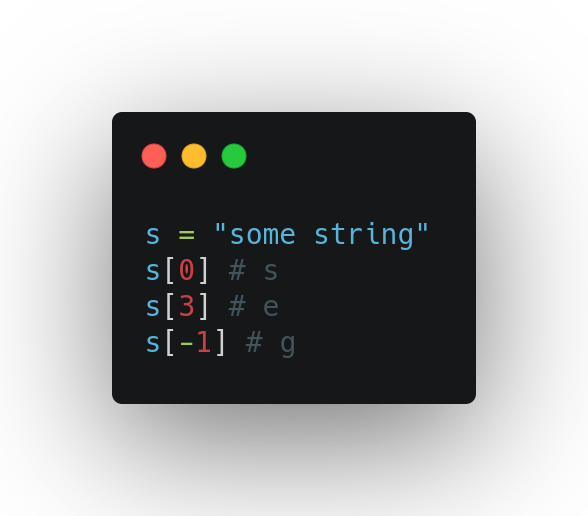
**Indexação e Fatiamento**

As strings são consideradas *listas* em Python, portanto, os caracteres dentro delas podem ser acessados via índice ou *fatiamento*.

**Índices:**

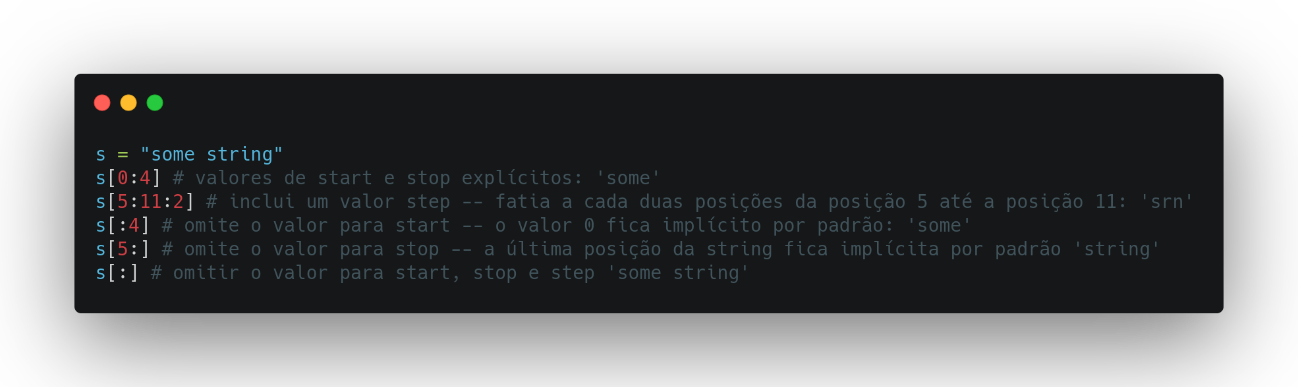
Para acessar um caractere em uma posição específica dentro de uma string, coloque um inteiro que representa a posição do caractere entre colchetes, [ ]. Os números positivos começam pela esquerda, pelo 0, e os números negativos começam pela direita, pelo -1.



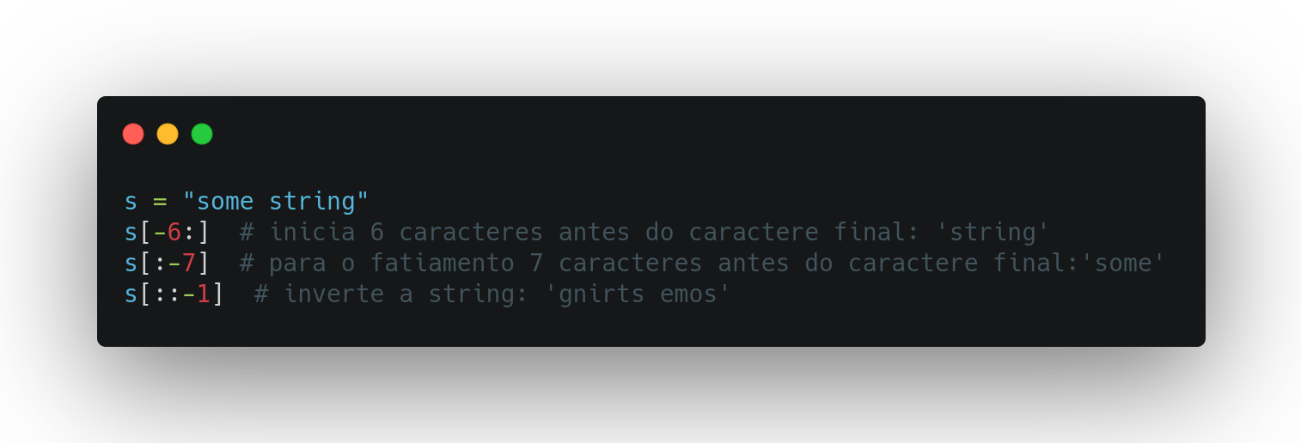
**Fatiamento**

Para acessar grupos de caracteres dentro de uma string, chamados *fatias*, use dois pontos (:) para separar a posição inicial da posição final. A sintaxe se parece com isso string[start:stop:step], onde *start*, *stop* e *step* são inteiros.

* A posição *start* está inclusa na fatia resultante.
* A posição *stop* **não** está inclusa na fatia resultante.
* O valor de *step* indica quantos caracteres devem ser pulados por vez.
* Todos os valores são opcionais, desde que seja incluído um dois pontos para representar a operação de fatiamento.



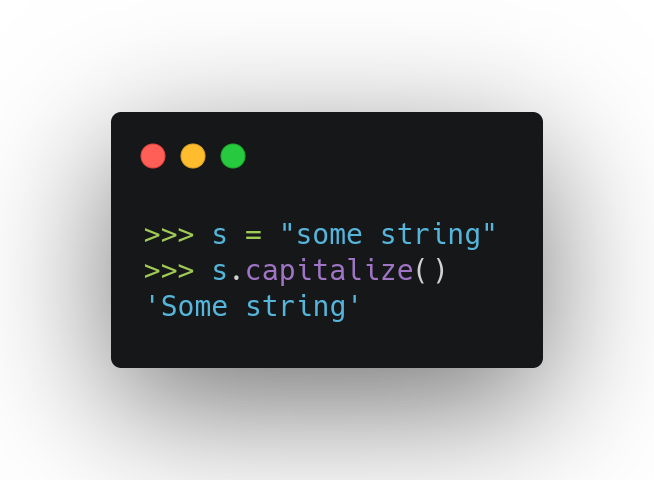
Você pode usar até mesmo valores negativos para criar fatias interessantes.



**Métodos de Strings**

Vamos explorar alguns métodos incorporados que vem com cada objeto string.

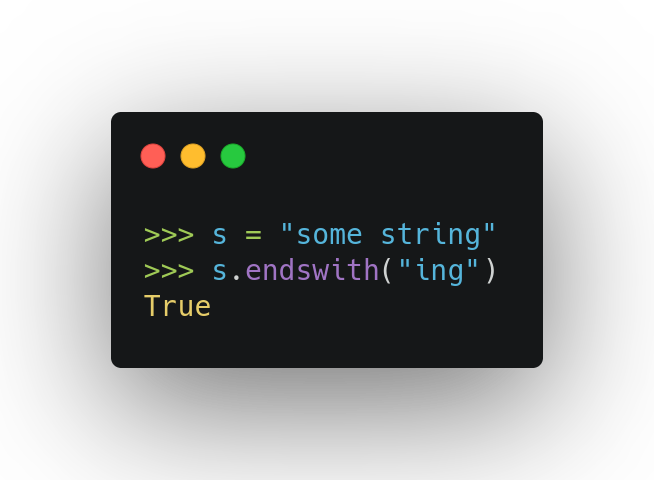
* **capitalize()** - muda o primeiro caractere da string para maiúsculo



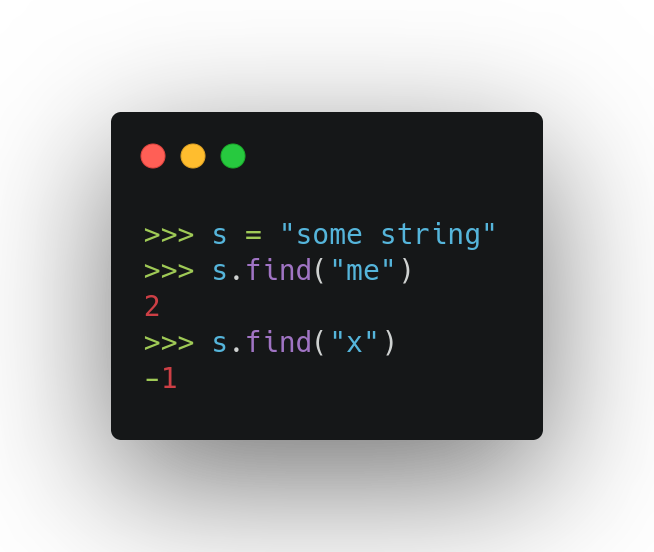
* **count()** - retorna o número de ocorrências de uma determinada sub-string da string



* **endswith()** - informa se a string termina com uma determinada sub-string

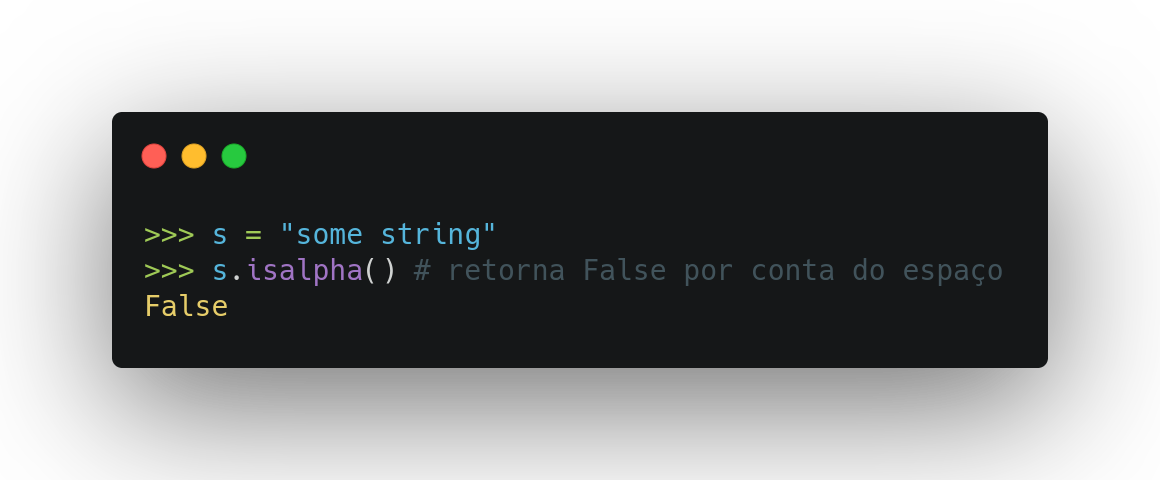


* **find()** - retorna o primeiro índice de onde uma determinada sub-string é encontrada dentro da string (retorna -1 se não encontrar nada)



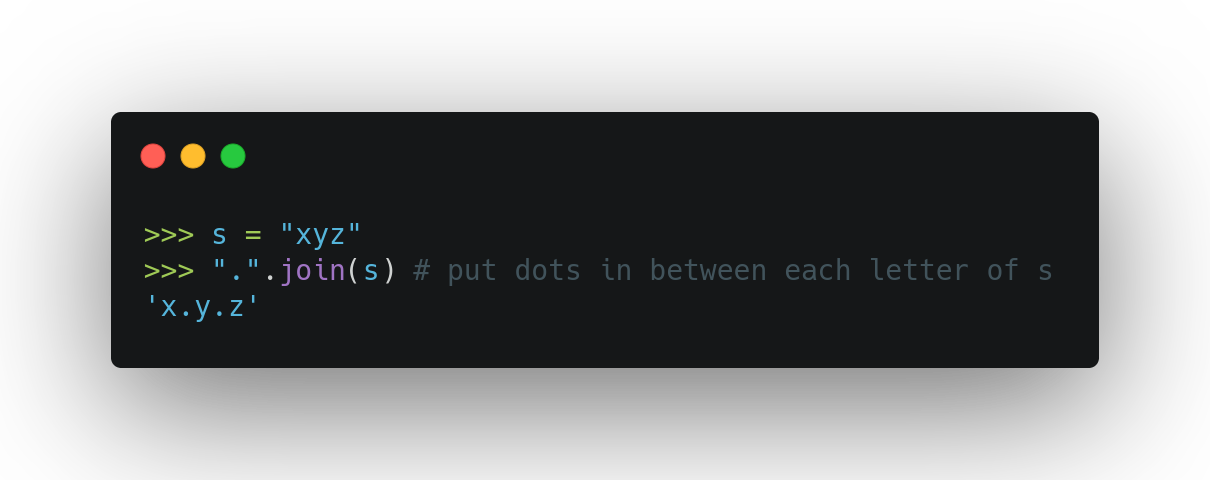
* **index()** - o mesmo que find(), mas retorna uma exceção ValueError caso a sub-string não seja encontrada
* **isalpha(), isdigit(), isspace ()** - informa se a string satisfaz os critérios de alfanumérico, número e espaço

Há um monte desses tipos de métodos que verificam as características de uma string. Digite **dir(str)** no REPL para consultá-los.

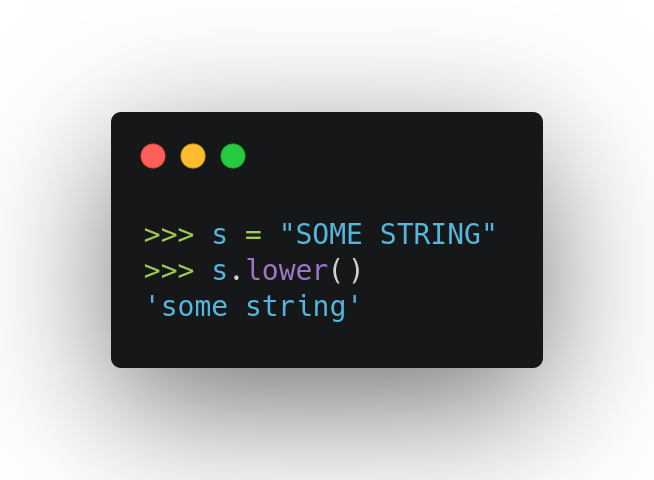


* **join()** - insere a string entre cada item de um determinado dado iterável

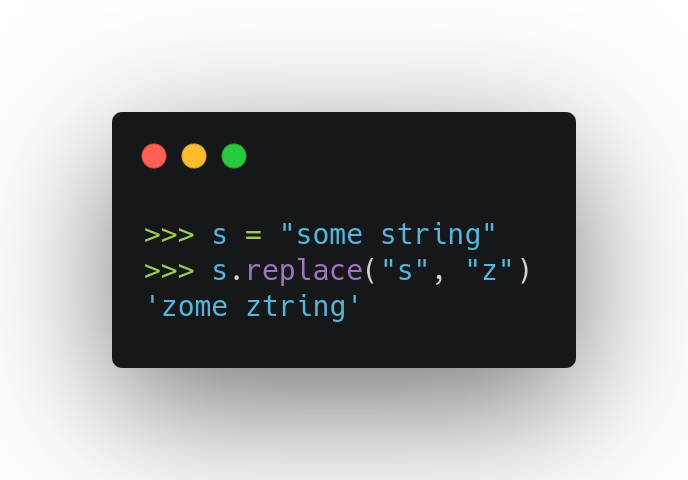
Este método é frequentemente usado anexado diretamente a uma string literal para ser usado como um separador para outros dados.



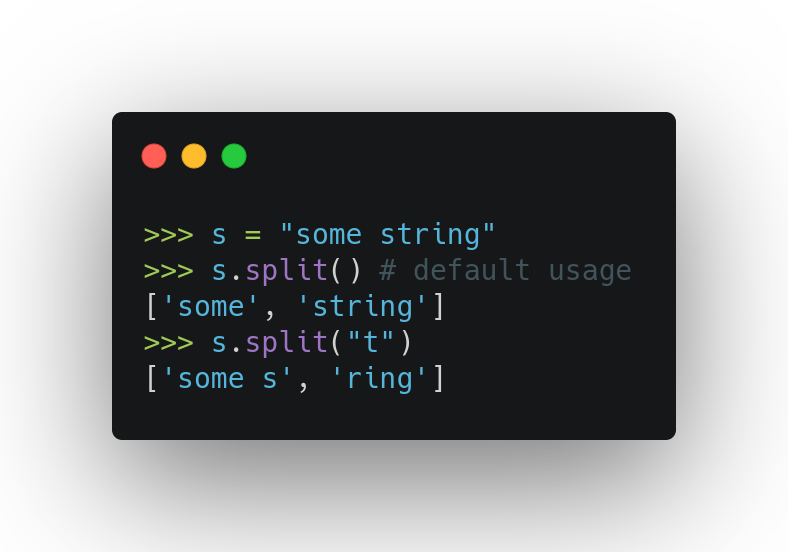
* **lower()** - retorna uma versão em letras minúsculas da string



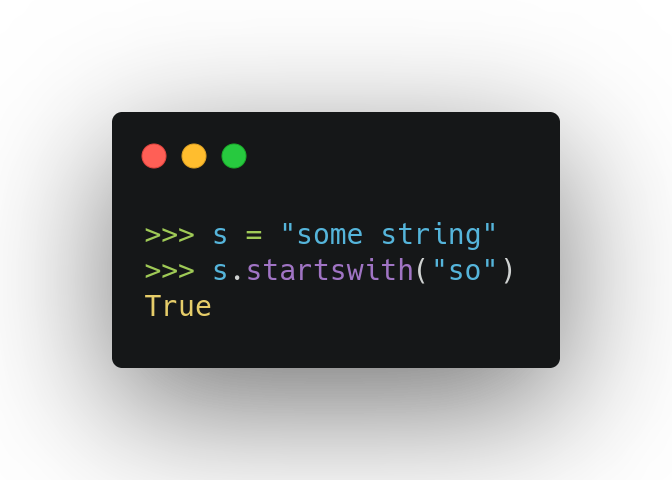
* **replace()** - substitui todas as ocorrências de uma sub-string da string por uma determinada string



* **split()** - retorna uma lista de palavras de uma string, com um separador opcional para ser usado como delimitador

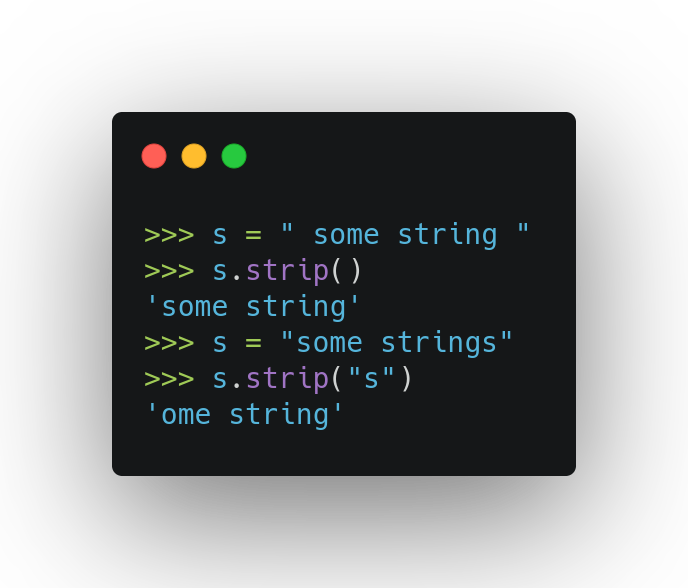


* **startswith()** - informa se uma string começa com uma determinada sub-string



* **strip()** - remove os espaços em branco iniciais e finais do texto

Opcionalmente, você pode indicar um caractere alternativo para remover do início e do final do texto.



* **upper()** - retorna uma versão maiúscula da string