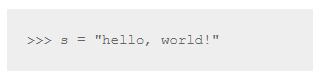
Atividade Python

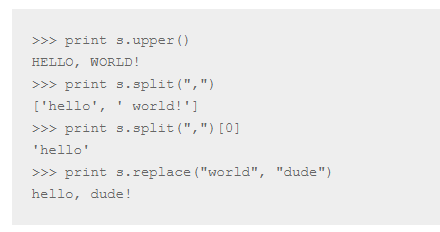
Matheus Andreoli V. 2ºY Nº20

FATIAMENTO (SLICING) DE STRINGS EM PYTHON

Strings em Python são objetos como outros quaisquer. Podem ser construídos com uma atribuição simples:



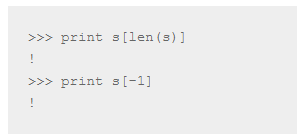
O objeto **s** possui disponíveis vários métodos:



Acima, são mostrados apenas alguns dos métodos disponíveis para as strings, nativamente em Python. Assim como em outras linguagens, elementos individuais de uma string podem ser acessados via índice:



Para acessar o último elemento de uma string, podemos proceder de duas formas:



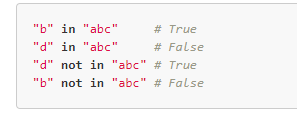
Uma operação muito interessante que Python fornece para manipulação de strings é o fatiamento (slicing). Fatiamento significa extrair apenas uma parte da string, ou seja, uma substring. Com essa operação, podemos delimitar os limites inferior e superior do pedaço da string que queremos acessar. Por exemplo, se quisermos acessar a substring da posição 0 até a posição 4 na string **s** original, podemos fazer o seguinte:



<https://pythonhelp.wordpress.com/2011/11/12/fatiamento-slicing-de-strings-em-python/>

### Pesquisando a string

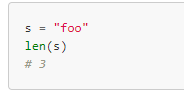
Podemos pesquisar um string com o **operador in**, exemplo:



## Métodos comuns para strings

### Método len()

Mostra o tamanho da string



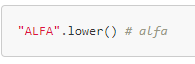
### Método upper()

Caixa alta.



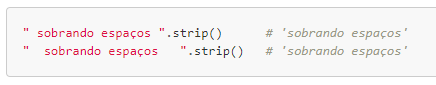
### Método lower()

Caixa baixa.



### Método strip()

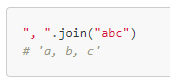
Retira espaços em branco no começo e no fim



### Método join()

Junta cada item da string com um delimitador especificado.

É o inverso do split().



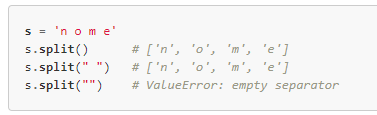
Aceita listas.



### Método split()

Separa uma string conforme um delimitador.

É o inverso do join().

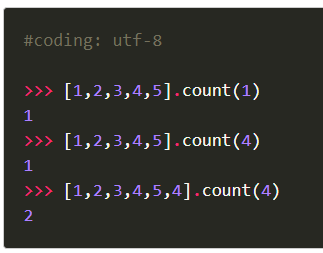


<http://devfuria.com.br/python/strings/>

## Função count()

A função count() retorna a quantidade de vezes que um mesmo elemento está contido numa lista. Essa é uma excelente maneira que evita a implementação de um Laço de Repetição em busca de elementos iguais.

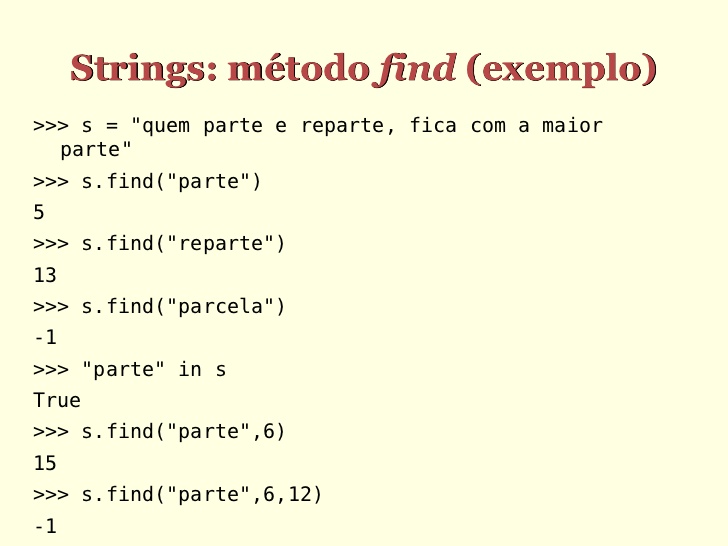
No código a seguir, defimos uma lista e em seguida, invocamos a função count(), pergunta assim, quantas vezes um determinado número existe nessa lista.



<http://excript.com/python/quantidade-de-itens-lista-python.html>

Método Find()

O método Find()retorna o index da 1º ocorrência da substring(se achar). Se não achar, retorna -1.

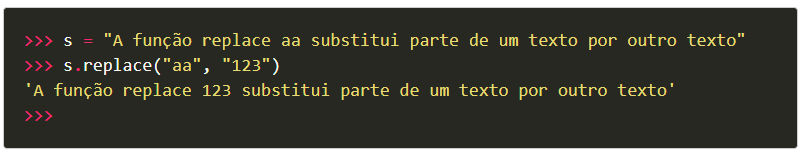


<https://www.programiz.com/python-programming/methods/string/find>

Método replace()

A função replace() substitui uma parte do texto por uma outra String. A palavra replace(), do Inglês, siginifca substituir e é isso que a função replace() da classe String do Python faz.

No código a seguir, iremos definir uma String e em seguida, substituiremos uma parte do texto por outra.



<http://excript.com/python/funcoes-string-python.html>

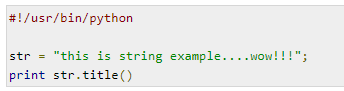
Método capitalize()

O método capitalize "troca" a primeira letra de uma string para maiúscula ex:  
>>>'python'.capitalize()  
Python

<http://python-iniciantes.blogspot.com/2012/08/dicas-12-trabalhando-strings-1.html>

Método title()

Esse método retorna uma cópia da string na qual a primeira letra de cada palavra vai estar em maiúsculo.



Quando executamos o código, aparece o seguinte resultado



<https://www.tutorialspoint.com/python/string_title.htm>

Método rstrip()

ele serve pra 'estripar' algum caractere, ou seja, retirar, dar um *strip* nele, e varrer da string.

3. Transformação

- Método Rstrip

- Método Lstrip