

Aprendendo ainda mais sobre Strings

Contando e pesquisando elementos em uma String

Aprendendo ainda mais sobre Strings

Contando e pesquisando elementos em uma String

Podemos contar elementos em uma string usando o método *count*.

Aprendendo ainda mais sobre Strings

Contando e pesquisando elementos em uma String

```
>>> texto = "Python é uma linguagem de programação. Python é simples. Python é organizado.  
Python é uma excelente linguagem."  
>>> texto.count("Python")  
4  
>>> texto.count("linguagem")  
2  
>>> texto.count("a")  
8  
>>> |
```

python.exe

Search 🔍 + ▾ 📄 🔒 📁 ≡

Aprendendo ainda mais sobre Strings

Contando e pesquisando elementos em uma String

Para um exemplo prático, pesquisei no Google por filetype:txt e baixei um arquivo do link:

http://www.konnexion.biz/discos_estoque.txt

Vamos abrir este arquivo e contar as ocorrências de uma determinada palavra.

Aprendendo ainda mais sobre Strings

Contando e pesquisando elementos em uma String

```
contador = 0
palavra = "volvo"
nome_arquivo = "discos_estoque.txt"
with open(nome_arquivo, "r") as arquivo:
    for linha in arquivo:
        contador = contador + linha.upper().count(palavra.upper())

print("Palavra pesquisada: " + palavra.upper())
print("Arquivo pesquisado: " + nome_arquivo)
print("Total de palavras encontradas: " + str(contador))
```

Aprendendo ainda mais sobre Strings

Contando e pesquisando elementos em uma String

Para que o programa possa ser reutilizado vamos possibilitar que o usuário informe a palavra a ser pesquisada e o nome do arquivo.

Aprendendo ainda mais sobre Strings

Contando e pesquisando elementos em uma String

```
contador = 0
palavra = input("Informe a palavra a ser pesquisada: ")
nome_arquivo = input("Informe o nome do arquivo: ")
with open(nome_arquivo, "r") as arquivo:
    for linha in arquivo:
        contador = contador + linha.upper().count(palavra.upper())

print("Palavra pesquisada: " + palavra.upper())
print("Arquivo pesquisado: " + nome_arquivo)
print("Total de palavras encontradas: " + str(contador))
```

Aprendendo ainda mais sobre Strings

Contando e pesquisando elementos em uma String

Podemos utilizar o método *find* para obter a posição da primeira ocorrência de uma String. O Python retorna a posição do primeiro caractere da String. Caso não encontre a ocorrência, o Python retorna -1.

Aprendendo ainda mais sobre Strings

Contando e pesquisando elementos em uma String

```
>>> texto = "Python é uma linguagem de programação. Python é simples. Python é organizado.  
Python é uma excelente linguagem."  
>>> texto.find("lingua")  
13  
>>> texto.find("Olá")  
-1
```

Aprendendo ainda mais sobre Strings

Contando e pesquisando elementos em uma String

Podemos utilizar o método *rfind* para realizar a pesquisa da direita para a esquerda.

Aprendendo ainda mais sobre Strings

Contando e pesquisando elementos em uma String

```
>>> texto = "Python é uma linguagem de programação. Python é simples. Python é organizado.  
Python é uma excelente linguagem."  
>>> texto.rfind("lingua")  
101  
>>> |
```

Aprendendo ainda mais sobre Strings

Contando e pesquisando elementos em uma String

Tanto *find* quanto *rfind* possuem dois parâmetros para que sejam delimitados o início e o fim para realização da pesquisa.

```
>>> texto = "Python é uma linguagem de programação. Python é simples. Python é organizado. Python é uma excelente linguagem."
>>> texto.find("Python", 25, 50)
39
>>> texto.rfind("Python", 25, 80)
57
>>> |
```

python.exe

Search 🔍 + ▾ 🔒 📄 ≡

Aprendendo ainda mais sobre Strings

Contando e pesquisando elementos em uma String

Para localizar o índice de ocorrência de uma string, podemos usar também o ***index*** (já apresentado na seção “Variáveis e tipos de dados”, na aula “Aprendendo mais sobre strings – aula 2”) e ***rindex***, porém, caso a substring não seja localizada, esses métodos geram uma exceção ***ValueError***.

```
>>> texto = "Python é uma linguagem de programação. Python é simples. Python é organizado. Python é uma excelente linguagem."
>>> texto.index("é")
7
>>> texto.rindex("é")
85
>>> texto.index("Teste")
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
ValueError: substring not found
>>> texto.rindex("Teste")
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
ValueError: substring not found
>>> |
```

FIM