

TECHMIMO

Autor: Rafael Pereira da Silva

Seguem alguns recados para ajudá-los e para contribuir com o curso:

- Fiquem à vontade para me contatar pelo Linkdin, costumo responder por lá também:
<https://www.linkedin.com/in/rafael-pereira-da-silva-23890799/> (<https://www.linkedin.com/in/rafael-pereira-da-silva-23890799/>)
- Fiquem a vontade para compartilharem o certificado do curso no LinkedIn. Eu costumo curtir e comentar para dar mais credibilidade
- Vocês podem usar esses notebooks para resolver os exercícios e desafios
- Não se esqueçam de avaliar o curso e dar feedback, eu costumo criar conteúdos baseado nas demandas de vocês
- Se tiverem gostando do curso, recomendem aos amigos, pois isso também ajuda a impulsionar e a crescer a comunidade
- Bons estudos e grande abraços!

Seção 6 - Complementar

Esta seção preenche algumas lacunas que podem ter sido deixadas. Este curso não é voltado para orientação a objetos, mas é importante que vocês tenham uma noção de como funciona.

6.1 Noção sobre objetos

Sintaxe:

```
class Nome_da_classe: #cria um novo objeto
```

```
    **def \__init\__ (self,argumentos):** #Magic method que cria as propriedades
```

<https://docs.python.org/3/tutorial/classes.html> (<https://docs.python.org/3/tutorial/classes.html>).

In [19]:

```
class carro:
    def __init__(self, cor_do_carro, marca_do_carro, ano_do_carro):
        self.cor = cor_do_carro
        self.marca = marca_do_carro
        self.ano = ano_do_carro
        self.donos = []

    def add_dono(self, novo_dono):
        self.donos.append(novo_dono)

meu_carro = carro('cinza', 'vw', 2000)
meu_carro.add_dono('Rafael')
meu_carro.add_dono('João')
meu_carro.donos

len(meu_carro.donos)
```

Out[19]:

2

In [33]:

#Exemplo do SoloLearn

```
class vetor_2d:
    def __init__(self, x, y):
        self.x = x
        self.y = y

    def norma(vet):
        return ((vet.x**2 + vet.y**2)**(1/2))

    def __add__(vet1, vet2):
        return vetor_2d(vet1.x + vet2.x, vet1.y + vet2.y)

v_1 = vetor_2d(3, 3)
v_2 = vetor_2d(4, 5)

v_3 = v_1 + v_2
v_3.norma()
```

Out[33]:

10.63014581273465

In []:

In []:

In []:

In []:

6.2 Executando módulos como scripts

Estrutura:

```
if __name__ == "__main__":
```

Código

In [18]:

```
__name__
```

Out[18]:

```
'__main__'
```

In [19]:

```
#codigo_1
```

```
def funcao1():  
    print('funcao1')  
def funcao2():  
    print('funcao2')  
  
if __name__ == '__main__':  
    funcao1()  
    funcao2()  
    print('Este código está sendo executado como script')  
else:  
    print('Este código está sendo chamado como módulo')
```

```
funcao1
```

```
funcao2
```

```
Este código está sendo executado como script
```

In []:

```
import codigo_1 as cd
```

```
cd.funcao1()
```

In []:

In []:

In []:

In []:

In []:

6.3 Manipulando strings

Alguns métodos de manipulação de strings:

| Método | Descrição |
|--------------------------|--|
| upper() | retorna letras maiúsculas |
| lower() | retorna letras minúsculas |
| replace(arg_ant,arg_nov) | substitui arg_ant pelo arg_nov |
| find(arg) | retorna a posição da primeira ocorrência |
| rfind() | retorna a posição da última ocorrência |

Atenção: Esses métodos não modificam a string original se ela estiver armazenada em uma variável.

fonte: <https://panda.ime.usp.br/pensepy/static/pensepy/08-Strings/strings.html>
(<https://panda.ime.usp.br/pensepy/static/pensepy/08-Strings/strings.html>)

In [3]:

```
string_1 = 'python'
string_2 = string_1.upper()
string_2
```

Out[3]:

```
'PYTHON'
```

In [7]:

```
string_3 = 'asdfasdfasdfasdf'
string_3.find('s')
```

Out[7]:

1

In [11]:

```
string_4 = 'pyathoan'
string_5 = string_4.replace('a', '')
string_5
```

Out[11]:

'python'

In [15]:

```
string_6 = string_4.split('a')
string_6
```

Out[15]:

['py', 'tho', 'n']

In []:

6.4 Escrevendo arquivos

- Estrutura para ler:

with open(caminho_string) as f:

```
    variavel = **f.read()**

    **f.close**
```

- Estrutura para escrever:

with open(nome_novo_arquivo) as f:

```
    f.write('nova_string','w')

    **f.close**
```

In [21]:

```
old_file = 'Arquivo_sujo.txt'

with open(old_file) as f:
    linhas = f.read()
    f.close

linhas_new = linhas.replace('p','').replace(' ','')
linhas_new

new_file = 'Arquivo_limpo.txt'

with open(new_file, 'w') as f:
    f.write(linhas_new)
    f.close
```

In []: