

Programação em Python



Prof. Daniel Santos

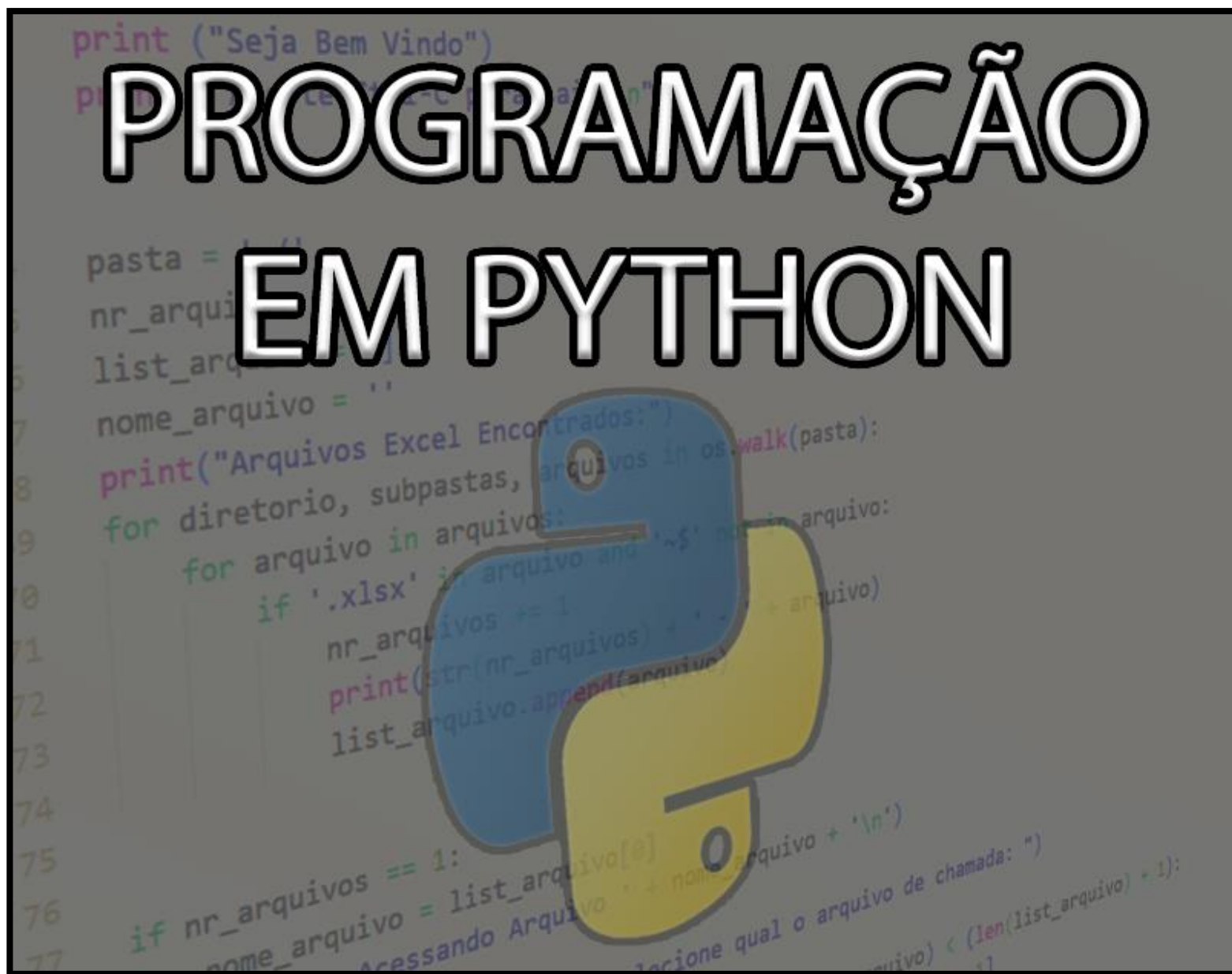
daniel.sampaio@sp.senai.br

Senai Roberto Simonsen

R. Monsenhor Andrade, 298 - Brás, São Paulo - SP

PROGRAMAÇÃO

EM PYTHON



Prof. Daniel Santos

Manipulação de arquivos em Python

Nativamente, o Python possui a capacidade de abrir e editar arquivos simples como **txt**.

Para criar um arquivo numa subpasta (previamente criada) chamada **Arquivos**, basta utilizar o código a seguir:

```
with open('Arquivos/arquivo.txt','w') as arq:  
    arq.write('Olá mundo\n')
```

O comando **with open** manipula a abertura de um arquivo e fecha ele caso haja algum problema durante a execução do código, facilitando a operação com arquivos.

O primeiro parâmetro é o diretório onde será salvo, pode ser apenas passado o diretório atual ou o caminho completo para o diretório.

(ex: C:/Users/dansa/Desktop/Senai/FIC/Python/Codes/Pycharm/Codes/Arquivos/arquivo.txt)

Manipulação de arquivos em Python



Para abrir um arquivo, você pode selecionar um dos seguintes modos:

	r	r+	w	w+	a	a+
ler	*	*		*		*
escrever		*	*	*	*	*
criar			*	*	*	*
deletar dados			*	*		
posiciona no começo	*	*	*	*		
posiciona no final					*	*

Manipulação de arquivos em Python

Exemplo de criação e leitura de arquivo:

Criar arquivo novo (se existir, deleta o conteúdo)

```
with open('Arquivos/arquivo.txt','w') as file:  
    file.write('teste\n')
```

Abrir para Leitura

```
with open('Arquivos/arquivo.txt','r') as file:  
    print(file.read())
```

Abrir para Adicionar dados

```
with open('Arquivos/arquivo.txt','a') as file:  
    file.write('Nova Linha')
```

Abrir para Leitura

```
with open('Arquivos/arquivo.txt','r') as file:  
    print(file.read())
```

CSV



Esta é uma biblioteca (**Library**) nativa em Python, utilizada para ler e escrever arquivos em **CSV** (**C**omma **S**eparated **V**alues → Valores Separados por Vírgula).

Esta biblioteca permite tratar os dados armazenados em um arquivo **CSV** como uma lista:

```
import csv
lista_de_nomes = ['joao', 'daniel', 'ana']
with open('Arquivos/pessoas.csv', 'w') as file:
    csv.writer(file).writerow(lista_de_nomes)
```

Podemos criar uma **variável** para armazenar o modo **writer**:

```
import csv
lista_de_nomes = ['joao', 'daniel', 'ana']
with open('Arquivos/pessoas.csv', 'w') as file:
    writer = csv.writer(file)
    writer.writerow(lista_de_nomes)
```


CSV

Quando lemos um arquivo em **CSV** ele retorna como um objeto **iterável** do tipo **csv.reader**. Se fizermos uma **for** dentro do objeto conseguimos extrair os dados por linha:

```
import csv

with open('Arquivos/pessoas.csv','r') as file:
    reader = csv.reader(file)
    for linha in reader:
        print(linha)
```

OBS: Os dados extraídos retornam como uma lista.

Maiores informações: <https://www.programiz.com/python-programming/csv>
<https://docs.python.org/3/library/csv.html>

Instalando bibliotecas não nativas

Algumas bibliotecas precisam ser instaladas para poderem ser utilizadas.

Para isto pode-se utilizar a instalação direta no ambiente **core** do sistema, no qual a biblioteca estará disponível para qualquer projeto que seja elaborado.

Também é possível instalar utilizando algum **ambiente virtual**, no qual a biblioteca será instalada apenas para aquele projeto.

Para instalar no sistema, basta abrir o **prompt de comando**.

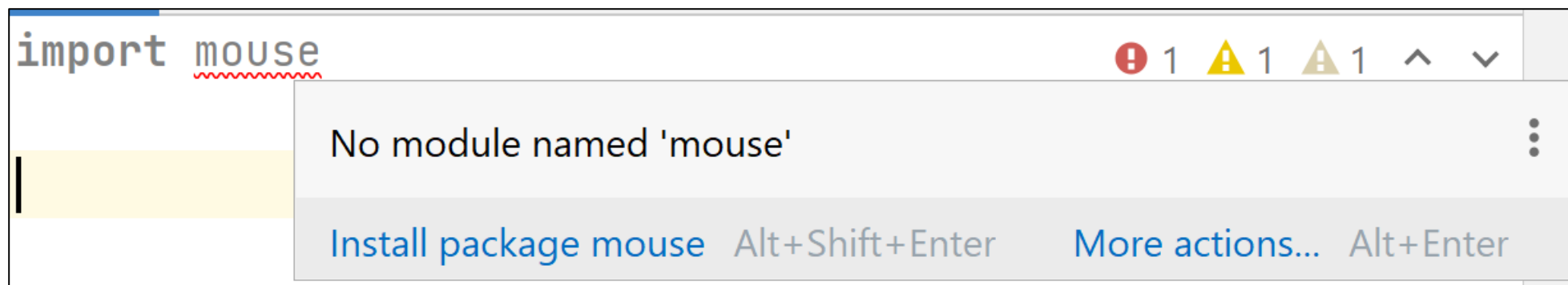
Exemplo: **pip3 install Mouse**

```
C:\Users\dansa>pip3 install Mouse
Collecting Mouse
  Using cached mouse-0.7.1-py2.py3-none-any.whl (16 kB)
Installing collected packages: Mouse
Successfully installed Mouse-0.7.1
```


Instalando bibliotecas não nativas

Para instalar dentro de um projeto criado previamente em um **ambiente virtual (Virtual Env)** basta utilizar a IDE com o projeto aberto.

Pycharm:



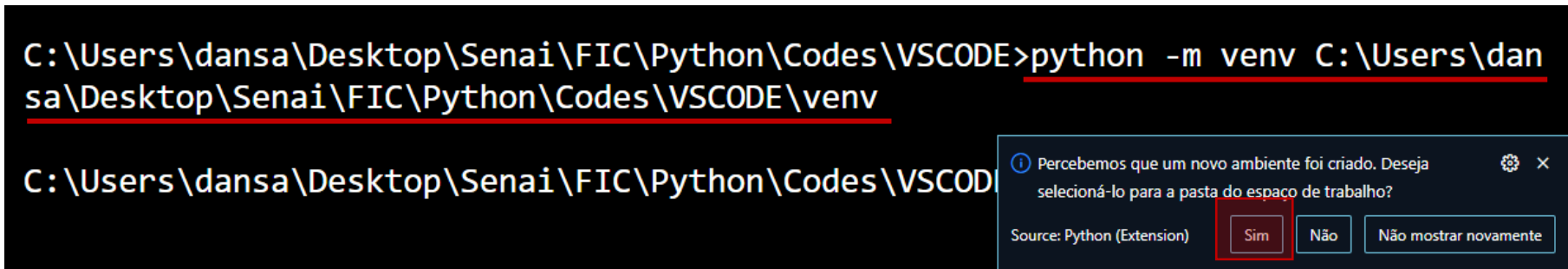
Instalando bibliotecas não nativas

No **VSCode** precisa abrir o ambiente virtual por terminal, e então instalar pelo próprio terminal:

1. Criar ambiente virtual dentro do **workplace**:

python -m venv DIRETÓRIO_DO_SEU_WORKPLACE\venv

2. Clicar em **Sim**

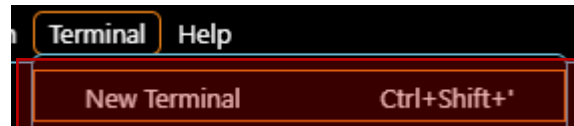


```
C:\Users\dansa\Desktop\Senai\FIC\Python\Codes\VSCODE>python -m venv C:\Users\dansa\Desktop\Senai\FIC\Python\Codes\VSCODE\venv
```

Percebemos que um novo ambiente foi criado. Deseja selecioná-lo para a pasta do espaço de trabalho?

Source: Python (Extension) **Sim** Não Não mostrar novamente

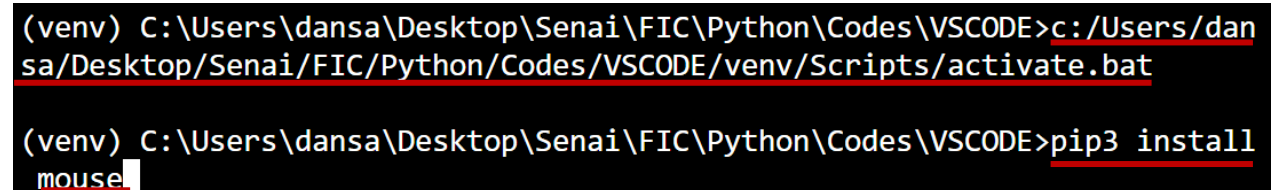
3. Abrir **Novo Terminal**



4. Ativar o terminal e instalar a biblioteca:

DIRETÓRIO_DO_SEU_WORKPLACE\venv\Scripts\activate.bat

pip3 install mouse



```
(venv) C:\Users\dansa\Desktop\Senai\FIC\Python\Codes\VSCODE>c:/Users/dansa/Desktop/Senai/FIC/Python/Codes/VSCODE/venv/Scripts/activate.bat
```

```
(venv) C:\Users\dansa\Desktop\Senai\FIC\Python\Codes\VSCODE>pip3 install mouse
```

Installing collected packages: mouse
Successfully installed mouse-0.7.1

MOUSE



Esta biblioteca permite ler e comandar o **mouse**.

Para ler basta seguir o exemplo:

```
import mouse
import time

while True:
    if mouse.is_pressed('left'):
        print(mouse.get_position())
        time.sleep(0.5)
    elif mouse.is_pressed('middle'):
        break
```

MOUSE



Para comandar o mouse:

```
import mouse
```

```
mouse.drag(0, 0, -200, 200, absolute=False, duration=1)
```

```
mouse.click('right')
```

Para gravar todos os eventos do mouse até clicar o botão **direito**, pode-se utilizar:

```
import mouse
```

```
events = mouse.record()
```

```
print(events)
```

Maiores informações: <https://www.thepythoncode.com/article/control-mouse-python>

KEYBOARD

Esta biblioteca permite ler e comandar o **teclado**.

Referência: <https://stackabuse.com/guide-to-pythons-keyboard-module/>

Para ler basta seguir o exemplo:

```
import keyboard
```

```
rec = keyboard.record(until='Esc')  
print(rec)
```

Para controlar, pode se utilizar algumas funções:

```
import keyboard
```

```
import time
```

```
keyboard.press("a")  
time.sleep(1)  
keyboard.release("a")  
keyboard.press_and_release('\n, b, \n')  
keyboard.write('python')
```

KEYBOARD

Exemplo de como adicionar . e - no CPF automaticamente:

```
import keyboard
from time import sleep

cpf = ""
while True:
    event = keyboard.read_event()
    if event.event_type == keyboard.KEY_DOWN:
        number = event.name
        if number == 'enter':
            break
        elif number.isdigit():
            cpf += number
            #print(len(cpf))
            sleep(0.1)
            if len(cpf) == 3:
                cpf += '.'
                print('.',end="")
            elif len(cpf) == 7:
                cpf += '.'
                print('.',end="")
            elif len(cpf) == 11:
                cpf += '-'
                print('-',end="")

print('\nCPF:', cpf)
```