



escola
britânica de
artes criativas
& tecnologia

Analista de Dados

Python: Scripting

Módulo | Python: Scripting

Caderno de Aula

Professor [André Perez](#)

Tópicos

1. Instalando o Python na própria máquina;
 2. Executando código Python no terminal;
 3. Criando arquivos de script Python.
-

Aulas

1. Instalando o Python na própria máquina

1.1. Google Colab

Python e o PIP já vem instalado na máquina virtual provisionada. **Exemplo:** Conferir a versão do Python via Python.

```
In [ ]: import platform
```

```
In [ ]: print(platform.python_version())
```

```
In [ ]: print(platform.system())
```

Exemplo: Conferir a versão do Python via Bash (linguagem nativa do Linux).

```
In [ ]: !python -v
```

Exemplo: Conferir a versão do PIP via Bash (linguagem nativa do Linux).

```
In [ ]: !pip -V
```

1.2. Máquina Local

Verifique se você já tem o Python e o PIP instalado na sua máquina. Para verificar a instalação, abrir o terminal (windows `cmd` ou linux/macOS `terminal`) e executar o seguinte comando.

```
python -V
pip -V
```

Se não o tiver instalado, acessar o site oficial do Python ([link](#)) e instalar o software para o seu sistema operacional. Um excelente tutorial se encontra neste [link](#). Para verificar a instalação, abrir o terminal (windows `cmd` ou linux/macOS `terminal`) e executar o seguinte comando.

```
python -V
pip -V
```

2. Executando código Python no terminal

2.1. Google Colab

O software do Python já vem "aberto" para ser utilizado.

```
In [ ]: from datetime import datetime

print(datetime.now().year)
```

2.2. Máquina Local

1. Criar uma pasta (sugestão);
2. Abrir o terminal nesta pasta (windows `shift + right click` ou linux/macOS `right click`);
3. Executar o seguinte comando:

```
python
```

Para 'sair' do Python, basta fechar o terminal ou digitar o seguinte código:

```
quit()
```

3. Criando arquivos de script Python

3.1. Google Colab

1. Criar um arquivo de *script* com a extensão `py` seguindo as boas práticas do PEP 8 ([link](#));
2. Executar com o seguinte comando:

```
!python arquivo.py
```

Exemplo: Script para extrair a taxa CDI do site da B3.

In []:

```
import os
import json
from random import random
from datetime import datetime

import requests

URL = 'https://www2.cetip.com.br/ConsultarTaxaDi/'+
      'ConsultarTaxaDICetip.aspx'

# Criando a variável data e hora

data_e_hora = datetime.now()
data = datetime.strftime(data_e_hora, '%Y/%m/%d')
hora = datetime.strftime(data_e_hora, '%H:%M:%S')

# Captando a taxa CDI do site da B3

try:
    response = requests.get(URL)
    response.raise_for_status()
except requests.HTTPError as exc:
    print("Dado não encontrado, continuando.")
    cdi = None
except Exception as exc:
    print("Erro, parando a execução.")
    raise exc
else:
    dado = json.loads(response.text)
    cdi = float(
        dado['taxa'].replace(',', '.')
    ) + (random() - 0.5)

# Verificando se o arquivo "taxa-cdi.csv" existe

if os.path.exists('./taxa-cdi.csv') == False:

    with open(file='./taxa-cdi.csv', mode='w', encoding='utf8')
    as fp:
        fp.write('data,hora,taxa\n')

# Salvando dados no arquivo "taxa-cdi.csv"

with open(file='./taxa-cdi.csv', mode='a', encoding='utf8')
as fp:
    fp.write(f'{data},{hora},{cdi}\n')

print("Sucesso")
```

Sucesso

Vamos criar o arquivo de *script* `extrair-cdi.py`.

```
In [ ]: !python extrair-cdi.py
```

Sucesso

Exemplo: Script com argumentos.

```
In [ ]: from sys import argv

print(argv)
print(type(argv))
```

Vamos criar o arquivo de *script* `args.py`.

```
In [ ]: !python args.py andre True 10 20

['args.py', 'andre', 'True', '10', '20']
<class 'list'>
0
args.py
<class 'str'>
1
andre
<class 'str'>
2
True
<class 'str'>
3
10
<class 'str'>
4
20
<class 'str'>
```

Exemplo: *Script* para gerar um grafico da taxa CDI do site da B3.

```
In [ ]: import csv
        from sys import argv

        import seaborn as sns

        # Extraíndo as colunas hora e taxa

        horas = []
        taxas = []

        with open(file='./taxa-cdi.csv', mode='r', encoding='utf8') as fp:
            linha = fp.readline()
            linha = fp.readline()
            while linha:
                linha_separada = linha.split(sep=',')
                hora = linha_separada[1]
                horas.append(hora)
                taxa = float(linha_separada[2])
                taxas.append(taxa)
                linha = fp.readline()

        # Salvando no grafico

        grafico = sns.lineplot(x=horas, y=taxas)
        grafico.get_figure().savefig(f'{argv[1]}.png')
```

Vamos criar o arquivo de *script* `cdi-grafico.py`.

```
In [ ]: !python cdi-grafico.py dia-10
```

3.2. Máquina Local

Mesma dinâmica!

3.3. Ferramenta de Desenvolvimento Local

As IDEs (Integrated Development Environment) são ferramentas completas de desenvolvimento de código em software.

- **PyCharm** da JetBrains ([link](#));
- **Visual Studio Code** da Microsoft ([link](#)).