



escola
britânica de
artes criativas
& tecnologia

Módulo 07 | Python: Programação Orientada a Objetos

Caderno de **Exercícios**

Professor [André Perez](#)

Tópicos

1. from / import / as;
 2. Módulo;
 3. Pacote;
 4. Baixando pacotes.
-

Exercícios

0. Preparação do ambiente

Neste exercício vamos utilizar a base de dados de ações da bolsa de valores dos EUA, a Dow Jones. Os dados estão disponíveis para *download* neste [link](#). Vamos utilizar o pacote `wget` para fazer o *download* dos dados.

- Instalando o pacote `wget` na versão 3.2.

```
In [ ]: !pip install wget==3.2
```

- Fazendo o download dos dados no arquivo compactado `dados.zip`.

```
In [ ]: import wget

wget.download(
    url='https://archive.ics.uci.edu/ml/machine-learning-databases/' + \
        '00312/dow_jones_index.zip',
    out='./dados.zip'
)
```

- Descompactando os `dados` na pasta dados com o pacote nativo `zipfile`.

```
In [ ]: import zipfile

with zipfile.ZipFile('./dados.zip', 'r') as fp:
    fp.extractall('./dados')
```

Verifique a pasta dados criada, ela deve conter dois arquivos:

- **dow_jones_index.data**: um arquivo com os dados;
- **dow_jones_index.names**: um arquivo com a descrição completa dos dados.

É possível observar que o arquivo de dados é um arquivo separado por virgulas, o famoso `csv`. Vamos renomear o arquivo de dados para que ele tenha a extensão `csv` com o pacote nativo `os`.

- Renomeando o arquivo com o pacote nativo `os`.

```
In [ ]: import os

os.rename('./dados/dow_jones_index.data', './dados/dow_jones_index.csv')
```

Pronto! Abra o arquivo e o Google Colab irá apresentar uma visualização bem legal dos dados.

1. Pandas

Para processar os dados, vamos utilizar o pacote `pandas` na versão `1.1.5`. A documentação completa por ser encontrada neste [link](#)

```
In [ ]: !pip install pandas==1.1.5
```

Vamos importar o pacote com o apelido (alias) `pd`.

```
In [ ]: import pandas as pd
```

Estamos prontos para ler o arquivo.

```
In [ ]: df = pd.read_csv('./dados/dow_jones_index.csv')
```

O pandas trabalha com o conceito de dataframe, uma estrutura de dados com muitos métodos e atributos que aceleram o processamento de dados. Alguns exemplos:

- Visualizando as `n` primeiras linhas:

```
In [ ]: df.head(n=10)
```

- Visualizando o nome das colunas:

```
In [ ]: df.columns.to_list()
```

- Verificando o número de linhas e colunas.

```
In [ ]: linhas, colunas = df.shape
        print(f'Número de linhas: {linhas}')
        print(f'Número de colunas: {colunas}')
```

Vamos selecionar os valores de abertura, fechamento, máximo e mínimo das ações do McDonalds, listado na Dow Jones como MCD:

- Selecionando as linha do dataframe original `df` em que a coluna `stock` é igual a `MCD`.

```
In [ ]: df_mcd = df[df['stock'] == 'MCD']
```

- Selecionando apenas as colunas de data e valores de ações.

```
In [ ]: df_mcd = df_mcd[['date', 'open', 'high', 'low', 'close']]
```

Excelente, o problema é que as colunas com os valores possuem o caráter `$` e são do tipo texto (`object` no `pandas`).

```
In [ ]: df_mcd.head(n=10)
```

```
In [ ]: df_mcd.dtypes
```

Vamos limpar as colunas com o método `apply`, que permite a aplicação de uma função anônima (`lambda`) qualquer. A função `lambda` remove o caracter `$` e faz a conversão do tipo de `str` para `float`.

```
In [ ]: for col in ['open', 'high', 'low', 'close']:
        df_mcd[col] = df_mcd[col].apply(
            lambda value: float(value.split(sep='$')[-1])
        )
```

Verifique novamente os dados e seus tipos.

```
In [ ]: df_mcd.head(n=10)
```

```
In [ ]: df_mcd.dtypes
```

Excelente, agora podemos explorar os dados visualmente.

Agora é a sua vez! Conduza o mesmo processo para extrair e tratar os dados da empresa Coca-Cola (`stock` column igual a `KO`).

```
In [ ]: # extração e tratamento dos dados da empresa Coca-Cola.
```

2. Seaborn

Para visualizar os dados, vamos utilizar o pacote `seaborn` na versão `0.11.1`. A documentação completa por ser encontrada neste [link](#)

```
In [ ]: !pip install seaborn==0.11.1
```

Vamos importar o pacote com o apelido (alias) `sns`.

```
In [ ]: import seaborn as sns
```

Vamos visualizar o os valores de abertura das ações ao longo do tempo.

```
In [ ]: plot = sns.lineplot(x="date", y="open", data=df_mcd)
_ = plot.set_xticklabels(labels=df_mcd['date'], rotation=90)
```

Vamos também visualizar o os valores de fechamento das ações ao longo do tempo.

```
In [ ]: plot = sns.lineplot(x="date", y="close", data=df_mcd)
_ = plot.set_xticklabels(labels=df_mcd['date'], rotation=90)
```

Para facilitar a comparação, vamo visualizar os quatro valores no mesmo gráfico.

```
In [ ]: plot = sns.lineplot(
    x="date",
    y="value",
    hue='variable',
    data=pd.melt(df_mcd, ['date'])
)

_ = plot.set_xticklabels(labels=df_mcd['date'], rotation=90)
```

Para finalizar, vamos salvar o gráfico numa figura.

```
In [ ]: plot.figure.savefig("./mcd.png")
```

Agora é a sua vez, faça o gráfico acima para a empresa Coca-Cola e salve a imagem com o nome `ko.png`.

```
In [ ]: # visualização dos dados da Coca-Cola.
```

Analise as duas imagens e escreva pelo menos um *insight* que você consegue extrair dos dados. Fique a vontade para escrever quantos *insights* você quiser.

Insight #1: ...
