

ORIENTAÇÃO DE INSTALAÇÃO E CONFIGURAÇÃO DE AMBIENTE PARA UTILIZAÇÃO DE PANDAS

PROGRAMAÇÃO EM PYTHON

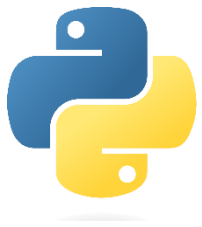
ORIENTADOR : BRUNO TOMAZ DOS SANTOS





Sumário

1	Bibliotecas em python	2
2	Passo 01 – O que é PANDAS ?	2
3	primeiros passos	2
4	Utilizando pandas	5



1 BIBLIOTECAS EM PYTHON

As bibliotecas em Python são conjuntos de módulos e funções que simplificam o desenvolvimento de programas, economizando tempo e esforço. Com mais de 137 mil bibliotecas disponíveis, elas oferecem diversas funcionalidades, desde tratamento de dados até automação. Utilizá-las amplia a capacidade de Python, tornando-o uma linguagem versátil e eficiente, acelerando o desenvolvimento de software e permitindo a resolução de uma ampla variedade de tarefas com facilidade e precisão.

2 PASSO 01 – O QUE É PANDAS ?

A biblioteca Pandas é uma poderosa ferramenta em Python para análise de dados. Ela oferece estruturas de dados flexíveis, como DataFrames, que permitem armazenar e manipular informações de maneira eficiente. Com o Pandas, é possível realizar tarefas como carregar, limpar, transformar e analisar dados de forma rápida e precisa. Sua versatilidade a torna essencial para cientistas de dados, analistas e desenvolvedores, facilitando a exploração e o processamento de dados de maneira eficaz. Além disso, o Pandas é amplamente utilizado em tarefas de limpeza, transformação e análise de dados em projetos de ciência de dados e análise estatística.

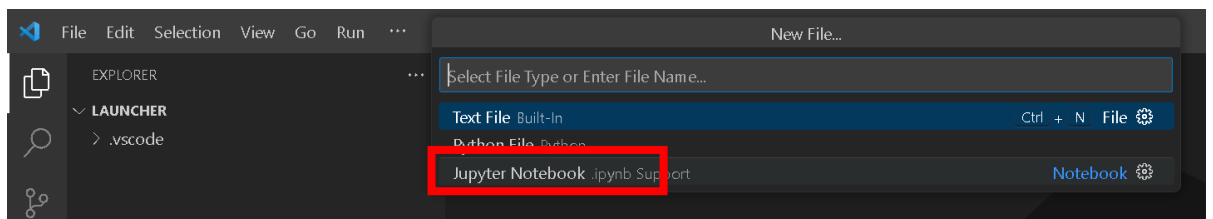
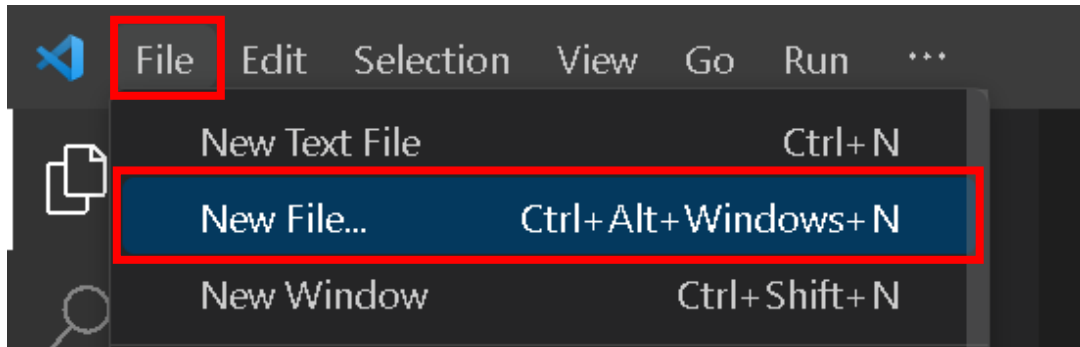
3 PRIMEIROS PASSOS

Para começar a entender a estrutura do pandas e seu funcionamento é recomendada a leitura da documentação oficial.

<https://pandas.pydata.org/docs/>

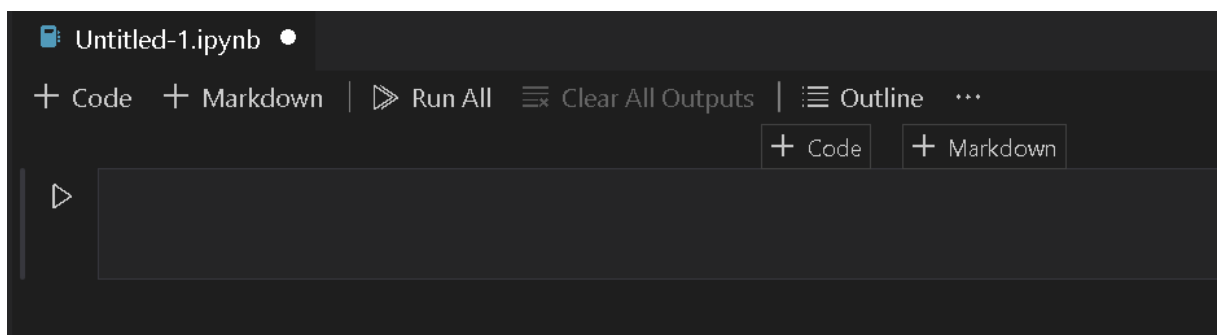


Para esclarecer alguns conceitos e dar os primeiros passos vamos iniciar instalando o pandas no ambiente jupyter utilizando o comando pip:



(Caso a execução gere algum tipo de erro tente ler a mensagem de erro e ver qual foi o problema)

Caso tudo ocorra bem o que deve aparecer na sua tela é:



Agora use o comando pip para instalar a biblioteca pandas em seu visual studio code.



```

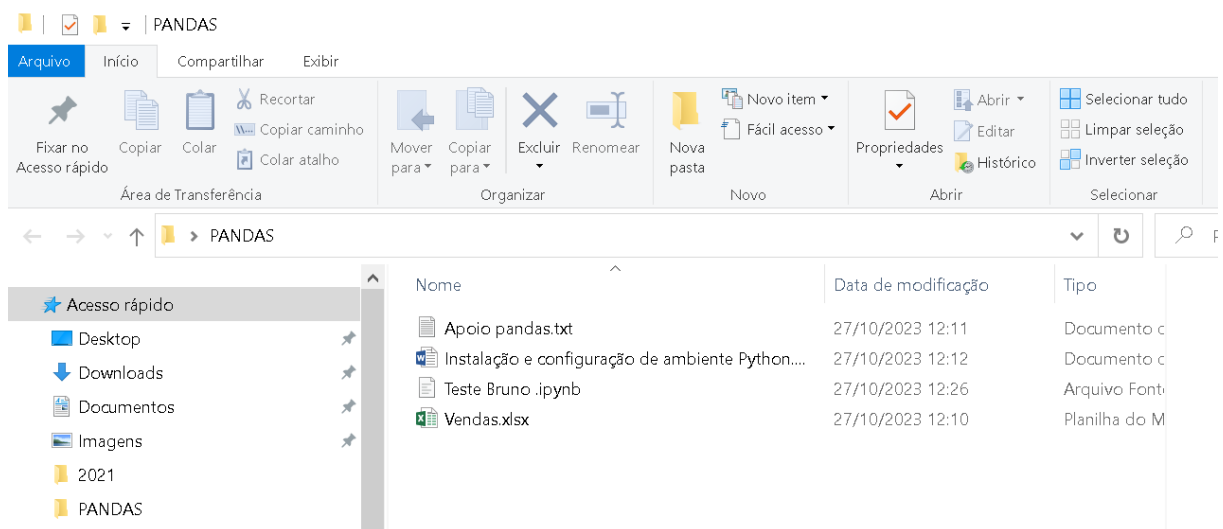
Untitled-1.ipynb •
+ Code + Markdown | ▶ Run All ↺ Restart ☰ Clear All Outputs | ☰ Outline ...
+ Code + Markdown

[1] ✓ 3.9s

... Requirement already satisfied: pandas in c:\users\instrutor\appdata\local\programs\python\python39\python.exe
Requirement already satisfied: numpy>=1.23.2 in c:\users\instrutor\appdata\local\programs\python\python39\python.exe
Requirement already satisfied: python-dateutil>=2.8.2 in c:\users\instrutor\appdata\local\programs\python\python39\python.exe
Requirement already satisfied: pytz>=2020.1 in c:\users\instrutor\appdata\local\programs\python\python39\python.exe
Requirement already satisfied: tzdata>=2022.1 in c:\users\instrutor\appdata\local\programs\python\python39\python.exe
Requirement already satisfied: six>=1.5 in c:\users\instrutor\appdata\roaming\python\python39\python.exe
Note: you may need to restart the kernel to use updated packages.

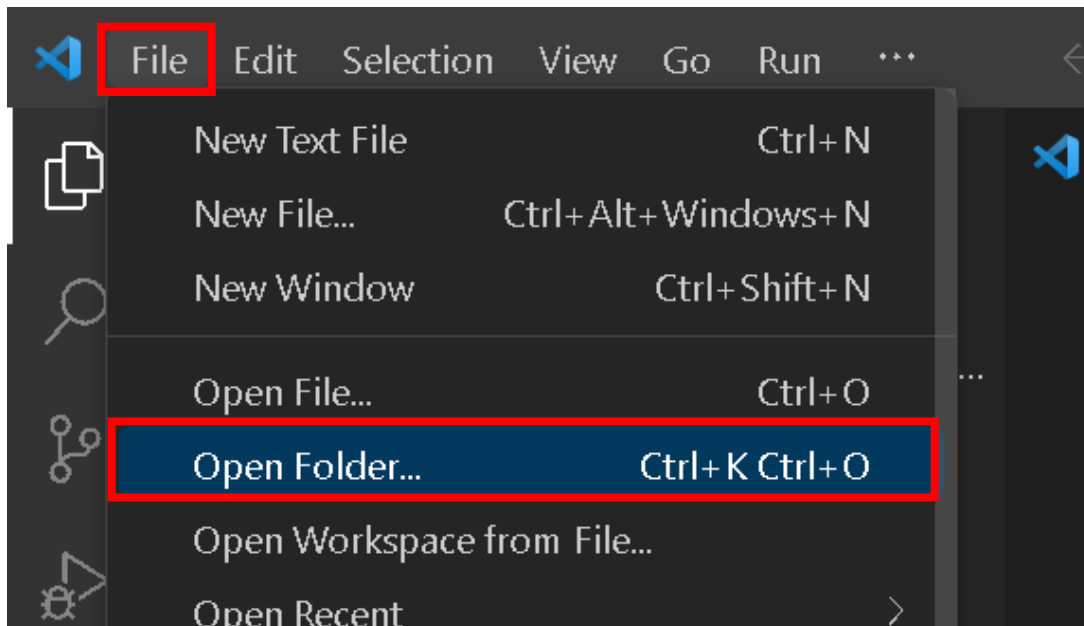
[notice] A new release of pip is available: 23.1.2 -> 23.3.1
[notice] To update, run: python.exe -m pip install --upgrade pip
    
```

Perfeito, esta tudo certo, agora vmaos preparar nosso ambiente. Salve o arquivo pandas que voce esta utilizando na pasta baixada disponibilizada pelo professor.





Seguindo o passo a passa abaixo iremos criar uma ligação entre o VS code e a pasta que iremos utilizar na atividade.



Após clicar em “open folder” basta selecionar a pasta com os arquivos que vamos utilizar.

4 UTILIZANDO PANDAS

Agora que já temos o ambiente pronto vamos começar a testar a biblioteca, primeiramente vamos tentar criar um data frame com informações digitadas manualmente.

Para isso digite o código disponível abaixo ou utilize do recurso de copiar e colar o mesmo.



```
import pandas as pd
df2 = pd.DataFrame(
    {
        "A": 1.0,
        "B": 5.9,
        "C": [25,26,59,26],
        "D": 5.6,
        "E": 2.0,
        "F": [3,56,36,8],
    }
)
print(df2)
```

Ao inserir este código na celular e o executar o resultado esperado é o seguinte:



```
import pandas as pd
df2 = pd.DataFrame(
    {
        "A": 1.0,
        "B": 5.9,
        "C": [25, 26, 59, 26],
        "D": 5.6,
        "E": 2.0,
        "F": [3, 56, 36, 8],
    }
)

print(df2)
```

[1] ✓ 3.5s

	A	B	C	D	E	F
0	1.0	5.9	25	5.6	2.0	3
1	1.0	5.9	26	5.6	2.0	56
2	1.0	5.9	59	5.6	2.0	36
3	1.0	5.9	26	5.6	2.0	8

Caso a saída seja igual a da imagem acima quer dizer que está tudo funcionando corretamente e já podemos começar a entender melhor este código.

Caso algum erro ocorra, **leia** a mensagem de erro para resolvê-lo.

Vamos entender um pouco melhor este código.

```
# Importa a biblioteca Pandas e a renomeia como "pd" para facilitar o uso.
import pandas as pd

# Cria um DataFrame chamado df2 com várias colunas de dados.
df2 = pd.DataFrame(
    {
        "A": 1.0,          # Coluna A contendo valores numéricos.
```




```
"B": 5.9,          # Coluna B contendo valores numéricos.
"C": [25, 26, 59, 26], # Coluna C contendo uma lista de números
"D": 5.6,          # Coluna D contendo valores numéricos.
"E": 2.0,          # Coluna E contendo valores numéricos.
"F": [3, 56, 36, 8]  # Coluna F contendo uma lista de números
}
)

# Imprime o DataFrame df2 na saída padrão.
print(df2)
```

5 LER ARQUIVOS EXCEL COM O PANDAS

Utilizando da documentação oficial e suas habilidades adquiridas de pesquisa exploratória, encontre a melhor solução para associar um arquivo Excel a um data frame pandas.

Para realizar os testes utilize o arquivo “Vendas.xlsx”



```
df = pd.read_excel("Vendas.xlsx")
```

[7] ✓ 0.4s

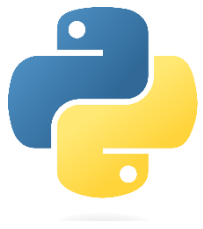
▶ df

[8] ✓ 0.0s

...

	SKU	Dia	Mês	Ano	Tipo	Marca	Faturamento	Lucro
0	SKU 353	2021-01-01	1	2021	Camisa	Reserva	208.955	94.080
1	SKU 563	2021-01-02	1	2021	Camisa	Reserva	231.035	171.696
2	SKU 384	2021-01-01	1	2021	Casaco	CalopStore	62.215	12.880
3	SKU 661	2021-01-04	1	2021	Camiseta	MarcoGPT	45.195	10.080
4	SKU 353	2021-01-01	1	2021	Sandália	CalopStore	140.645	93.072
...
368	SKU 466	2021-12-27	12	2021	Chinelo	MarcoGPT	17.710	7.616
369	SKU 681	2021-12-26	12	2021	Sapato	MarcoGPT	9.545	1.904
370	SKU 256	2021-12-27	12	2021	Camiseta	MarcoGPT	147.085	26.096
371	SKU 757	2021-12-26	12	2021	Casaco	Reserva	307.165	158.032
372	SKU 367	2021-12-27	12	2021	Camisa	CalopStore	94.875	27.328

373 rows × 8 columns



6 EXPLORAÇÃO INICIAL DE DADOS

Comandos básicos:

```
vendas.head()
```

[14] ✓ 0.0s

	SKU	Dia	Mês	Ano	Tipo	Marca	Faturamento	Lucro
0	SKU 353	2021-01-01	1	2021	Camisa	Reserva	208.955	94.080
1	SKU 563	2021-01-02	1	2021	Camisa	Reserva	231.035	171.696
2	SKU 384	2021-01-01	1	2021	Casaco	CalopStore	62.215	12.880
3	SKU 661	2021-01-04	1	2021	Camiseta	MarcoGPT	45.195	10.080
4	SKU 353	2021-01-01	1	2021	Sandália	CalopStore	140.645	93.072

Para exibir as primeiras cinco linhas do DataFrame. Isso dá uma visão inicial de como os dados estão estruturados.

```
vendas.info()
```

[15] ✓ 0.0s

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 373 entries, 0 to 372
Data columns (total 8 columns):
#   Column          Non-Null Count  Dtype
---  ---
0   SKU              373 non-null    object
1   Dia              373 non-null    datetime64[ns]
2   Mês              373 non-null    int64
3   Ano              373 non-null    int64
4   Tipo             373 non-null    object
5   Marca            373 non-null    object
6   Faturamento     373 non-null    float64
7   Lucro            373 non-null    float64
dtypes: datetime64[ns](1), float64(2), int64(2), object(3)
memory usage: 23.4+ KB
```

Para obter um resumo conciso do DataFrame. Esta função mostra o número de entradas, a quantidade de colunas, os tipos de dados de cada coluna, e a quantidade de valores não nulos.



```
vendas.describe()
```

[16] ✓ 0.0s

	Dia	Mês	Ano	Faturamento	Lucro
count	373	373.000000	373.0	373.000000	373.000000
mean	2021-06-30 06:10:36.997318912	6.506702	2021.0	90.933244	38.377866
min	2021-01-01 00:00:00	1.000000	2021.0	5.520000	1.120000
25%	2021-04-04 00:00:00	4.000000	2021.0	30.245000	7.168000
50%	2021-06-23 00:00:00	6.000000	2021.0	59.455000	16.576000
75%	2021-09-15 00:00:00	9.000000	2021.0	130.525000	54.432000
max	2021-12-27 00:00:00	12.000000	2021.0	427.915000	200.256000
std	NaN	3.412199	0.0	82.270208	45.633262

Para visualizar estatísticas descritivas que resumem a tendência central, dispersão e forma da distribuição dos dados numéricos.

7 MANIPULAÇÃO BÁSICA DE DADOS

```
# Selecionando as colunas 'SKU' e 'Faturamento'
vendas_selecionadas = vendas[['SKU', 'Faturamento']]
print(vendas_selecionadas.head())
```

[17] ✓ 0.0s

	SKU	Faturamento
0	SKU 353	208.955
1	SKU 563	231.035
2	SKU 384	62.215
3	SKU 661	45.195
4	SKU 353	140.645

Selecionar colunas específicas de um DataFrame. Isso é útil para simplificar a visualização de dados ou para preparar os dados para análises específicas.



```
# Filtrando vendas da 'MarcaX'
vendas_marca_x = vendas[vendas['Marca'] == 'Reserva']
print(vendas_marca_x.head())
```

[19] ✓ 0.0s

...	SKU	Dia	Mês	Ano	Tipo	Marca	Faturamento	Lucro
0	SKU 353	2021-01-01	1	2021	Camisa	Reserva	208.955	94.080
1	SKU 563	2021-01-02	1	2021	Camisa	Reserva	231.035	171.696
9	SKU 679	2021-01-10	1	2021	Casaco	Reserva	263.350	121.968
12	SKU 699	2021-01-13	1	2021	Camisa	Reserva	191.360	110.768
15	SKU 195	2021-01-14	1	2021	Casaco	Reserva	148.465	87.248

Filtrar linhas de um DataFrame com base em condições específicas. Por exemplo, selecionar todas as vendas de uma marca específica ou de um determinado ano.

Qual foi o faturamento total da 'Reserva' no ano de 2020?