

## Data Science – Princípios e Técnicas

Agosto  
2024



# Data Science



Onde me encontrar:

<https://www.linkedin.com/in/marco-mialaret-junior/>

e

<https://github.com/MatmJr>

# Dicionários em Python

# Data Science

---

Dicionários são a implementação do Python de uma estrutura de dados conhecida mais genericamente como um array associativo. Um dicionário consiste em uma coleção de pares de chave-valor. Cada par de chave-valor mapeia a chave ao seu valor associado.

# Data Science

---

Você pode definir um dicionário colocando uma lista de pares de chave-valor separados por vírgulas entre chaves ({}). Dois pontos (:) separam cada chave de seu valor associado:

# Data Science

---

```
d = {  
    <chave>: <valor>,  
    <chave>: <valor>,  
    .  
    .  
    .  
    <chave>: <valor>  
}
```

```
timesMascotes = {  
    'Sport' : 'Leão',  
    'Náutico' : 'Timbu',  
    'Santa Cruz' : 'Cobra Coral'  
}
```

# Data Science

---

Um valor é recuperado de um dicionário especificando sua chave correspondente entre colchetes '[]', por exemplo:

```
timesMascotes['Sport']
```

# Data Science

---

Um valor é recuperado de um dicionário especificando sua chave correspondente entre colchetes '[]', por exemplo:

```
timesMascotes['Sport']
```



# Data Science

---

Para adicionar uma nova chave e um respectivo valor você usar o padrão `variável['nova_chave'] = 'novo_valor'`. Exemplo:

```
timesMascotes['Palmeiras'] = 'Porco'
```

# Aplicação Prática

# Data Science

---

Sistemas de cadastros são comumente utilizados para armazenar e gerenciar informações estruturadas de pessoas, produtos, ou qualquer outro tipo de entidade relevante para uma organização.

# Data Science

---

Em termos de programação, uma lista de dicionários como a que criamos anteriormente pode ser vista como uma representação básica de um sistema de cadastro. Cada dicionário atua como um registro ou entrada no cadastro, armazenando dados de uma entidade específica, como uma pessoa.

# Data Science

---

```
cadastros = []
```

```
def adicionar_cadastro(nome, idade, profissao):  
    cadastros.append({"nome": nome, "idade": idade, "profissão":  
profissao})
```

# Data Science

---

```
adicionar_cadastro("Ana", 28, "Engenheira")  
adicionar_cadastro("Carlos", 35, "Professor")
```

# Data Science

---

```
def consultar_cadastro(nome):  
    for cadastro in cadastros:  
        if cadastro["nome"] == nome:  
            return cadastro  
    return "Cadastro não encontrado."
```

# Carregando arquivos



# Data Science

---

## Carregando arquivos

Para abrir um arquivo de texto em Python e ler seu conteúdo, você pode usar a declaração `with` junto com a função `open()`. A vantagem de usar `with` é que ele garante que o arquivo seja fechado automaticamente após o bloco de código dentro dele ser executado, mesmo se ocorrer um erro.

# Data Science

---

**Isso é chamado de gerenciamento de contexto e é uma prática recomendada para trabalhar com arquivos, pois ajuda a prevenir erros, como esquecer de fechar um arquivo.**

# Data Science

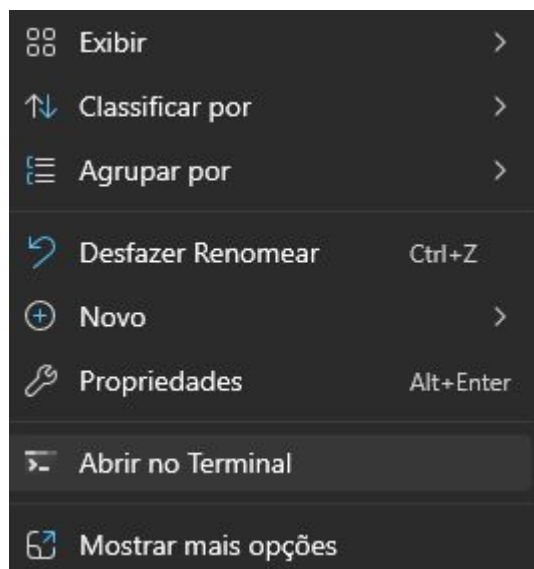
---

```
with open('caminho_para_o_arquivo.txt', 'r') as  
arquivo:  
    conteudo = arquivo.read()  
    print(conteudo)
```

# Criação do Ambiente de Trabalho

# Big Data

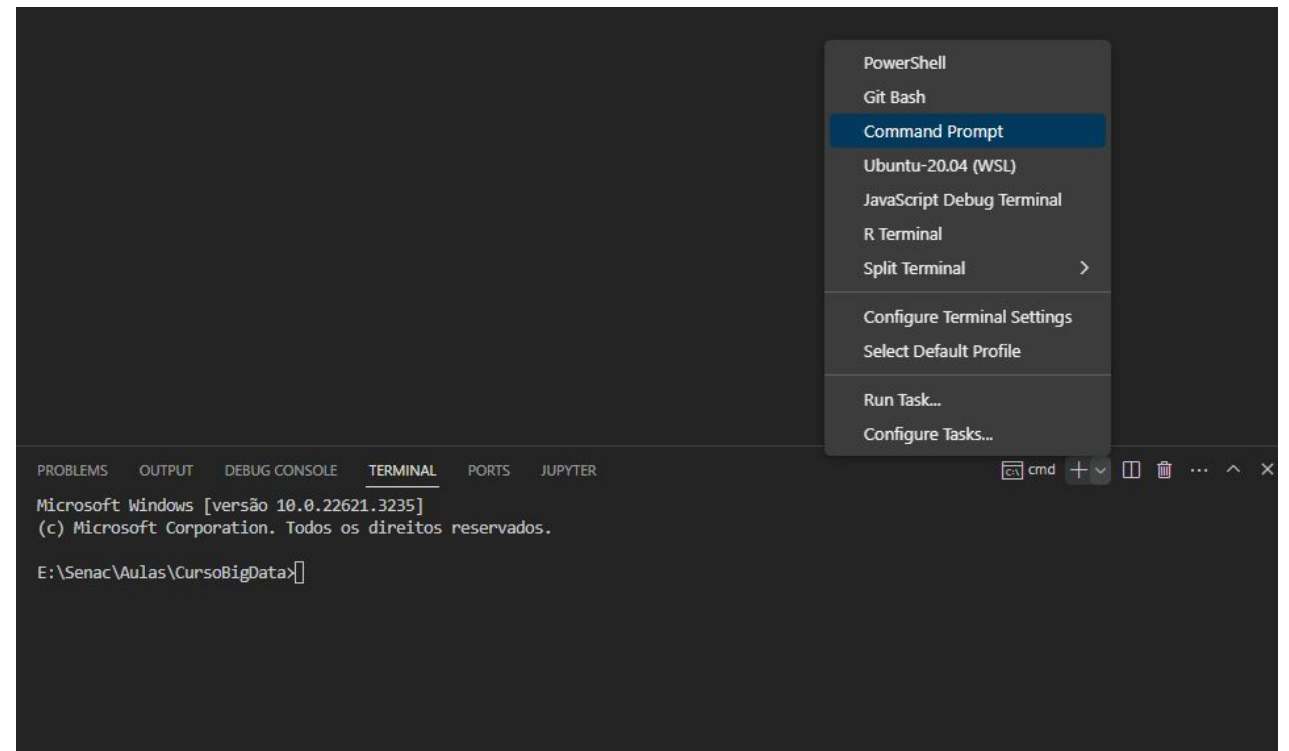
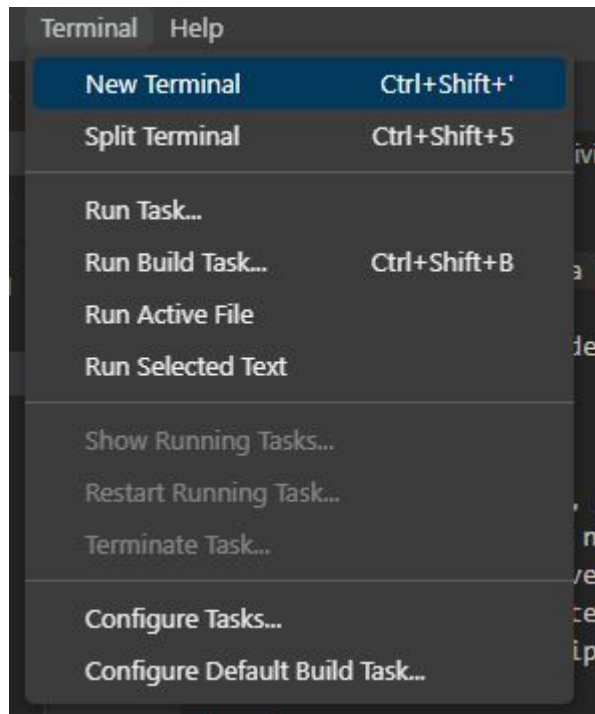
Após criar a pasta chamada BigData, clicar com o botão do lado direito e selecionar a opção abrir no terminal



```
regamento de perfis pessoais e do sistema levou 729ms.  
PS E:\Senac\Aulas\Slides\BigData\BigData> |code .
```

# Big Data

No VSCode:



# Big Data

---

Criando o ambiente:

```
E:\Senac\Aulas\CursoBigData>python -m venv venv
```

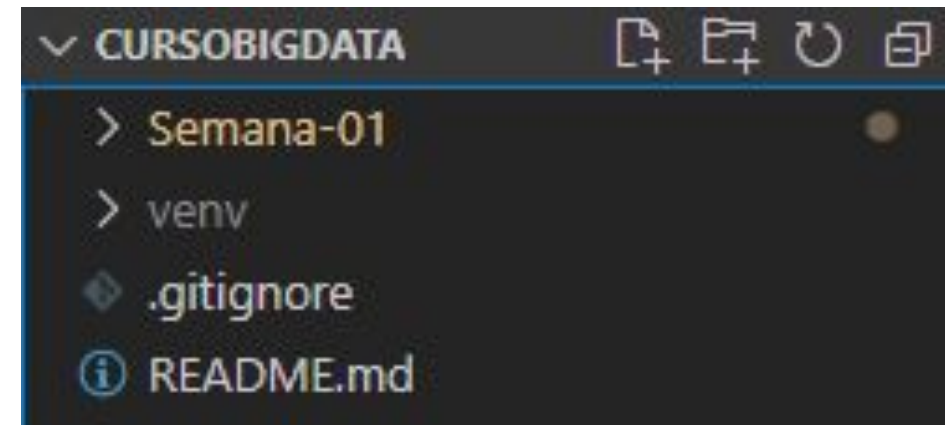
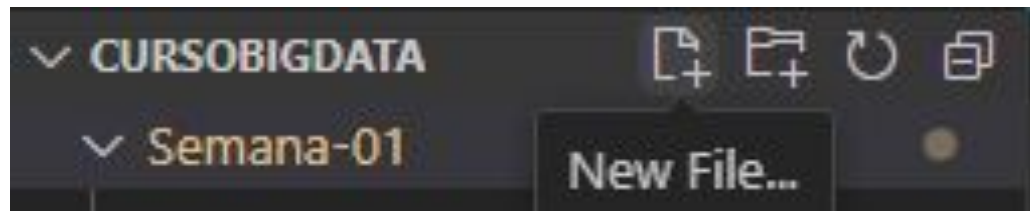
Ativando o ambiente:

```
E:\Senac\Aulas\CursoBigData>.\venv\Scripts\activate
```

```
(venv) E:\Senac\Aulas\CursoBigData>|
```

# Big Data

Criação de arquivos no VSCode:





# Big Data



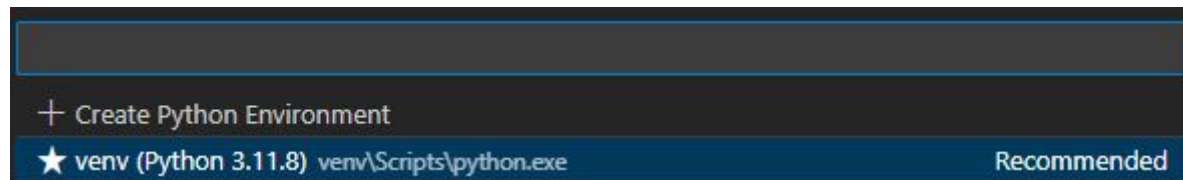
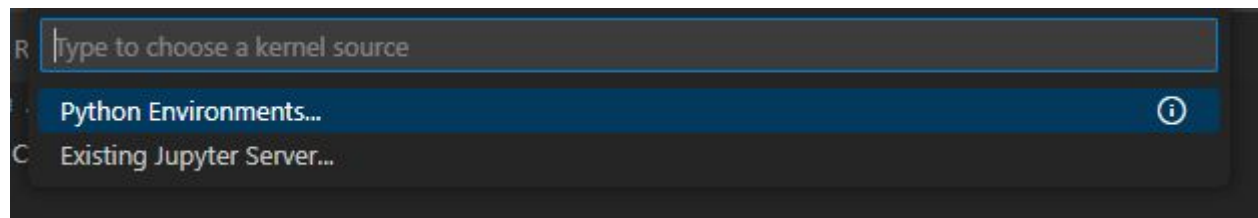
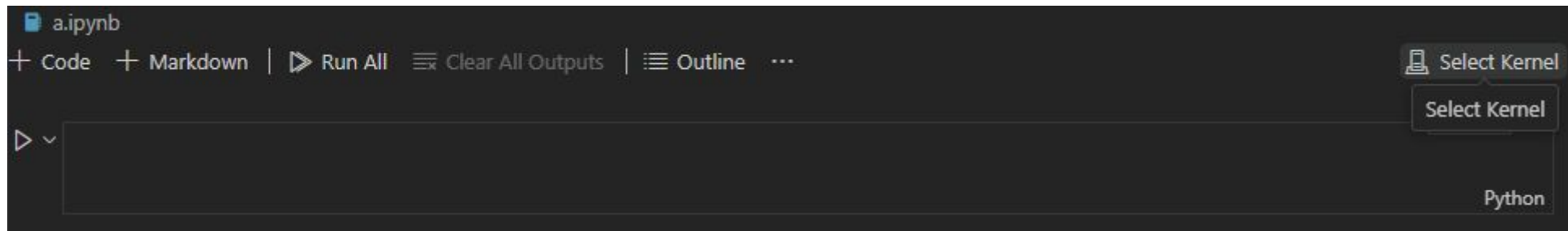
Crie um arquivo chamado aula01.ipynb.

No terminal, instale o pandas no ambiente virtual:

```
(venv) E:\Senac\Aulas\CursoBigData>pip install pandas
Collecting pandas
  Using cached pandas-2.2.1-cp311-cp311-win_amd64.whl.metadata (19 kB)
Collecting numpy<2,>=1.23.2 (from pandas)
  Using cached numpy-1.26.4-cp311-cp311-win_amd64.whl.metadata (61 kB)
```

# Big Data

## Ativando o ambiente virtual no notebook Jupyter



# Dúvidas?

---



**Marco Mialaret, MSc**

**Telefone:**

**81 98160 7018**

**E-mail:**

**marcomialaret@gmail.com**

