

**Educação
Profissional
Paulista**

Técnico em
**Ciência de
Dados**

Manipulação de arquivos

Escrita de arquivos em Python

Aula 2

Código da aula: [DADOS]ANO1C2B3S17A2

**Programação
aplicada à Ciência
de Dados**

Mapa da unidade 4 Componente 3

semana

16

Leitura de arquivos
em Python

semana

17

Você está aqui!

Escrita de arquivos
em Python

**Visualização
de dados**

**Mapa
da unidade 3
Componente 4**

Você está aqui!

Escrita de arquivos em Python

17

Aula 2

Código da aula: [DADOS]ANO1C2B3S17A2



Objetivos da aula

- Conhecer as várias formas de importar bibliotecas.



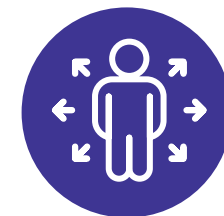
Recursos didáticos

- Recurso audiovisual para exibição de vídeos e imagens.
- Acesso ao laboratório de informática e/ou à internet.
- Software Anaconda/Jupyter Notebook instalado ou similar.



Duração da aula

50 minutos



Competências técnicas

- Ser proficiente em linguagens de programação para manipular e analisar grandes conjuntos de dados.
- Usar técnicas para explorar e analisar dados, aplicar modelos estatísticos, identificar padrões, realizar inferências e tomar decisões com base em evidências.



Competências socioemocionais

- Colaborar efetivamente com outros profissionais, como cientistas de dados e engenheiros de dados; trabalhar em equipes multifuncionais, colaborando com colegas, gestores e clientes.

Ponto de
partida

Como extrair dados de um site a partir do **web scraping**?

Alunos aprovados

| Número da inscrição | Nome | Email |
|---------------------|--------------------------------------|--------------------------------|
| 17236 | Margarida Were Vomsirnjol | tanderson@yahoo.com |
| 99746 | Guimar Sanches Valverde Aoson | taylorbarron@hotmail.com |
| 11123 | Rose Jéssica Hedo Ecemson | uoneal@hotmail.com |
| 74376 | Gigi Melanie Migoya Kornfirnblosi | marshrichard@hotmail.com |
| 29756 | Elisângela Tainá Cidunu de Bragança | madsionowens@gmail.com |
| 15562 | Caio Itamar Pinhão | joshua33@hotmail.com |
| 29178 | Max de Alencar Naves | carrteresa@yahoo.com |
| 50741 | Andréia Lina de França | ujohnson@gmail.com |
| 96397 | Luiz Mário de Fonseca | jasonwilliams@gmail.com |
| 10869 | Rodolfo Wesley de Guimarães Júnior | christine18@yahoo.com |
| 95556 | Dionísio de Ayres dos Santos | tracyjackson@hotmail.com |
| 59455 | Omar Benedito Tufauzov Gatetani | lisa42@gmail.com |
| 60777 | Rita Genam | christopherrobbins@hotmail.com |
| 91002 | Reginaldo de Paiva Teles | paige43@yahoo.com |
| 70587 | Aparecida Daiane da Silva de Moreira | cgarcia@yahoo.com |
| 83366 | Juan Eduardo Emmi Banhos de Souza | richard21@hotmail.com |
| 93213 | Jackson Baltazar Ximenes Neto | teresa38@yahoo.com |

The screenshot shows a Google Sheet titled "site_vestibular" with columns labeled "Número da inscrição", "Nome", and "Email". The data rows correspond exactly to the table provided.

Elaborado especialmente para o curso.

Construindo
o **conceito**

Como extrair dados da internet?

Já vimos que, com o
web scraping, é possível
extrair dados da internet.

Como fazer isso no Python?

Veja as linhas 1 e 2 na
imagem ao lado.
O que têm de diferente?

```
import requests
from bs4 import BeautifulSoup

url = "https://sites.google.com/view/vestibular123"

response = requests.get(url)
content = response.content

soup = BeautifulSoup(content, "html.parser")
meta_tags = soup.find_all('meta', {'property': 'og:description'})

texto_salvar = meta_tags[0]['content']

print(texto_salvar)
```

| Número da inscrição | Nome |
|--------------------------|--------------------------------|
| Email | |
| 17236 | Margarida Were Vomsirnjol |
| tanderson@yahoo.com | |
| 99746 | Guiomar Sanches Valverde Aason |
| taylorbarron@hotmail.com | |
| 11123 | Rose Jéssica Hedó Ecemson |
| uoneal@hotmail.com | |
| 74376 | Gigi Melanie Migoya |

Elaborado especialmente para o curso com a ferramenta Jupyter Notebook.

Construindo o conceito

Importar uma biblioteca

Em Python, as bibliotecas são módulos reutilizáveis que contêm funções, classes e métodos que podem ser importados em seus programas para fornecer funcionalidades adicionais.

1. Importação simples

Para importar uma biblioteca inteira, você pode usar a palavra-chave `import` seguida do nome da biblioteca: `import math`.

```
1 import math
2
3 # Calculando a raiz quadrada de 25
4 raiz_quadrada = math.sqrt(25)
5
6 print("A raiz quadrada de 25 é:", raiz_quadrada)
```

A raiz quadrada de 25 é: 5.0

Nesse exemplo, é exibido como importar a biblioteca **math**, que fornece funções matemáticas.

Elaborado especialmente para o curso com a ferramenta Jupyter Notebook.

Construindo
o **conceito**

Importar uma biblioteca

Importação simples de bibliotecas comuns em Ciência de Dados

```
1 # Importando a biblioteca pandas para manipulação de dados tabulares
2 import pandas
3
4 # Importando a biblioteca numpy para manipulação de arrays e operações matemáticas
5 import numpy
6
7 # Importando a biblioteca matplotlib para visualização de dados em gráficos
8 import matplotlib
9
10 # Importando a biblioteca seaborn para visualização de dados estatísticos em gráficos mais avançados
11 import seaborn
12
13 # Importando a biblioteca scipy para funções matemáticas e estatísticas mais avançadas
14 import scipy
15
16 # Importando a biblioteca sklearn (scikit-learn) para aprendizado de máquina e mineração de dados
17 import sklearn
18
```

Elaborado especialmente para o curso com a ferramenta Jupyter Notebook.

Construindo o conceito

Importar uma biblioteca

Exemplos de uso de importação simples

```
1 import pandas
2
3 # Criando um DataFrame simples
4 data = {'Nome': ['Alice', 'Bob', 'Charlie'],
5         'Idade': [25, 30, 35]}
6 df = pandas.DataFrame(data)
7
8 df
```

| | Nome | Idade |
|---|---------|-------|
| 0 | Alice | 25 |
| 1 | Bob | 30 |
| 2 | Charlie | 35 |

```
1 import numpy
2
3 # Criando um array NumPy
4 arr = numpy.array([1, 2, 3, 4, 5])
5
6 arr
```

array([1, 2, 3, 4, 5])

Elaborado especialmente para o curso com a ferramenta Jupyter Notebook.

Construindo o conceito

Importar uma biblioteca

Exemplos de uso de importação simples

```
1 import sklearn.linear_model
2
3 # Criando um objeto de regressão linear
4 regression_model = sklearn.linear_model.LinearRegression()
5
6 print("Objeto de regressão linear criado:", regression_model)
```

Objeto de regressão linear criado: LinearRegression()

```
1 import scipy
2
3 # Calculando o fatorial de 5
4 fatorial = scipy.special.factorial(5)
5
6 print("Fatorial de 5 é:", fatorial)
```

Fatorial de 5 é: 120.0

Elaborado especialmente para o curso com a ferramenta Jupyter Notebook

Construindo o conceito

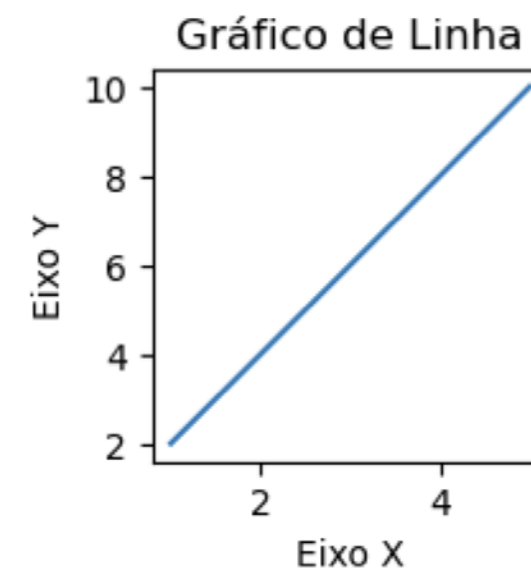
Importar uma biblioteca

Exemplos de uso de importação simples

```
1 import seaborn
2 import numpy
3 import matplotlib
4
5 # Dados
6 x = numpy.random.normal(size=100)
7 y = numpy.random.normal(size=100)
8
9 # Plotando o gráfico de dispersão
10 matplotlib.pyplot.figure(figsize=(2,2))
11 seaborn.scatterplot(x=x, y=y)
12 matplotlib.pyplot.xlabel('X')
13 matplotlib.pyplot.ylabel('Y')
14 matplotlib.pyplot.title('Gráfico de Dispersão')
15 matplotlib.pyplot.show()
```



```
1 import matplotlib
2
3 # Dados
4 x = [1, 2, 3, 4, 5]
5 y = [2, 4, 6, 8, 10]
6
7 # Plotando o gráfico de linha
8 matplotlib.pyplot.figure(figsize=(2,2))
9 matplotlib.pyplot.plot(x, y)
10 matplotlib.pyplot.xlabel('Eixo X')
11 matplotlib.pyplot.ylabel('Eixo Y')
12 matplotlib.pyplot.title('Gráfico de Linha')
13 matplotlib.pyplot.show()
```



Elaborado especialmente para o curso com a ferramenta Jupyter Notebook.

Construindo o conceito

Importar uma biblioteca – Renomeando

2. Renomeando bibliotecas

Você pode renomear bibliotecas ao importá-las, o que é útil para reduzir a digitação ou para evitar conflitos de nomes:

```
import math as m
```

Agora, você pode usar a biblioteca `math` referindo-se a ela como `m`.

Compare com a importação simples.

```
1 import math as m
2
3 # Calculando a raiz quadrada de 25
4 raiz_quadrada = m.sqrt(25)
5
6 print("A raiz quadrada de 25 é:", raiz_quadrada)
```

A raiz quadrada de 25 é: 5.0

```
1 import math
2
3 # Calculando a raiz quadrada de 25
4 raiz_quadrada = math.sqrt(25)
5
6 print("A raiz quadrada de 25 é:", raiz_quadrada)
```

A raiz quadrada de 25 é: 5.0

Elaborado especialmente para o curso com a ferramenta Jupyter Notebook.

Construindo o conceito

Importar uma biblioteca – Renomeando

Podemos renomear com qualquer nome, porém existem algumas convenções que podem ajudar.

Não estranhe essa notação, pois vamos entendê-la melhor ao longo do curso.

Convencionalmente, não renomeamos a biblioteca sklearn.

```
1 # Importando a biblioteca pandas para manipulação de dados tabulares
2 import pandas as pd
3
4 # Importando a biblioteca numpy para manipulação de arrays e operações matemáticas
5 import numpy as np
6
7 # Importando a biblioteca matplotlib para visualização de dados em gráficos
8 import matplotlib.pyplot as plt
9
10 # Importando a biblioteca seaborn para visualização de dados estatísticos em gráficos mais avançados
11 import seaborn as sns
12
13 # Importando a biblioteca scipy para funções matemáticas e estatísticas mais avançadas
14 import scipy as sp
15
16 # Importando a biblioteca sklearn (scikit-learn) para aprendizado de máquina e mineração de dados
17 import sklearn as qualquernome
```

Nesse código, cada biblioteca foi **renomeada conforme as convenções comuns de alias** utilizadas na comunidade Python.

Isso torna o código mais legível e fácil de entender, especialmente quando você está lidando com várias bibliotecas em um único script.

Elaborado especialmente para o curso com a ferramenta Jupyter Notebook.

Construindo o conceito

Importar uma biblioteca – Renomeando

Compare com a importação simples e diga qual é melhor.

```
1 import pandas as pd
2
3 # Criando um DataFrame simples
4 data = {'Nome': ['Alice', 'Bob', 'Charlie'],
5         'Idade': [25, 30, 35]}
6 df = pd.DataFrame(data)
7
8 df
```

| | Nome | Idade |
|---|---------|-------|
| 0 | Alice | 25 |
| 1 | Bob | 30 |
| 2 | Charlie | 35 |

```
1 import numpy as np
2
3 # Criando um array NumPy
4 arr = np.array([1, 2, 3, 4, 5])
5
6 arr
```

array([1, 2, 3, 4, 5])

```
1 import scipy as sp
2
3 # Calculando o fatorial de 5
4 fatorial = sp.special.factorial(5)
5
6 print("Fatorial de 5 é:", fatorial)
```

Fatorial de 5 é: 120.0

Elaborado especialmente para o curso com a ferramenta Jupyter Notebook.

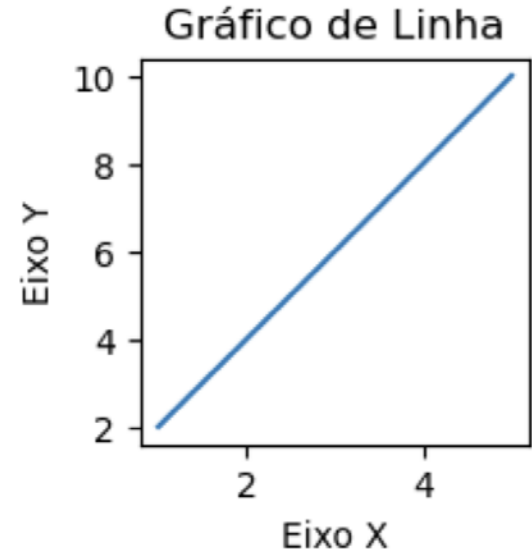


Construindo
o **conceito**

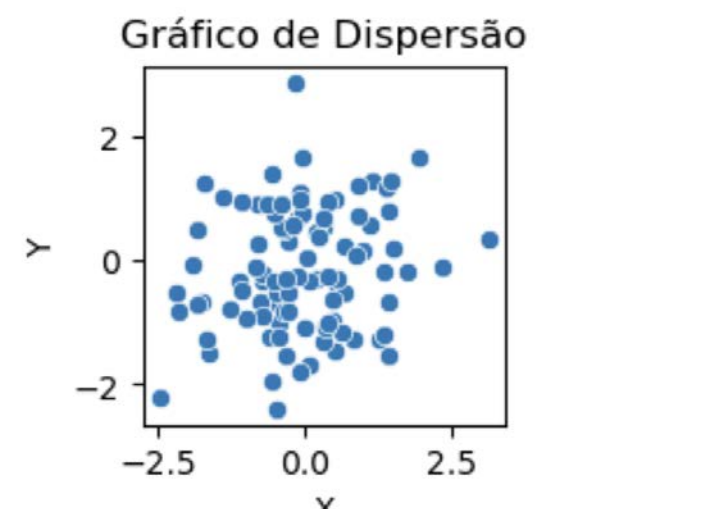
Importar uma biblioteca – Renomeando

Compare com a importação simples e diga qual é melhor.

```
1 import matplotlib.pyplot as plt
2
3 # Dados
4 x = [1, 2, 3, 4, 5]
5 y = [2, 4, 6, 8, 10]
6
7 # Plotando o gráfico de linha
8 plt.figure(figsize=(2,2))
9 plt.plot(x, y)
10 plt.xlabel('Eixo X')
11 plt.ylabel('Eixo Y')
12 plt.title('Gráfico de Linha')
13 plt.show()
```



```
1 import seaborn as sns
2 import numpy as np
3 import matplotlib.pyplot as plt
4
5 # Dados
6 x = np.random.normal(size=100)
7 y = np.random.normal(size=100)
8
9 # Plotando o gráfico de dispersão
10 plt.figure(figsize=(2,2))
11 sns.scatterplot(x=x, y=y)
12 plt.xlabel('X')
13 plt.ylabel('Y')
14 plt.title('Gráfico de Dispersão')
15 plt.show()
```



Elaborado especialmente para o curso com a ferramenta Jupyter Notebook.

Construindo o conceito

Importar parte de uma biblioteca

3. Importar parte da biblioteca

Você também pode importar apenas partes específicas de uma biblioteca:

```
from math import sqrt
```

Dessa forma, importa-se apenas a função `sqrt` da biblioteca `math`, permitindo que você use `sqrt()` diretamente sem prefixá-la com `math`.

Para um código limpo, importar apenas a parte da biblioteca que se vai usar faz muita diferença.

```
1 from math import sqrt
2
3 # Calculando a raiz quadrada de 25
4 raiz_quadrada = sqrt(25)
5
6 print("A raiz quadrada de 25 é:", raiz_quadrada)
```

A raiz quadrada de 25 é: 5.0

Elaborado especialmente para o curso com a ferramenta Jupyter Notebook.

Construindo
o **conceito**

Importar parte de uma biblioteca

Alguns exemplos:

```
1 # Importando apenas a função sqrt da biblioteca math
2 from math import sqrt
3
4 # Importando apenas a função array da biblioteca numpy
5 from numpy import array
6
7 # Importando apenas a função scatterplot da biblioteca seaborn
8 from seaborn import scatterplot
9
10 # Importando apenas a função factorial da biblioteca scipy
11 from scipy.special import factorial
12
13 # Importando apenas o módulo LinearRegression da biblioteca sklearn
14 from sklearn.linear_model import LinearRegression
```

Elaborado especialmente para o curso com a ferramenta Jupyter Notebook.

Construindo o conceito

Importar parte de uma biblioteca

```
1 from pandas import DataFrame
2
3 # Criando um DataFrame simples
4 data = {'Nome': ['Alice', 'Bob', 'Charlie'],
5         'Idade': [25, 30, 35]}
6 df = DataFrame(data)
7
8 df
```

| | Nome | Idade |
|---|---------|-------|
| 0 | Alice | 25 |
| 1 | Bob | 30 |
| 2 | Charlie | 35 |

```
1 from scipy.special import factorial
2
3 # Calculando o fatorial de 5
4 fatorial = factorial(5)
5
6 print("Fatorial de 5 é:", fatorial)
```

Fatorial de 5 é: 120.0

```
1 from numpy import array
2
3 # Criando um array NumPy
4 arr = array([1, 2, 3, 4, 5])
5
6 arr
```

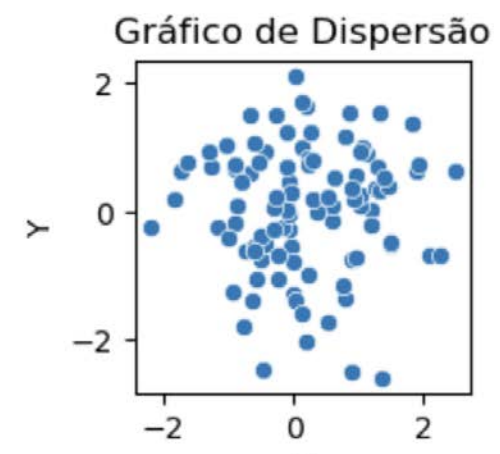
array([1, 2, 3, 4, 5])

Elaborado especialmente para o curso com a ferramenta Jupyter Notebook.

Construindo
o **conceito**

Importar parte de uma biblioteca

```
1 # Importando apenas a função scatterplot da biblioteca seaborn
2 from seaborn import scatterplot
3
4 # Importando apenas a função normal da biblioteca numpy
5 from numpy.random import normal
6
7 # Importando apenas as funções show e figure da biblioteca matplotlib.pyplot
8 from matplotlib.pyplot import show, figure, xlabel, ylabel, title
9
10 # Dados
11 x = normal(size=100)
12 y = normal(size=100)
13
14 # Plotando o gráfico de dispersão
15 figure(figsize=(2,2))
16 scatterplot(x=x, y=y)
17 xlabel('X') # Usando xlabel diretamente
18 ylabel('Y') # Usando ylabel diretamente
19 title('Gráfico de Dispersão') # Usando title diretamente
20 show()
```

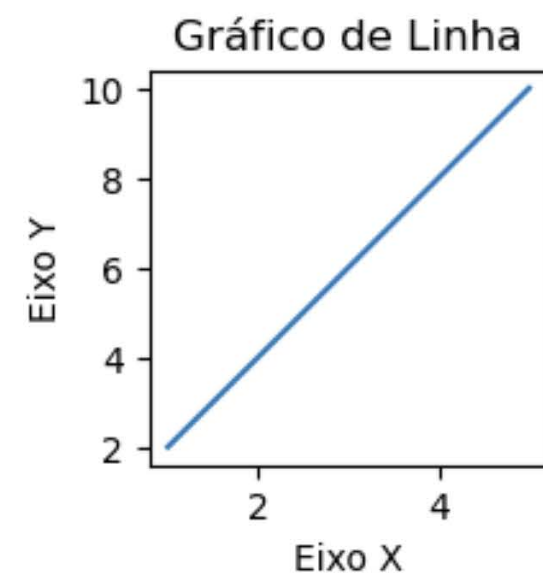


Elaborado especialmente para o curso com a ferramenta Jupyter Notebook.

Construindo o conceito

Importar parte de uma biblioteca

```
1 # Importando apenas a função plot da biblioteca matplotlib.pyplot
2 from matplotlib.pyplot import figure, plot, xlabel, ylabel, title, show
3
4 # Dados
5 x = [1, 2, 3, 4, 5]
6 y = [2, 4, 6, 8, 10]
7
8 # Plotando o gráfico de linha
9 figure(figsize=(2,2))
10 plot(x, y)
11 xlabel('Eixo X') # Não precisamos mais usar plt, pois xlabel foi importado diretamente
12 ylabel('Eixo Y') # Não precisamos mais usar plt, pois ylabel foi importado diretamente
13 title('Gráfico de Linha') # Não precisamos mais usar plt, pois title foi importado diretamente
14 show() # Não precisamos mais usar plt, pois show foi importado diretamente
```



Elaborado especialmente para o curso com a ferramenta Jupyter Notebook.

Construindo o **conceito**

Importar tudo de uma biblioteca

4. Importar tudo de uma biblioteca

Você pode importar todas as funções e todas as classes de uma biblioteca usando o caractere *. Assim, importará todas as funções e todas as classes da biblioteca math:

```
from math import *
```

No entanto, tenha cuidado ao usar este método, que pode poluir o espaço de nomes, causar conflitos de nomes e diminuir o desempenho de execução.



Tome nota

Em Python, `import math` importa o módulo math. Qualquer função que você utilize terá que ser usada como `math.sqrt`, `math.ceil` etc. Já `from math import *` importa todas as funções do módulo math. Qualquer função que você utilize pode ser usada como `sqrt`, `ceil` etc.

Lembrando que a segunda opção pode não ser a melhor prática de programação, pois pode importar muitas funções desnecessárias e levar a conflitos de nomes de funções.



Colocando em **prática**

Identificando as bibliotecas

```
1 import numpy as np
2 import pandas as pd
3 import matplotlib.pyplot as plt
4 import seaborn as sns
5 from sklearn import datasets
6 from sklearn.model_selection import train_test_split
7 from sklearn.metrics import accuracy_score
8 from sklearn.tree import DecisionTreeClassifier
9
10 # Carregando um conjunto de dados de exemplo
11 iris = datasets.load_iris()
12
13 # Separando os dados em um conjunto de treino e um conjunto de teste
14 X_train, X_test, y_train, y_test = train_test_split(iris.data, iris.target, test_size=0.2, random_state=42)
15
16 # Criando um modelo de Árvore de Decisão
17 model = DecisionTreeClassifier(max_depth=2)
18
19 # Treinando o modelo com os dados de treino
20 model.fit(X_train, y_train)
21
22 # Testando a acurácia do modelo com os dados de teste
23 y_pred = model.predict(X_test)
24 accuracy = accuracy_score(y_test, y_pred)
25
26 # Imprimindo a acurácia do modelo
27 print(f"A acurácia do modelo foi de {accuracy:.2f}.")
```



Documento de texto



10 minutos

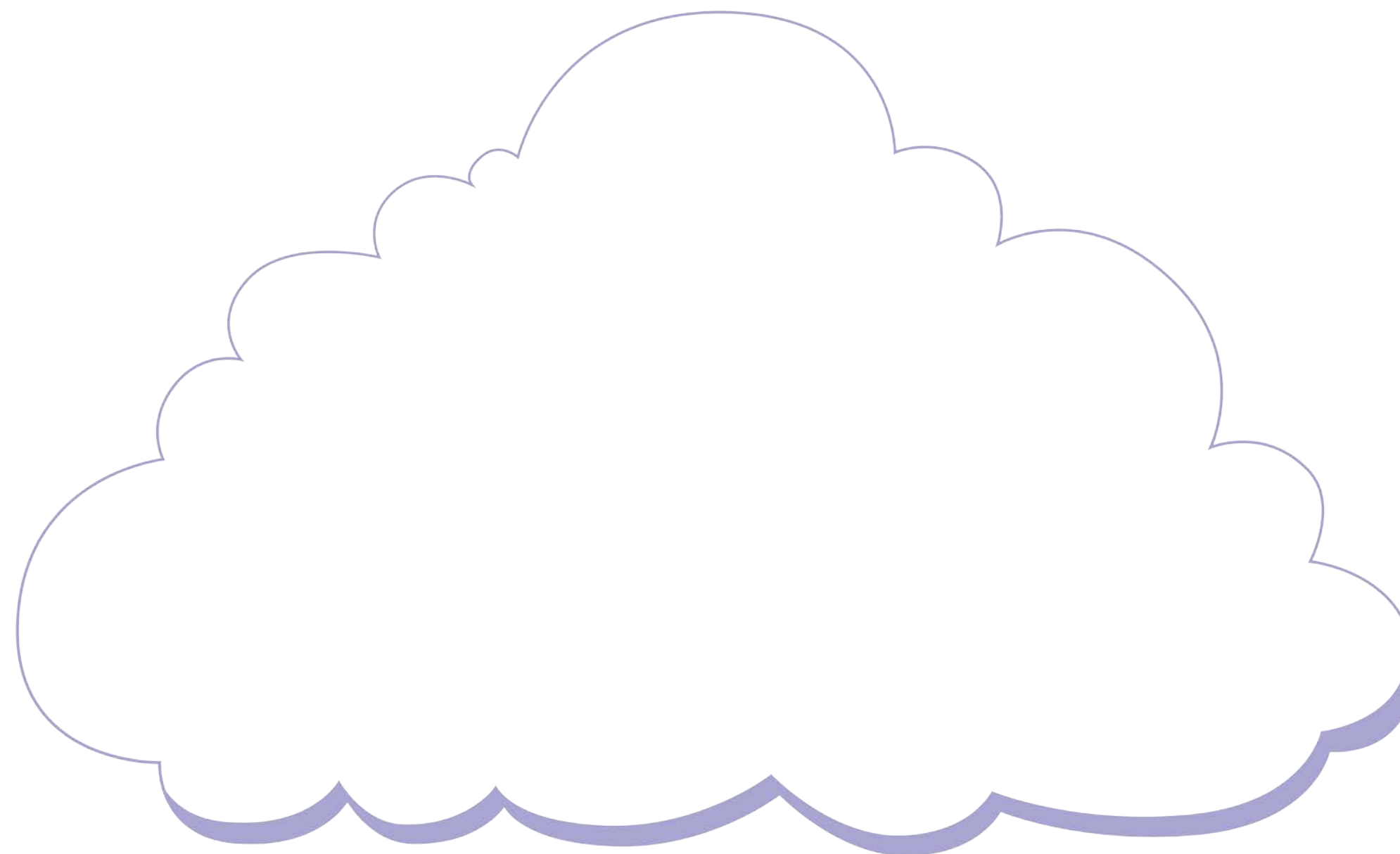


Em grupo de quatro
alunos

1. Analise o código a seguir, que é um exemplo real de um uso de aprendizado de máquina que classifica um conjunto de dados em três classes diferentes.
2. Elabore um documento, comentando cada linha e indicando o tipo de importe e como é usado.

Elaborado para o curso com a ferramenta Jupyter Notebook.

Nuvem de palavras



© Getty Images

O que nós
**aprendemos
hoje?**



© Getty Images

O que nós
**aprendemos
hoje?**

Então ficamos assim...

- 1** Existem várias formas de importar uma biblioteca e, de acordo com sua forma, o uso é ligeiramente diferente.
- 2** Vimos que, em Python, as bibliotecas são módulos reutilizáveis que contêm funções, classes e métodos que podem ser importados em seus programas para fornecer funcionalidades adicionais.
- 3** Vimos algumas bibliotecas usuais de Ciência de Dados.

Saiba mais

Quer saber mais sobre como ler bibliotecas e entender como elas funcionam?

Neste vídeo de 13 minutos do curso de bibliotecas da Alura, você vai ficar por dentro de tudo isso. Aproveite!

ALURA. *Python para Data Science*: trabalhando com funções, estruturas de dados e exceções. 03

Importando Bibliotecas. Disponível em:

<https://cursos.alura.com.br/course/python-data-science-funcoes-estruturas-dados-excecoes/task/125894>. Acesso em: 29 abr. 2024.

Referências da aula

MENEZES, N. N. C. *Introdução à programação com Python: algoritmos e lógica de programação para iniciantes*. São Paulo: Novatec, 2019.

PYTHON ORG. 6. *Módulos*. Disponível em: <https://docs.python.org/pt-br/3/tutorial/modules.html>. Acesso em: 29 abr. 2024.

PYTHON ORG. 7.11. *A instrução Import*. Disponível em: https://docs.python.org/pt-br/3/reference/simple_stmts.html#the-import-statement. Acesso em: 29 abr. 2024.

PYTHON ORG. 5. *O sistema de importação*. Disponível em: <https://docs.python.org/pt-br/3/reference/import.html#importsystem>. Acesso em: 29 abr. 2024.

Identidade visual: Imagens © Getty Images.

**Educação
Profissional
Paulista**

Técnico em
**Ciência de
Dados**