



AULA DE REVISÃO

MÓDULOS 1 E 2





Material produzido pela Daviny Letícia

dispensa apresentações



30
minutos

VAMOS A NOSSA REVISÃO

PLAY



Estruturas de dados

`['BANANA', 'LARANJA']`

coleção de elementos ordenada
e mutável

LISTA

`(1, 2, 3, 4, "DANIEL")`

coleção ordenada e imutável de
elementos

TUPLA

`{'YAN': '14', 'PEDRO': '99'}`

Um dicionário é uma coleção
não ordenada de elementos,
onde cada elemento é um par
chave-valor

DICIONÁRIO



Estrutura de repetição

Proximo capítulo



while (enquanto)

O while é uma estrutura de repetição que executa um bloco de código enquanto uma condição específica for verdadeira.

```
Contador = 0  
while contador < 5:  
    print(contador)  
    contador += 1
```

for (para)

O for é uma estrutura de repetição utilizada para iterar sobre uma sequência (como listas, tuplas, strings, dicionários, etc.) ou qualquer objeto iterável.

```
frutas = ['maçã', 'banana', 'laranja']  
for fruta in frutas:  
    print(fruta)
```

for com range()

O for com range() é uma forma específica do for que gera uma sequência de números. range() é uma função que retorna uma sequência imutável de números entre o início até um número especificado (não incluído).

```
for i in range(início, fim, passo):  
    # código a ser repetido para cada valor de i gerado por range()
```

```
for i in range(1, 6, 2):  
    print(i)
```


Estrutura de seleção



if ternário

Python possui uma forma compacta e eficiente de realizar operações condicionais em uma única linha, conhecida como expressão condicional ou operador ternário condicional.

valor_se_verdadeiro **if** condição **else** valor_se_falso



Condicionais Compostas (if-else)

A estrutura condicional composta if-else permite executar um bloco de código se uma condição for verdadeira e outro bloco de código se a condição for falsa.

```
if condição:
```

```
    # código a ser executado se a condição for verdadeira
```

```
else:
```

```
    # código a ser executado se a condição for falsa
```



Condicionais Encadeadas (if-elif-else)

A estrutura condicional encadeada if-elif-else permite verificar múltiplas condições em sequência.

```
if condição1:  
    # código a ser executado se a condição1 for verdadeira  
elif condição2:  
    # código a ser executado se a condição1 for falsa e a condição2 for verdadeira  
else:  
    # código a ser executado se todas as condições anteriores forem falsas
```



Nested if (if aninhado)

O if aninhado é quando você coloca uma estrutura condicional dentro de outra estrutura condicional.

```
idade = 20
if idade >= 18:
    if idade <= 65:
        print("Você é um adulto.")
    else:
        print("Você é um idoso.")
else:
    print("Você é menor de idade.")
```





Função

Definição de Função

Em Python, uma função é definida usando a palavra-chave `def`, seguida pelo nome da função, parênteses que podem conter parâmetros, e um bloco de código indentado que constitui o corpo da função.

```
def nome_da_funcao(parametros):  
    # bloco de código  
    return valor_de_retorno
```



Chamada de Função

Para executar uma função, você faz uma chamada de função, passando os argumentos necessários entre parênteses.

PARÂMETROS

```
def saudacao(mensagem)
    print(mensagem)

saudacao("olá Katia")
```

ARGUMENTOS





SQL

- DDL, DM, join



DDL (Data Definition Language)

DDL é usado para definir e modificar a estrutura dos objetos em um banco de dados, como tabelas, índices, e visões. Os comandos DDL incluem:

CREATE: Cria objetos no banco de dados, como tabelas, índices, e visões.

ALTER: Modifica a estrutura de um objeto existente no banco de dados.

```
-- Cria uma tabela chamada 'clientes'
```

```
CREATE TABLE clientes (  
    id INT PRIMARY KEY,  
    nome VARCHAR(100),  
    email VARCHAR(100)  
);
```

```
-- Adiciona uma nova coluna 'telefone' à tabela 'clientes'
```

```
ALTER TABLE clientes ADD telefone VARCHAR(20);
```




DML (Data Manipulation Language)

DML é usado para manipular os dados dentro das tabelas. Os comandos DML incluem:

- SELECT: Recupera dados das tabelas.
- INSERT: Insere novos dados nas tabelas.
- UPDATE: Atualiza dados existentes nas tabelas.
- DELETE: Remove dados das tabelas.

```
-- Insere um novo registro na tabela 'clientes'
INSERT INTO clientes (id, nome, email) VALUES (1, 'João', 'joao@example.com');

-- Atualiza o email do cliente com id 1
UPDATE clientes SET email = 'joao_novo@example.com' WHERE id = 1;

-- Seleciona todos os registros da tabela 'clientes'
SELECT * FROM clientes;

-- Remove o cliente com id 1
DELETE FROM clientes WHERE id = 1;
```




Join

Os joins são uma operação fundamental em SQL utilizada para combinar dados de duas ou mais tabelas com base em uma condição relacionada.

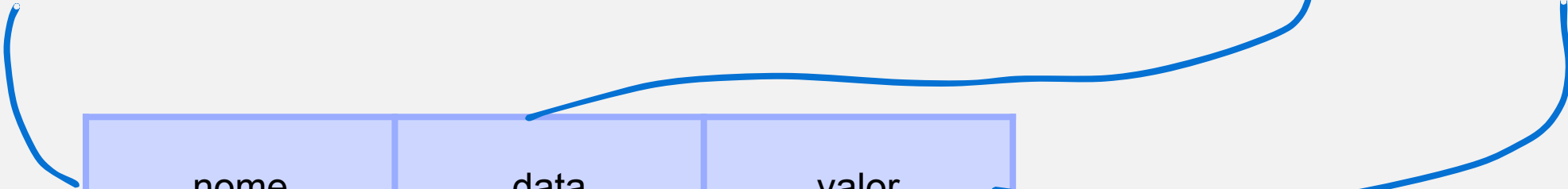
Tabela Perfil

Id	CPF	Nome

Tabela venda

id	id_perfil	data	valor

nome	data	valor



Join

Os joins são uma operação fundamental em SQL utilizada para combinar dados de duas ou mais tabelas com base em uma condição relacionada.

Tabela Perfil

Id	CPF	Nome
1	434.32	Letícia
2		

Tabela venda

id	id_perfil	data	valor
1	2	7/8/20	45.00
2			

nome	data	valor
Letícia	7/8/20	45.00



Join

```
SELECT employees.name, departments.department_name  
FROM employees  
INNER JOIN departments  
ON employees.department_id = departments.department_id;
```

Numpy, Padas, matplotlib

Proximo capítulo



Pandas vs Numpy vs Matplotlib

- **Pandas** é amplamente usado para manipulação de DataFrames em Python
 - Um **DataFrame** é uma estrutura de dados bidimensional, similar a uma tabela, amplamente usada para análise e manipulação de dados tabulares.
- **NumPy** é uma biblioteca fundamental para a computação científica em Python. Ela fornece suporte para arrays e matrizes de grandes dimensões, além de uma coleção extensa de funções matemáticas para operar esses arrays de forma eficiente.
 - Numpy usa mais numpy array
- **Matplotlib** é uma das bibliotecas mais utilizadas para criação de gráficos e visualizações em Python.
 - Matplotlib usa figure



ATIVIDADE

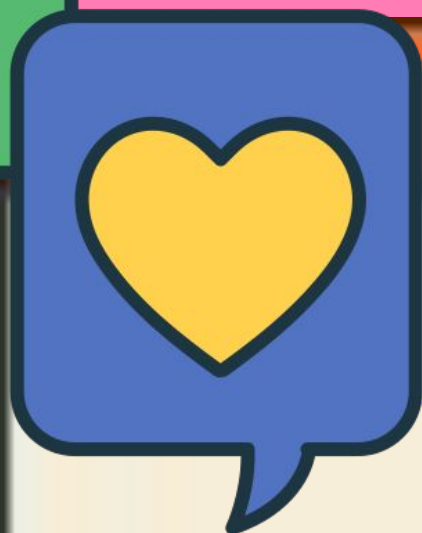
1
horas

Segunda Parte

vamos usar o vscode com jupyter integrado

PLAY





OBRIGADA

