

Introdução ao Python

1



Bibliotecas populares do Python:

- Matplotlib
- Seaborn
- Numpy
- Pandas
- TensorFlow
- PyTorch
- Scikit-learn



Ambientes de desenvolvimento

populares em Python

- Jupyter Notebook
- Google Colab
- PyCharm
- VS Code
- Spyder

Alguns **domínios** em que Python é aplicado

- Financeiro
- Saúde
- Marketing
- Tecnologia

2

Principais **tipos** de **dados**:

- **int** (Inteiro): 1,2,3,4
- **float** (ponto flutuante): 3.14
- **str** (string): "Olá"
- **bool** (booleano): True ou False

Operadores aritméticos

- **Soma**: +
- **Subtração**: -
- **Multiplicação**: *
- **Divisão**: /
- **Divisão inteira**: //
- **Resto**: %
- **Exponenciação**: **

Operadores relacionais e lógicos:

- **Igual** a: ==
- **Diferente de**: !=
- **Maior que**: >
- **Menor que**: <
- **Maior ou igual**: >=
- **Menor ou igual**: <=

Funções built-in úteis:

- **len()**: Retorna o comprimento de um objeto;
- **type()**: Retorna o tipo de um objeto;
- **int()**, **float()**, **str()**: Convertem o tipo;
- **input()**: Lê uma entrada do usuário;
- **print()**: Exibe uma saída.

3

Listas: coleções **ordenadas** e **mutáveis**. Úteis para armazenar itens em sequência.

Tuplas: coleções **ordenadas** e **imutáveis**. Utilizadas para armazenar dados que não devem ser alterados.

Exemplo de dicionário:

Dicionário	
Chaves	Valores
"nome"	"Gandalf"
"classe"	"Wizard"
"ordem"	"Istari"
"altura"	"1.85"

Exemplo de conjunto:



Lembre-se que os critérios **mutabilidade**, **unicidade**, **indexação** e **desempenho** são usados para a escolha adequadas das estruturas de dados

4



Condicionais:

- **if**: Executa um bloco de código se a condição for verdadeira.
- **elif**: Adiciona condições adicionais, executando o bloco se a condição anterior for falsa e a nova for verdadeira.
- **else**: Executa um bloco de código se nenhuma condição anterior for satisfeita.

Bloco match-case:

- **match** define a variável cujo valor será comparado.
- **case** define um valor específico com o qual o valor da variável será comparado.

Estruturas de repetição:

- **for**: Utilizada para iterar sobre uma sequência
- **while**: Executa um bloco de código enquanto uma condição for verdadeira.

5

Funções nomeadas:

- Funções sem parâmetros e sem retorno: **apenas** executam **uma série** de instruções
- Funções sem parâmetros e com retorno: **retornam** um **valor** ao final da execução

As funções lambda são úteis quando você precisa de uma **função** simples por um curto período de tempo e **que não será reutilizada** em outro lugar que não seja o código atual.



Escopo de variáveis:

- Global: a **variável** pode ser **acessada** em **qualquer parte** do código
- Local: só pode ser **acessada dentro** da **função** em que foi declarada.

6

Algumas operações com arquivos:

- **open**(arquivo, 'r') abre o arquivo especificado em modo de leitura ('r');
- **as file** atribui o objeto de arquivo aberto à variável file (pode ser escolhido outro nome ou identificador);
- **file.read()** lê todo o conteúdo do arquivo e o armazena na variável conteúdo;
- **'w'** (write): Abre o arquivo para escrita. Se o arquivo já existir, ele será truncado (todo o conteúdo será apagado). Se o arquivo não existir, ele será criado;
- **'a'** (append): Abre o arquivo para escrita em modo de adição. Os dados são escritos no final do arquivo sem truncá-lo. Se o arquivo não existir, ele será criado.

7

Funções: **blocos** de código **reutilizáveis**. São definidas com def e podem receber parâmetros e retornar valores.



O uso do bloco **with** serve para **abrir** arquivos, garantindo que sejam **fechados corretamente** após a operação. Modos como 'r', 'w', 'a', 'b', 't', '+' são usados conforme a necessidade.

Métodos como read, readline, readlines e write são **essenciais** para manipulação de dados em arquivos. Operações incluem leitura de todo o arquivo, linha a linha e escrita de dados.

As exceções try, except, else, e finally são usadas para **capturar** e **tratar exceções**, garantindo que o programa continue funcionando mesmo em caso de erros.