

Programação Python - Nível Básico

O Que é Python?

Python é uma linguagem de programação de alto nível, interpretada, de propósito geral e fácil aprendizado. Criada por Guido van Rossum e lançada em 1991, Python se destaca pela sua sintaxe clara e legibilidade, o que a torna ideal para iniciantes. É uma linguagem versátil, utilizada em diversos campos como desenvolvimento web, análise de dados, inteligência artificial, automação e ciência de dados.

A filosofia de design do Python enfatiza a legibilidade do código com o uso significativo de espaços em branco (indentação). Sua abordagem de "baterias inclusas" significa que a biblioteca padrão já vem com muitas funcionalidades úteis, reduzindo a necessidade de código adicional.

Variáveis e Tipos de Dados Fundamentais

Em Python, variáveis são espaços na memória que armazenam dados. Diferentemente de outras linguagens, não é necessário declarar o tipo de uma variável antes de usá-la, pois Python é dinamicamente tipada - o tipo é determinado automaticamente com base no valor atribuído.

Os principais tipos de dados fundamentais em Python são:

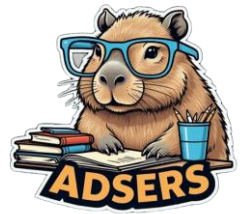
- Strings (str): Sequências de caracteres delimitadas por aspas simples ou duplas. Exemplo: "Olá, mundo!" .
- Inteiros (int): Números inteiros sem parte decimal. Exemplo: 42 .
- Ponto Flutuante (float): Números com parte decimal. Exemplo: 3.14 .
- Booleanos (bool): Representam valores de verdadeiro (True) ou falso (False).
- None: Representa a ausência de valor ou um valor nulo.

Operadores Matemáticos e de Comparação

Python oferece uma variedade de operadores para realizar operações matemáticas e comparações lógicas.

Operadores Matemáticos:

- Adição (+): Soma dois valores. Exemplo: 5 + 3 resulta em 8 .



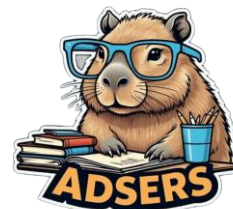
- Subtração (-): Subtrai o segundo valor do primeiro. Exemplo: 5 - 3 resulta em 2 .
- Multiplicação (*): Multiplica dois valores. Exemplo: 5 * 3 resulta em 15 .
- Divisão (/): Divide o primeiro valor pelo segundo, resultando em um float. Exemplo: 5 / 2 resulta em 2.5 .
- Divisão inteira (//): Divide e retorna apenas a parte inteira. Exemplo: 5 // 2 resulta em 2 .
- Módulo (%): Retorna o resto da divisão. Exemplo: 5 % 2 resulta em 1 .
- Exponenciação (**): Eleva o primeiro valor à potência do segundo. Exemplo: 2 ** 3 resulta em 8 . Operadores de Comparação:
- Igual a (==): Verifica se dois valores são iguais.
- Diferente de (!=): Verifica se dois valores são diferentes.
- Maior que (>), Menor que (<): Comparam valores.
- Maior ou igual a (>=), Menor ou igual a (<=): Comparam valores incluindo igualdade.

Estruturas Condicionais em Python (if, elif, else)

Estruturas condicionais permitem que o programa tome decisões baseadas em condições específicas, executando diferentes blocos de código dependendo se essas condições são verdadeiras ou falsas.

Estrutura if:

```
if condição:  
    # código executado se a condição for verdadeira
```



Estrutura if-else:

```
if condição:
    # código executado se a condição for verdadeira
else:
    # código executado se a condição for falsa
```

Estrutura if-elif-else:

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

Python também suporta operadores ternários, que são formas concisas de expressar condicionais em uma única linha:

```
resultado = valor_se_verdadeiro if condição else valor_se_falso
```

Referências:

Lopes, E. (2023). *If/Elif/Else: as Estruturas Condicionais do Python*. Python Academy. Disponível em: <https://pythonacademy.com.br/blog/estruturas-condicionais-no-python> - DataCamp. (2024). *if...elif...else in Python Tutorial*. Disponível em: <https://www.datacamp.com/tutorial/elif-statements-python>

Ribeiro, E. (2023). *Do Básico ao Complexo: Aprendendo a Programar em Python com ChatGPT*. Universidade Federal do Tocantins. - Python Software Foundation. (2024). *Documentação Python 3.13.3*. Disponível em: <https://docs.python.org/pt-br/3/reference/index.html>

DataCamp. (2025). *Tipos de dados Python explicados: Um guia para iniciantes*. Disponível em: <https://www.datacamp.com/pt/blog/python-data-types>

De Vit, A. R. D. (2023). *Introdução à Programação e Estruturas de Dados Fundamentais com Python para Programadores C*. Editora Synapse.