

Guia Rápido – Fundamentos Teóricos do Python

Python é uma linguagem de programação interpretada, de alto nível e de tipagem dinâmica. É multiparadigma, suportando programação estruturada, funcional e orientada a objetos. Sua sintaxe simples e próxima do inglês a torna uma linguagem ideal para iniciantes e poderosa para especialistas.

1. Estrutura Básica de um Programa

Um programa Python pode ser formado por uma sequência de instruções simples. O exemplo clássico imprime uma mensagem no console.

Exemplo:

```
print("Olá, mundo!")
```

2. Variáveis e Tipos de Dados

Variáveis em Python funcionam como caixas que guardam valores. O tipo é definido automaticamente pelo interpretador, conforme o valor atribuído.

3. Escopo de Variáveis: Globais x Locais

O escopo define onde a variável pode ser acessada. Variáveis globais são criadas fora de funções e podem ser acessadas em todo o programa. Variáveis locais são criadas dentro de funções e só existem ali. Se uma variável global for modificada dentro de uma função, é necessário o uso da palavra-chave 'global'.

Exemplo ilustrativo:

```
x = 10 # global def funcao(): global x x = 20 # altera a variável global
```

4. Estruturas de Controle

Python usa indentação para organizar blocos de código. São fundamentais as estruturas condicionais (if/else) e os laços de repetição (for, while).

5. Estruturas de Dados

Python possui estruturas nativas para organização de dados, como listas, tuplas, dicionários e conjuntos. Cada uma tem suas características: listas são mutáveis, tuplas imutáveis, dicionários guardam pares chave-valor e conjuntos não permitem duplicatas.

6. Funções

Funções são blocos reutilizáveis que organizam o código. Podem receber parâmetros e retornar valores. Funções anônimas (lambda) são usadas em casos simples.

7. Programação Orientada a Objetos

Python permite criar classes (modelos) e objetos (instâncias). O construtor `__init__` inicializa atributos, e métodos definem comportamentos. Isso ajuda a estruturar sistemas complexos.

8. Tratamento de Erros (Exceções)

Para evitar que o programa pare em caso de erro, utiliza-se try/except. Também existe o bloco finally, que sempre é executado.

9. Aplicações Práticas do Python

Python é usado em diversas áreas: desenvolvimento web, integração com bancos de dados, ciência de dados, inteligência artificial, automação de tarefas, cibersegurança, interfaces gráficas e consumo de APIs.

10. Como Rodar Arquivos Python

No Windows (CMD):

- Salvar o arquivo, ex.: `hello.py`, com o código desejado.
- Abrir o Prompt de Comando e navegar até a pasta com `cd`.
- Executar: `python hello.py` ou `py hello.py`.

No Linux (Terminal):

- Salvar o arquivo `hello.py`.
- Abrir o terminal e usar `cd` para ir até a pasta.
- Executar: `python3 hello.py`.
- Opcional: tornar o arquivo executável com `chmod +x hello.py` e rodar `./hello.py`.

Conclusão

Python combina simplicidade com poder. Seus fundamentos envolvem sintaxe clara, variáveis dinâmicas, escopo bem definido (local x global), estruturas de dados nativas, suporte a múltiplos paradigmas e aplicações em praticamente todas as áreas da tecnologia.