

Introdução à Linguagem Python

Python é uma linguagem de programação amplamente utilizada em ciência de dados, computação científica, automação e inteligência artificial. Assim como o R, Python é usado para transformar dados em informação por meio de código executável. Isso faz de Python uma ferramenta científica, e não apenas uma linguagem de programação genérica.



Eduardo Ogasawara

eduardo.ogasawara@cefet-rj.br

<https://eic.cefet-rj.br/~eogasawara>

Programação, Algoritmos e Linguagens

Linguagem de Programação

Sistema formal usado para escrever instruções que um computador pode executar, descrevendo cálculos, transformações de dados e processos lógicos.

Algoritmo

Sequência finita de passos que descreve como resolver um problema, como calcular uma média ou classificar dados.

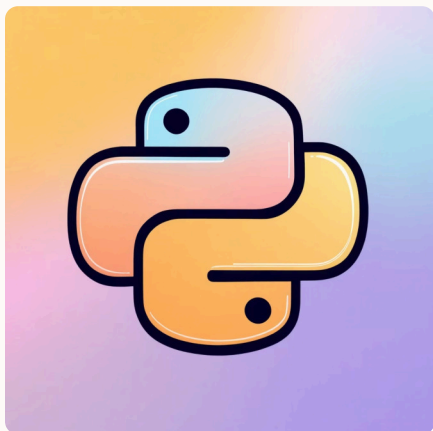
Programa

Implementação de um algoritmo em uma linguagem de programação que o computador pode executar.

Quando escrevemos um programa em Python, estamos transformando um algoritmo em algo que o computador pode executar.



O que é a Linguagem Python



Python é uma linguagem **interpretada**, de **alto nível** e projetada para ser simples e legível. O código é executado linha por linha pelo interpretador Python, em vez de ser previamente compilado.

Cada comando digitado é analisado, executado e o resultado é devolvido imediatamente. Esse modelo favorece a experimentação e a análise exploratória, de forma análoga ao R.



Interpretada

Execução linha por linha sem necessidade de compilação prévia.



Legível

Sintaxe clara e intuitiva, próxima da linguagem natural.



Exploratória

Ideal para experimentação e análise de dados.

O Console Python

O console Python é um ambiente interativo onde comandos podem ser digitados e executados imediatamente. Ele permite testar ideias, verificar valores e explorar dados rapidamente.

1

Digitação

O usuário digita um comando no console Python.

2

Execução

O interpretador analisa e executa o comando imediatamente.

3

Resultado

O resultado é devolvido instantaneamente na tela.

No Visual Studio Code, o console aparece no terminal integrado ou em janelas interativas. O console é o principal meio de interação entre o usuário e a linguagem Python.



▶ PRÁTICA

Primeiros Comandos no Console

No console Python, podemos executar expressões diretamente e obter resultados imediatos. Cada comando executado no console é uma instrução de um programa.

O interpretador recebe a expressão, executa o cálculo e devolve o resultado instantaneamente, permitindo experimentação rápida.

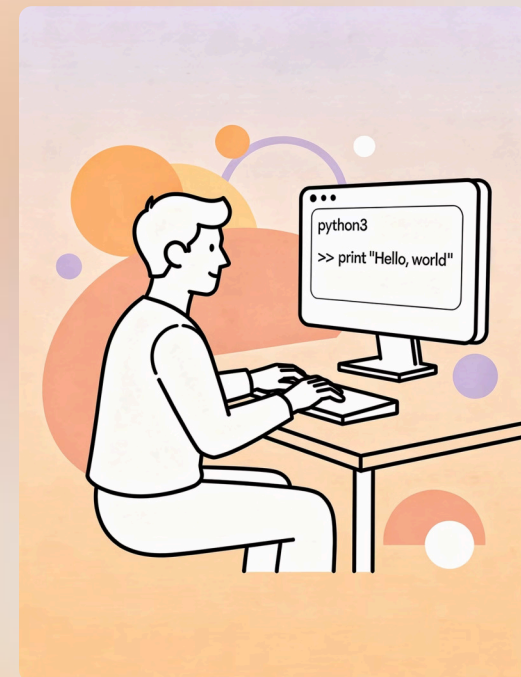


Exemplo Básico

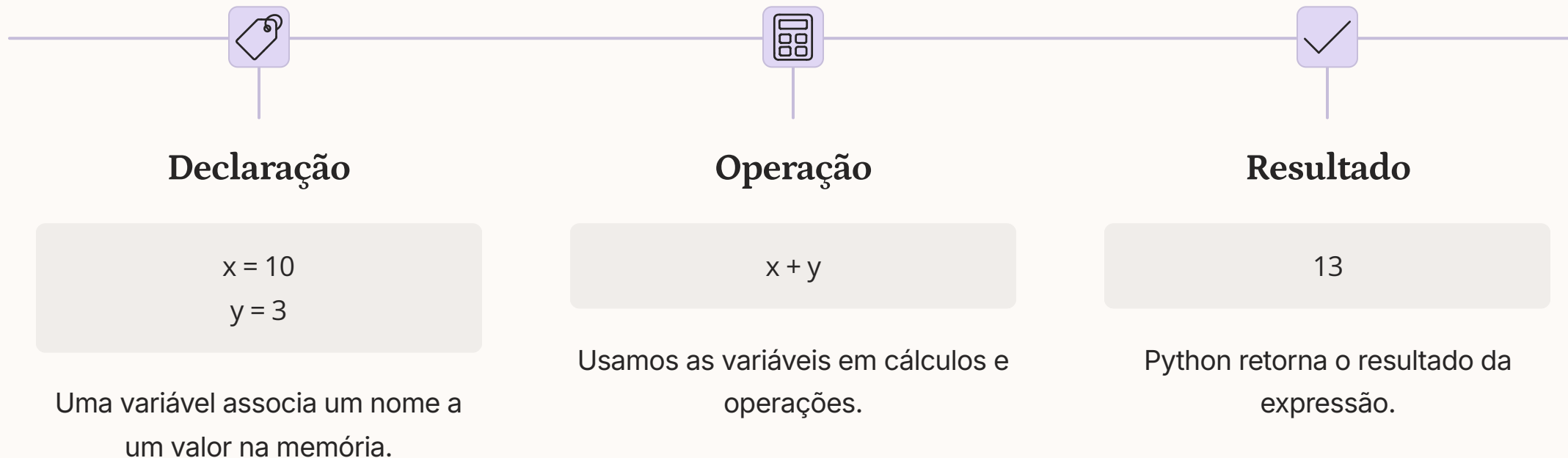
```
>>> 2 + 2
```

```
4
```

O símbolo `>>>` indica o prompt do console Python.



Variáveis e Execução Sequencial



Python executa os comandos na ordem em que aparecem, e cada linha depende do estado criado pelas anteriores. Essa execução sequencial é fundamental para entender como os programas funcionam.

Programas em Python

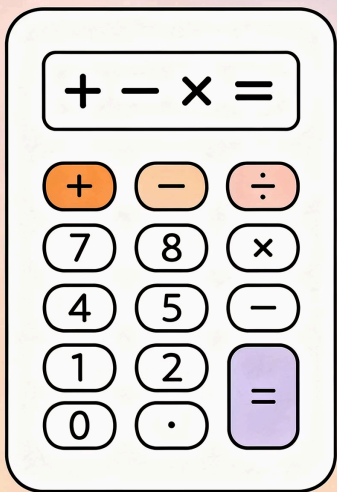
Exemplo de Programa



```
x = 10  
y = 20  
z = x + y  
z
```

Este programa implementa o algoritmo de soma de dois números e já representa um modelo formal de cálculo.

Um conjunto de comandos forma um programa. Cada linha é executada sequencialmente, construindo o resultado final através de passos lógicos.



Operações Aritméticas em Python

Python pode ser usado como uma calculadora científica poderosa. Essas operações formam a base de cálculos estatísticos e científicos.

Adição e Subtração

```
2 + 3  
10 - 4
```

Operações básicas de soma e diferença.

Multiplicação e Divisão

```
5 * 6  
8 / 2
```

Multiplicação usa * e divisão usa /.

Potenciação

```
2 ** 3
```

O operador ** calcula potências ($2^3 = 8$).

Vetores em Python

Listas Python

```
v = [10, 20, 30, 40]  
v
```

Sequências de valores podem ser representadas como listas nativas do Python.

Vetores NumPy

```
import numpy as np  
x = np.array([10, 20, 30, 40])  
x
```

Para computação científica, usamos vetores do NumPy, que permitem operações matemáticas eficientes.

Vetores são fundamentais para manipulação de conjuntos de dados e operações matemáticas vetorizadas, essenciais em ciência de dados.

Funções em Python

Funções encapsulam algoritmos reutilizáveis, permitindo organizar e modularizar o código de forma eficiente.

Definindo uma Função

```
def soma(a, b):  
    return a + b
```

Chamando a Função

```
soma(3, 5)  
# Retorna: 8
```

Funções permitem escrever código uma vez e reutilizá-lo múltiplas vezes, tornando os programas mais organizados e fáceis de manter.

Parâmetros e Argumentos

Parâmetros

São os nomes definidos na declaração da função:

```
def media(x, y):  
    return (x + y) / 2
```

x e **y** são os parâmetros da função.

Argumentos

São os valores passados na chamada da função:

```
media(10, 20)
```

10 e **20** são os argumentos fornecidos.

Esse mecanismo é essencial para criar algoritmos flexíveis que podem processar diferentes conjuntos de dados usando a mesma lógica.

Scripts em Python

O que é um Script?

Um arquivo contendo um programa completo, geralmente com extensão `.py`.

Como Criar?

Scripts podem ser criados no VS Code e executados pelo interpretador Python.

Por que Usar?

Permitem registrar análises, reproduzir resultados e organizar projetos científicos.

Scripts são fundamentais para trabalho profissional em ciência de dados, garantindo que análises sejam documentadas e reproduzíveis ao longo do tempo.

Instalando o Python

Passos para Instalação

1 Acesse o site oficial

Visite <https://www.python.org> para baixar a versão mais recente.

3 Execute a instalação

Siga as instruções do instalador. Marque a opção "Add Python to PATH".

2 Baixe o instalador

Escolha a versão adequada para seu sistema operacional (Windows, macOS ou Linux).

4 Verifique a instalação

Após a instalação, o comando `python` estará disponível no terminal do sistema.

Visual Studio Code para Python

O Visual Studio Code é o ambiente de desenvolvimento usado para trabalhar com Python. Assim como o RStudio para o R, ele se torna o laboratório de trabalho para Python.

Editor de Código

Interface poderosa com realce de sintaxe, autocompletar e detecção de erros.

Terminal Integrado

Execute comandos Python diretamente sem sair do editor.

Integração Python

Suporte nativo para interpretador, debugging e extensões Python.

O VS Code pode ser obtido em: <https://code.visualstudio.com>

Ambientes Virtuais (Environments)

O que é um Environment?

Define quais bibliotecas e qual interpretador Python um projeto utiliza, isolando dependências.

Ativando o Environment

Deve ser ativado para que o projeto use suas próprias dependências isoladas.

Criando um Environment

```
python -m venv venv
```

Este comando cria um ambiente virtual chamado "venv".

Benefícios

Evita conflitos entre projetos e garante reprodutibilidade das análises.

Reprodutibilidade e Boas Práticas

Em ciência de dados, um resultado só é válido se puder ser reproduzido.



Use Scripts

Documente todas as análises em arquivos executáveis e versionados.



Environments Isolados

Mantenha dependências específicas para cada projeto.



Bibliotecas Explícitas

Registre todas as bibliotecas e versões utilizadas.



Código Limpo

Organize código, use nomes claros e documente funções.

Usar scripts, environments e bibliotecas explícitas garante que análises possam ser refeitas no futuro, mantendo a integridade científica do trabalho.



ATENÇÃO

Erros Comuns de Iniciantes

Erros de Digitação

Nomes de variáveis incorretos, parênteses faltando ou operadores errados são muito frequentes no início.

Esquecimento de Imports

Usar bibliotecas como NumPy ou Pandas sem importá-las primeiro gera erros de nome não definido.

Conflitos de Environment

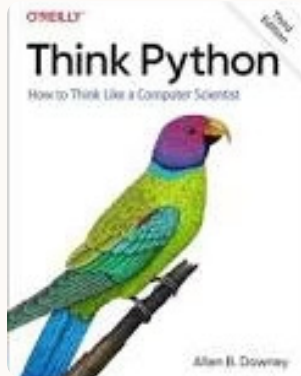
Misturar projetos diferentes no mesmo environment causa conflitos de versão e comportamentos inesperados.

Execução Fora de Ordem

Executar linhas de código fora de sequência pode produzir resultados errados ou erros de variável não definida.

Aprender a ler mensagens de erro é parte fundamental do processo de aprendizagem. Os erros são oportunidades de compreender melhor como o Python funciona.

Referências



Think Python

Downey, A. *Think Python: How to Think Like a Computer Scientist*. O'Reilly Media.

An essential introduction to programming fundamentals and computational thinking using Python.



Python Data Science Handbook

VanderPlas, J. *Python Data Science Handbook*. O'Reilly Media.

A comprehensive guide to essential tools for working with data in Python, including NumPy, Pandas, and visualization libraries.



Data Science from Scratch

Grus, J. *Data Science from Scratch*. O'Reilly Media.

Learn data science fundamentals by building algorithms and tools from the ground up using Python.