

Introdução ao Python

CATALOGUE

Contents

- Características da Linguagem
- Recursos Avançados
- Trabalhando com Variáveis em Python

Características da Linguagem

PART 01

Foco na Interação

Estrutura case sensitive

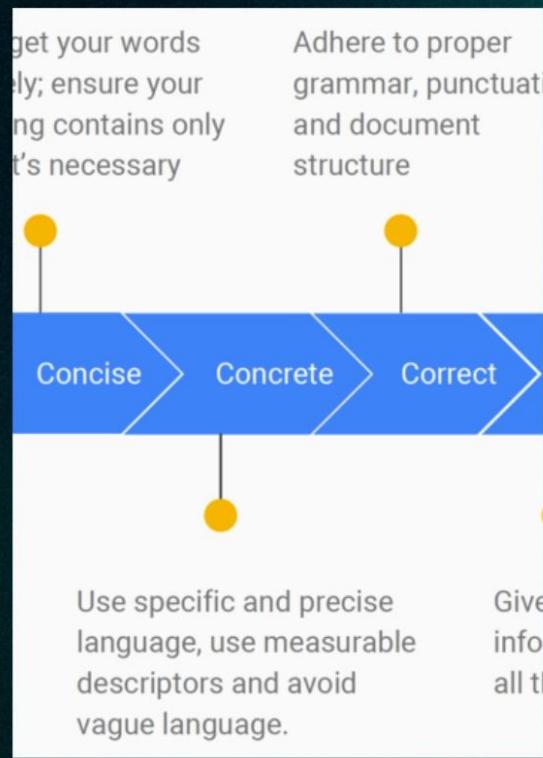
Python diferencia maiúsculas e minúsculas, o que exige atenção do programador para evitar erros relacionados a nomes de variáveis e funções.

Tipagem dinâmica

Permite que variáveis alterem seu tipo durante a execução, oferecendo maior flexibilidade na manipulação de dados e no desenvolvimento de aplicações.

Sintaxe clara e menos verbosa

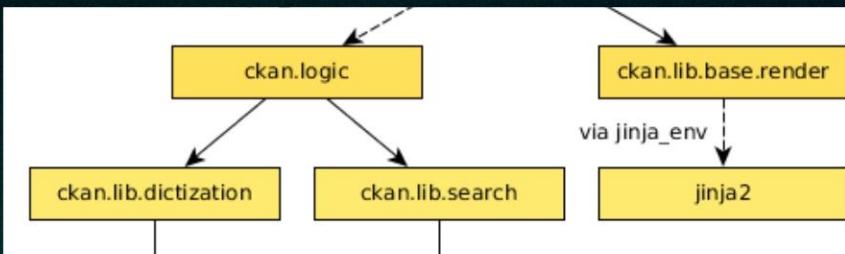
Python prioriza a simplicidade no código, reduzindo a necessidade de comandos complexos e facilitando a interação com o computador.



Orientação a Objetos e Código Organizado

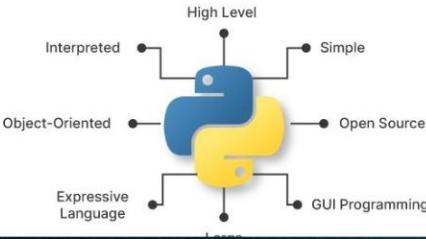
Modelo totalmente orientado a objetos

Em Python, todos os elementos do código, incluindo números e funções, são tratados como objetos, promovendo uma abordagem consistente e estruturada.



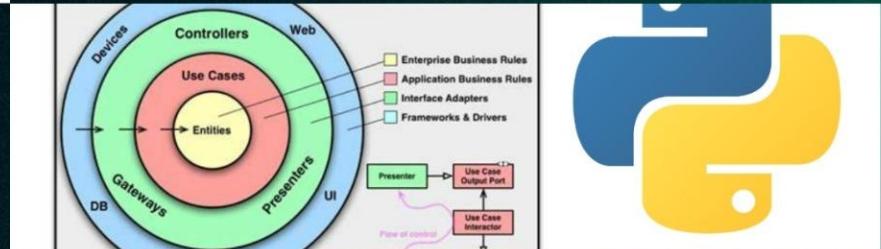
Foco na legibilidade

Python promove boas práticas de escrita, como o uso de nomes semânticos e convenções como o snake case, tornando o código mais fácil de compreender e manter.



Indentação obrigatória

A organização do código é garantida por meio de indentação, eliminando a necessidade de delimitadores manuais, como chaves, para indicar blocos de código.



Recursos Avançados

PART 02

Recursos Avançados

01 Supporte multiplataforma

Python é compatível com diferentes sistemas operacionais como Windows, macOS e Linux, proporcionando flexibilidade ao programador para desenvolver e executar aplicações em diversas plataformas sem necessidade de modificações no código.

02 Modo interativo

Python permite execução de código de forma interativa, facilitando a prototipagem rápida e testes de funcionalidades sem a necessidade de criar arquivos separados, ideal para aprendizado e desenvolvimento ágil.

03 Gerenciamento automático de memória

O sistema de gerenciamento de memória do Python inclui coleta de lixo integrada, permitindo a alocação e liberação automática de recursos, otimizando a performance e reduzindo erros de código relacionados à memória.

Recursos Avançados

Módulos de extensão diversos

A vasta biblioteca de módulos e pacotes de extensão permite que desenvolvedores implementem funcionalidades avançadas como análise de dados (NumPy, pandas), aprendizado de máquina (TensorFlow, scikit-learn), gráficos (Matplotlib, Plotly), entre outras.

Integração com outras linguagens

Python oferece ferramentas e bibliotecas, como Cython e Jython, que facilitam a integração com linguagens como C++, Java ou até linguagens web. Isso amplia sua aplicabilidade em projetos multidisciplinares.

Trabalhando com Variáveis em Python

PART 03

Tipos de Dados Embutidos

Int (inteiros)

Representam números inteiros, como -1, 0 e 100. São amplamente utilizados para cálculos matemáticos e operações que necessitem de números sem casas decimais.

Float (ponto flutuante)

Permitem representar números com casas decimais, como 3.14 ou -0.01. São fundamentais para operações que exigem maior precisão numérica.

Str (cadeias de caracteres)

Manipulam textos e caracteres, como "Python" ou "123". São utilizados em processamento de texto, mensagens de interface, e entradas de usuário.

Bool (valores lógicos)

Trabalham com valores True e False, sendo muito úteis em decisões condicionais e operações de controle de fluxo em programas.

Trabalhando com Constantes

Imutabilidade simulada

Constantes podem ser simuladas em Python ao usar variáveis que não são alteradas, recomendando boas práticas de codificação para mantê-las fixas no programa.

Uso da biblioteca `math`

Para valores matemáticos predefinidos (como π e e), a biblioteca `math` oferece constantes confiáveis para cálculos complexos em diversas aplicações.

Convenção por capitalização

Embora não sejam protegidas no nível da linguagem, uma prática comum é usar nomes em MAIÚSCULAS para identificar variáveis designadas como constantes.

Garantia adicional com ferramentas

Frameworks e pacotes externos podem ser usados para implementar constantes realmente imutáveis, tornando seu comportamento mais robusto.

Impacto no desempenho

O uso adequado de constantes reforça a legibilidade do código e pode evitar bugs, promovendo consistência em cálculos e configurações.

Obrigado