

Linguagem Python

João G. Santos — RA: 2411612877

Introdução

O Python é uma linguagem de programação interpretada, multiparadigma e dinamicamente tipada criada em 1991, aproximadamente 20 anos após a criação do C[1]. Devido à sua versatilidade e interoperabilidade com linguagens de mais baixo nível como o C, o Python é amplamente utilizado por profissionais de diversas áreas, como Matemática Computacional, Ciência de Dados, e Computer Vision. Esse artigo visa expôr os fatores que contribuíram para a popularidade da linguagem, que perdura até os dias atuais.

Desenvolvimento

Características

O Python é caracterizado por possuir tipagem dinâmica. Um PEP (Python Enhancement Proposal) sugerindo um sistema de tipagem estática foi no entanto aceito e mergeado na linguagem. Ferramentas de análise estática como o `pyright` e o `mypy` podem ser usadas para verificar os tipos.

A linguagem também oferece suporte a paralelismo com threads e um sistema de queues que permite a comunicação bidirecional entre threads. Paralelismo baseado em `async/await` também é possível com a biblioteca `asyncio`.

Ecossistema

Um dos grandes destaques da linguagem é o massivo ecossistema de módulos e ferramentas criados ao redor da mesma. Parte disso se dá pela interoperabilidade da linguagem com o C, o que fez com que muitas interfaces programáveis para softwares existentes fossem criadas e distribuídas.

A publicação de módulos do Python é feita através do PIP, o repositório oficial de pacotes Python. Qualquer desenvolvedor pode publicar seu módulo de forma gratuita, sendo exigida apenas a criação de uma conta e configuração da autenticação de dois fatores. Existem também formas padronizadas de se documentar um projeto Python.

Pacotes famosos do Python incluem:

- pandas
- opencv
- redis
- mongodb

Design

Um dos objetivos do design da linguagem foi a facilidade de uso e de aprendizado. Possui uma semântica simples com poucas palavras chave e uma sintaxe que prima pela legibilidade do código, fazendo uso de whitespaces ao invés de brackets.

A distribuição regular do Python inclui um pequeno texto em forma de aforismas que condensa os princípios de design da linguagem:

The Zen of Python, by Tim Peters

```
Beautiful is better than ugly.  
Explicit is better than implicit.  
Simple is better than complex.  
Complex is better than complicated.  
Flat is better than nested.
```

Sparse is better than dense.
Readability counts.
Special cases aren't special enough to break the rules.
Although practicality beats purity.
Errors should never pass silently.
Unless explicitly silenced.
In the face of ambiguity, refuse the temptation to guess.
There should be one-- and preferably only one --obvious way to do it.
Although that way may not be obvious at first unless you're Dutch.
Now is better than never.
Although never is often better than *right* now.
If the implementation is hard to explain, it's a bad idea.
If the implementation is easy to explain, it may be a good idea.
Namespaces are one honking great idea -- let's do more of those!

Considerações finais

Há com efeito diversos fatores que concorrem para o sucesso do Python na computação acadêmica e nas indústrias.

- [1] - [https://en.wikipedia.org/wiki/Python_\(programming_language\)](https://en.wikipedia.org/wiki/Python_(programming_language))
- [2] - https://en.wikipedia.org/wiki/Zen_of_Python