



Python 101

Introdução a linguagem de programação



Principais características

- Não é necessário compilação para executar a linguagem;
- Pode ser executada em quase todos os ambientes : Windows, Linux, MacOS...;
- Por padrão, linguagem totalmente orientada a objetos. Até os tipos primitivos são objetos;
- Software Open Source;
- Grande comunidade e uso, tanto no mundo open source quanto em empresas;
- Multi-porpuse language;



Vantagens x Desvantagens

1. Vantagens

- a. Sintaxe simples e estruturada;
- b. Multi-paradigma;
- c. Linguagem dinâmica (REPL);
- d. Linguagem de alto-nível;
- e. Apesar de não ser tipada, existem exceptions que podem ser tratadas na combinação de tipos;

2. Desvantagens

- a. Performance;
- b. Em projetos muito grandes, pode ficar complexo a falta da definição de tipos (vem melhorado nas versões mais novas da linguagem);
- c. Não existe formalmente uma empresa que suporte a linguagem, cabendo apenas a comunidade.

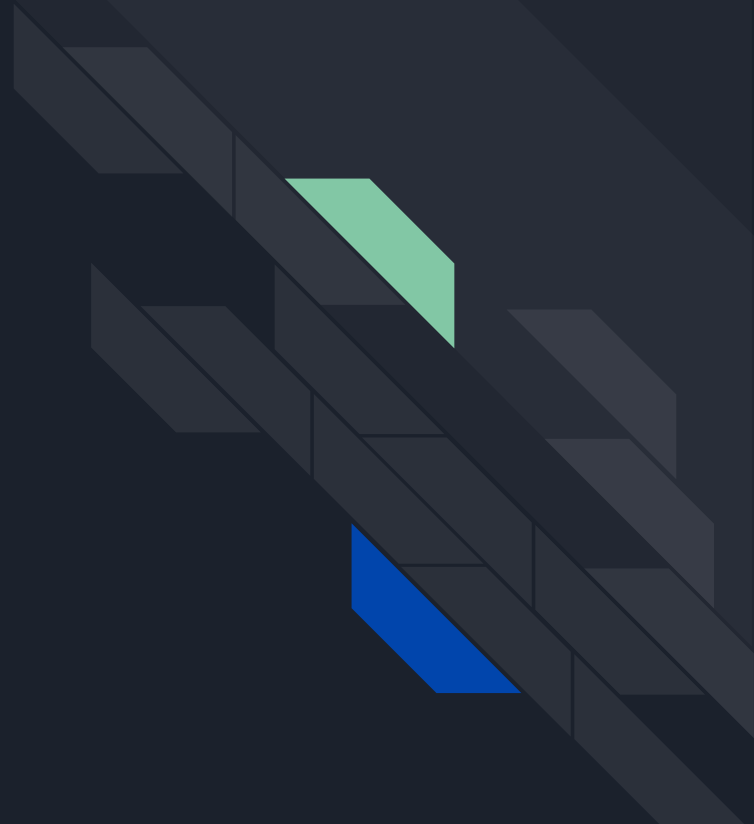


Marcos da linguagem

- 1991 - Criada em meados dos anos 80 (1986) por Guido Von Rossum na Holanda, mas efetivamente publicada em 1991, **versão 0.9.0**;
- 1994 **versão 1.0**;
- 1999 - O CMS **Plone** foi criado;
- 2001 **versão 2.0**;
- 2002 - Projeto **Twisted** foi lançado;
- 2005 - **Django** e **numpy** foram ambos criados;
- 2008 - **Python 3.0** foi lançado;
- 2010 - Projeto **scikit-learn** e **Flask** foram criados;
- 2011 - **IPython** (posteriormente nomeado para **Jupyter**) foi criado;
- 2015 - **TensorFlow** foi criado pelo Google;
- 2018/2019 - **The Steering Council Model** (PEP 8016)

Linguagem

ver Jupyter Notebook Python101*.ipynb





Ementa

- Introdução
 - PEP's
 - Zen of Python
 - PyPi
 - Legacy python
- Ferramentas : IDE's x Jupyter
- Sintaxe e instruções lógicas
 - Estruturas condicionais e Laço de repetição
 - Funções
 - File I/O
- Estruturas de dados
 - Listas, tuplas, dicionários, sets
 - List Comprehension
- Tratamento de Exceções
- Gerenciamento de pacotes
 - Virtual Environment
 - pyenv



Package Management

- pip
 - Gerenciador de pacotes do python (como em outras linguagens temos: maven, npm, composer);
- virtualenv
 - **Ambientes python isolados**, permitindo a instalação de bibliotecas específicas (versões específicas) para aquele ambiente.
- pyenv
 - Maneira de se instalar múltiplas versões da linguagem python no computador e conseguir comutar entre elas sem afetar o funcionamento.
 - Possibilidade de criar sistemas (junto com virtualenv) em determinada versão da linguagem com bibliotecas específicas, sem a necessidade de muita configuração.

Dúvidas?

