



Python for Data Engineer

Aula 1 - Apresentação da Disciplina e Introdução

Leandro Mendes Ferreira

leandro.ferreira@faculdadeimpacta.com.br



AGENDA!



APRESENTAÇÃO



OBJETIVOS



CONHECENDO PYTHON



PONTOS FORTES E FRACOS



PYTHON E BIG DATA



BIBLIOGRAFIA

Objetivos do Treinamento



- Visão Geral da Linguagem
- Apresentação dos conceitos básicos aos intermediários
- Base para poder continuar o curso de Spark com Python
- Curso intercala aula teórica com exercícios

Python

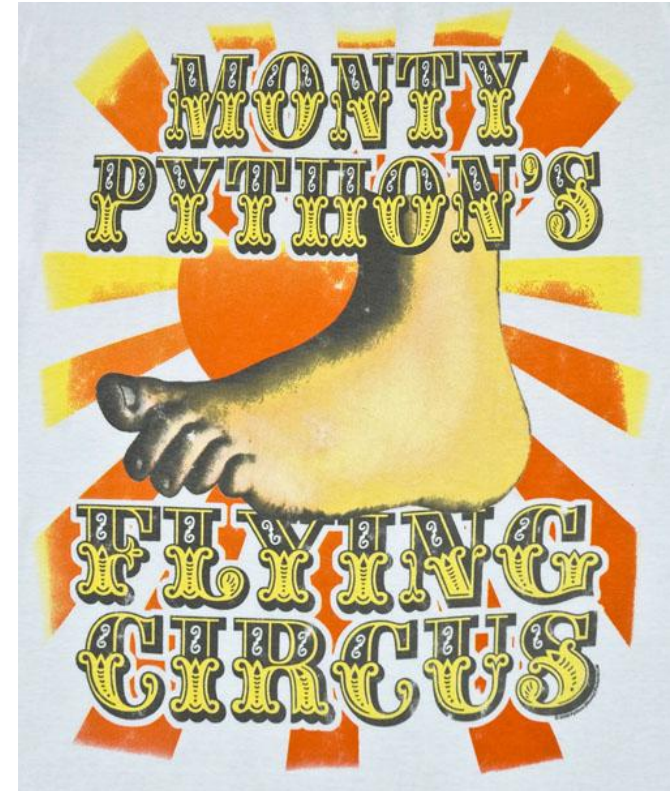


“Python é uma linguagem de alto nível, orientada a objetos, de programação de alto nível e de semântica dinâmica. Sua construção baseada em estruturas de dados de alto nível, combinada com sua “tipagem” e ligações dinâmicas tornam-na muito atraente tanto para o desenvolvimento rápido de aplicações quanto para criação de scripts ou como linguagem para interligar componentes já existentes. Python é simples, possui uma sintaxe fácil de aprender que dá ênfase à legibilidade, reduzindo o custo de manutenção do código”

Guido van Rossum

Características

- Criado por Guido van Rossum
- Derivada da linguagem ABC
 - C++, Java, Scheme
- Distribuições Python:
 - Cpython
 - Pypy
 - Jython
 - Cython
 - pySpark*
 - py4J
- **Spam, Eggs, Bacon** -> ao invés de foo, bar, baz
 - <https://www.youtube.com/watch?v=anwy2MPT5RE>



Um pouco da história do Python

A linguagem foi escrita inicialmente por [Guido van Rossum](#).

Sobre a origem do Python ele disse:

“ Há mais de seis anos, em dezembro de 1989, eu estava procurando por um projeto de programação como "hobby" que me mantivesse ocupado durante a semana próxima ao [Natal](#). Meu escritório... estaria fechado, mas eu tinha um computador em casa, e não muito mais do que isso em mãos. Eu decidi escrever um interpretador para a nova linguagem de scripting sobre a qual eu vinha pensando ultimamente: uma descendente da ABC que agradaria a hackers de Unix/C. Eu escolhi Python como um título provisório para o projeto, sendo que eu estava num humor um pouco irreverente (e sendo também um grande fã do [Monty Python's Flying Circus](#)). ”



Hello World

... 'hello world' - Java, C, PHP, Pascal

```
program helloworld;  
begin  
    writeln<'Hello World!';>  
end
```

```
class HelloWorld  
{  
    public static void main(String[] args)  
    {  
        System.out.println("Hello World!");  
    }  
}
```

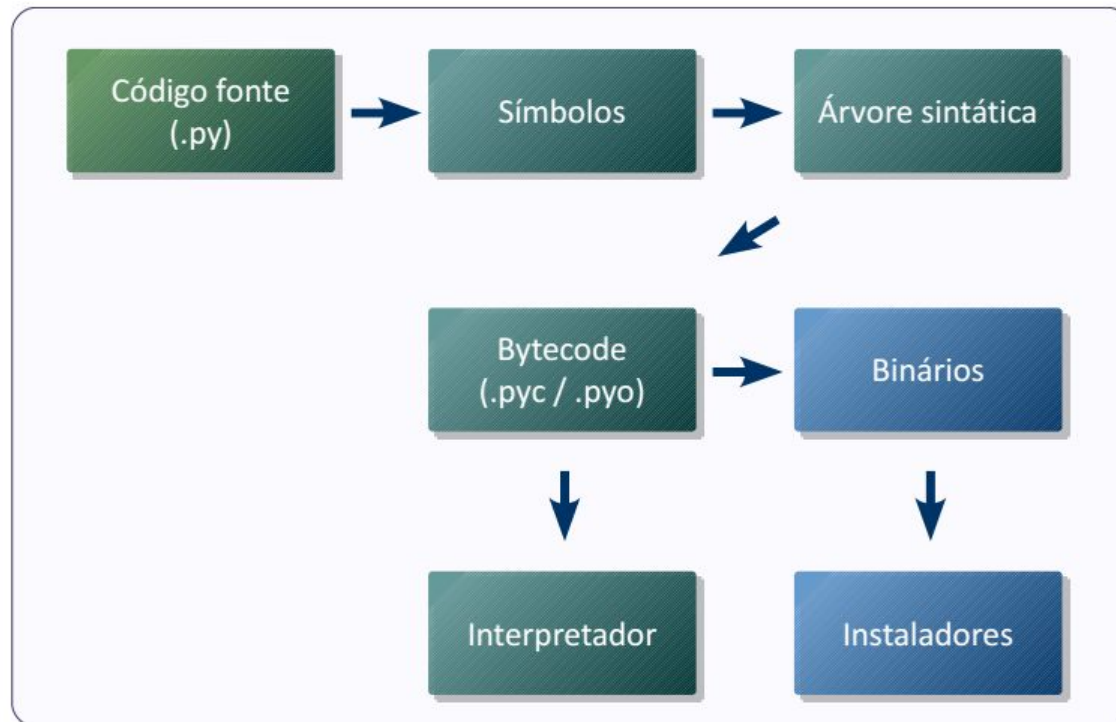
```
#include <stdio.h>  
int main (void)  
{  
    printf("Hello World");  
    return 0;  
}
```

```
<?php  
    echo "Hello World";  
?>
```

Mateus Vanzo de
Pádua

Hello World em Python

```
print("Hello  
World")
```

Filosofia

- Você deve escrever códigos Python de forma “*Pythonica*”
- *Import this*

The Zen of Python, by Tim Peters

```
01 Bonito é melhor que feio.
02 Explícito é melhor que implícito.
03 Simples é melhor que complexo.
04 Complexo é melhor que complicado.
05 Plano é melhor que aninhado.
06 Esparso é melhor que denso.
07 Legibilidade conta.
08 Casos especiais não são especiais o bastante para se quebrar as regras.
09 Embora a simplicidade supere o purismo.
10 Erros nunca deveriam passar silenciosamente.
11 A menos que explicitamente silenciados.
12 Ao encarar a ambiguidade, recuse a tentação de adivinhar.
13 Deveria haver uma - e preferencialmente apenas uma - maneira óbvia de se fazer isto.
14 Embora aquela maneira possa não ser óbvia à primeira vista se você não for holandês.
15 Agora é melhor que nunca.
16 Embora nunca, seja muitas vezes melhor que pra já.
17 Se a implementação é difícil de explicar, é uma má idéia.
18 Se a implementação é fácil de explicar, pode ser uma boa idéia.
19 Namespaces são uma idéia estupenda - vamos fazer mais deles!
```



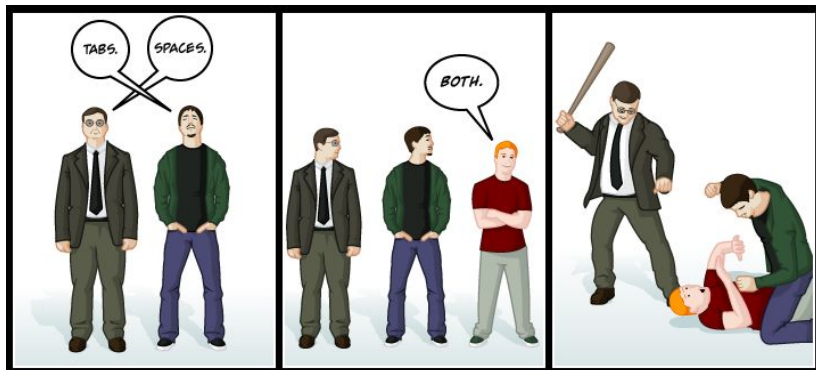
BY LUISGUS

Características da Linguagem

- Linguagem Interpretada (Script?)
 - Pré Analisada
 - Sintática e
 - Heurística
 - byte-compilada
 - .pyc
- Orientada a objetos e imperativa
 - Tudo em Python é objeto
- Suporta programação procedural
 - Possível programar em script e orientado a procedimento
 - Não é necessário criar classes ou interfaces
- Suporta programação funcional
 - Funções de Primeira Classe
 - Cloujure
 - Lambda
- Tipagem forte e dinâmica



Características da Linguagem



- Suporta Programação concorrente e Paralela
 - Threads
 - Asyncio
 - Canais
- Suporta conceitos complexos de computação
 - recursividade
 - yield
 - iterators
 - lazy
 - decorators
 - desempacotamento de tuplas
 - sobrecarga de operadores
 - gerenciador de contexto (with)
 - assert
 - dataframes (através de pacotes de terceiros)
- Só roda se o código estiver indentado
 - PEP 8

Pontos Fortes do Python

- Simples mas robusto
- Leitura geralmente fácil
- Modulo Interativo (Shell)
- Sistema de auto documentação de código fácil
- Ampla documentação
- Muita base de conhecimento
- Linguagem madura e estabelecida a muitos anos (criado nos anos 1990)
- Baixa curva de aprendizado
- Comunidade fortemente estabelecida
- Informação de erros razoável
- Ampla biblioteca nativa
- Muitos pacotes disponíveis
- Suporte a computação científica
- Amplo suporte para Big Data



Problemas Python

- Python em certas aplicações é LENTO
 - Interpretado
 - Objetos complexos
 - Representação interna dos objetos como dicionário (MAP)
- Duas versões de Python ainda vigentes (2.7 e a 3.X)
 - Nem todos pacotes de terceiro foram migrados
- Grave problema com encoding
 - encoding até do script de código
- Algumas coisas são abstraídas demais
- Programação paralela e concorrente com Python é complexa
 - Melhorou a partir da versão 3.5
 - Conceitos e abordagem ainda é adaptada para programação concorrente
- Sistema de dependências do Python é pouco sofisticado e com suporte limitado comparado a (MAVEN, SBT, até npm)
- pip e easyinstall



Big Data e Python

- Problemas com alguns conceitos fortes de Orientação a Objetos
 - Não suporta encapsulamento forte
 - Faz pseudo-encapsulamento com os `__` e decorators
 - Suporte a poucas versões a interfaces e não de forma natural
 - Não possui sobrecarga de métodos
 - Self explícito em métodos
 - Class explícito em métodos de Classe e na instanciação de Superclasses
- Problemas com alguns conceitos de Programação funcional
 - Lambda explícito
 - Objetos mutáveis
 - Não possui árvores de tipos fortes
- Não possui ponteiros
- Não possui tipagem explícita
- Métodos padrões com “*underling underling*” são horríveis de ler



Big Data e Python

- Fortemente usado com Apache Spark
- Integração com diversas ferramentas do mundo de Big Data
 - Hadoop
 - Kafka
 - Storm
- Dezenas de bibliotecas para Machine Learning
 - scikit-learn
 - Pandas
 - nltk
 - Anaconda
 - TensorFlow
 - Orange
 - OpenCV
 - ETC



PS: - Bibliotecas Python

- Biblioteca Padrão

- os
- sys
- json, xml, csv
- xmlrpclib
- unittest e doctest
- logging
- iterators
- gzip, zlib, bz2, zipfile e tarfile
- math
- datetime, time
 - trabalha muito bem e de forma fácil com horas
- timit (medição de desempenho)
- asyncio (3.5)
- threading
- re (regex)
- Template (strings templates),
- urllib2 socket
- struct - empacotamento binario
- collections
- decimal

- Algumas Bibliotecas de Terceiros

- requests
- curl
- beautifulsoup
- Selenium
- Flask
- Django
- Plone
- Zope
- SQLAlchemy
- pyUnit, UnitPy, pytest
- PyQt, QtPy, Qt - PyGtk, Kivy, wxPython
- xlwt, xlrd, XlsxWriter
- pillow
- Scrapy
- OpenCV
- pandas
- scipy
- Matplotlib
- numpy
- scikit
- nltk
- <http://pypi-ranking.info/>

PS: - Quem usa Python

• ~~Sites Conhecidos (Core)~~

- ~~Youtube~~
- ~~Instagran~~
- ~~DropBox~~
- ~~Quora~~
- ~~Disqus~~
- ~~Reddit~~
- ~~Pinterest~~
- ~~NASA~~
- ~~Washington Post~~
- ~~Prezi~~
- ~~Bitbucket~~
- ~~Mozilla Firefox~~
- ~~Globe.com~~
- ~~Garas.com~~

• ~~Maiores Projetos Conhecidos~~

- ~~OpenStack~~
- ~~Ansible~~
- ~~YUM~~
- ~~Mercurial~~
- ~~Sentry~~
- ~~Ubuntu Software Center~~
- ~~Luigi~~
- ~~Airflow~~
- ~~Superset~~
- ~~OpenERP/Odoo~~
- ~~reCAPTCHA~~
- ~~Calibre~~
- ~~Jupyter Notebook~~
- ~~TensorFlow / Theano / Keras / Caffe 2~~

PS: - Quem usa Python

~~• Jogos~~

- ~~• 1024~~
- ~~• Civilization IV~~
- ~~• Frets on Fire~~
- ~~• Pirates of the Caribbean Online~~
- ~~• Disney's Toontown Online~~

~~• Usam Python em Partes~~

- ~~• Google~~
- ~~• Spotify~~
- ~~• Yahoo! Groups~~
- ~~• Battlefield 2~~
- ~~• The Sims 4~~
- ~~• Open Office/LibreOffice~~
- ~~• Blender~~
- ~~• QGIS~~
- ~~• Sublime~~
- ~~• RSS~~

~~• Aceitam Scripts em Python~~

- ~~• SQL Server~~
- ~~• PowerBI~~
- ~~• Open Office/LibreOffice~~
- ~~• ArcGIS~~
- ~~• Autodesk MotionBuilder(Autocad)~~
- ~~• GIMP~~
- ~~• Minecraft~~
- ~~• MySQL~~
- ~~• Vim~~
- ~~• Notepad++~~
- ~~• Blender~~

PS: - Quem usa Python

Todo mundo usa Python



[Home](#) > [Notícias](#) > [ComputerWorld](#)

Python está prestes a se tornar a linguagem de programação mais popular

A linguagem Python está mais perto do que nunca da posição número 1 do índice TIOBE

Redação

10:09 am - 14 de setembro de 2021