



python<sup>TM</sup>

## PYTHON: COMO USAR!?!



# A Origem da simplicidade!!!



- Tudo começou com este cara...



*Guido van Rossum, Anos 90.*

- No início tudo era trevas(C, C++, JAVA, etc).

- Ano da luz(1989) e o nome da criança é?

- Python, matemática e o filho pródigo.

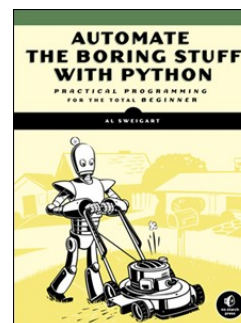
- Versões e o Pythonlabs

- Hoje em dia....

*Monty Python e o significado da vida(1983)*



*Data Science*



*Automatização*



*Python Web*



# Python e as empresas

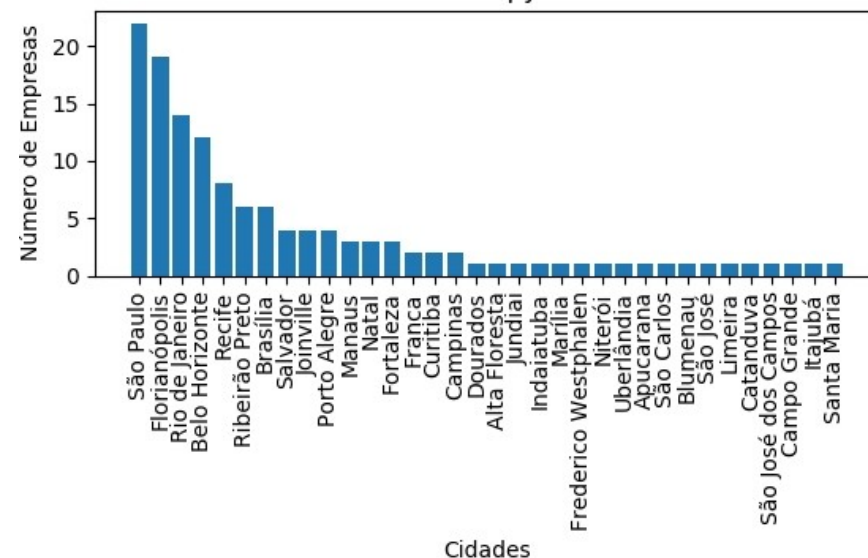


- **Rankings Brasileiros**

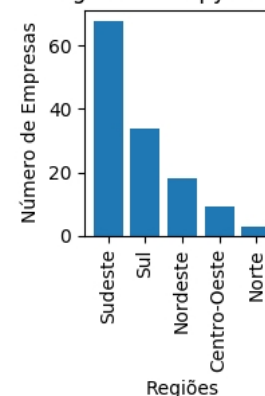
- Regiões do Brasil
- Estados
- Cidades
- Exemplos de empresas:

<http://python.org.br/empresas>

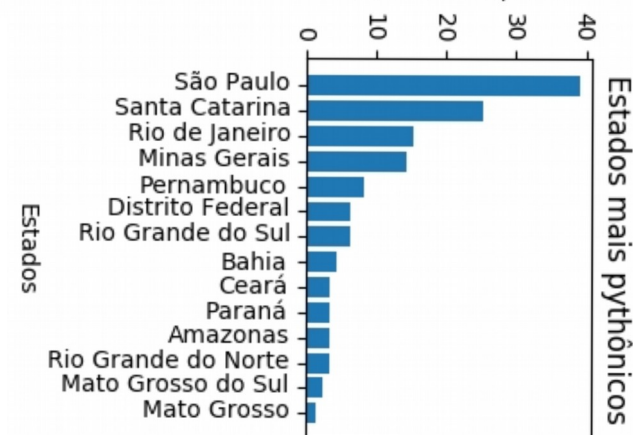
Cidades mais pythônicas



Regiões mais pythônicas



Estados mais pythônicos



# Guia Completo de Instalação



- **Tutorial no Windows**

- Download do python:
  - Python simples:
    - <https://www.python.org/downloads/>
  - Anaconda:
    - <https://www.anaconda.com/download/#windows>
- Configuração ao Vivo!!!!

# VAMOS COMEÇAR A BRINCAR!!!



- **Sintaxe Básica:**

- Regras do Python – Zen of Python
- Variáveis:
  - Tipos – Strings, Números(Decimal, Inteiros), Boolean
- Operadores lógicos
- Operadores aritméticos
- Laços:
  - Relacional
  - Loops
- Conjuntos – Listas, Tuplas, Dicionários e Conjuntos.

# VAMOS COMEÇAR A BRINCAR!!!



- **Sintaxe Avançada:**

- Estrutura de Dados:

- Ponteiros
    - Pilhas
    - Filas
    - Listas Encadeadas



- Programação Orientada a Objeto:

- Funções
    - Classes
    - Heranças, Construtores, Polimorfismos

# CHEGOU A HORA DE PRATICAR!



- **EXERCÍCIOS**

- Exercícios do básico ao avançado
  - Ficha de prática
    - [https://github.com/victorpenna/oficina\\_python\\_estacio/blob/master/ficha\\_oficina-python.pdf](https://github.com/victorpenna/oficina_python_estacio/blob/master/ficha_oficina-python.pdf)
- Dúvidas constantes, e agora!?!
  - Links interessantes:
    - <http://pythonclub.com.br/>
    - <https://pythonhelp.wordpress.com/>
  - Links da documentação:
    - <https://docs.python.org/3/index.html>
    - <http://www3.ifrn.edu.br/~jurandy/fdp/doc/aprenda-python/index.html>

# RECOMENDAÇÕES



- **Cursos:**

- Python para zumbis – Prof.º Fernando Masanori
- Udemy – Buscar por python...

- Indico links:

- <https://www.udemy.com/complete-python-bootcamp/>
    - <https://www.udemy.com/python-and-django-full-stack-web-developer-bootcamp/>

- Exercícios:

- <http://codingbat.com/python>
    - <https://www.reddit.com/r/dailyprogrammer/>
    - <http://www.pythonchallenge.com/>
    - [http://www.codeabbey.com/index/task\\_list](http://www.codeabbey.com/index/task_list)
    - <https://projecteuler.net/archives>



# NUNCA SE ESQUEÇAM!!!!!!



## • Zen of Python e as boas práticas.

```
import this
"""The Zen of Python, by Tim Peters. (poster by Joachim Jablon)"""

1 Beautiful is better than ugly.
2 Explicit is better than impl..
3 Simple is better than complex.
4 Complex is better than cOmp1|c@ted.
5 Flat is better than nested.
6 Sparse is better than dense.
7 Readability counts.
8 Special cases aren't special enough to break the rules.
9 Although practicality beats purity.
10 raise PythonicError("Errors should never pass silently.")
11 # Unless explicitly silenced.
12 In the face of ambiguity, refuse the temptation to guess.
13 There should be one-- and preferably only one --obvious way to do it.
14 # Although that way may not be obvious at first unless you're Dutch.
15 Now is better than ... never.
16 Although never is often better than rightnow.
17 If the implementation is hard to explain, it's a bad idea.
18 If the implementation is easy to explain, it may be a good idea.
19 Namespaces are one honking great idea -- let's do more of those!
```

– <https://zenorocha.com/licoes-aprendidas-com-o-mundo-python/>



**FINALIZANDO  
COM  
HELLO WORLD NO  
DJANGO.**



**OBRIGADO PELA ATENÇÃO.**

Meus contatos

**<https://github.com/victorpenna>**

**<https://www.facebook.com/victor.penna.29>**

Email:

**[victorp.info@yahoo.com.br](mailto:victorp.info@yahoo.com.br)**