



Introdução a Python e seu Ecossistema



Python?

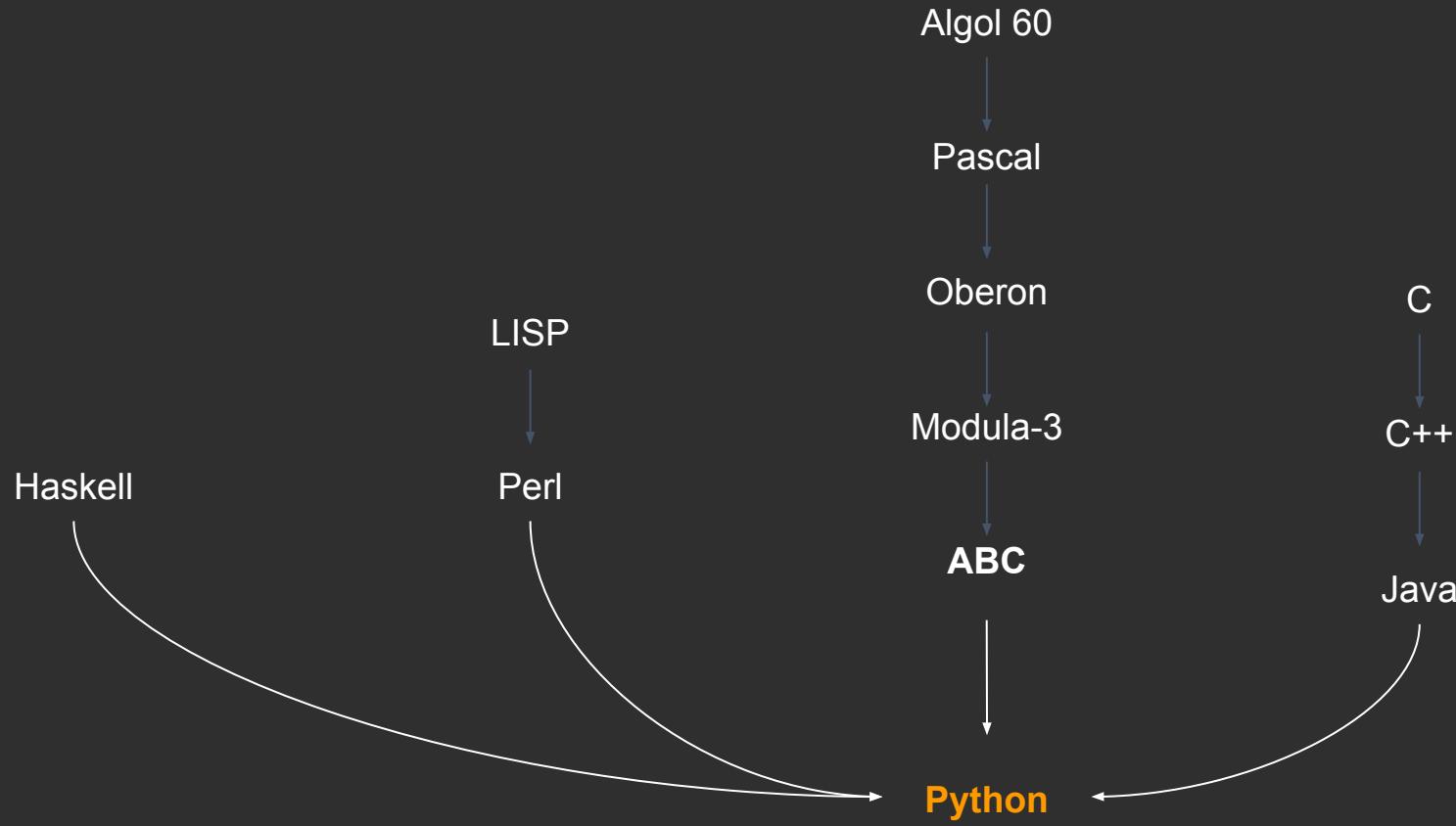


- Criada por Guido van Rossum e lançada em 1991
- Multi-paradigma
- Interpretada
- Tipagem dinâmica
- Expressiva
- O nome veio do programa de TV “*Monty Python’s Flying Circus*”

Motivações

- “*I decided to write an interpreter for the new scripting language I had been thinking about lately: a descendant of ABC that would appeal to Unix/C hackers*” (Guido van Rossum)
- Simplicidade acima de complexidade
- Ênfase na legibilidade do código
- Fácil de aprender

Origens



Filosofia

```
In [1]: import this
```

```
The Zen of Python, by Tim Peters
```

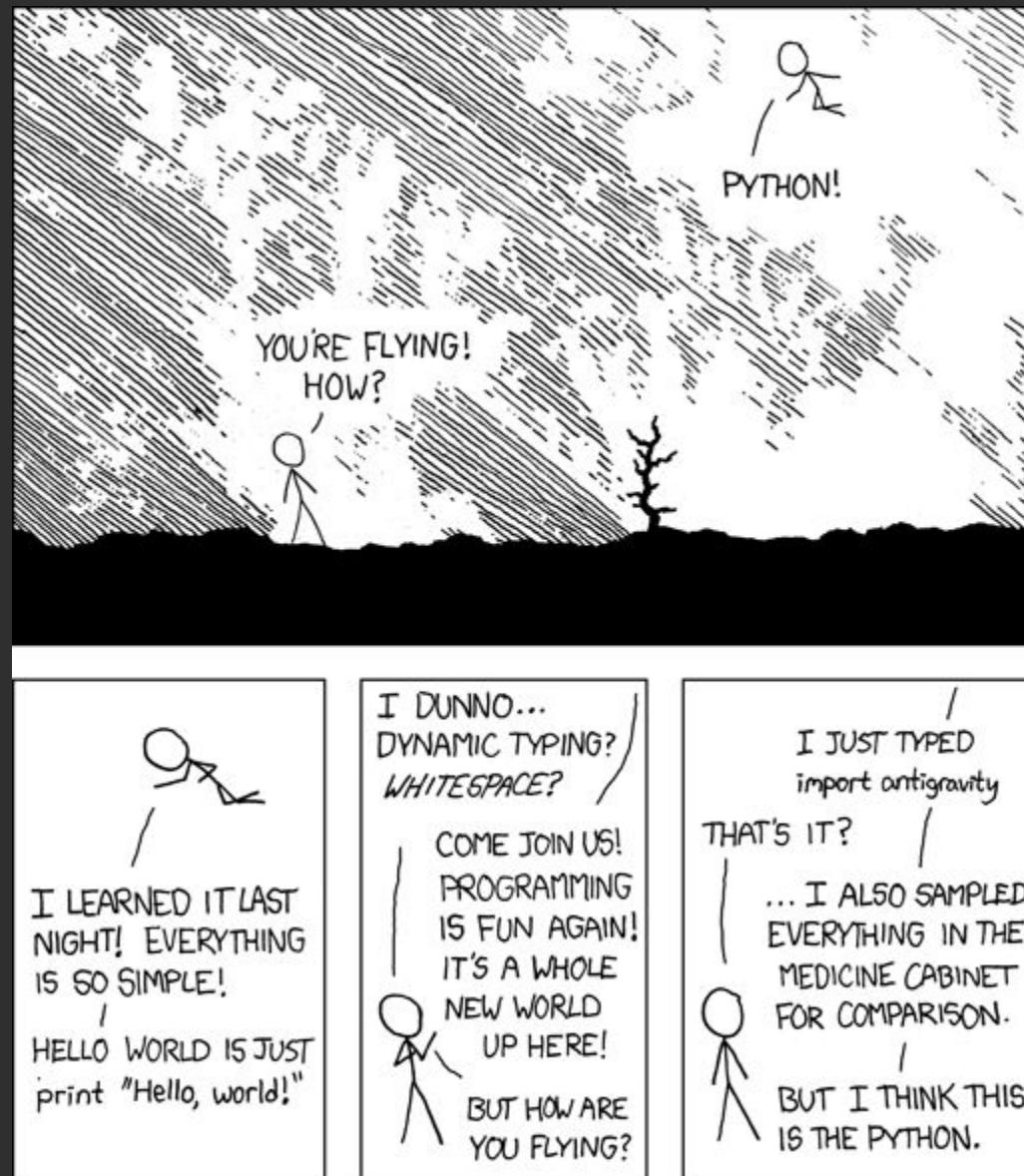
```
Beautiful is better than ugly.  
Explicit is better than implicit.  
Simple is better than complex.  
Complex is better than complicated.  
Flat is better than nested.  
Sparse is better than dense.  
Readability counts.  
Special cases aren't special enough to break the rules.  
Although practicality 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!
```

PEP8 (Python Enhancement Proposals)

PEP 8 -- Style Guide for Python Code

PEP:	8
Title:	Style Guide for Python Code
Author:	Guido van Rossum <guido at python.org>, Barry Warsaw <barry at python.org>, Nick Coghlan <ncoghlan at gmail.com>
Status:	Active
Type:	Process
Created:	05-Jul-2001
Post-History:	05-Jul-2001, 01-Aug-2013

Expressividade

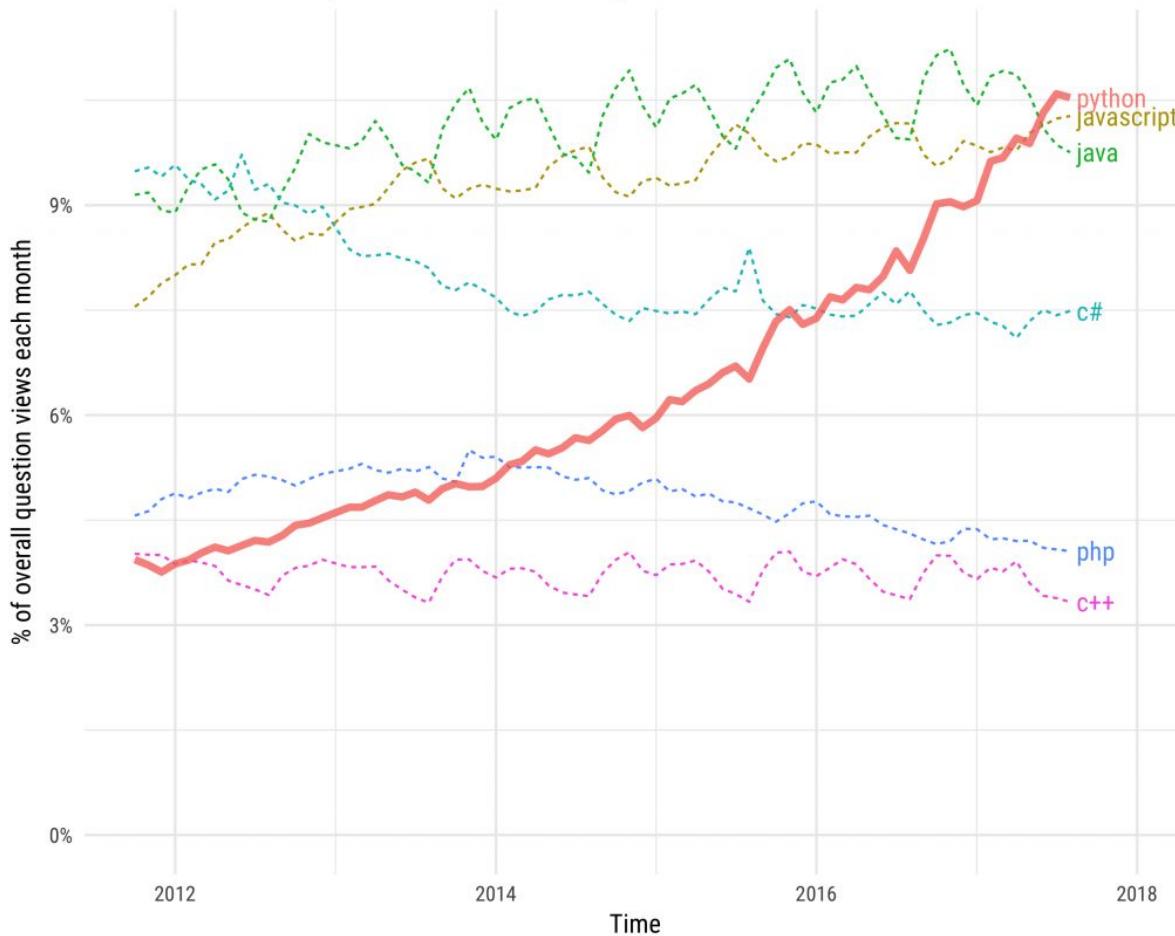


<https://xkcd.com/353/>

Popularidade

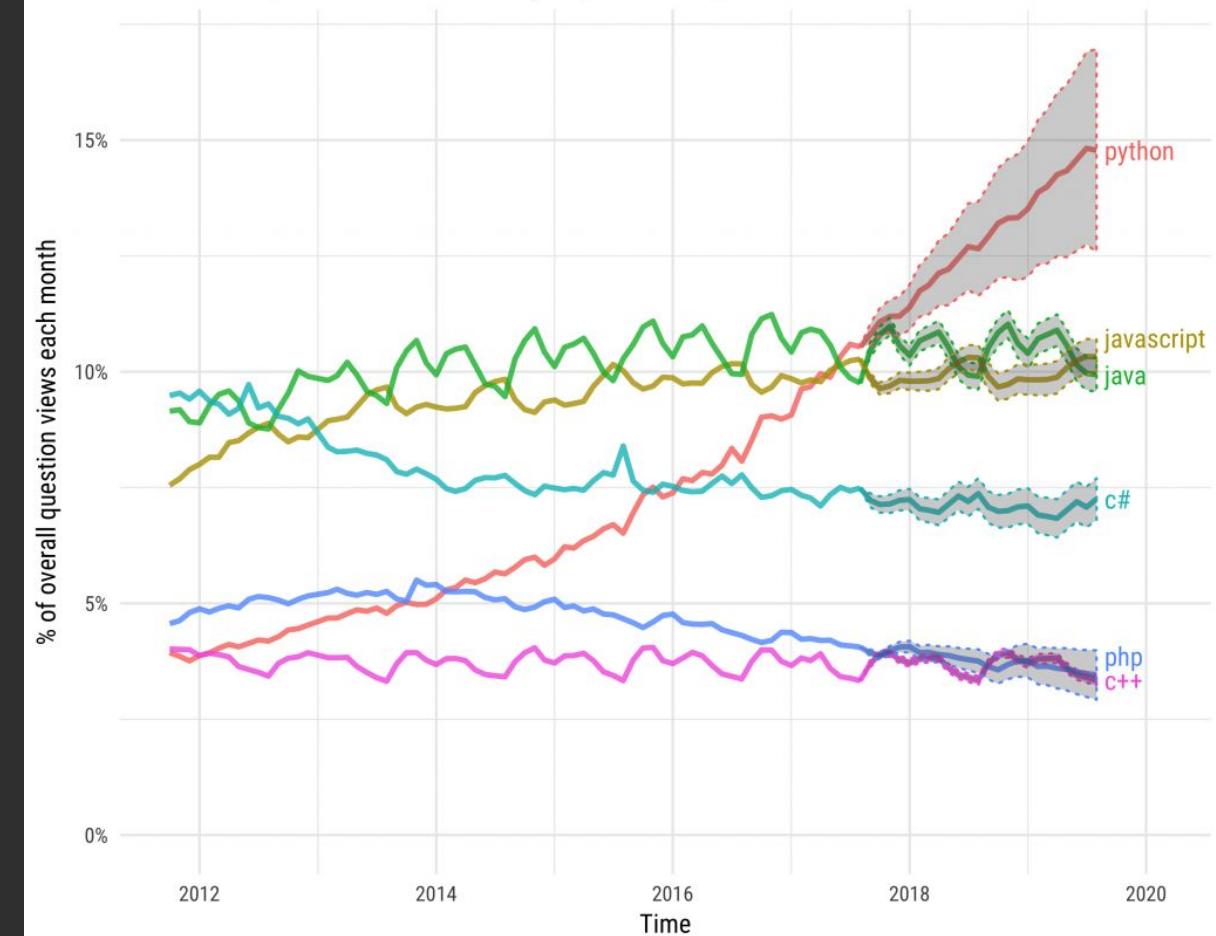
Growth of major programming languages

Based on Stack Overflow question views in World Bank high-income countries



Projections of future traffic for major programming languages

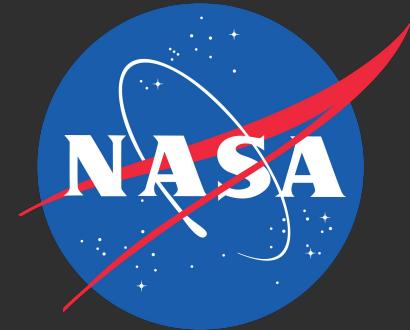
Future traffic is predicted with an STL model, along with an 80% prediction interval.



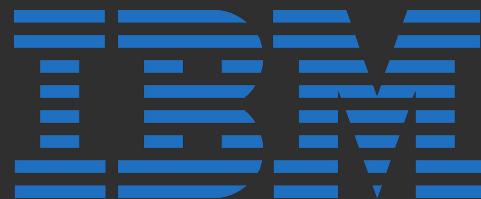
Quem usa?



moz://a



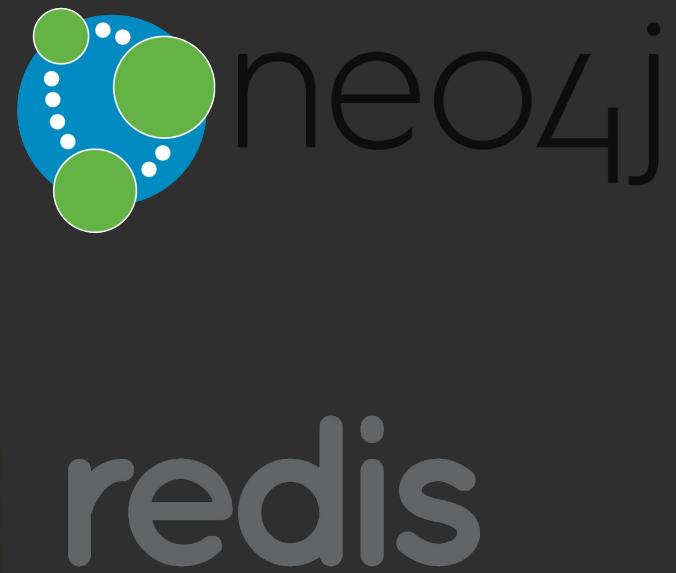
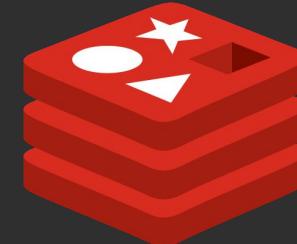
Google



Bibliotecas - Bancos de dados



PostgreSQL



Bibliotecas - Desenvolvimento Web

WEB2PY



django



gunicorn



Flask

web development,
one drop at a time

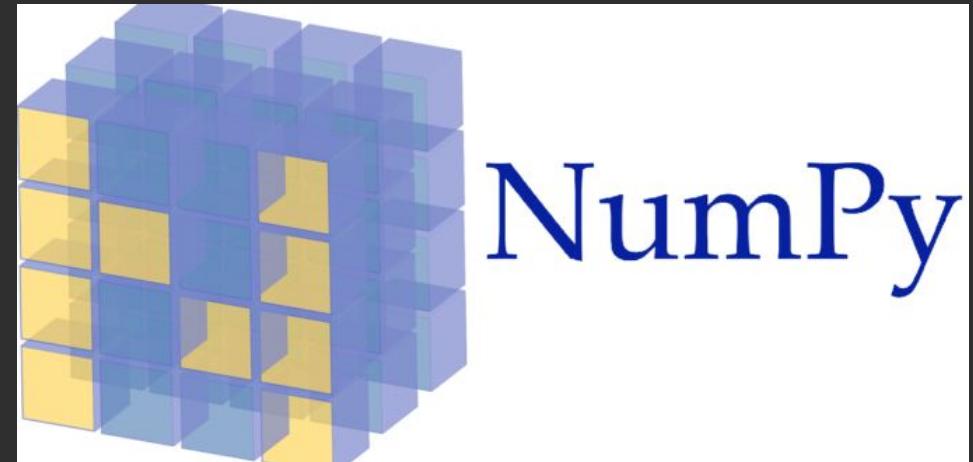
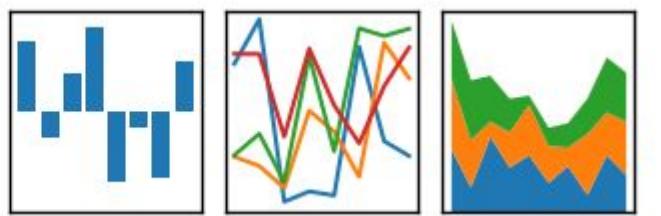


Jinja2

Bibliotecas - Análise de dados

pandas

$$y_{it} = \beta' x_{it} + \mu_i + \epsilon_{it}$$





IP[y]: IPython
Interactive Computing



Bibliotecas - Machine Learning



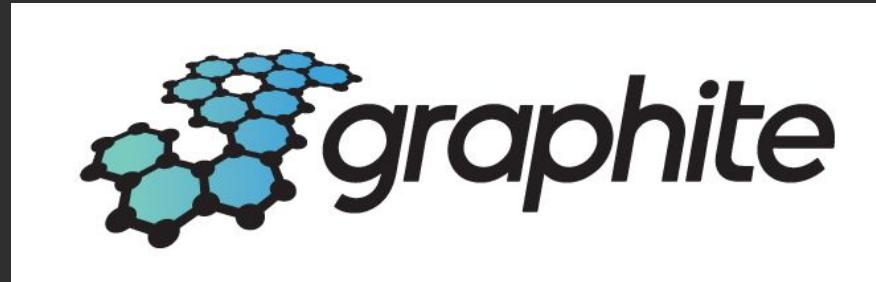
theano

↑GRPC↓

ØMQ



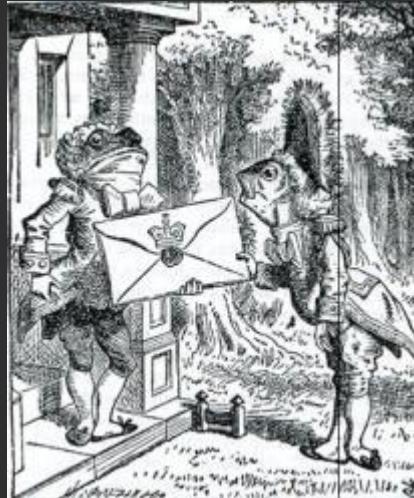
Bibliotecas - Monitoramento



Bibliotecas - Web Scraping



Scrapy



Requests
http for humans

<docopt>

pip

virtualenv

easy_install



Implementações

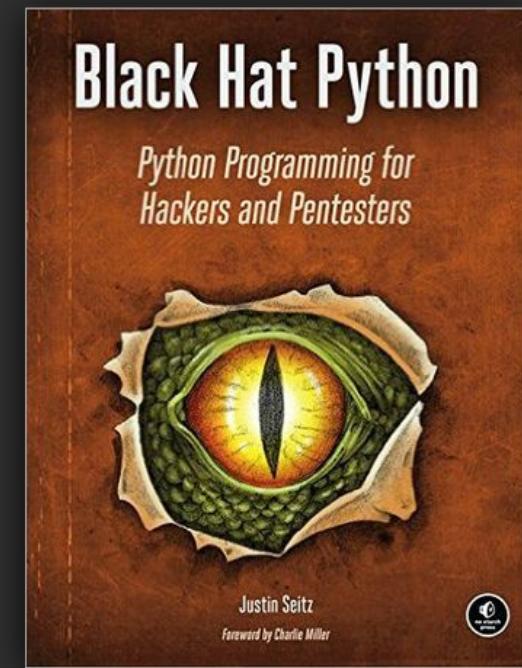
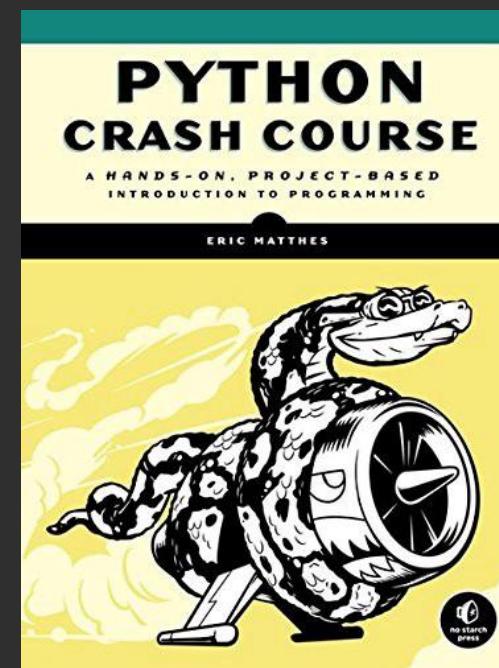


IronPython

Onde aprender mais?

- <https://github.com/vinta/awesome-python>
- <https://www.codecademy.com/learn/learn-python>
- <https://learnxinyminutes.com/docs/python/>
- <http://docs.python-guide.org/en/latest/>
- <https://www.fullstackpython.com>

Livros



Documentação

The screenshot shows a documentation page for PyToolz. At the top left is a sidebar with a blue header containing the 'Toolz' logo and the word 'latest'. Below the header is a search bar labeled 'Search docs'. To the right of the sidebar, the main content area has a header 'Docs » PyToolz API Documentation' and a 'Edit on GitHub' button. The main title 'PyToolz API Documentation' is centered above a paragraph of text. Below the text, there's a section titled 'Functional Virtues' with a bulleted list. Further down, another section is introduced with its own bulleted list. At the bottom of the page, a summary sentence is followed by a 'Read the Docs' footer.

Docs » PyToolz API Documentation

>Edit on GitHub

PyToolz API Documentation

Toolz provides a set of utility functions for iterators, functions, and dictionaries. These functions interoperate well and form the building blocks of common data analytic operations. They extend the standard libraries *itertools* and *functools* and borrow heavily from the standard libraries of contemporary functional languages.

Toolz provides a suite of functions which have the following functional virtues:

- **Composable:** They interoperate due to their use of core data structures.
- **Pure:** They don't change their inputs or rely on external state.
- **Lazy:** They don't run until absolutely necessary, allowing them to support large streaming data sets.

Toolz functions are *pragmatic*. They understand that most programmers have deadlines.

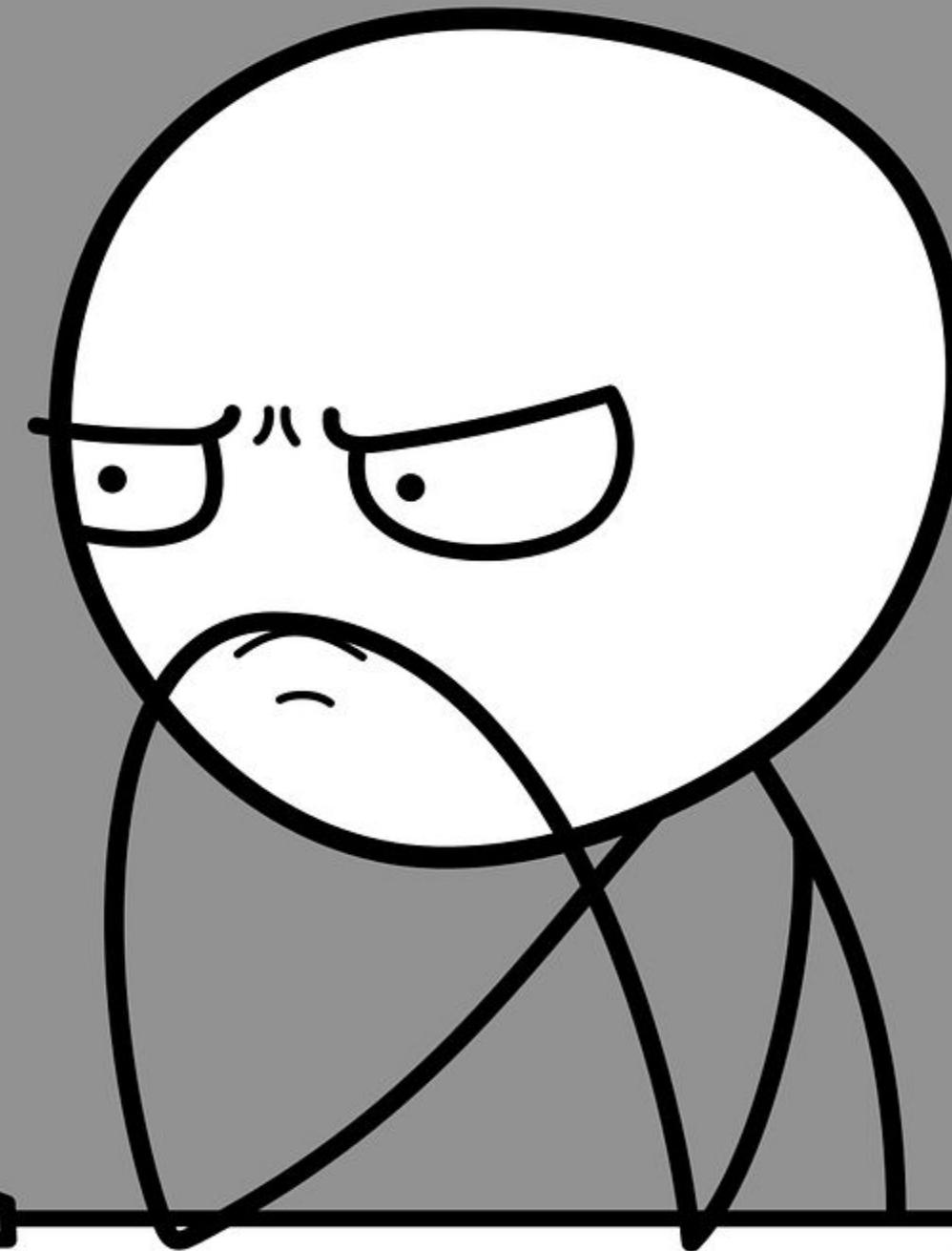
- **Low Tech:** They're just functions, no syntax or magic tricks to learn
- **Tuned:** They're profiled and optimized
- **Serializable:** They support common solutions for parallel computing

This gives developers the power to write *powerful* programs to solve *complex problems* with

Read the Docs v: latest ▾



**ESTRUTURA E
SINTAXE**



Tipos de dados

- Booleano True ou False
- Números 3, 3.141592, 5 + 3j
- Strings “Isso é um texto”
- Listas [1, 2, 3, 4, 5]
- Tuplas (1, 2, 3, 4, 5)
- Sets {1, 2, 3, 4, 5}
- Dicionários { ‘a’: 1, ‘b’: 2, ‘c’: 3 }
- Bytes b‘The type of this is bytes’

Controle de fluxo

```
x = 10
if x == 10:
    print('x is equal to 10')
else:
    print('x is NOT equal to 10')
```

```
x = 0
while x <= 100:
    print(x)
    x += 1
```

```
numbers = [1, 2, 3, 4, 5, 6, 7, 8, 9]

for number in numbers:
    print(number)
```

```
numbers = [1, 2, 3, 4, 5, 6, 7, 8, 9]

for i in range(len(numbers)):
    print(numbers[i])
```

```
even = []
for x in range(1000):
    if x % 2 == 0:
        even.append(x)
    if x == 113:
        break
```

Funções

```
def greet():
    print("Hello")

def greet_person(name):
    print("Hello " + name + "!")

def greet_person(name='Pessoa'):
    print("Hello " + name + "!")

def greet_people(*people):
    for person in people:
        greet_person(person)
```

Funções

```
def fn(*args, **kwargs):
    return args, kwargs

fn('hello', 1, 2, 3, 4, ['bye'], x=1, y=2, z=3)
#(('hello', 1, 2, 3, 4, ['bye']), {'x': 1, 'y': 2, 'z': 3})
```

Listas

```
l1 = []

l2 = [1, 3, 5, 7, 9, 11]

l3 = [3, 5, 3.141592, "Hello", dict()]

l4 = [[1, 2, 3], [4, 5, 6], [7, 8, 9]]

l5 = [x for x in range(12) if x % 2 != 0]
# [1, 3, 5, 7, 9, 11]

l6 = [x*x for x in l2]
# [1, 9, 25, 49, 81, 121]

l6 = [str(x*x) for x in l2]
# ['1', '9', '25', '49', '81', '121']
```

Dicionários (Hash-map)

```
d1 = {}

d2 = {'a': 1, 'b': 2, 'c': 3}

d3 = {1: 'a', '1': 'b', True: 3}

d4 = {'a': {'b': 13}, 'c': [1, 2, 4]}

d5 = {k: v for k,v in [( 'a', 1), ('b', 2), ('c',3)]}
# {'a': 1, 'b': 2, 'c': 3}

d6 = dict(zip(['a', 'b', 'c'], [1, 2, 3]))
# {'a': 1, 'b': 2, 'c': 3}
```

Exceções

```
In [1]: def divide_by_number(number):
...:     return 1/number
...:

In [2]: divide_by_number(0)

ZeroDivisionError                                 Traceback (most recent call last)
<ipython-input-2-9887ff30e17a> in <module>()
----> 1 divide_by_number(0)

<ipython-input-1-4e74b51e7696> in divide_by_number(number)
      1 def divide_by_number(number):
----> 2     return 1/number

ZeroDivisionError: division by zero
```

Exceções

```
In [1]: def divide_by_number(number):
...:     try:
...:         return 1/number
...:     except ZeroDivisionError as err:
...:         return f'divide_by_number failed: {str(err)}'
...:

In [2]: divide_by_number(0)
Out[2]: 'divide_by_number failed: division by zero'
```

I/O - Formatação

```
In [1]: "Bom dia, %s. O que você comeu no dia %s?" % ("Felipe", "27 de janeiro")
```

```
Out[1]: 'Bom dia, Felipe. O que você comeu no dia 27 de janeiro?'
```

```
In [2]: "Bom dia, {}. O que você comeu no dia {}?".format("Felipe", "27 de janeiro")
```

```
Out[2]: 'Bom dia, Felipe. O que você comeu no dia 27 de janeiro?'
```

```
In [3]: name, day = "Felipe", "27 de janeiro"
```

```
In [4]: f"Bom dia, {name}. O que você comeu no dia {day}?"
```

```
Out[4]: 'Bom dia, Felipe. O que você comeu no dia 27 de janeiro?'
```

I/O - Arquivos

```
In [1]: import json

In [2]: from pprint import pprint

In [3]: with open('config.json', 'r') as f:
...:     raw_file = f.read()
...:

In [4]: pprint(raw_file)
({'\n':
  '    "URL": "https://raccoon.ag",\n'
  '    "API_PORT": 10000,\n'
  '    "MAX_ATTEMPTS": 5,\n'
  '    "BACKUP_COMMAND": "tar cvf backup.tar.gz /var/lib/data"\n'
} )\n')

In [5]: json.loads(raw_file)
Out[5]:
{'API_PORT': 10000,
 'BACKUP_COMMAND': 'tar cvf backup.tar.gz /var/lib/data',
 'MAX_ATTEMPTS': 5,
 'URL': 'https://raccoon.ag'}
```

I/O - Logging

```
In [1]: import sys

In [2]: import logging

In [3]: logger = logging.getLogger('cinac_logger', level=logging.INFO)
....: handler = logging.StreamHandler(stream=sys.stdout)
....: fmt = '[%(levelname)s][%(asctime)s] | %(message)s'
....: handler.setFormatter(logging.Formatter(fmt))
....: logger.addHandler(handler)
....:

In [4]: logger.info("I'm an INFO message!")
[INFO][2018-05-15 18:01:13,335] | I'm an INFO message!

In [5]: logger.error("I'm an ERROR message!")
[ERROR][2018-05-15 18:01:26,679] | I'm an ERROR message!
```

I/O - STDIN e STDOUT

```
In [1]: name = input("What's your name?\n")
```

```
What's your name?
```

```
João
```

```
In [2]: home = input("Where do you live?\n")
```

```
Where do you live?
```

```
Logo ali
```

```
In [3]: f"Hello, {name} que mora {home}"
```

```
Out[3]: 'Hello, João que mora Logo ali'
```

Classes e métodos

```
class Polygon:  
    def __init__(self, sides, name):  
        self.sides = sides  
        self.name = name  
  
    def area(self):  
        raise NotImplementedError
```

```
class Square(Polygon):  
    def __init__(self, side):  
        super().__init__(4, "square")  
        self.side  
  
    def area(self):  
        return side * side  
  
class Triangle(Polygon):  
    def __init__(self, base, height):  
        super().__init__(3, "triangle")  
        self.base = base  
        self.height = height  
  
    def area(self):  
        return (base * height)/2
```

Imports

```
In [1]: import math  
  
In [2]: math.sqrt(9)  
Out[2]: 3.0  
  
In [3]: from math import sqrt  
  
In [4]: sqrt(9)  
Out[4]: 3.0  
  
In [5]: from math import sqrt as MyAwEs0m3sQr7  
  
In [6]: MyAwEs0m3sQr7(9)  
Out[6]: 3.0
```

Imports

```
bisca@blackeye:~/Projetos/awesome-project$ tree
.
└── pkg1
    └── __init__.py
├── pkg2
    └── __init__.py
└── pkg3
    └── main.py
└── run.py

3 directories, 4 files
```

```
In [1]: import pkg1
Olar, eu sou o pkg1

In [2]: import pkg2
Olar, eu sou o pkg2

In [3]: import pkg3
Olar, eu sou o pkg3

In [4]: from pkg3 import main
Olar, eu sou o pkg3

In [5]: 
```

virtualenv e pip

```
bisca@blackeye:~/Projetos/awesome-project$ virtualenv -p python3.6 env
Already using interpreter /usr/local/bin/python3.6
Using base prefix '/usr/local'
New python executable in /home/bisca/Projetos/awesome-project/env/bin/python3.6
Also creating executable in /home/bisca/Projetos/awesome-project/env/bin/python
Installing setuptools, pip, wheel...done.
bisca@blackeye:~/Projetos/awesome-project$ which python
/usr/bin/python
bisca@blackeye:~/Projetos/awesome-project$ source env/bin/activate
(env) bisca@blackeye:~/Projetos/awesome-project$ which python
/home/bisca/Projetos/awesome-project/env/bin/python
(env) bisca@blackeye:~/Projetos/awesome-project$ ls env/
bin  include  lib  pip-selfcheck.json
(env) bisca@blackeye:~/Projetos/awesome-project$ █
```

virtualenv e pip

```
(env) bisca@blackeye:~/Projetos/awesome-project$ pip install flask
Collecting flask
  Downloading https://files.pythonhosted.org/packages/7f/e7/08578774ed4536d3242b14dacb4696386634607af824ea99
    100% |██████████| 92kB 1.9MB/s
Collecting Jinja2>=2.10 (from flask)
  Downloading https://files.pythonhosted.org/packages/7f/ff/ae64bacdfc95f27a016a7bed8e8686763ba4d277a78ca761
    100% |██████████| 133kB 2.9MB/s
Collecting itsdangerous>=0.24 (from flask)
Collecting Werkzeug>=0.14 (from flask)
  Downloading https://files.pythonhosted.org/packages/20/c4/12e3e56473e52375aa29c4764e70d1b8f3efa6682bef8d0a
    100% |██████████| 327kB 3.5MB/s
Collecting click>=5.1 (from flask)
  Downloading https://files.pythonhosted.org/packages/34/c1/8806f99713ddb993c5366c362b2f908f18269f8d792aff1a
    100% |██████████| 71kB 5.7MB/s
Collecting MarkupSafe>=0.23 (from Jinja2>=2.10->flask)
Installing collected packages: MarkupSafe, Jinja2, itsdangerous, Werkzeug, click, flask
Successfully installed Jinja2-2.10 MarkupSafe-1.0 Werkzeug-0.14.1 click-6.7 flask-1.0.2 itsdangerous-0.24
```

virtualenv e pip

```
(env) bisca@blackeye:~/Projetos/awesome-project$ pip freeze
click==6.7
Flask==1.0.2
itsdangerous==0.24
Jinja2==2.10
MarkupSafe==1.0
Werkzeug==0.14.1
(env) bisca@blackeye:~/Projetos/awesome-project$ 
```

```
# requirements.txt

click==6.7
Flask==1.0.2
itsdangerous==0.24
Jinja2==2.10
MarkupSafe==1.0
Werkzeug==0.14.1
```

```
(env) bisca@blackeye:~/Projetos/awesome-project$ pip install -r requirements.txt
Collecting click==6.7 (from -r requirements.txt (line 3))
  Using cached https://files.pythonhosted.org/packages/34/c1/8806f99713ddb993c5366c362b2f
Collecting Flask==1.0.2 (from -r requirements.txt (line 4))
  Using cached https://files.pythonhosted.org/packages/7f/e7/08578774ed4536d3242b14dacb46
Collecting itsdangerous==0.24 (from -r requirements.txt (line 5))
Collecting Jinja2==2.10 (from -r requirements.txt (line 6))
  Using cached https://files.pythonhosted.org/packages/7f/ff/ae64bacdfc95f27a016a7bed8e86
Collecting MarkupSafe==1.0 (from -r requirements.txt (line 7))
Collecting Werkzeug==0.14.1 (from -r requirements.txt (line 8))
  Using cached https://files.pythonhosted.org/packages/20/c4/12e3e56473e52375aa29c4764e70
Installing collected packages: click, MarkupSafe, Jinja2, itsdangerous, Werkzeug, Flask
Successfully installed Flask-1.0.2 Jinja2-2.10 MarkupSafe-1.0 Werkzeug-0.14.1 click-6.7 i
(env) bisca@blackeye:~/Projetos/awesome-project$ 
```



**SHUT UP AND SHOW ME
THE CODE!**

Exemplos

<https://github.com/felipemocruha/semcomp-beta>

Obrigado

alcantara@raccoon.ag

