Python hasta para cepillarse los dientes

Agenda

- Usos de Python
- Beneficios
- Comunidad
- Perlas
- Mythbuster/FAQ

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!
```

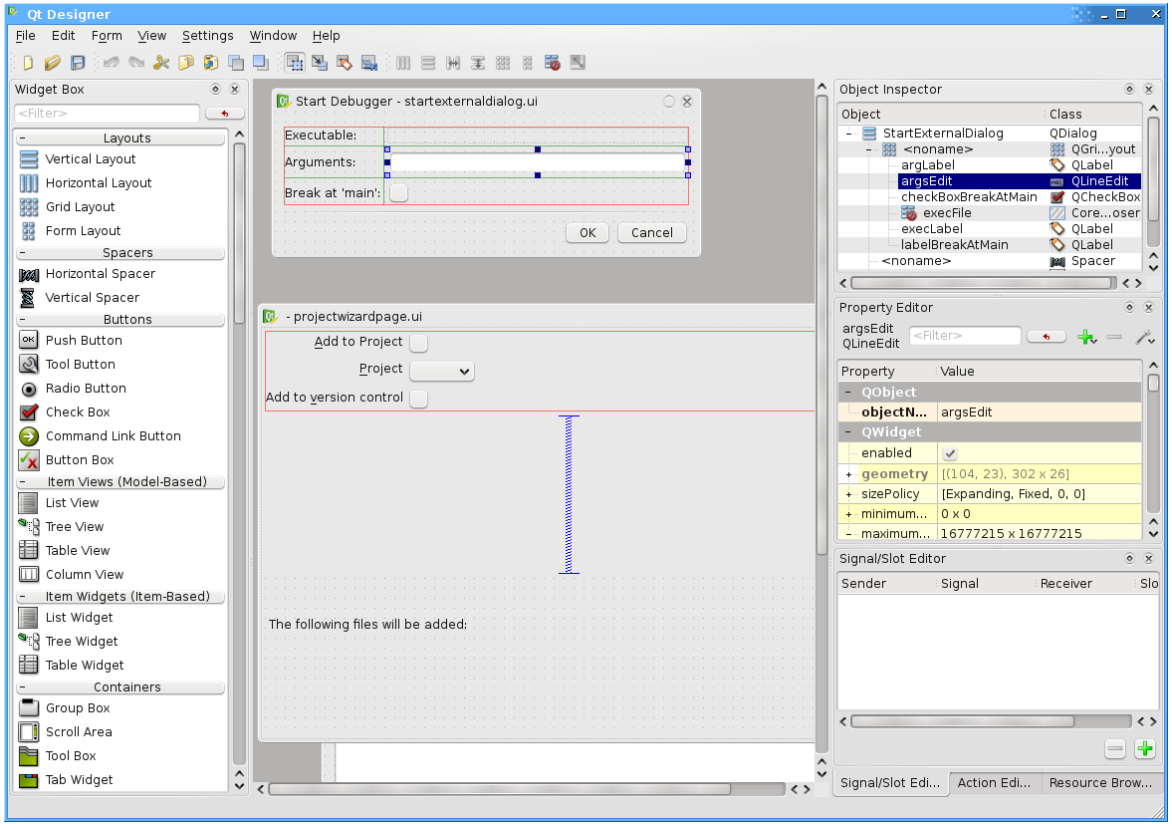
Usos de Python - Scripting

Uno de los usos inciales y más comunes de Python

```
#!/usr/bin/python
import os
source_dir = 'images'
target_dir = 'new_images/'
# Move and rename all images in the source directory
for num, fname in enumerate(os.listdir(source_dir)):
    source_file = os.path.join(source_dir, fname)
    new_file = os.path.join(target_dir,
                            "%i-cool_image.JPG" % num)
    print("%s -> %s" %  (source_file, new_file))
    os.rename(os.path.join(source_dir, fname),
              new_file)
```

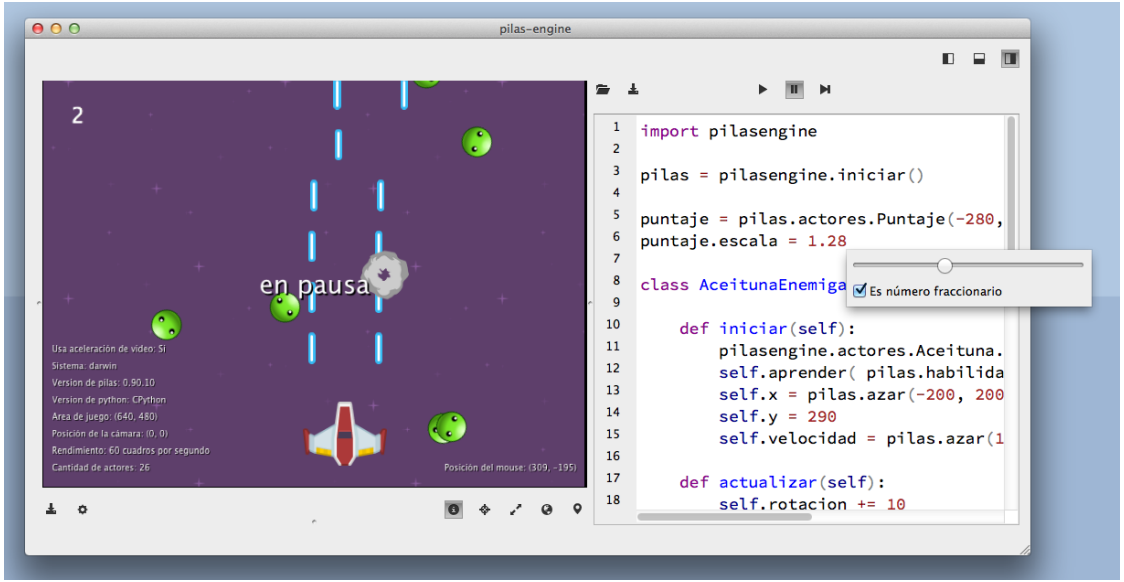
Usos de Python - Web

Usos de Python - Desktop



- pyQt
- wxPython
- tkinter

Usos de Python - Juegos

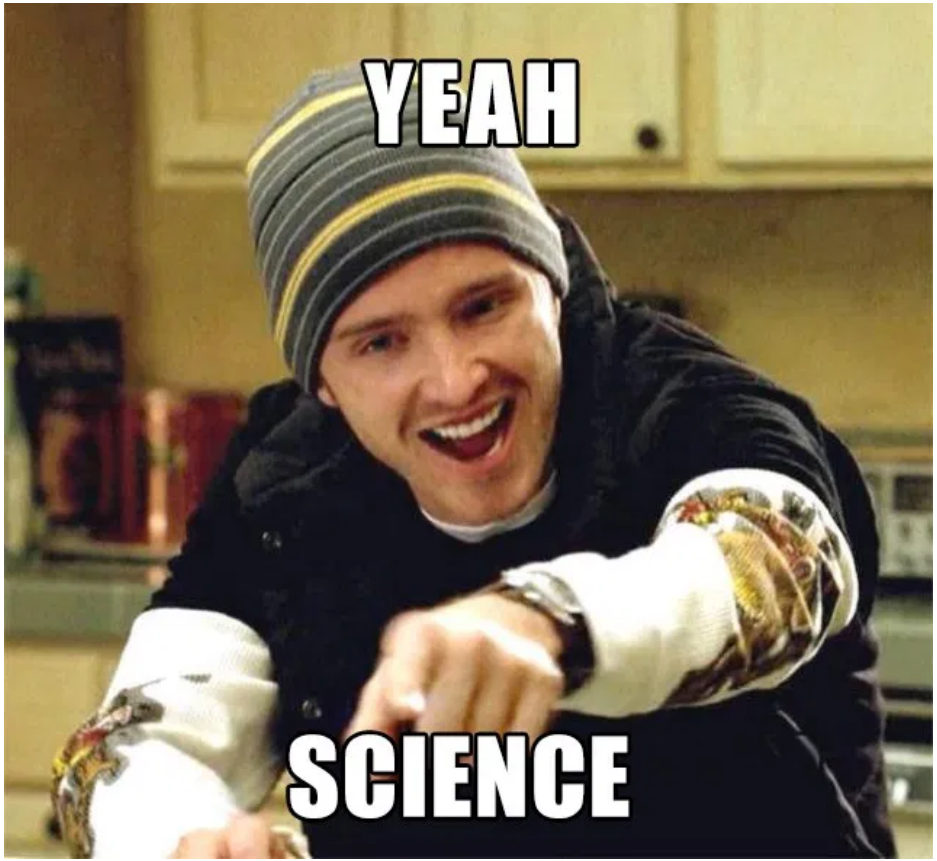


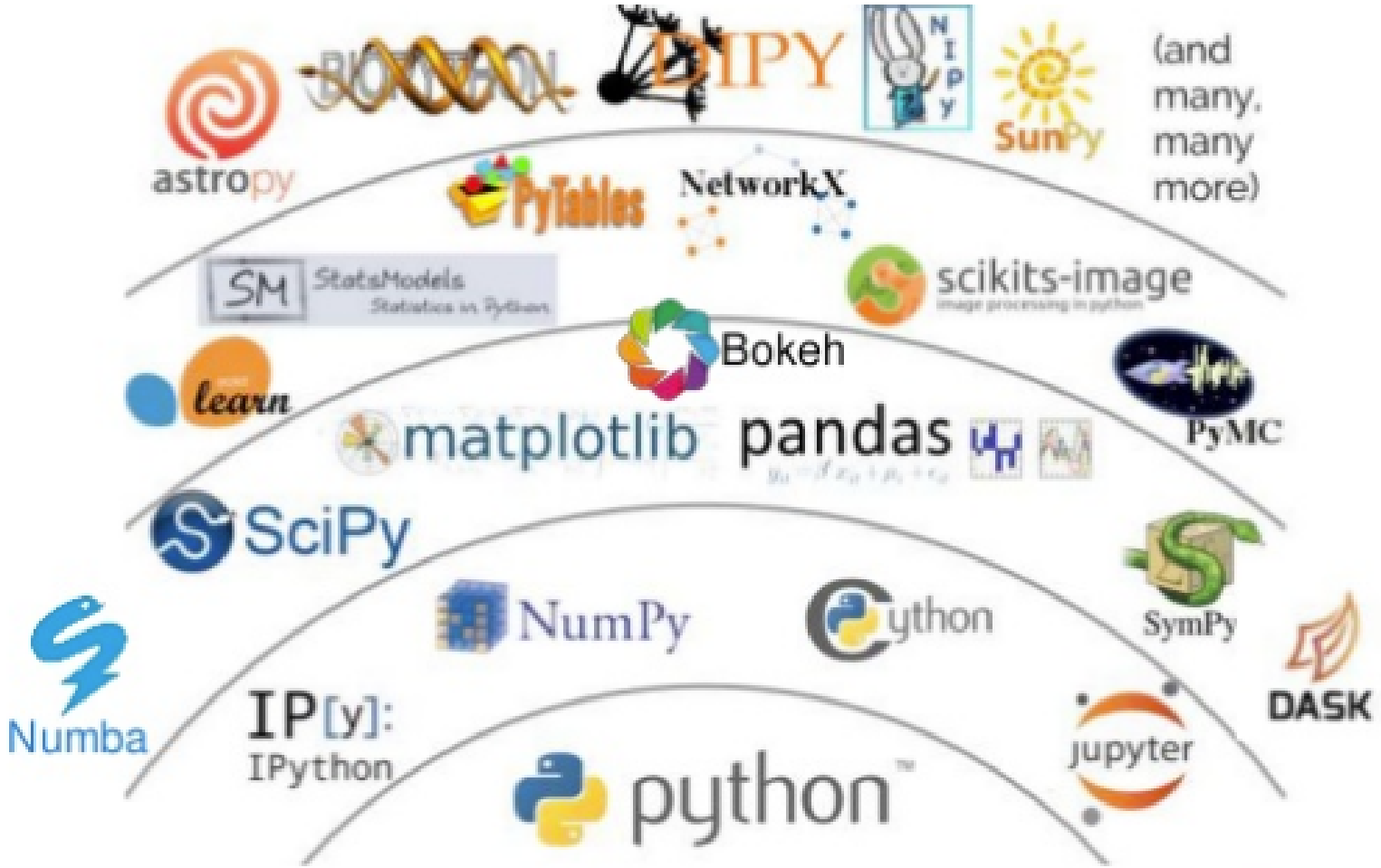
- pygame
- Pilas Engine (arg)
- Cocos2d (arg)

Usos de Python - Mobile

kivy

Usos de Python – Ciencia




IPY
(and many, many more)
astropy
PyTables
NetworkX
StatsModels
Statistics in Python
scikits-image
Bokeh
learn
matplotlib
pandas
PyMC
SciPy
NumPy
Cython
SymPy
DASK
Numba
IP[y]:
IPython
jupyter
python

Beneficios

Entrega rápida

Fácil de aprender

Open source! Legible

Debugging y REPL

Baterías incluidas

- python -m http.server
- CSV, sqlite3, tempfile
- request, os, unittest
- email, json
- asyncio, cryptografia

Comunidad


python
SOFTWARE
FOUNDATION

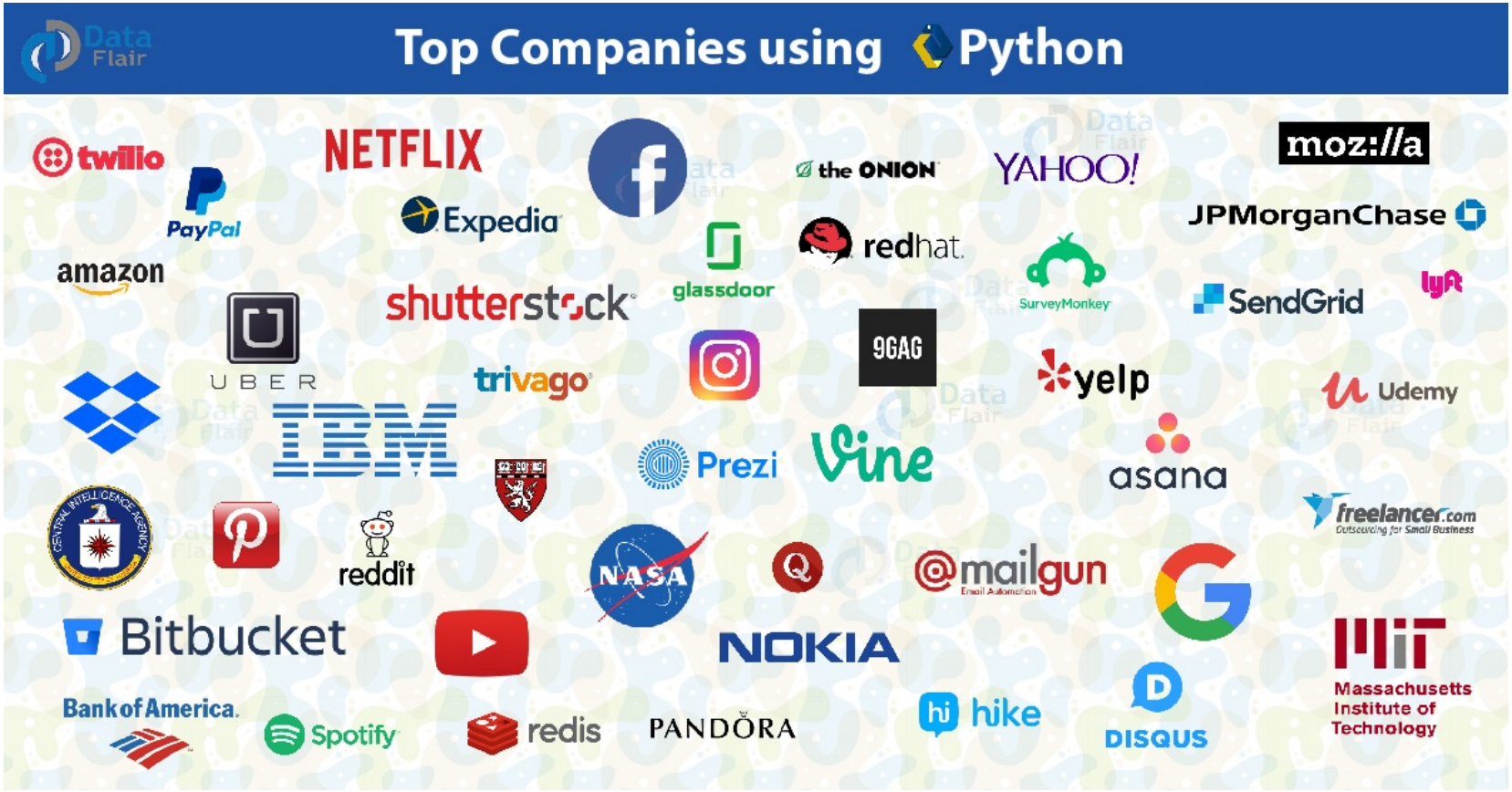

PyAr


django
girls


2019
CABA
PyConAr

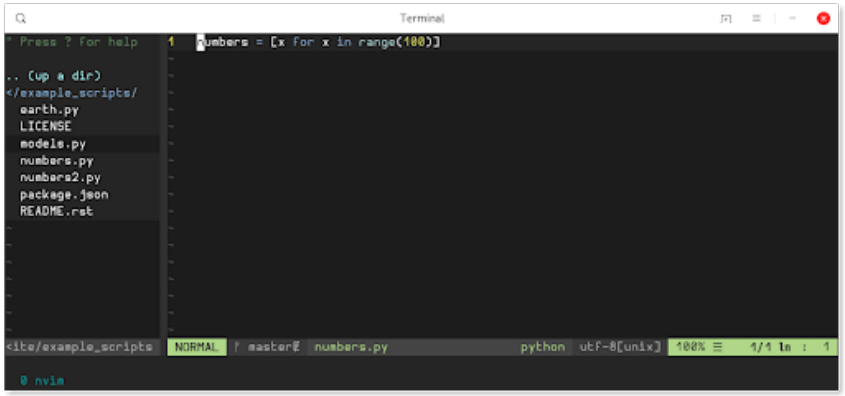

PyData

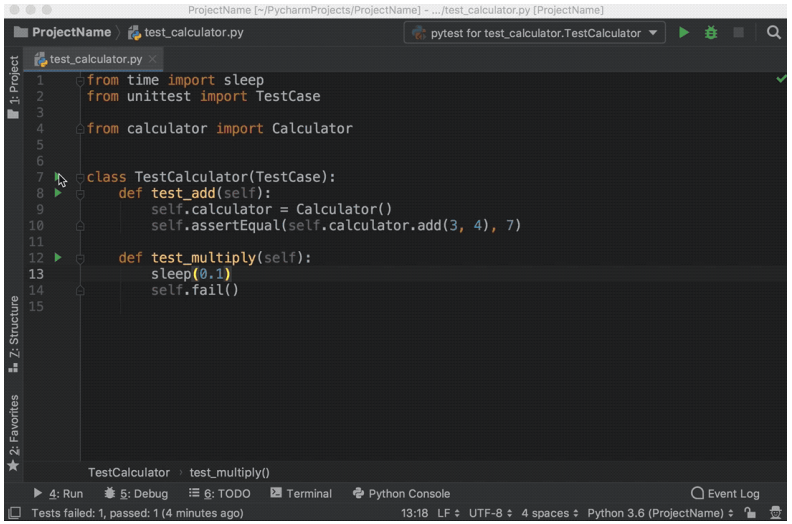
Empresas

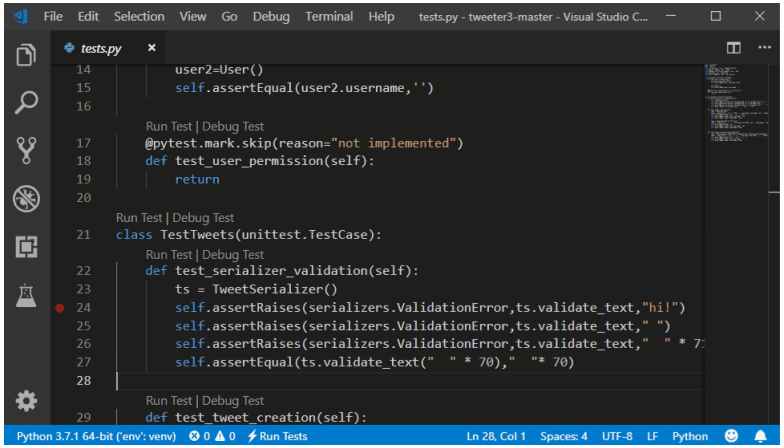

Data Flair
Top Companies using Python
twilio
PayPal
NETFLIX
Expedia
the ONION
YAHOO!
moz://a
JPMorganChase
amazon
glassdoor
redhat.
SurveyMonkey
SendGrid
lyft
shutterstock
UBER
trivago
9GAG
yelp
Udemy
IBM
Prezi
Vine
asana
freelancer.com
Outsourcing for Small Business
reddit
NASA
@mailgun
Email Automation
Bitbucket
NOKIA
Massachusetts Institute of Technology
Bank of America
Spotify
redis
PANDORA
hike
DISQUS

Empresas Argentinas

Editores


Terminal
" Press ? for help
.. (up a dir)
</example_scripts/
earth.py
LICENSE
models.py
numbers.py
numbers2.py
package.json
README.rst
numbers = [x for x in range(100)]
<ite/example_scripts
NORMAL
master
numbers.py
python
utf-8[unix]
100%
1/1 ln : 1
" nvim


ProjectName [~/PycharmProjects/ProjectName] - .../test_calculator.py [ProjectName]
ProjectName
test_calculator.py
pytest for test_calculator.TestCalculator
test_calculator.py
1: Project
2: Favorites
Z: Structure
from time import sleep
from unittest import TestCase
from calculator import Calculator
class TestCalculator(TestCase):
def test_add(self):
self.calculator = Calculator()
self.assertEqual(self.calculator.add(3, 4), 7)
def test_multiply(self):
sleep(0.1)
self.fail()
TestCalculator
test_multiply()
4: Run
5: Debug
6: TODO
Terminal
Python Console
Event Log
Tests failed: 1, passed: 1 (4 minutes ago)
13:18
LF
UTF-8
4 spaces
Python 3.6 (ProjectName)


File
Edit
Selection
View
Go
Debug
Terminal
Help
tests.py - tweeter3-master - Visual Studio C...
tests.py
user2=User()
self.assertEqual(user2.username,'')
Run Test | Debug Test
@pytest.mark.skip(reason="not implemented")
def test_user_permission(self):
return
Run Test | Debug Test
class TestTweets(unittest.TestCase):
Run Test | Debug Test
def test_serializer_validation(self):
ts = TweetSerializer()
self.assertRaises(serializers.ValidationError,ts.validate_text,"hi!")
self.assertRaises(serializers.ValidationError,ts.validate_text," ")
self.assertRaises(serializers.ValidationError,ts.validate_text," " * 7
self.assertEqual(ts.validate_text(" " * 70)," "* 70)
Run Test | Debug Test
def test_tweet_creation(self):
Python 3.7.1 64-bit ('env': venv)
Run Tests
Ln 28, Col 1
Spaces: 4
UTF-8
LF
Python

Mythbuster/FAQ

- Soporta async
- Chequeo de tipado
- Optimizable a C/ C++
- Se compila a .pyc

PREGUNTAS?

RESPUESTAS?

Matias Barriento

- Programador / Payaso / Amante de listar cosas
- Socio de Python Argentina
- Trabajo en Kilimo
- Doy charlas (si, caíste de orto)

¡MUCHAS GRACIAS!