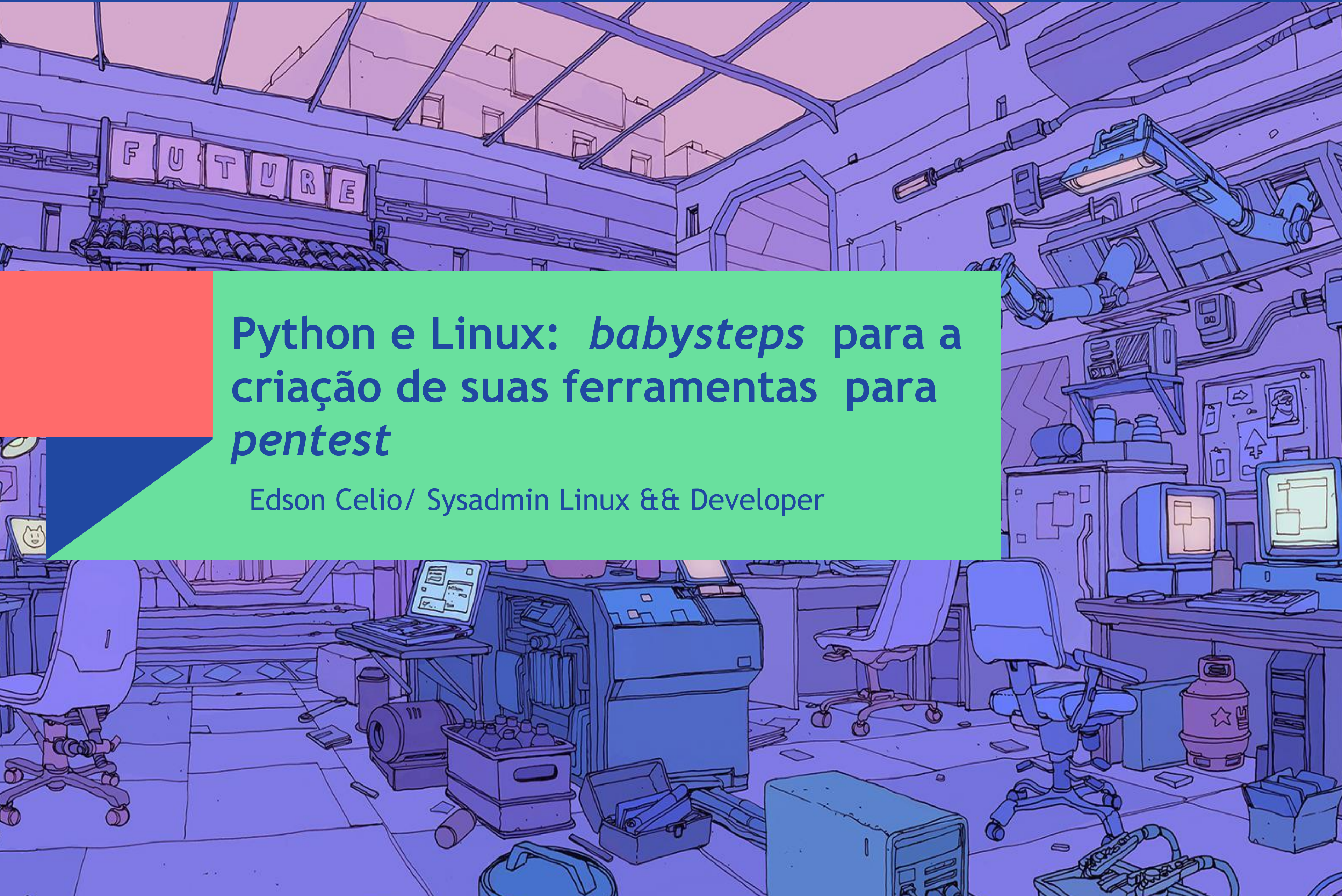
ROADSEC MAIS UM EVENTO: Flipside SECURITY BEYOND TECHNOLOGY REALIZAÇÃO: Green Helmet



# Python e Linux: *babysteps* para a criação de suas ferramentas para *pentest*

Edson Celio/ Sysadmin Linux && Developer

## EdsonCelio --help

- Graduando em Engenharia da Computação - UFC Campus Sobral;
- Responsável pelo GELEC - Grupo de Estudos em Linux da Engenharia da Computação (https://gelec.org);
- Administrador de sistemas Linux;
- Desenvolvedor Python, Shellscript e C#;

## Por que Linux?

- Variedade de distribuições;
- Controle maior de ferramentas e do sistema;
- Python é nativo de distribuições Linux;
- Praticidade na instalação de bibliotecas Python via PIP:

  (e.g) `sudo pip3 install requests`
- Fácil configuração do ambiente de desenvolvimento;

## Porque Python?

- Multiplataforma
- Muitas bibliotecas e ferramentas focadas em segurança
- Comunidade ativa
- Documentação

## Python2 ou Python3x?

- Apesar de ter sido o padrão por muito tempo, o Python2 está em processo de descontinuação, logo todas as novas funcionalidades estão sendo acrescentadas ao Python3;
- Python2 receberá atualização somente até 2020;
- Para mais detalhes sobre as mudanças no Python3, consultar: https://docs.python.org/3.0/whatsnew/3.0.html

## Modo Interativo vs Modo Script

- Modo Interativo ou Linha de Comando (shell mode):

```
user@programmer$ python3
Python 3.4.5 (default, Jan 24 2017, 17:55:08)
[GCC 4.9.3] on linux
Type "help", "copyright", "credits" or "license" for more
information.
>>>
```

## Modo Interativo vs Modo Script

- Modo Script (script mode):

```
user@programmer$ python3 my_pentest_tools.py
“Hello, it’s my first pentest tool!”
```

## Variável

- Um nome que aponta para algum dado armazenado na memória:

```
>> port   = 21
>> banner = “FTP Server”
```

## Estruturas de dados -> Strings

- Sequência de caracteres,onde cada caractere pode ser acessado por vez:

```
>> banner = "FTP Server"
>> banner[0]
>> len(banner)
>>banner.upper()
FTP SERVER
>>banner.replace('FTP','SSH')
SSH Server
```

## Estruturas de dados -> Lists

- Como uma string, uma lista é uma sequência de valores. Em uma string, os valores são caracteres; em uma lista, eles podem ser de qualquer tipo. Os valores em uma lista são chamados de elementos, ou, algumas vezes, de itens.

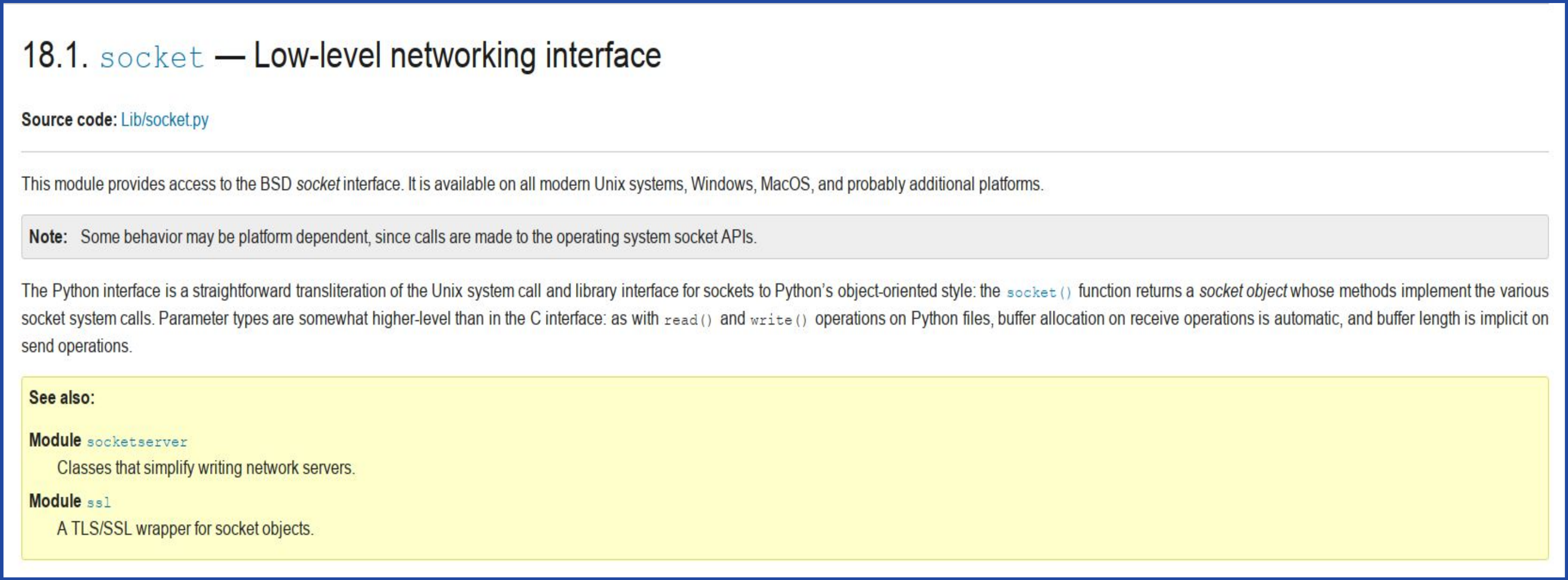
```
>> portList = []
>> portList = [21,80,443,22]
>> portList.append(5000)
>> print(portList)
[21,80,443,22,5000]
```

## Estruturas de dados -> Dicts

- Um dicionário contém uma coleção de índices, que se chamam chaves e uma coleção de valores. Cada chave é associada com um único valor. A associação de uma chave e um valor chama-se par chave-valor ou item.

```
>> services = {‘ftp’:21,’ssh’:22, ‘smtp’:25}
>> services.keys()
[‘ftp’,’ssh’,’smtp’]
>>services.items()
[(‘ftp’,21), (‘smtp’,25), (‘ssh’,22)]
```

## Bibliotecas -> Socket



18.1. socket — Low-level networking interface

**Source code:** Lib/socket.py

This module provides access to the BSD *socket* interface. It is available on all modern Unix systems, Windows, MacOS, and probably additional platforms.

**Note:** Some behavior may be platform dependent, since calls are made to the operating system socket APIs.

The Python interface is a straightforward transliteration of the Unix system call and library interface for sockets to Python's object-oriented style: the `socket()` function returns a *socket object* whose methods implement the various socket system calls. Parameter types are somewhat higher-level than in the C interface: as with `read()` and `write()` operations on Python files, buffer allocation on receive operations is automatic, and buffer length is implicit on send operations.

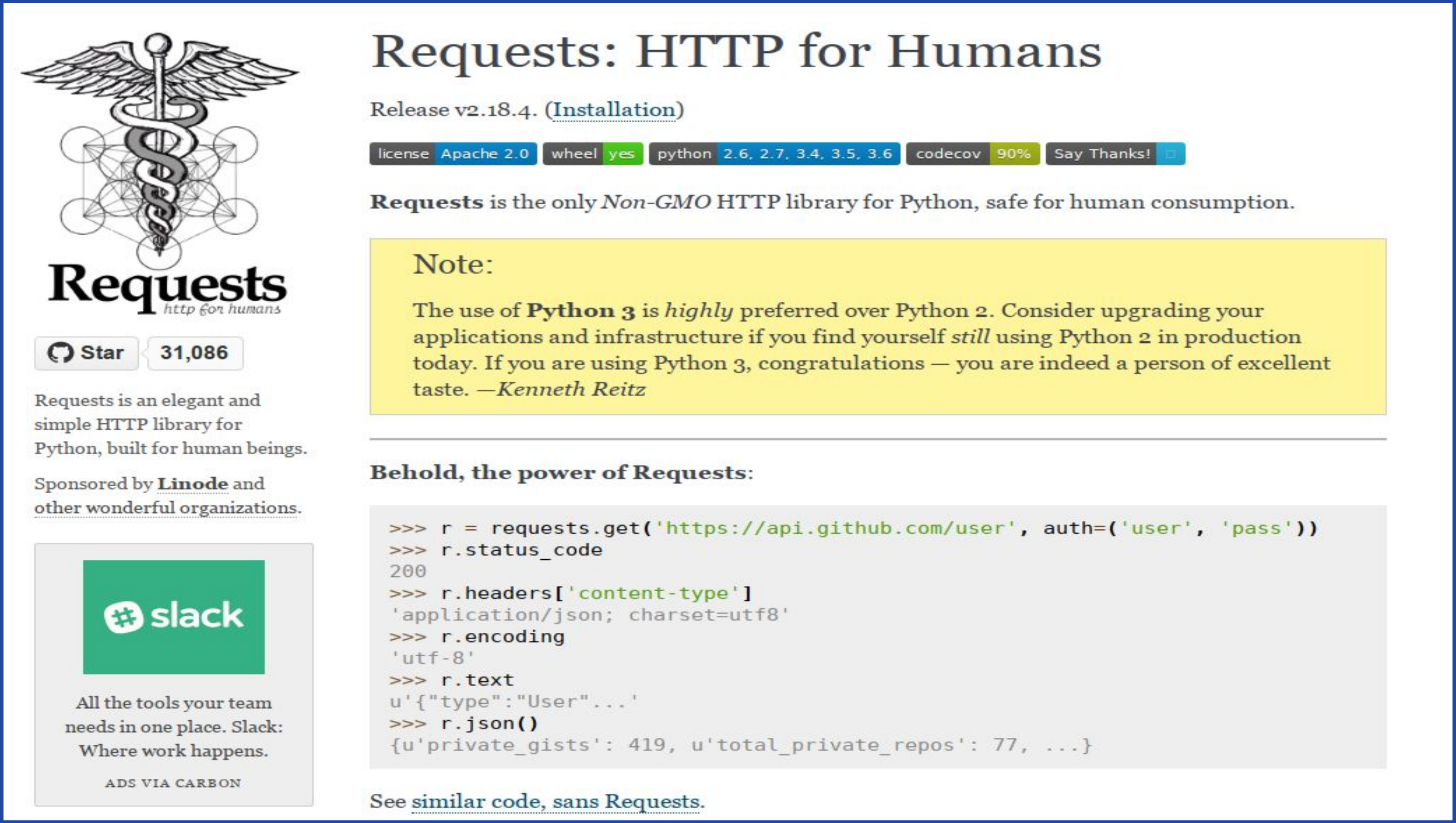
**See also:**

**Module** `socketserver`
Classes that simplify writing network servers.

**Module** `ssl`
A TLS/SSL wrapper for socket objects.

Instalação: nativa

## Bibliotecas -> Requests



# Requests: HTTP for Humans

Release v2.18.4. (Installation)

license Apache 2.0 | wheel yes | python 2.6, 2.7, 3.4, 3.5, 3.6 | codecov 90% | Say Thanks!

**Requests** is the only *Non-GMO* HTTP library for Python, safe for human consumption.

> Note:
>
> The use of **Python 3** is *highly* preferred over Python 2. Consider upgrading your applications and infrastructure if you find yourself *still* using Python 2 in production today. If you are using Python 3, congratulations — you are indeed a person of excellent taste. —*Kenneth Reitz*

**Behold, the power of Requests:**

```
>>> r = requests.get('https://api.github.com/user', auth=('user', 'pass'))
>>> r.status_code
200
>>> r.headers['content-type']
'application/json; charset=utf8'
>>> r.encoding
'utf-8'
>>> r.text
u'{"type":"User"...'
>>> r.json()
{u'private_gists': 419, u'total_private_repos': 77, ...}
```

See similar code, sans Requests.

Requests
http for humans

Star 31,086

Requests is an elegant and simple HTTP library for Python, built for human beings.

Sponsored by **Linode** and other wonderful organizations.

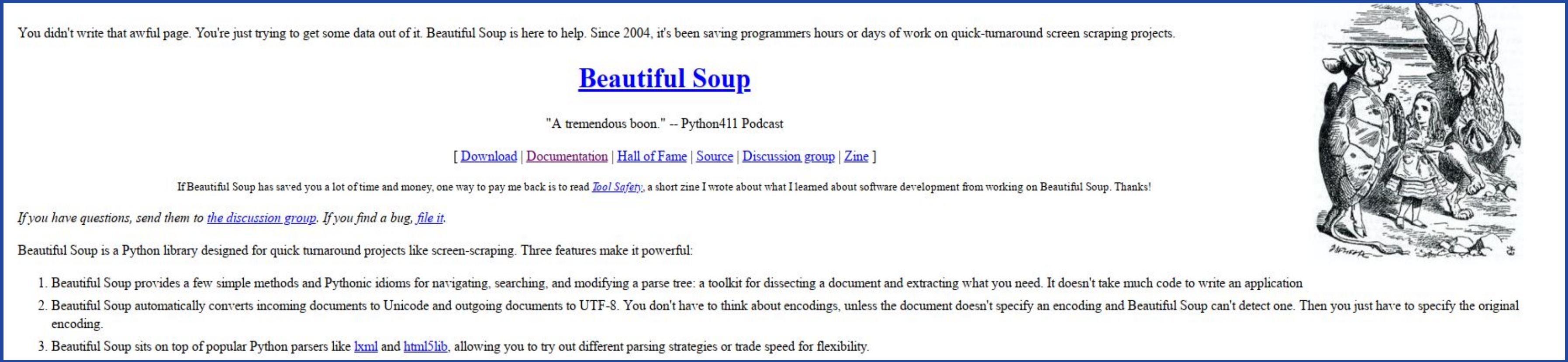
slack

All the tools your team needs in one place. Slack: Where work happens.

ADS VIA CARBON

Instalação: pip install requests

## Bibliotecas -> BeautifulSoup



You didn't write that awful page. You're just trying to get some data out of it. Beautiful Soup is here to help. Since 2004, it's been saving programmers hours or days of work on quick-turnaround screen scraping projects.

**Beautiful Soup**

"A tremendous boon." -- Python411 Podcast

[ Download | Documentation | Hall of Fame | Source | Discussion group | Zine ]

If Beautiful Soup has saved you a lot of time and money, one way to pay me back is to read *Tool Safety*, a short zine I wrote about what I learned about software development from working on Beautiful Soup. Thanks!

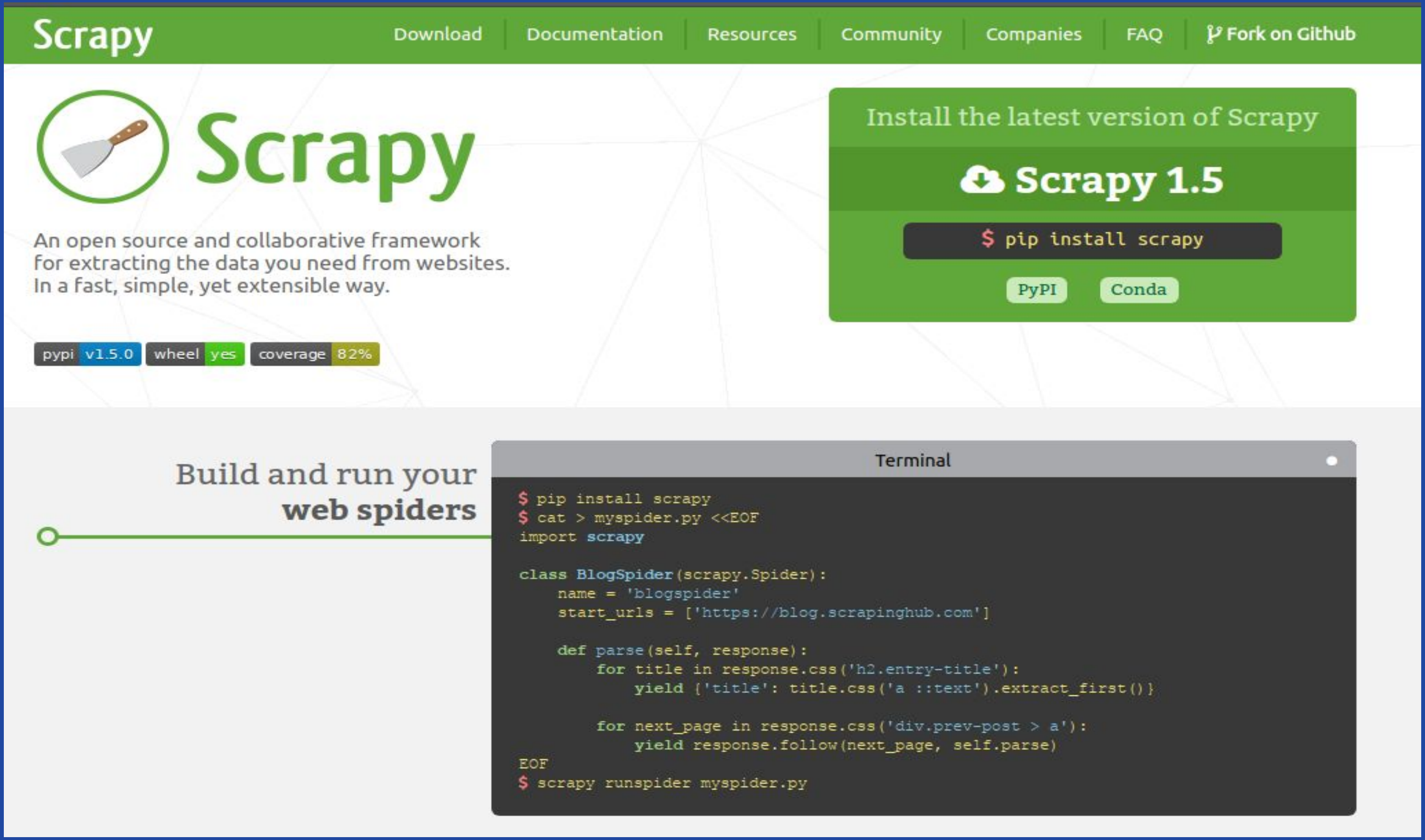
*If you have questions, send them to the discussion group. If you find a bug, file it.*

Beautiful Soup is a Python library designed for quick turnaround projects like screen-scraping. Three features make it powerful:

1. Beautiful Soup provides a few simple methods and Pythonic idioms for navigating, searching, and modifying a parse tree: a toolkit for dissecting a document and extracting what you need. It doesn't take much code to write an application
2. Beautiful Soup automatically converts incoming documents to Unicode and outgoing documents to UTF-8. You don't have to think about encodings, unless the document doesn't specify an encoding and Beautiful Soup can't detect one. Then you just have to specify the original encoding.
3. Beautiful Soup sits on top of popular Python parsers like lxml and html5lib, allowing you to try out different parsing strategies or trade speed for flexibility.

Instalação: pip install beautifulsoup4

## Bibliotecas -> Scrapy



An open source and collaborative framework for extracting the data you need from websites. In a fast, simple, yet extensible way.

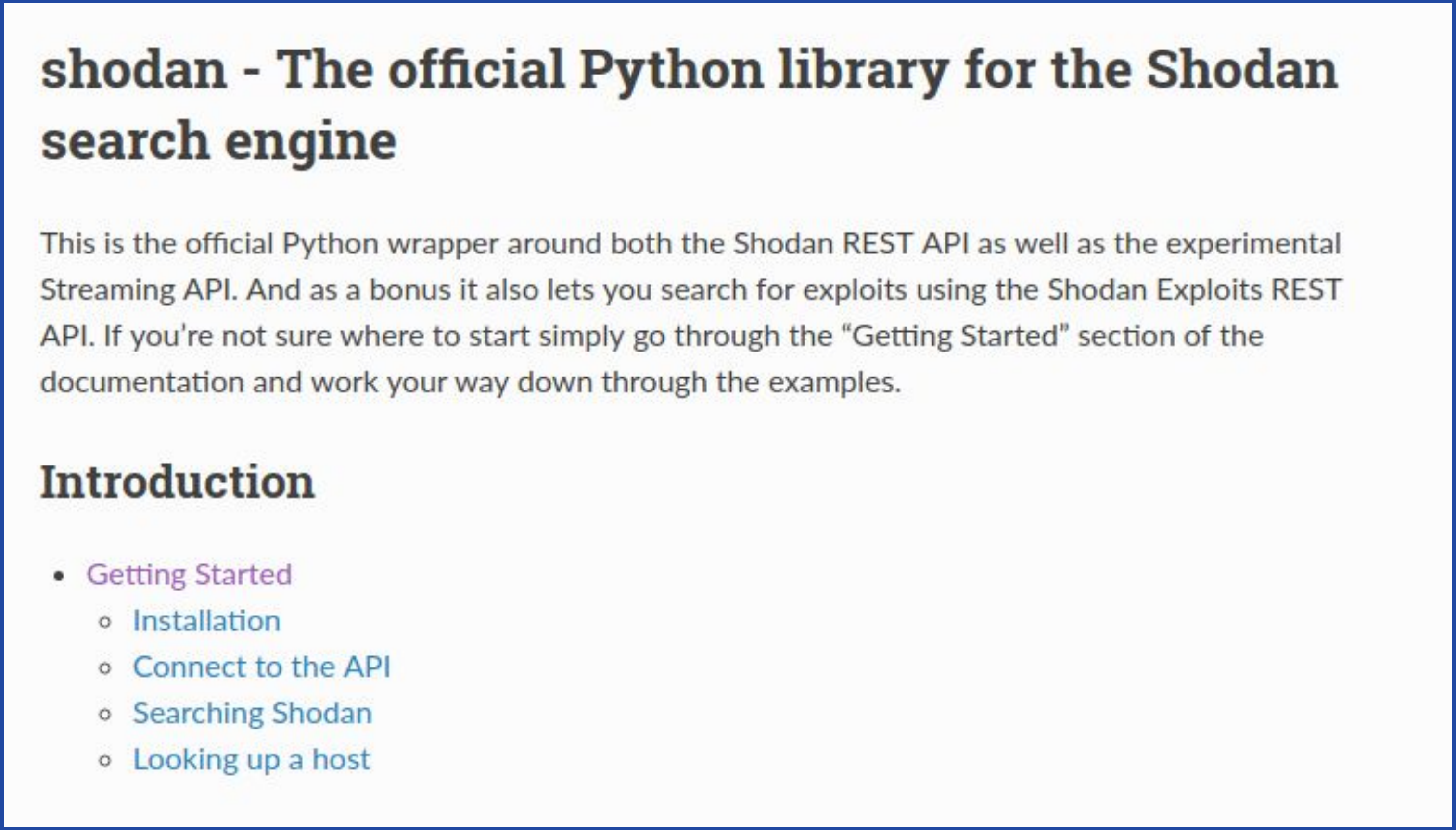
```
$ pip install scrapy
```

Build and run your **web spiders**

```
$ pip install scrapy
$ cat > myspider.py <<EOF
import scrapy

class BlogSpider(scrapy.Spider):
    name = 'blogspider'
    start_urls = ['https://blog.scrapinghub.com']

    def parse(self, response):
        for title in response.css('h2.entry-title'):
            yield {'title': title.css('a ::text').extract_first()}

        for next_page in response.css('div.prev-post > a'):
            yield response.follow(next_page, self.parse)
EOF
$ scrapy runspider myspider.py
```

Instalação: pip install scrapy

## Bibliotecas -> Shodan



### shodan - The official Python library for the Shodan search engine

This is the official Python wrapper around both the Shodan REST API as well as the experimental Streaming API. And as a bonus it also lets you search for exploits using the Shodan Exploits REST API. If you're not sure where to start simply go through the "Getting Started" section of the documentation and work your way down through the examples.

#### Introduction

- Getting Started
  - Installation
  - Connect to the API
  - Searching Shodan
  - Looking up a host

Instalação: `pip install shodan`

## Ambiente de Desenvolvimento

- Uma distribuição Linux de sua preferência (e.g Ubuntu, Linux Mint, Arch Linux, Fedora...) instalada ou em máquina virtual;
- Editor de texto gráfico ou terminal (e.g Nano, Vim ) ou ainda uma IDE (e.g Geany, Atom, PyCharm);
- Versão do Python instalada (de preferência python 3.x);
- Para evitar quebra de dependência ou bibliotecas desnecessárias instaladas, recomenda-se usar virtualenv: (https://virtualenv.pypa.io/en/stable/);

ROADSEC

## Exemplos Práticos

## Crack zipfile

```
import zipfile

filename = "test.zip"
dictionary = "passwordlist.txt"

password = None
file_to_open = zipfile.ZipFile(filename)
with open(dictionary, "r") as f:
    for line in f.readlines():
        password = line.strip("\n")

        try:
            file_to_open.extractall(pwd=password)
            password = ("Password found:{0}".format(password))
            print (password)

        #missing implementation
            except:
                pass
```

## NMAP Scan

```
import nmap
import sys

host = ‘127.0.0.1’
nmap = nmap.PortScanner()
nmap.scan(host, '1-1024')
print nmap.command_line()
print nmap.scaninfo()

for host in nmap.all_hosts():
    print("Host : {0} {1}".format(host, nmap[host].hostname()))
    print("State : {0}".format(nmap[host].state()))
for proto in nmap[host].all_protocols():
    print("Protocol : {0}".format(proto))

listport = nmap[host]['tcp'].keys()
listport.sort()

for port in listport:
    print("port : {0}\tstate : {1}".format(port,
nmap[host][proto][port]['state']))
```

## Crypt and Decrypt AES File

```
from Crypto.Cipher import AES

encrypt_AES = AES.new('key-12345', AES.MODE_CBC, 'This is an IV456')
message = "This is message "
ciphertext = encrypt_AES.encrypt(message)

print (ciphertext)

decrypt_AES = AES.new('key-12345', AES.MODE_CBC, 'This is an IV456')
message_decrypted =  decrypt_AES.decrypt(ciphertext)

print (message_decrypted)
```

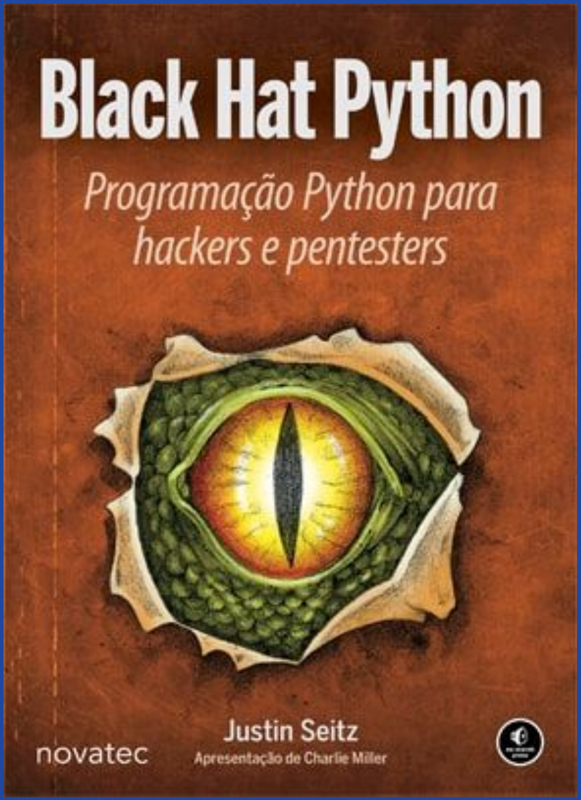
## Referências & Bibliotecas

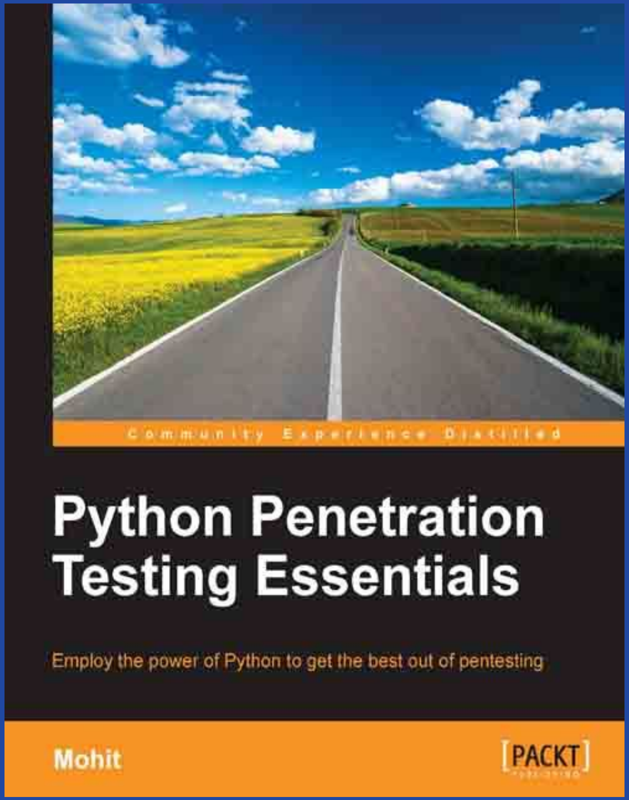
- Rede: socket [nativa], requests [externa]
- OS: os [nativa]
- Criptografia: crypt [nativa]
- Scanner de Portas: nmap [externa]
- FTP: ftplib [nativa]
- Geolocalização de IPs: pygeoip [externa da google]
- Scapy: scapy [externa]
- Expressões regulares: re [nativa]
- Mechanize: mechanize [externa da sourceforge]
- BeautifulSoup: beautifulsoup [externa]
- Arquivos Json: simplejson [externa]
- Conexão SSH: paramiko [externa]

Ajuda:

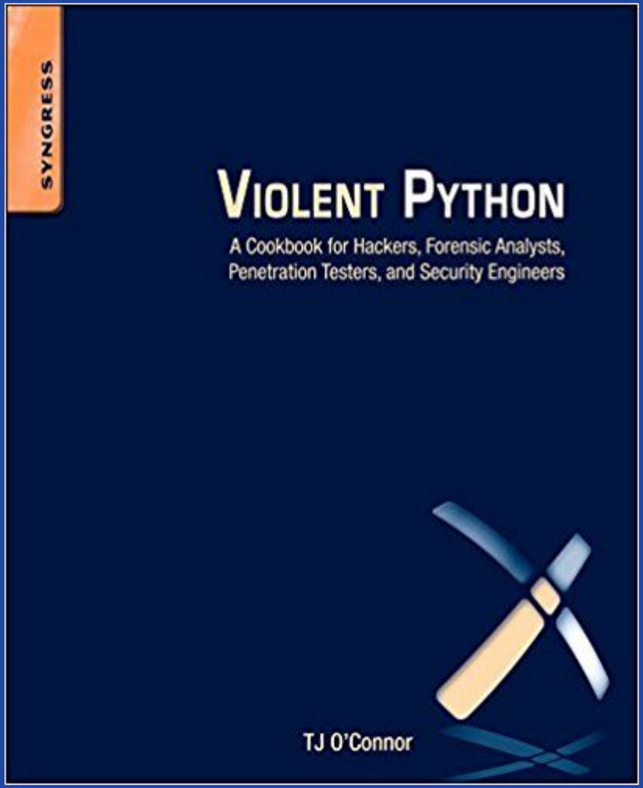
```
<nomedabiblioteca>.__doc__

>> (e.g) socket.__doc__
```

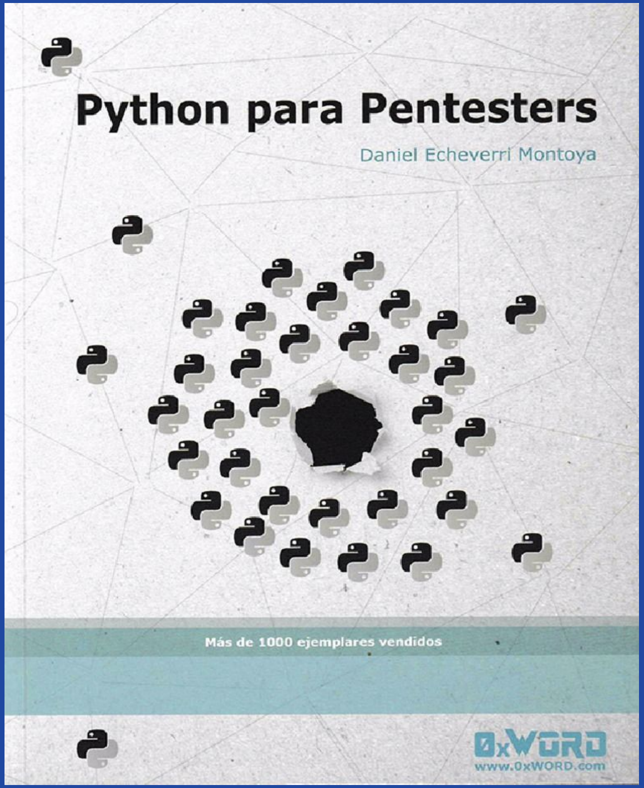
## Livros


Black Hat Python
Programação Python para hackers e pentesters
Justin Seitz
Apresentação de Charlie Miller
novatec


Community Experience Distilled
Python Penetration Testing Essentials
Employ the power of Python to get the best out of pentesting
Mohit
[PACKT]

## Livros


SYNGRESS
VIOLENT PYTHON
A Cookbook for Hackers, Forensic Analysts, Penetration Testers, and Security Engineers
TJ O'Connor


Python para Pentesters
Daniel Echeverri Montoya
Más de 1000 ejemplares vendidos
0xWORD
www.0xWORD.com

[Warning]: TODOS os códigos e referências aqui citados, são apenas para fins de ESTUDO!

link para acesso dos códigos e slides:
https://github.com/edsoncelio/palestra-roadsec2018

ROADSEC

## Obrigado!

edsoncelio@protonmail.com
@tuxpilgrim