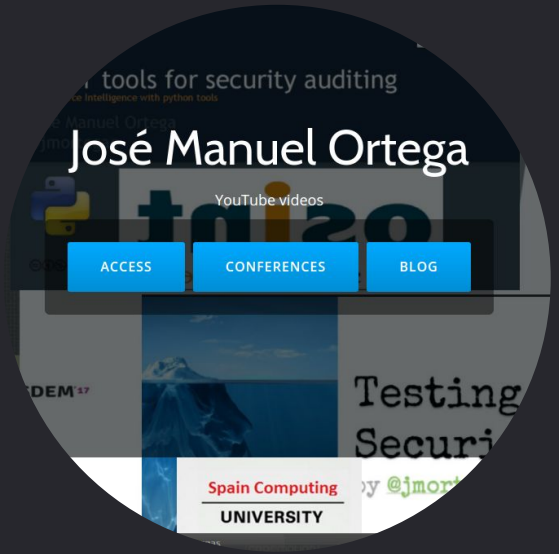
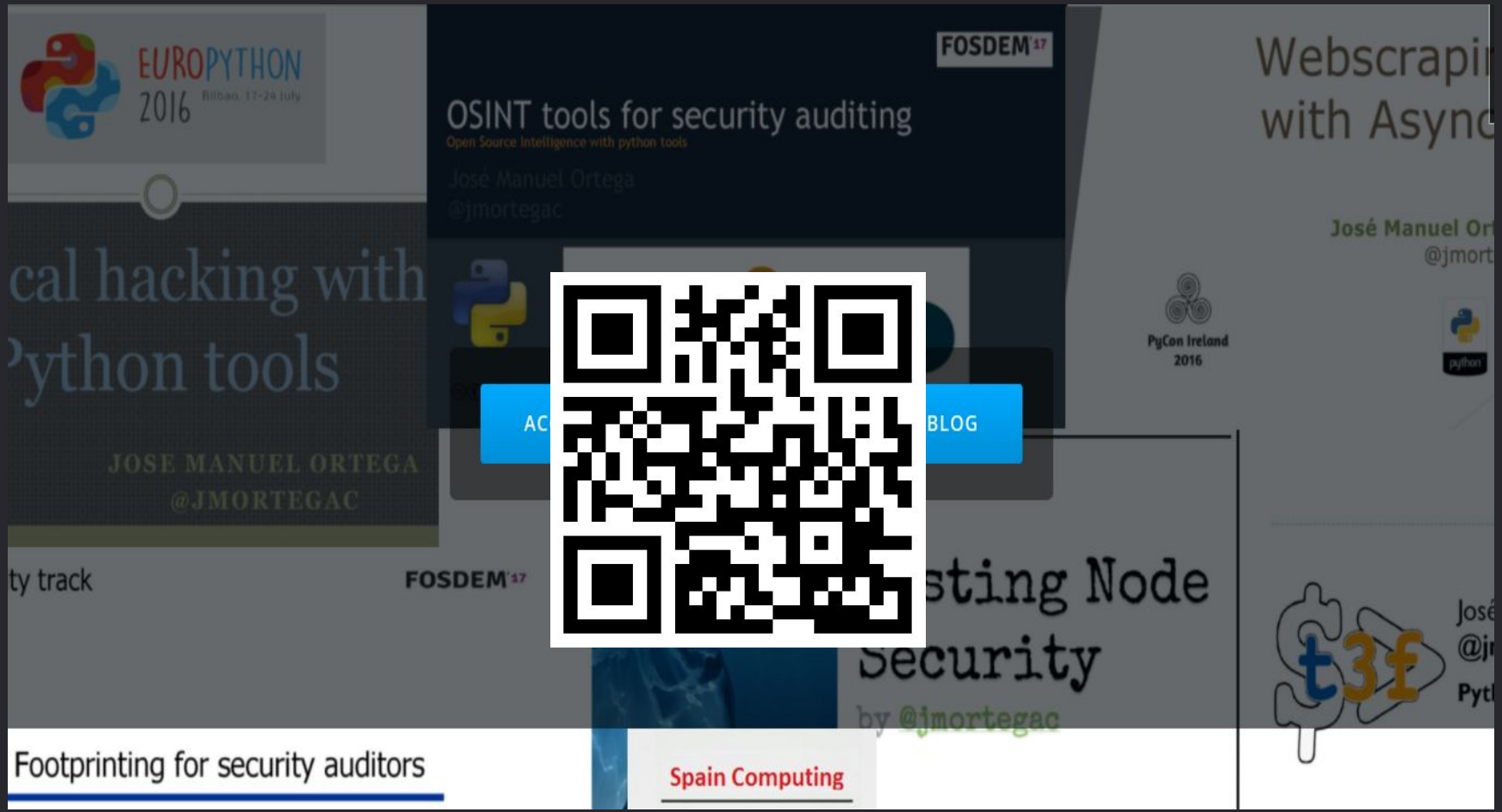
Introducción a python

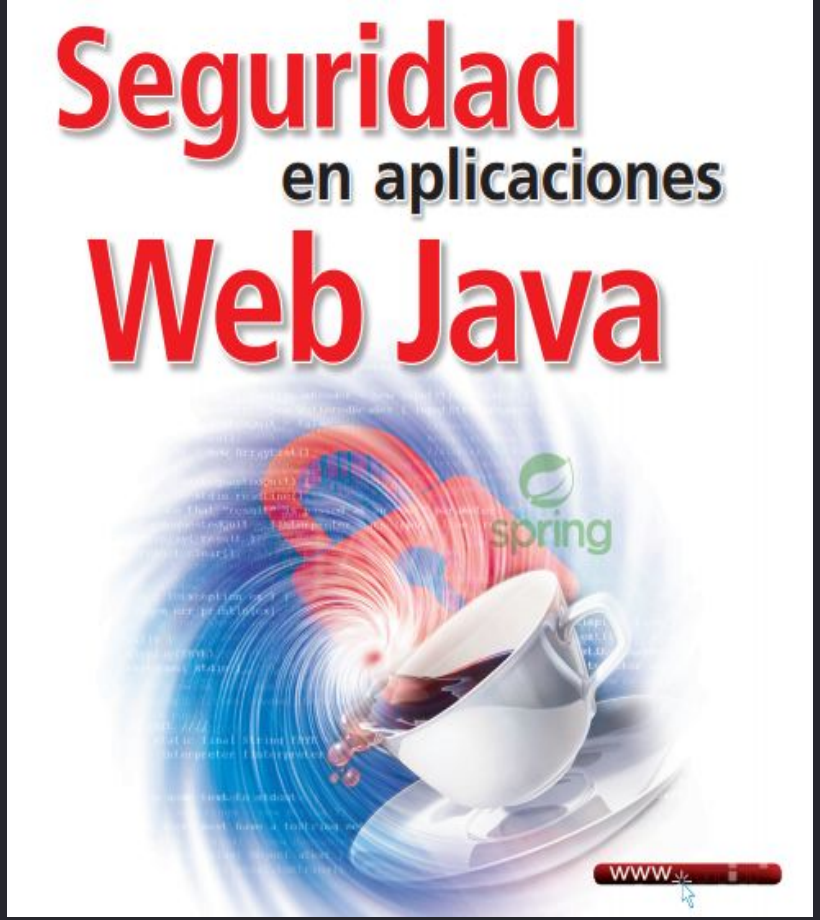


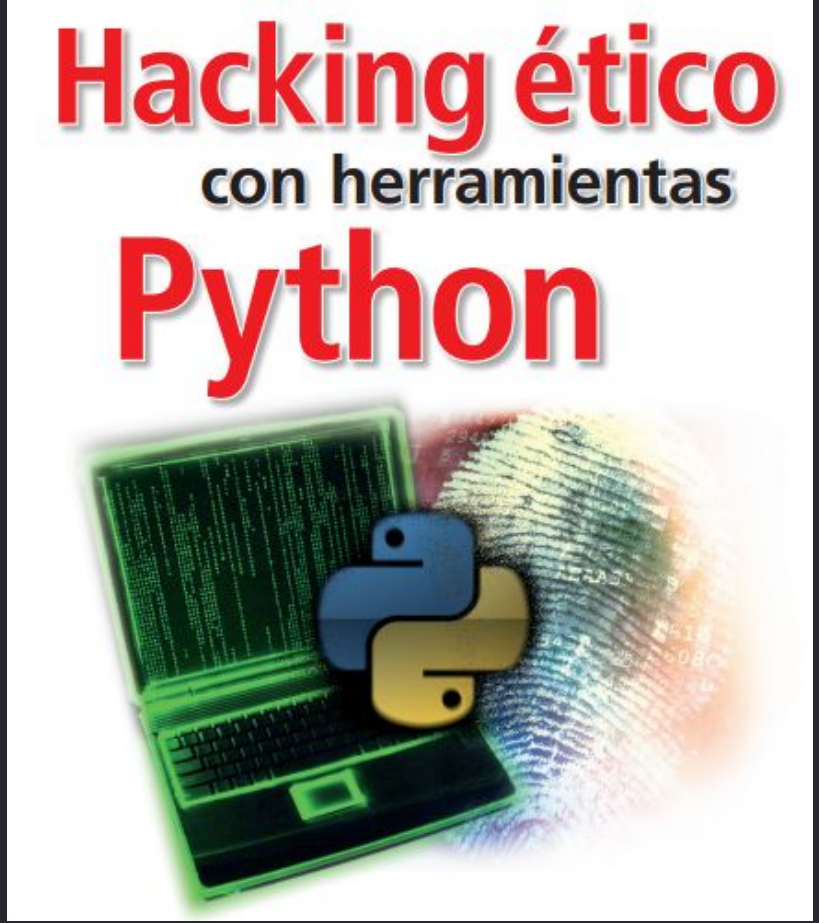
jmortega.github.io

about.me/jmortegac

- [EUROPYTHON 2016] Ethical hacking with Python tools
- [EUROPYTHON 2016] Hacking ético con herramientas Python
- [PYDATA 2016] Python tools for webscraping
- [TECHFEST 2016] Python para desarrolladores web - T3chFest2016
- [PYCONES 2015] Comparing Python ORM - Track Avanzado
- [PYCONES 2015] Seguridad y criptografía en Python - Track Científico


EUROPYTHON
2016
Bilbao, 17-24 July
FOSDEM'17
OSINT tools for security auditing
Open Source Intelligence with python tools
José Manuel Ortega
@jmortegac
Webscrapi
with Async
José Manuel Ort
@jmort
cal hacking with
ython tools
JOSE MANUEL ORTEGA
@JMORTEGAC
PyCon Ireland
2016
AC
BLOG
ty track
FOSDEM'17
sting Node
Security
by @jmortegac
t3f
Footprinting for security auditors
Spain Computing

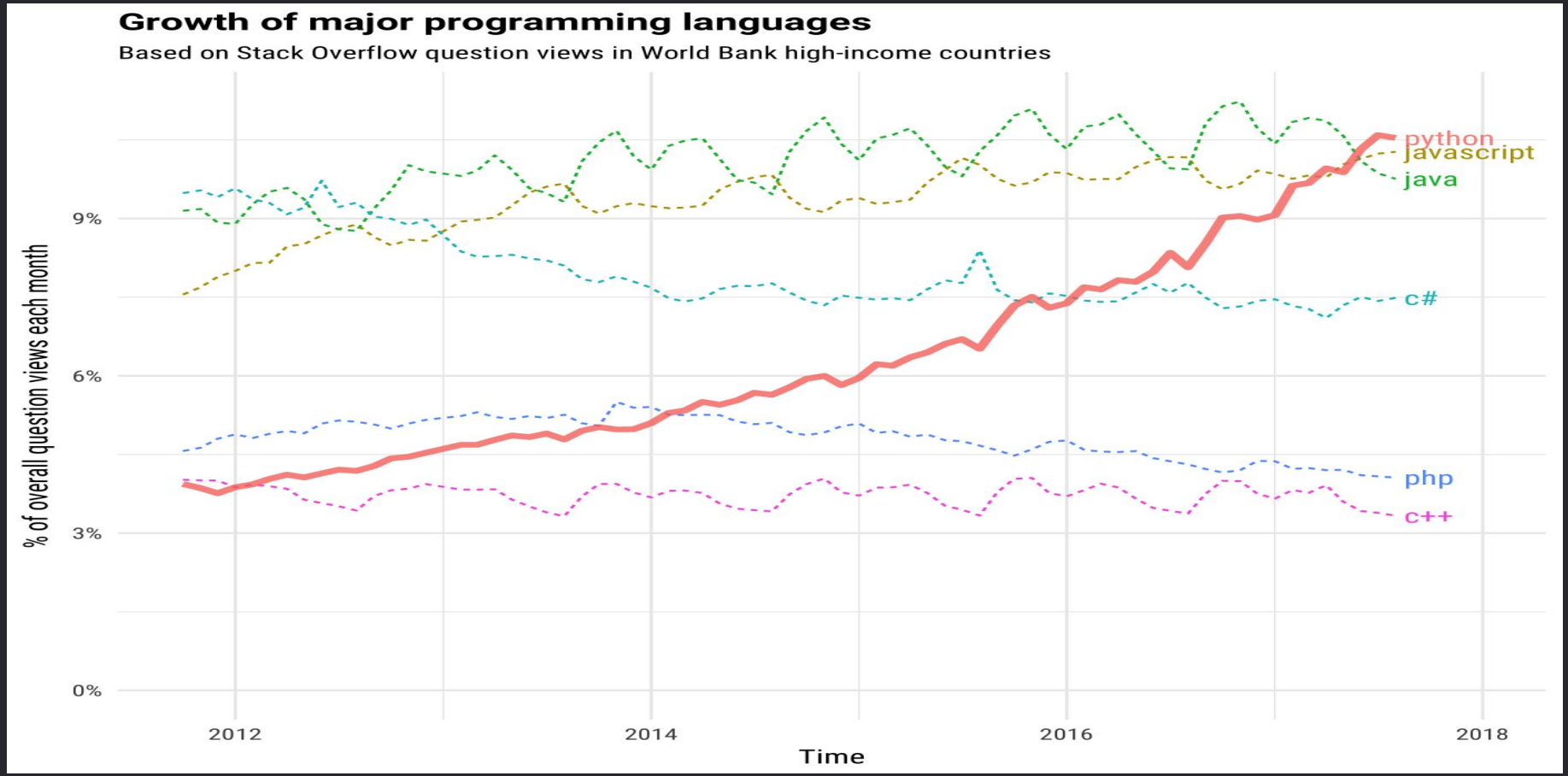

Seguridad
en aplicaciones
Web Java
spring
www


Hacking ético
con herramientas
Python

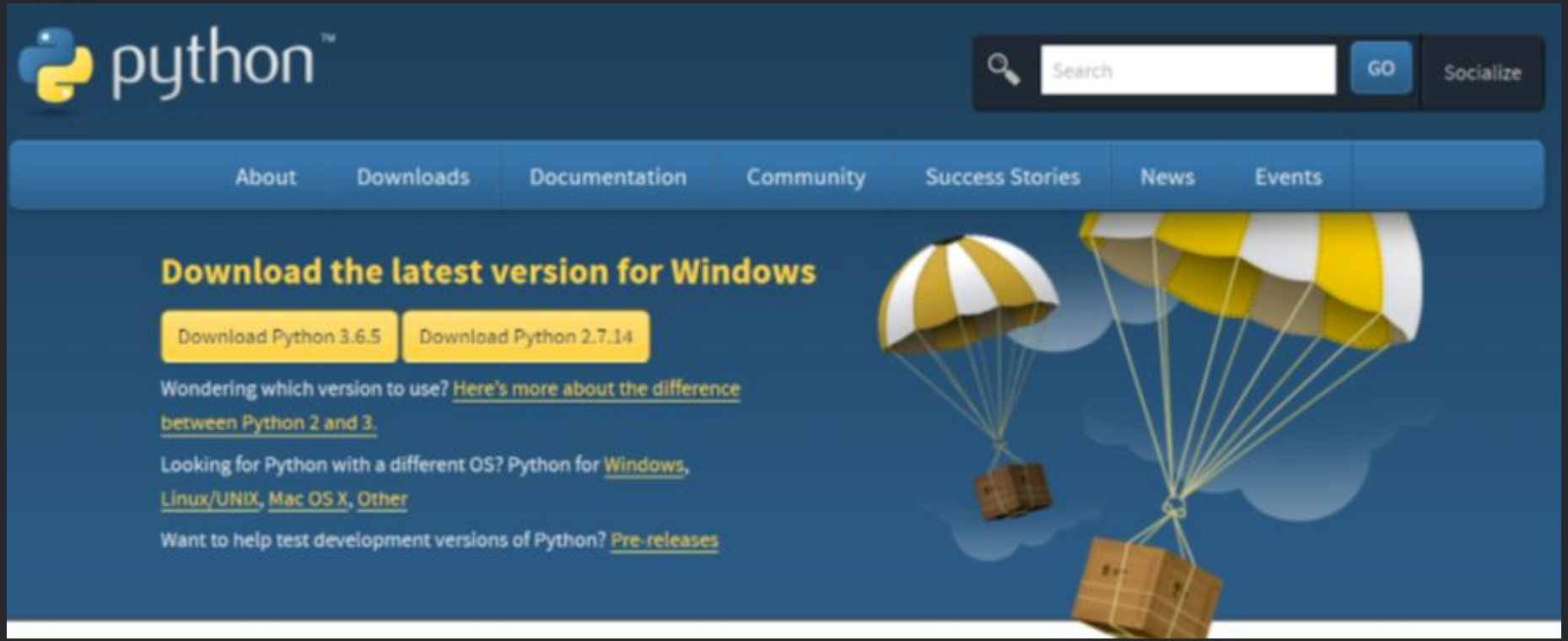
INTRODUCCIÓN A LA PROGRAMACIÓN CON PYTHON
METODOLOGÍA Y HERRAMIENTAS DE DESARROLLO
LIBRERÍAS Y MÓDULOS PARA REALIZAR PETICIONES
RECOLECCIÓN DE INFORMACIÓN CON PYTHON
EXTRACCIÓN DE INFORMACIÓN CON PYTHON
WEBSCRAPING CON PYTHON
ESCANEO DE PUERTOS Y REDES CON PYTHON
HERRAMIENTAS AVANZADAS

Agenda

- Python como lenguaje multiplataforma
- Principales estructuras de datos y colecciones
- Orientación a Objetos en python
- Gestionar paquetes y trabajar con virtualenv
- Entornos de desarrollo
- Módulo STB (Security Tools Builder)


Growth of major programming languages
Based on Stack Overflow question views in World Bank high-income countries
% of overall question views each month
9%
6%
3%
0%
2012
2014
2016
2018
Time
python
javascript
java
c#
php
c++

```
>>> 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!
```


python
Search
GO
Socialize
About
Downloads
Documentation
Community
Success Stories
News
Events
Download the latest version for Windows
Download Python 3.6.5
Download Python 2.7.14
Wondering which version to use? Here's more about the difference between Python 2 and 3.
Looking for Python with a different OS? Python for Windows, Linux/UNIX, Mac OS X, Other
Want to help test development versions of Python? Pre-releases

Python 2.7 will retire in...

1	6	16	3	25	17
Year	Months	Days	Hours	Minutes	Seconds

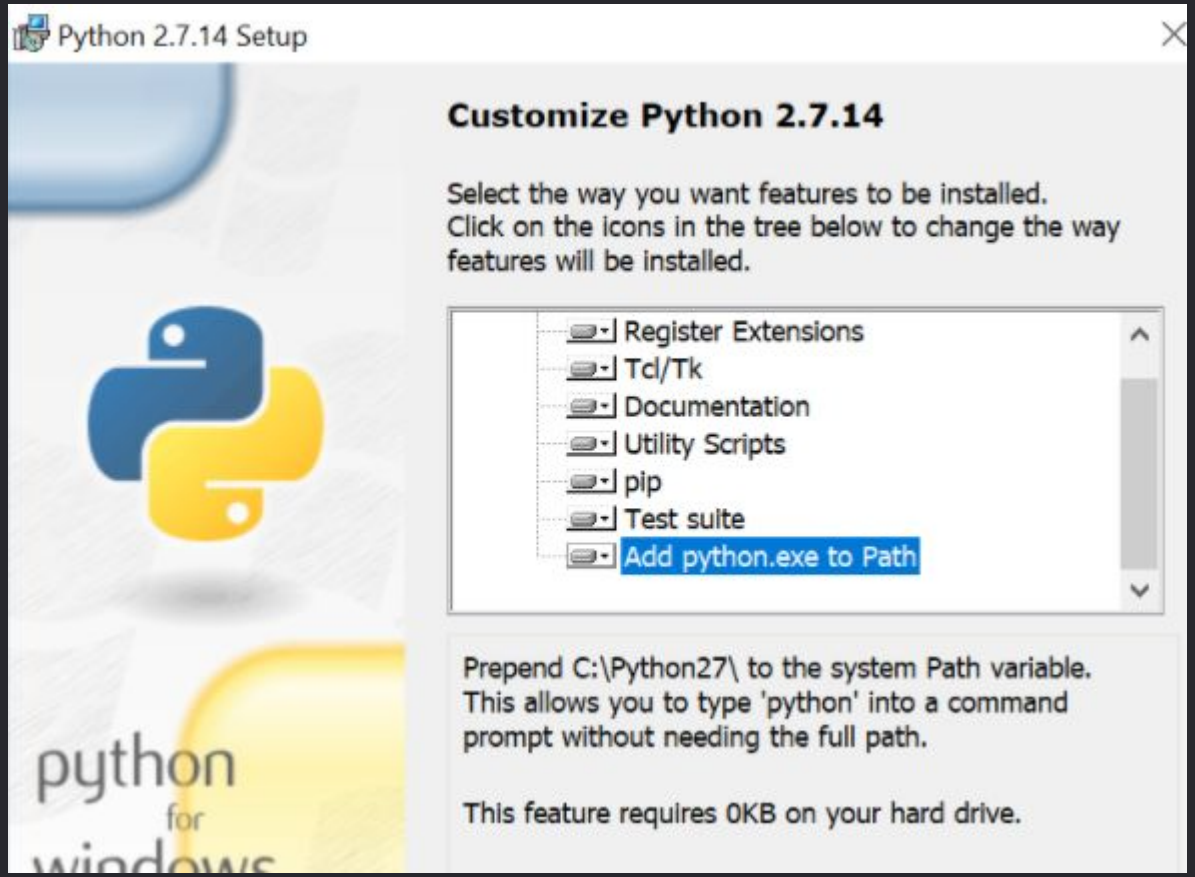
Enable Guido Mode Huh?

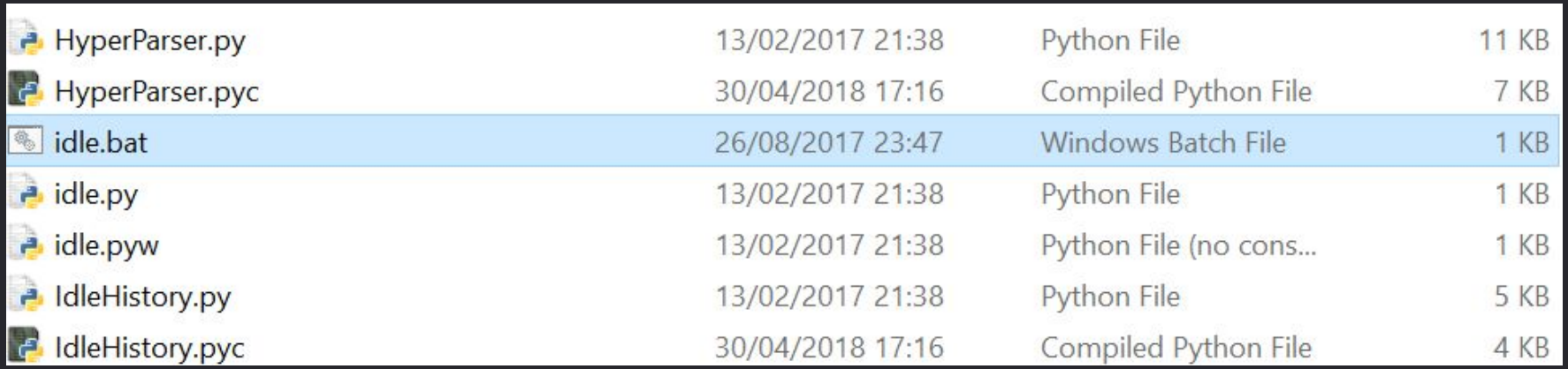
What's all this, then?

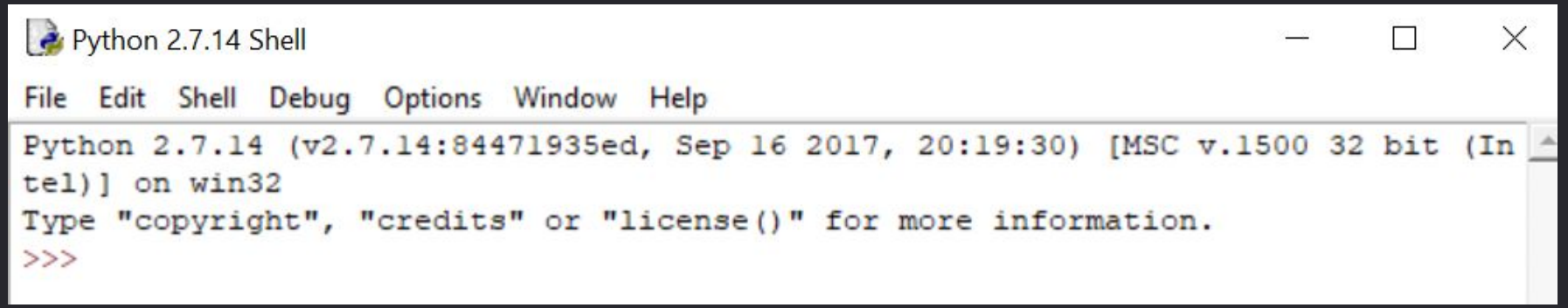
Python 2.7 will not be maintained past 2020. Originally, there was no official date. Recently, that date has been updated to January 1, 2020. This clock has been updated accordingly. My original idea was to throw a Python 2 Celebration of Life party at PyCon 2020, to celebrate everything Python 2 did for us. That idea still stands. (If this sounds interesting to you, email pythonclockorg@gmail.com).

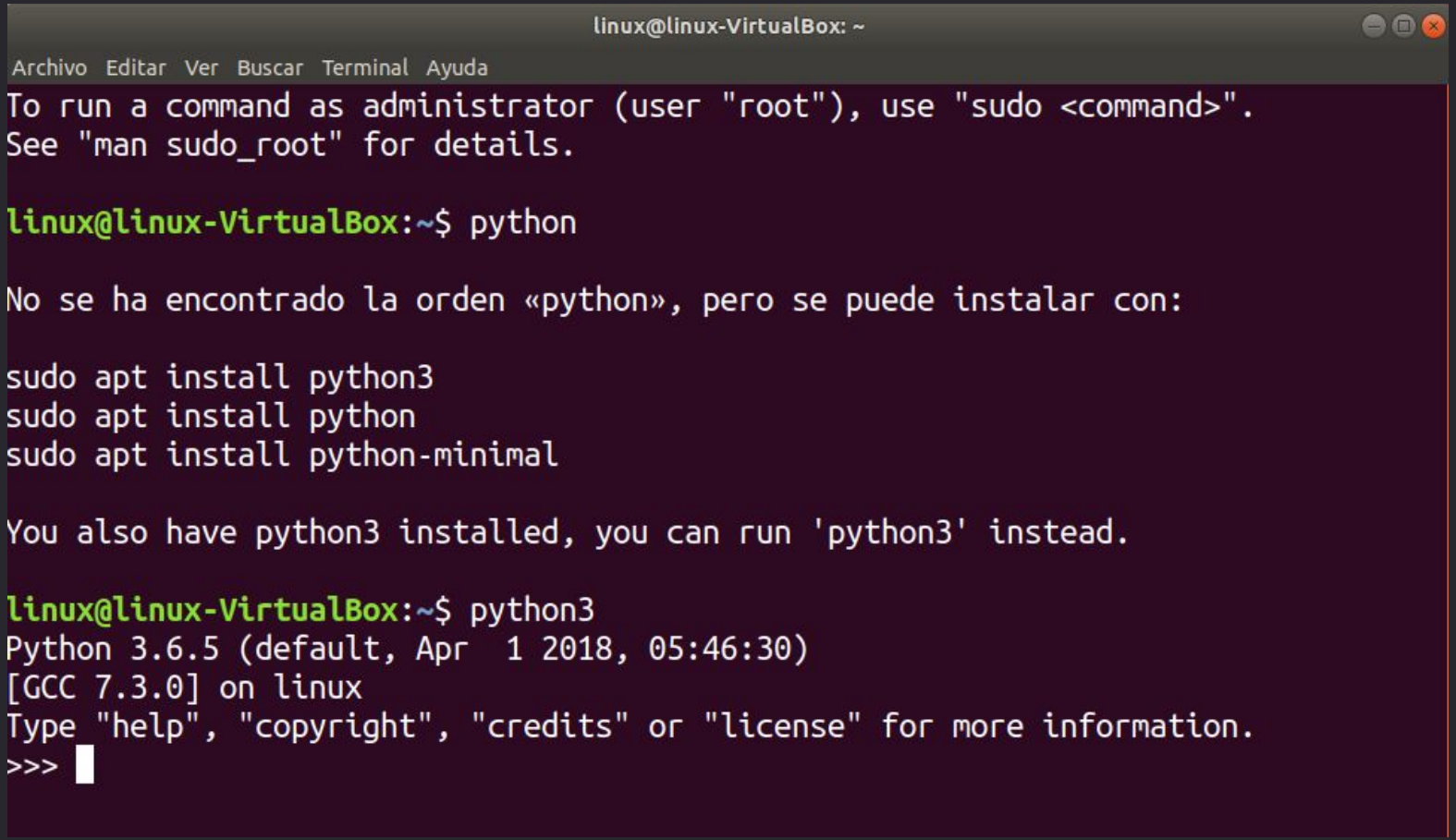
Python 2, thank you for your years of faithful service.

Python 3, your time is now.


Python 2.7.14 Setup
Customize Python 2.7.14
Select the way you want features to be installed.
Click on the icons in the tree below to change the way features will be installed.
Register Extensions
Tcl/Tk
Documentation
Utility Scripts
pip
Test suite
Add python.exe to Path
Prepend C:\Python27\ to the system Path variable. This allows you to type 'python' into a command prompt without needing the full path.
This feature requires 0KB on your hard drive.
python
for


HyperParser.py 13/02/2017 21:38 Python File 11 KB
HyperParser.pyc 30/04/2018 17:16 Compiled Python File 7 KB
idle.bat 26/08/2017 23:47 Windows Batch File 1 KB
idle.py 13/02/2017 21:38 Python File 1 KB
idle.pyw 13/02/2017 21:38 Python File (no cons... 1 KB
IdleHistory.py 13/02/2017 21:38 Python File 5 KB
IdleHistory.pyc 30/04/2018 17:16 Compiled Python File 4 KB


Python 2.7.14 Shell
File Edit Shell Debug Options Window Help
Python 2.7.14 (v2.7.14:84471935ed, Sep 16 2017, 20:19:30) [MSC v.1500 32 bit (In
tel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>


linux@linux-VirtualBox: ~
Archivo Editar Ver Buscar Terminal Ayuda
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.
linux@linux-VirtualBox:~$ python
No se ha encontrado la orden «python», pero se puede instalar con:
sudo apt install python3
sudo apt install python
sudo apt install python-minimal
You also have python3 installed, you can run 'python3' instead.
linux@linux-VirtualBox:~$ python3
Python 3.6.5 (default, Apr 1 2018, 05:46:30)
[GCC 7.3.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>>

Python idle

You can install IDLE for Python 3 with

```
sudo apt install idle
```

... or with ...

```
sudo apt install idle3
```

ubuntu packages

package names ▾ Search all options

» Ubuntu » Packages » bionic (18.04LTS) » python » idle3

[Source: python3-defaults]

[trusty] [xenial] [artful] [**bionic**] [cosmic]

Package: idle3 (3.6.5-3) [**universe**]

IDE for Python using Tkinter (transitional package)

Other Packages Related to idle3

depends | recommends | suggests | enhances

- idle (>= 3.6.5-3)
  IDE for Python using Tkinter (default version)
- python3 (>= 3.6.5-3)
  interactive high-level object-oriented language (default python3 version)

Download idle3

Architecture	Package Size	Installed Size	Files

Links for idle3

Ubuntu Resources:

- Bug Reports
- Ubuntu Changelog

Python idle

Python3-tools Download for Linux (rpm, i386, i586, i686, x86_64)

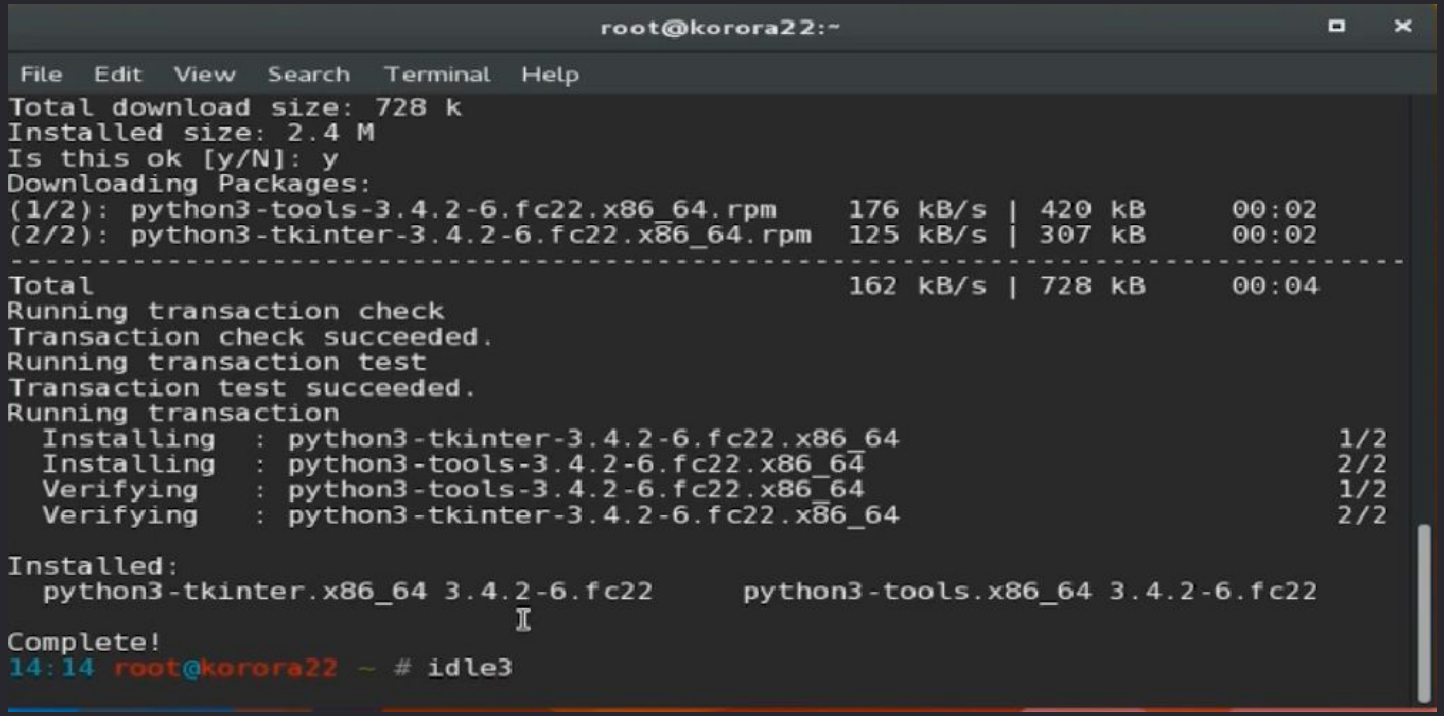
Download python3-tools linux packages for ALTLinux, CentOS, Fedora, openSUSE.

ALT Linux Sisyphus	
CentOS 7	
CentOS 6	
CentOS 5	
Fedora 28	
Fedora i386	
python3-idle-3.6.5-1.fc28.i686.rpm	A basic graphical development environment for Python
Fedora x86_64	
python3-idle-3.6.5-1.fc28.i686.rpm	A basic graphical development environment for Python
python3-idle-3.6.5-1.fc28.x86_64.rpm	A basic graphical development environment for Python

Python idle

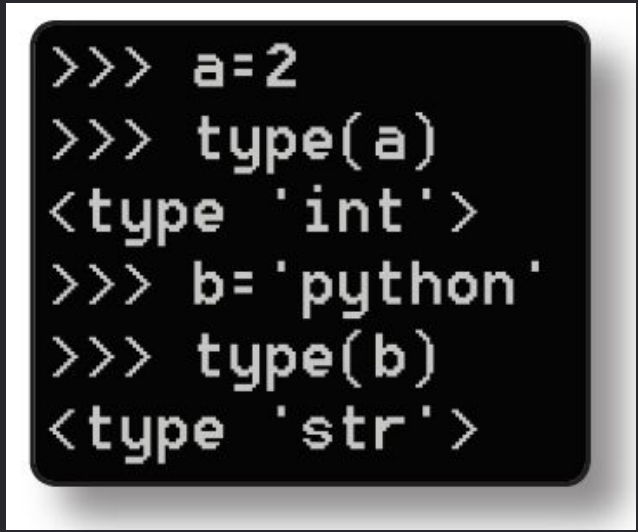

root@korora22:~
File Edit View Search Terminal Help
14:12 root@korora22 ~ # yum install python3-tools
Yum command has been deprecated, redirecting to '/usr/bin/dnf install python3-tools'.
See 'man dnf' and 'man yum2dnf' for more information.
To transfer transaction metadata from yum to DNF, run:
'dnf install python-dnf-plugins-extras-migrate && dnf-2 migrate'
Last metadata expiration check performed 0:52:57 ago on Mon Jan 18 13:20:01 2016.
Dependencies resolved.
Package Arch Version Repository Size
Installing:
python3-tkinter x86_64 3.4.2-6.fc22 updates 307 k
python3-tools x86_64 3.4.2-6.fc22 updates 420 k
Transaction Summary
Install 2 Packages
Total download size: 728 k
Installed size: 2.4 M
Is this ok [y/N]:

Python idle


root@korora22:~
File Edit View Search Terminal Help
Total download size: 728 k
Installed size: 2.4 M
Is this ok [y/N]: y
Downloading Packages:
(1/2): python3-tools-3.4.2-6.fc22.x86_64.rpm 176 kB/s | 420 kB 00:02
(2/2): python3-tkinter-3.4.2-6.fc22.x86_64.rpm 125 kB/s | 307 kB 00:02
Total 162 kB/s | 728 kB 00:04
Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
Installing : python3-tkinter-3.4.2-6.fc22.x86_64 1/2
Installing : python3-tools-3.4.2-6.fc22.x86_64 2/2
Verifying : python3-tools-3.4.2-6.fc22.x86_64 1/2
Verifying : python3-tkinter-3.4.2-6.fc22.x86_64 2/2
Installed:
python3-tkinter.x86_64 3.4.2-6.fc22 python3-tools.x86_64 3.4.2-6.fc22
Complete!
14:14 root@korora22 ~ # idle3

Tipos dinámicos

- Las variables no tienen tipo
- Tipos en t.ejecución



Estructuras de datos

- Strings
- Listas
- Tuplas
- Diccionarios

Strings

```
>>> cadena_texto="ejemplo"
>>> dir(cadena_texto)
['__add__', '__class__', '__contains__', '__delattr__', '__doc__', '__eq__', '__f
__', '__le__', '__len__', '__lt__', '__mod__', '__mul__', '__ne__', '__new__', '__
__', '_formatter_field_name_split', '_formatter_parser', 'capitalize', 'center',
ower', 'isspace', 'istitle', 'isupper', 'join', 'ljust', 'lower', 'lstrip', 'part
strip', 'swapcase', 'title', 'translate', 'upper', 'zfill']
```

```
find(...)
    S.find(sub [,start [,end]]) -> int

    Return the lowest index in S where substring sub is found,
    such that sub is contained within S[start:end].  Optional
    arguments start and end are interpreted as in slice notation.

    Return -1 on failure.
```

Strings

```
>>> protocol= "Hypertext Transfer Protocol"
>>> print protocol.upper()
HYPERTEXT TRANSFER PROTOCOL
>>> print protocol.lower()
hypertext transfer protocol
>>> print protocol.replace("Protocol","Protocol HTTP")
Hypertext Transfer Protocol HTTP
>>> print protocol.find("Protocol")
19
```

Listas

- Se instancian vacías con: []
- Se pueden instanciar con datos: [3, 4, 5]
- Métodos
  - append(x), extend(list), insert(i, x), remove(x), pop([i]), clear(), index(x), count(x), sort(), reverse(), copy()
- Insert, remove y sort devuelven “None”
  - Principio de diseño para todas las estructuras mutables en Python

Listas

```
>>> protocolList = []
>>> protocolList.append("ftp")
>>> protocolList.append("ssh")
>>> protocolList.append("smtp")
>>> protocolList.append("http")
>>> print protocolList
['ftp', 'ssh', 'smtp', 'http']
>>> protocolList.sort()
>>> print protocolList
['ftp', 'http', 'smtp', 'ssh']
>>> type protocolList
SyntaxError: invalid syntax
>>> type(protocolList)
<type 'list'>
>>> len(protocolList)
4
```

```
>>> for protocol in protocolList:
        print (protocol)


ftp
http
smtp
```

Listas

```
>>> position = protocolList.index("ssh")
>>> print "ssh position "+str(position)
ssh position 3
>>> protocolList.remove("ssh")
>>> print protocolList
['ftp', 'http', 'smtp']
>>> count= len(protocolList)
>>> print "Protocol elements "+str(count)
Protocol elements 3
```

Listas reversas

```
>>> protocolList.reverse()
>>> print protocolList
['smtp', 'http', 'ftp']
```

```
>>> for protocol in protocolList[::-1]:
        print(protocol)


smtp
http
ftp
```

Listas por compresion

- Azúcar sintáctico para crear listas
- Crear listas donde cada miembro es el resultado de una condición

```
>>> protocolList = ["FTP", "HTTP", "SNMP", "SSH"]
>>> protocolList_lower= [protocol.lower() for protocol in protocolList]
>>> print(protocolList_lower)
['ftp', 'http', 'snmp', 'ssh']
```

Iteradores

```
>>> iterador = iter(['python','java','c++'])
>>> print iterador
<listiterator object at 0x022724D0>
>>> for element in iterador:
...     print element
...
python
java
c++
```

```
>>> iterador = iter(['python','java','c++'])
>>> print iterador.next()
python
>>> print iterador.next()
java
>>> print iterador.next()
c++
>>> print iterador.next()
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
StopIteration
```

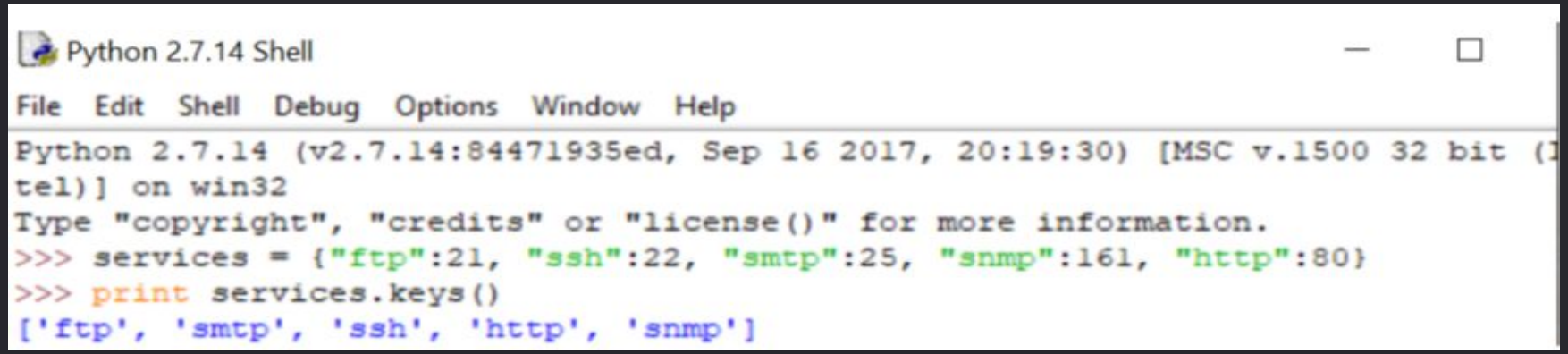
Tuplas

```
>>> tuple= ("ftp","ssh", "snmp","http")
>>> tuple[0]
'ftp'
>>> tuple[0]="FTP"

Traceback (most recent call last):
  File "<pyshell#91>", line 1, in <module>
    tuple[0]="FTP"
TypeError: 'tuple' object does not support item assignment
```

Diccionarios

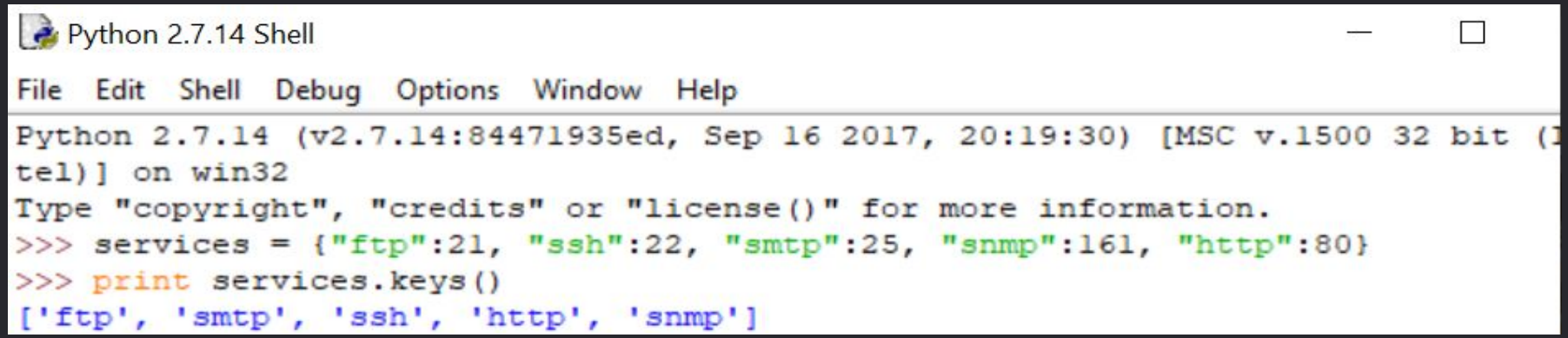
- .keys(): devuelve una tupla con las claves
- list(dicc.keys()): obtenemos una lista con las claves desordenadas
- sorted(list(dicc.keys())): obtenemos una lista ordenada de las claves
- x in dicc: comprueba que x existe en dicc

Diccionarios




```
>>> items = services.items()
>>> print items
[('ftp', 21), ('smtp', 25), ('ssh', 22), ('http', 80), ('snmp', 161)]
>>> items.sort()
>>> print items
[('ftp', 21), ('http', 80), ('smtp', 25), ('snmp', 161), ('ssh', 22)]
```

Diccionarios




```
>>> services = {"ftp":21, "ssh":22, "smtp":25, "http":80}
>>> services2 = {"ftp":21, "ssh":22, "snmp":161, "ldap":389}
>>> services.update(services2)
>>> print services
{'ftp': 21, 'http': 80, 'snmp': 161, 'smtp': 25, 'ssh': 22, 'ldap': 389}
```

```
>>> keys = services.keys()
>>> print keys
['ftp', 'smtp', 'ssh', 'http', 'snmp']
>>> keys.sort()
>>> print keys
['ftp', 'http', 'smtp', 'snmp', 'ssh']

>>> values = services.values()
>>> print values
[21, 25, 22, 80, 161]
>>> values.sort()
>>> print values
[21, 22, 25, 80, 161]

>>> services.has_key('http')
True
>>> services['http']
80
```

```
>>> for key,value in services.items():
        print key,value


ftp 21
smtp 25
ssh 22
http 80
snmp 161
```

Excepciones

```
>>> def divide (a, b):
        return a / b

>>> def calculate ():
        divide (1, 0)

>>> calculate()

Traceback (most recent call last):
  File "<pyshell#17>", line 1, in <module>
    calculate()
  File "<pyshell#16>", line 2, in calculate
    divide (1, 0)
  File "<pyshell#15>", line 2, in divide
    return a / b
ZeroDivisionError: integer division or modulo by zero
```

Excepciones

```
>>> try:
        print "[+] 10/0 = "+str(10/0)
except Exception, e:
        print "Error = "+str(e)


Error = integer division or modulo by zero
```

```
>>> try:
        f = file("file.txt")
except Exception,e:
        print "File not found =" +str(e)


File not found =[Errno 2] No such file or directory: 'file.txt'
```

Excepciones

```
>>> try:
...     lista =['python','java','c++']
...     print lista[0]
...     print lista[1]
...     print lista[2]
...     print lista[3]
... except Exception,e:
...     print str(e)
...
python
java
c++
list index out of range
```

Funciones

```
>>> def contains(sequence,item):
        for element in sequence:
                if element == item:
                        return True
        return False

>>>
>>> print contains([100,200,300,400],200)
True
>>> print contains([100,200,300,400],300)
True
>>> print contains([100,200,300,400],350)
False
```

Ficheros

- open(filename, mode)
  - retorna un objeto de tipo “file”
  - filename: nombre del fichero
  - mode: modo de apertura
    - Se pueden juntar: r+, w+, a+, rb, wb, …

Modo	Significado
'r'	Lectura (por defecto)
'w'	Escritura desde el principio del fichero
'x'	Abrir creando el fichero, falla si ya existe
'a'	Escritura desde el final del fichero
'b'	Modo binario
't'	Modo texto (por defecto)
'+'	Lectura y escritura
'U'	(Obsoleto)

Ficheros

```
>>> f = file('fichero','w')
>>> f.write('fichero abierto en modo escritura\n')
>>> f.write('por defecto se trata como un fichero de texto')
>>> f.close()
>>>
>>> for line in file('fichero'):
...     print line
...
fichero abierto en modo escritura

por defecto se trata como un fichero de texto
>>>
```

Expresiones regulares

```
Python 2.7.14 (v2.7.14:84471935ed, Sep 16 2017, 20:19:30) [MSC v.1500 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> import re
>>> data = "127.0.0.1"
>>> ipregex=re.compile("((25[0-5]|2[0-4][0-9]|[01]?[0-9][0-9]?)\\.){3}(25[0-5]|2[0-4][0-9]|[01]?[0-9][0-9]?)")
>>> match = re.match(ipregex, data)
>>> match
<_sre.SRE_Match object at 0x0352D3E0>
>>> match.group()
'127.0.0.1'
>>>
>>>
>>> data2 = "this is my ipaddress 127.0.0.1"
>>> search = re.search(ipregex,data2)
>>> search
<_sre.SRE_Match object at 0x0352DCA0>
>>> search.group()
'127.0.0.1'
>>>
>>>
>>> data3 = "this is my ipaddress 127.0.0000.1"
>>> search = re.search(ipregex,data3)
>>> search
>>> search.group()

Traceback (most recent call last):
  File "<pyshell#17>", line 1, in <module>
    search.group()
AttributeError: 'NoneType' object has no attribute 'group'
>>> print search
None
```

Funciones de orden superior

```
>>> list(map(lambda x: x**2, [1, 2, 3]))
[1, 4, 9]
>>> list(filter(lambda x: x > 1, [1, 2, 3]))
[2, 3]
>>> sorted([2, 1, 3], key=lambda x: x)
[1, 2, 3]
>>> sorted([1, 2, 3], key=lambda x: -x)
[3, 2, 1]
>>> from functools import reduce
>>> reduce(lambda x, y: x + y, [1, 2, 3])
6
```

Funciones de orden superior

```
>>> help(map)
Help on built-in function map in module __builtin__:

map(...)
    map(function, sequence[, sequence, ...]) -> list

    Return a list of the results of applying the function to the items of
    the argument sequence(s).  If more than one sequence is given, the
    function is called with an argument list consisting of the corresponding
    item of each sequence, substituting None for missing values when not all
    sequences have the same length.  If the function is None, return a list of
    the items of the sequence (or a list of tuples if more than one sequence).
```

Funciones de orden superior

```
>>> help(reduce)
Help on built-in function reduce in module __builtin__:

reduce(...)
    reduce(function, sequence[, initial]) -> value

    Apply a function of two arguments cumulatively to the items of a sequence
    from left to right, so as to reduce the sequence to a single value.
    For example, reduce(lambda x, y: x+y, [1, 2, 3, 4, 5]) calculates
    ((((1+2)+3)+4)+5).  If initial is present, it is placed before the items
    of the sequence in the calculation, and serves as a default when the
    sequence is empty.
```

POO

- Clases y objetos
- Paso de mensajes
- Clases definen comportamiento y estado
- Objetos como instancias de una clase

Clases

```
class Prueba:
    def __init__(self, x=2):
        self.x = x
        self.y = x**2
    def devuelve_datos(self):
        return “Con x=%i obtenemos: y=%i” % (self.x, self.y)
if __name__ == ‘__main__’:
    a1 = Prueba(2)
    print a1.devuelve_datos()
    a2 = Prueba(3)
    print a2.devuelve_datos()
```

Herencia

```
>>> class MyList(list):
...     def min_and_max(self):
...             return min(self),max(self)
...
>>> mylist = MyList()
>>> mylist.extend([100,150,200,250])
>>> print mylist
[100, 150, 200, 250]
>>> print mylist.min_and_max()
(100, 250)
```

Gestión de Paquetes

141,300 projects 990,028 releases 1,328,913 files 280,587 users

The Python Package Index (PyPI) is a repository of software for the Python programming language.

PyPI helps you find and install software developed and shared by the Python community. Learn about installing packages.

Package authors use PyPI to distribute their software. Learn how to package your Python code for PyPI.

Gestión de Paquetes

```
inux@linux-VirtualBox:~$ sudo apt install python-pip
eyendo lista de paquetes... Hecho
reando árbol de dependencias
eyendo la información de estado... Hecho
e instalarán los siguientes paquetes adicionales:
  build-essential dpkg-dev fakeroot g++ g++-7 gcc gcc-7 libalgorithm-diff-perl
  libalgorithm-diff-xs-perl libalgorithm-merge-perl libasan4 libatomic1
  libc-dev-bin libc6-dev libcilkrts5 libexpat1-dev libfakeroot libgcc-7-dev
  libitm1 liblsan0 libmpx2 libpython-all-dev libpython-dev libpython-stdlib
  libpython2.7-dev libquadmath0 libstdc++-7-dev libtsan0 libubsan0
  linux-libc-dev make manpages-dev python python-all python-all-dev
  python-asn1crypto python-cffi-backend python-crypto python-cryptography
  python-dbus python-dev python-enum34 python-gi python-idna python-ipaddress
  python-keyring python-keyrings.alt python-minimal python-pip-whl
  python-pkg-resources python-secretstorage python-setuptools python-six
  python-wheel python-xdg python2.7 python2.7-dev python2.7-minimal
aquetes sugeridos:
  debian-keyring g++-multilib g++-7-multilib gcc-7-doc libstdc++6-7-dbg
  gcc-multilib autoconf automake libtool flex bison gcc-doc gcc-7-multilib
  gcc-7-locales libgcc1-dbg libgomp1-dbg libitm1-dbg libatomic1-dbg
  libasan4-dbg liblsan0-dbg libtsan0-dbg libubsan0-dbg libcilkrts5-dbg
  libmpx2-dbg libquadmath0-dbg glibc-doc libstdc++-7-doc make-doc python-doc
```

```
linux@linux-VirtualBox:~$ pip list
DEPRECATION: The default format will switch to columns in
ns) (or define a format=(legacy|columns) in your pip.conf
ing.
asn1crypto (0.24.0)
cryptography (2.1.4)
enum34 (1.1.6)
idna (2.6)
ipaddress (1.0.17)
keyring (10.6.0)
keyrings.alt (3.0)
pip (9.0.1)
pycrypto (2.6.1)
pygobject (3.26.1)
pyxdg (0.25)
SecretStorage (2.3.1)
setuptools (39.0.1)
six (1.11.0)
wheel (0.30.0)
```

Pipreqs

Para generar el fichero requeriments.txt habría que ejecutar el comando **pipreqs <ruta_projecto>**

```
pipreqs - Generate pip requirements.txt file based on imports

Usage:
    pipreqs [options] <path>

Options:
    --use-local           Use ONLY local package info instead of querying PyPI
    --pypi-server         Use custom PyPi server
    --proxy               Use Proxy, parameter will be passed to requests library. You can also just set the
                          environments parameter in your terminal:
                          $ export HTTP_PROXY="http://10.10.1.10:3128"
                          $ export HTTPS_PROXY="https://10.10.1.10:1080"
    --debug               Print debug information
    --encoding <charset>  Use encoding parameter for file open
    --savepath <file>     Save the list of requirements in the given file
    --force               Overwrite existing requirements.txt
```

Requirements

- pip install -r requirements.txt
- pip freeze > requirements.txt

Piptools

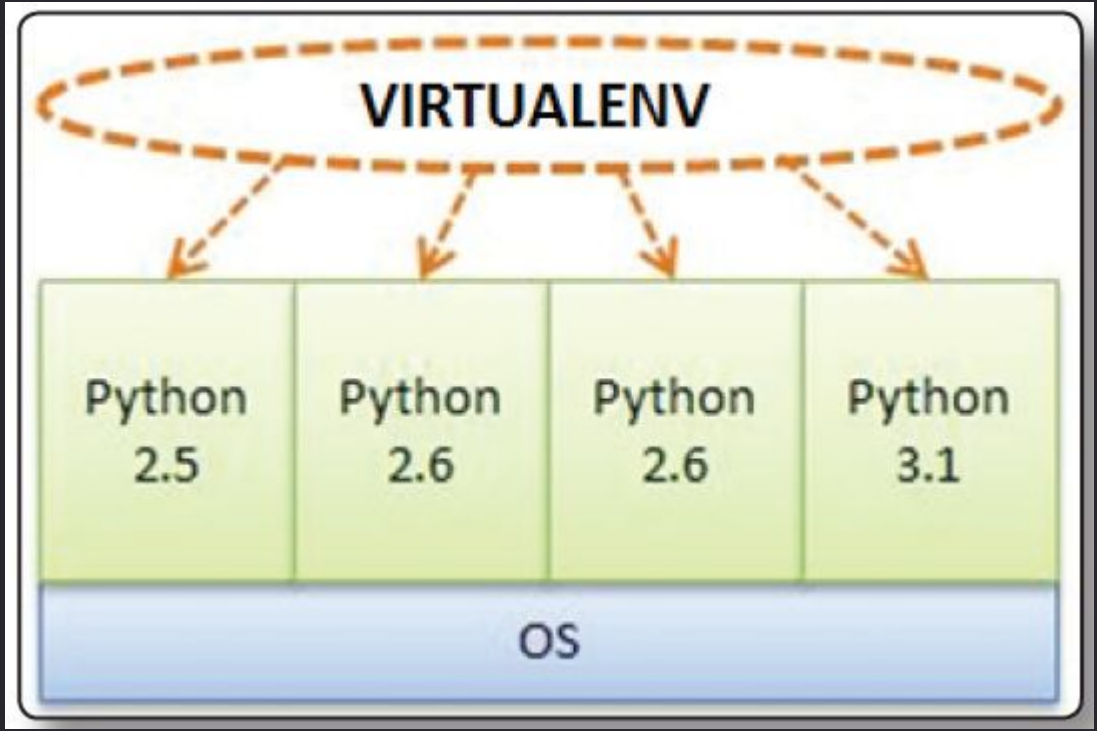
- https://github.com/jazzband/pip-tools
-

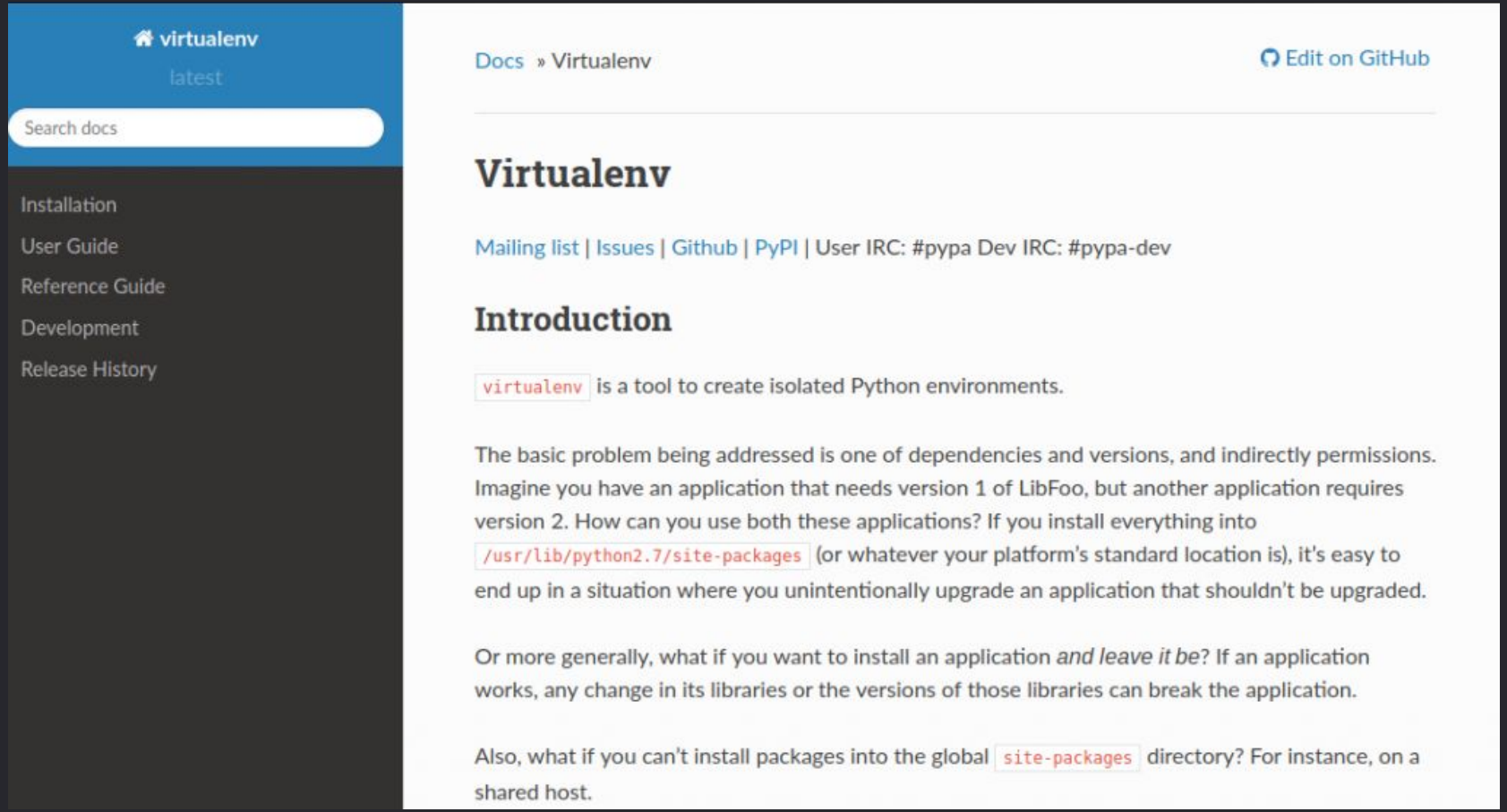
```
$ pip install pip-tools
...
$ pip-compile
#
# This file is autogenerated by pip-compile
# To update, run:
#
# pip-compile --output-file requirements.txt setup.py
#
arrow==0.10.0 # via stravalib
python-dateutil==2.6.0 # via arrow
pytz==2017.2 # via stravalib
requests==2.13.0 # via stravalib
```

VirtualEnv

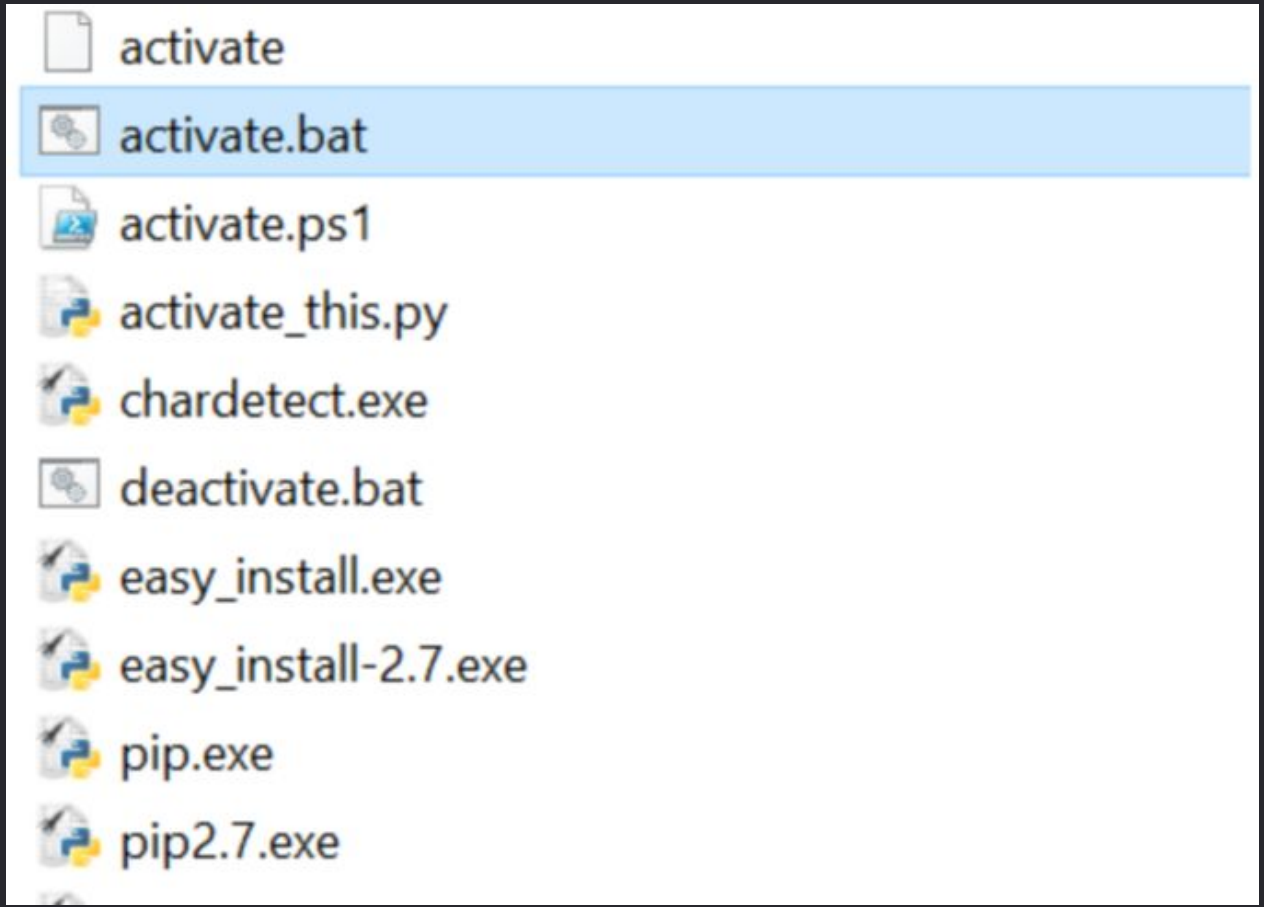
- Aislar nuestro proyecto a nivel de módulos y librerías instaladas, de los módulos que tenemos instalados a nivel global del sistema operativo.
- Ejecutar nuestro proyecto de forma aislada del resto de módulos y librerías del sistema operativo.

VirtualEnv



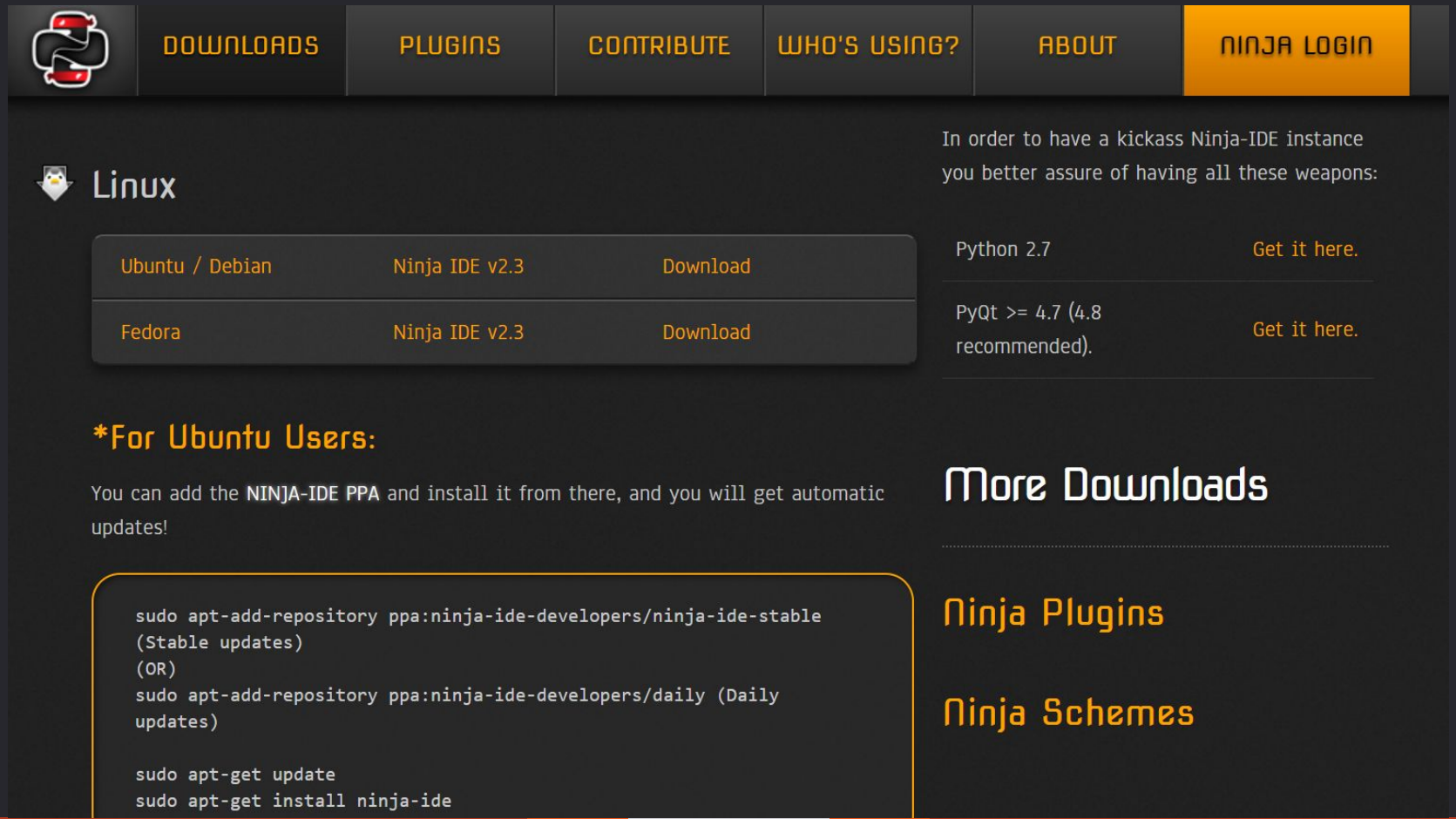

virtualenv
latest
Search docs
Installation
User Guide
Reference Guide
Development
Release History
Docs » Virtualenv
Edit on GitHub
Virtualenv
Mailing list | Issues | Github | PyPI | User IRC: #pypa Dev IRC: #pypa-dev
Introduction
virtualenv is a tool to create isolated Python environments.
The basic problem being addressed is one of dependencies and versions, and indirectly permissions. Imagine you have an application that needs version 1 of LibFoo, but another application requires version 2. How can you use both these applications? If you install everything into /usr/lib/python2.7/site-packages (or whatever your platform's standard location is), it's easy to end up in a situation where you unintentionally upgrade an application that shouldn't be upgraded.
Or more generally, what if you want to install an application and leave it be? If an application works, any change in its libraries or the versions of those libraries can break the application.
Also, what if you can't install packages into the global site-packages directory? For instance, on a shared host.

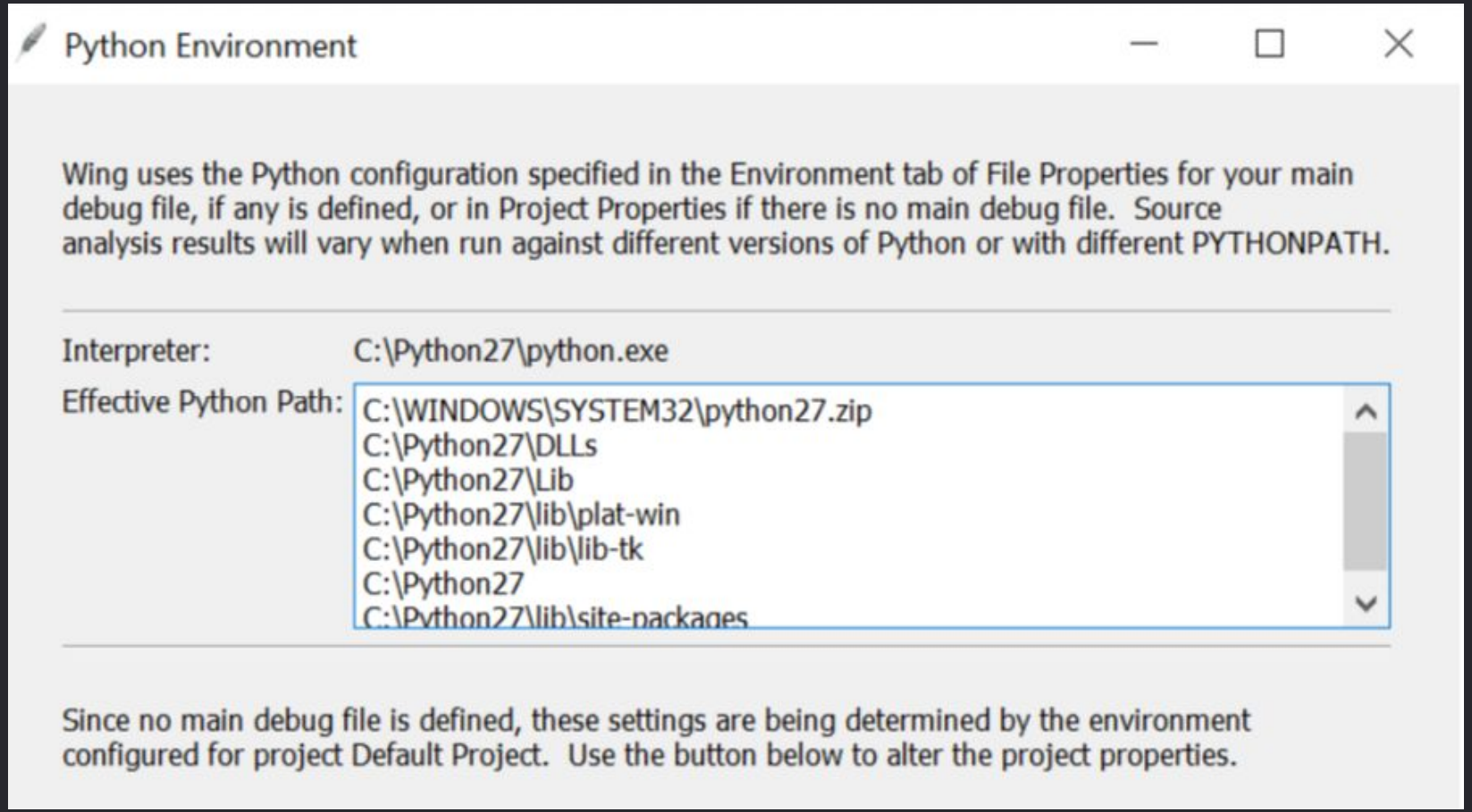
```
New python executable in venv\Scripts\python.exe
Installing setuptools, pip, wheel...done.
```


activate
activate.bat
activate.ps1
activate_this.py
chardetect.exe
deactivate.bat
easy_install.exe
easy_install-2.7.exe
pip.exe
pip2.7.exe

IDES

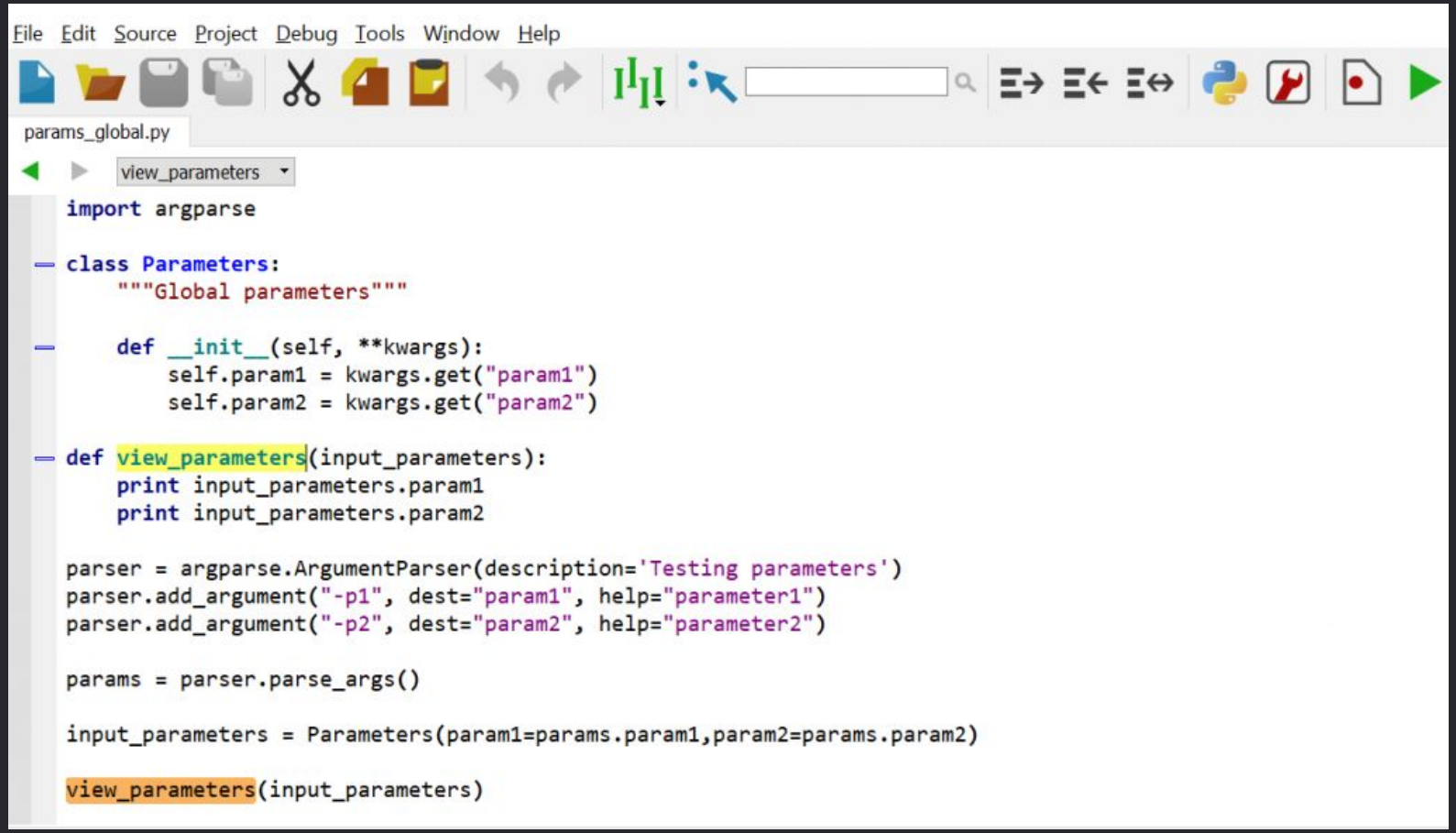

Creating singletons... BrowserMgr
PERSONAL 6
WINGWARE
PYTHON IDE
THE INTELLIGENT DEVELOPMENT ENVIRONMENT for PYTHON PROGRAMMERS
COPYRIGHT 1999-2018, ARCHAEOPTERYX SOFTWARE, INC D.B.A. WINGWARE

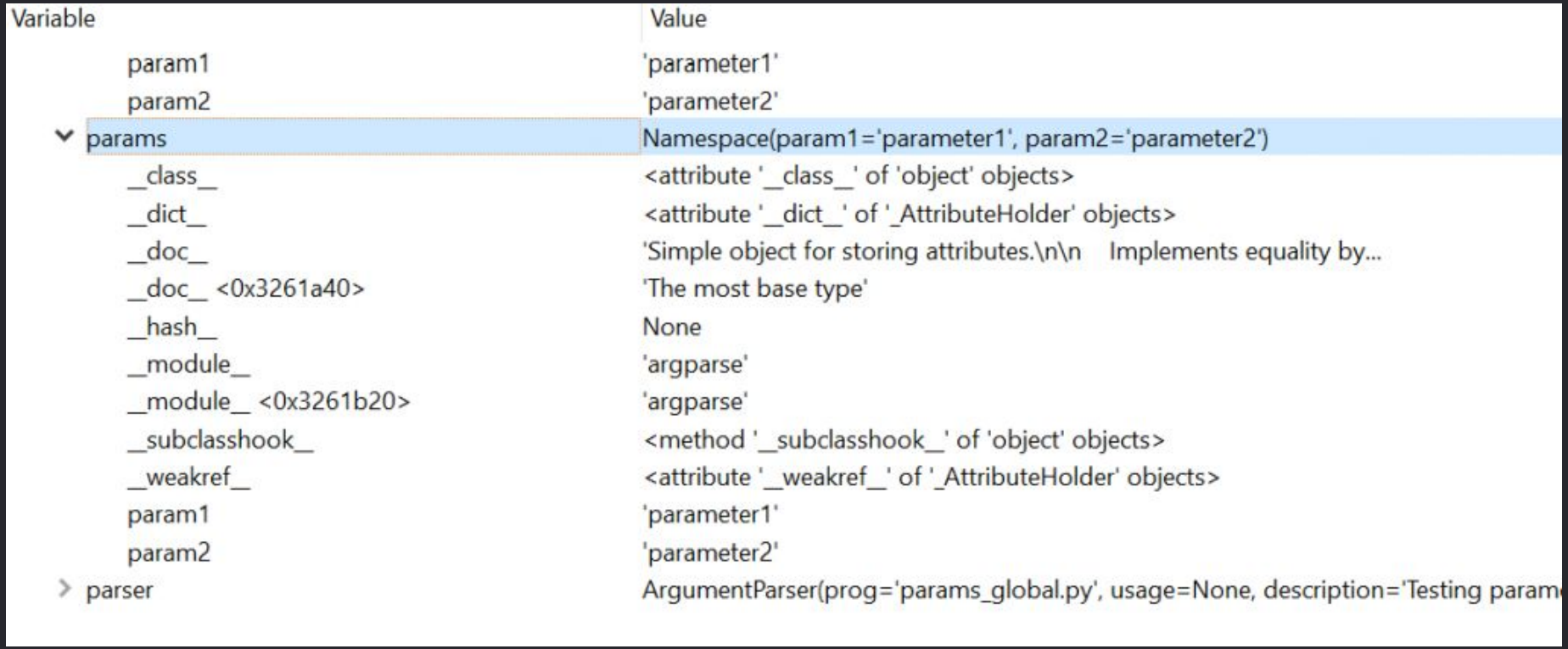

DOWNLOADS
PLUGINS
CONTRIBUTE
WHO'S USING?
ABOUT
NINJA LOGIN
Linux
Ubuntu / Debian
Ninja IDE v2.3
Download
Fedora
Ninja IDE v2.3
Download
*For Ubuntu Users:
You can add the NINJA-IDE PPA and install it from there, and you will get automatic updates!
sudo apt-add-repository ppa:ninja-ide-developers/ninja-ide-stable
(Stable updates)
(OR)
sudo apt-add-repository ppa:ninja-ide-developers/daily (Daily updates)
sudo apt-get update
sudo apt-get install ninja-ide
In order to have a kickass Ninja-IDE instance you better assure of having all these weapons:
Python 2.7
Get it here.
PyQt >= 4.7 (4.8 recommended).
Get it here.
More Downloads
Ninja Plugins
Ninja Schemes


Python Environment
Wing uses the Python configuration specified in the Environment tab of File Properties for your main debug file, if any is defined, or in Project Properties if there is no main debug file. Source analysis results will vary when run against different versions of Python or with different PYTHONPATH.
Interpreter:
C:\Python27\python.exe
Effective Python Path:
C:\WINDOWS\SYSTEM32\python27.zip
C:\Python27\DLLs
C:\Python27\Lib
C:\Python27\lib\plat-win
C:\Python27\lib\lib-tk
C:\Python27
C:\Python27\lib\site-packages
Since no main debug file is defined, these settings are being determined by the environment configured for project Default Project. Use the button below to alter the project properties.

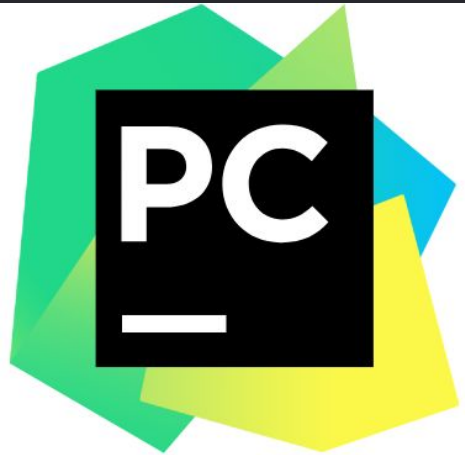
paso_parametros.py

```
import argparse
parser = argparse.ArgumentParser(description='Paso de parametros')
parser.add_argument("-p1", dest="param1", help="parameter1")
parser.add_argument("-p2", dest="param2", help="parameter2")
params = parser.parse_args()
print params.param1
print params.param2
```


File Edit Source Project Debug Tools Window Help
params_global.py
view_parameters
import argparse
class Parameters:
"""Global parameters"""
def __init__(self, **kwargs):
self.param1 = kwargs.get("param1")
self.param2 = kwargs.get("param2")
def view_parameters(input_parameters):
print input_parameters.param1
print input_parameters.param2
parser = argparse.ArgumentParser(description='Testing parameters')
parser.add_argument("-p1", dest="param1", help="parameter1")
parser.add_argument("-p2", dest="param2", help="parameter2")
params = parser.parse_args()
input_parameters = Parameters(param1=params.param1,param2=params.param2)
view_parameters(input_parameters)



Variable	Value
param1	'parameter1'
param2	'parameter2'
▼ params	Namespace(param1='parameter1', param2='parameter2')
__class__	<attribute '__class__' of 'object' objects>
__dict__	<attribute '__dict__' of '_AttributeHolder' objects>
__doc__	'Simple object for storing attributes.\n\n Implements equality by...
__doc__ <0x3261a40>	'The most base type'
__hash__	None
__module__	'argparse'
__module__ <0x3261b20>	'argparse'
__subclasshook__	<method '__subclasshook__' of 'object' objects>
__weakref__	<attribute '__weakref__' of '_AttributeHolder' objects>
param1	'parameter1'
param2	'parameter2'
▶ parser	ArgumentParser(prog='params_global.py', usage=None, description='Testing param



Download PyCharm

Windows macOS Linux

Version: 2018.1.4

Build: 181.5087.37

Released: May 31, 2018

System requirements

Installation Instructions

Previous versions

Professional

Full-featured IDE
for Python & Web
development

DOWNLOAD

Free trial

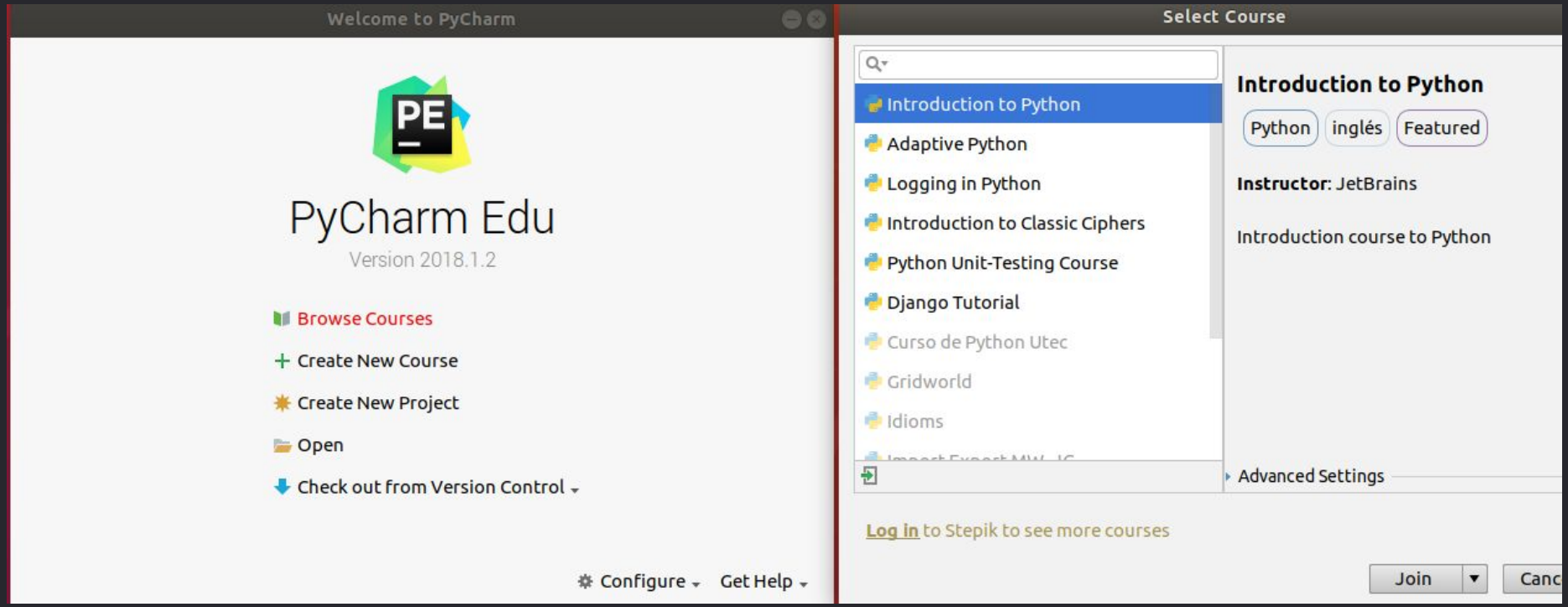
Community

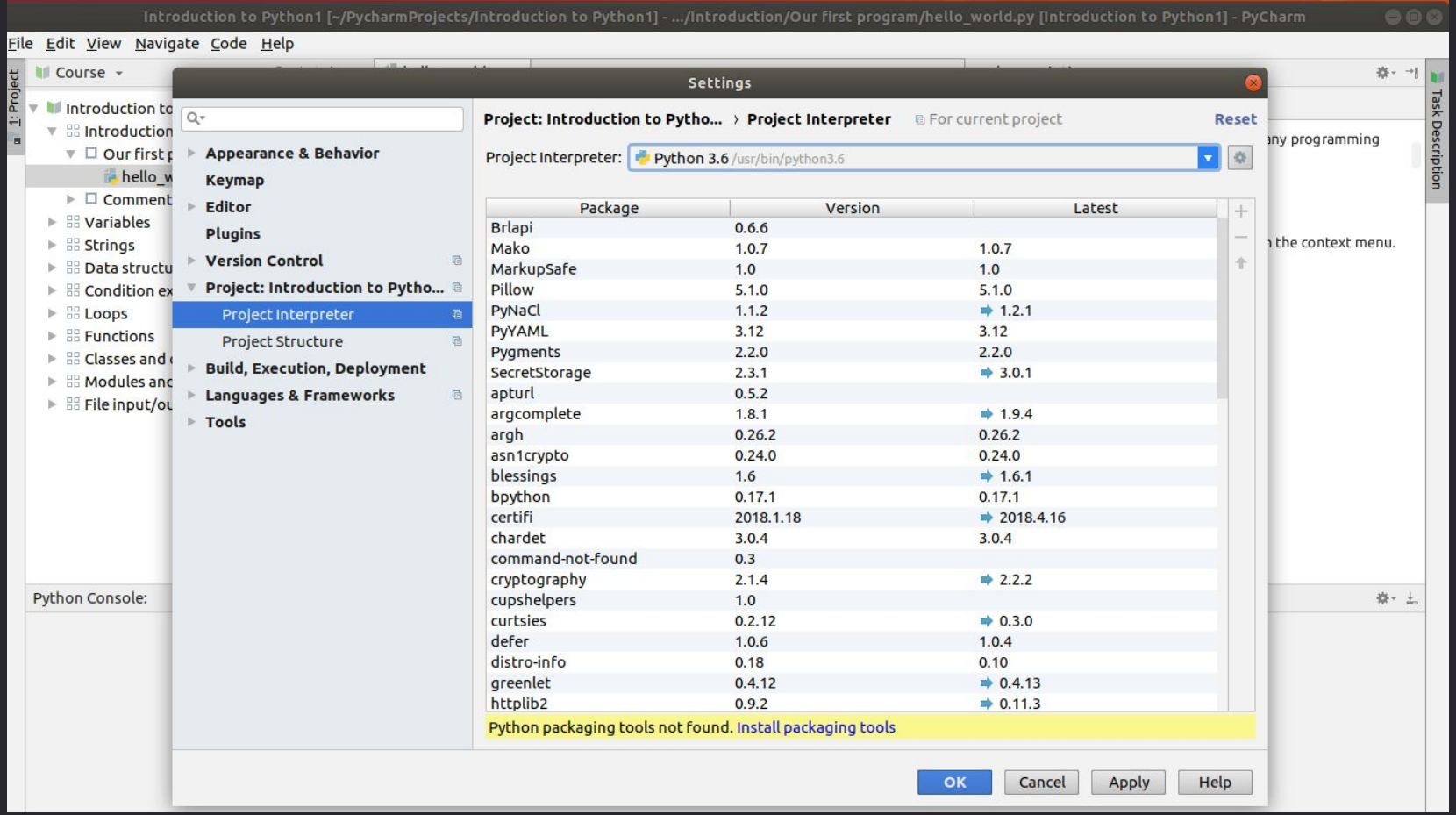
Lightweight IDE
for Python & Scientific
development

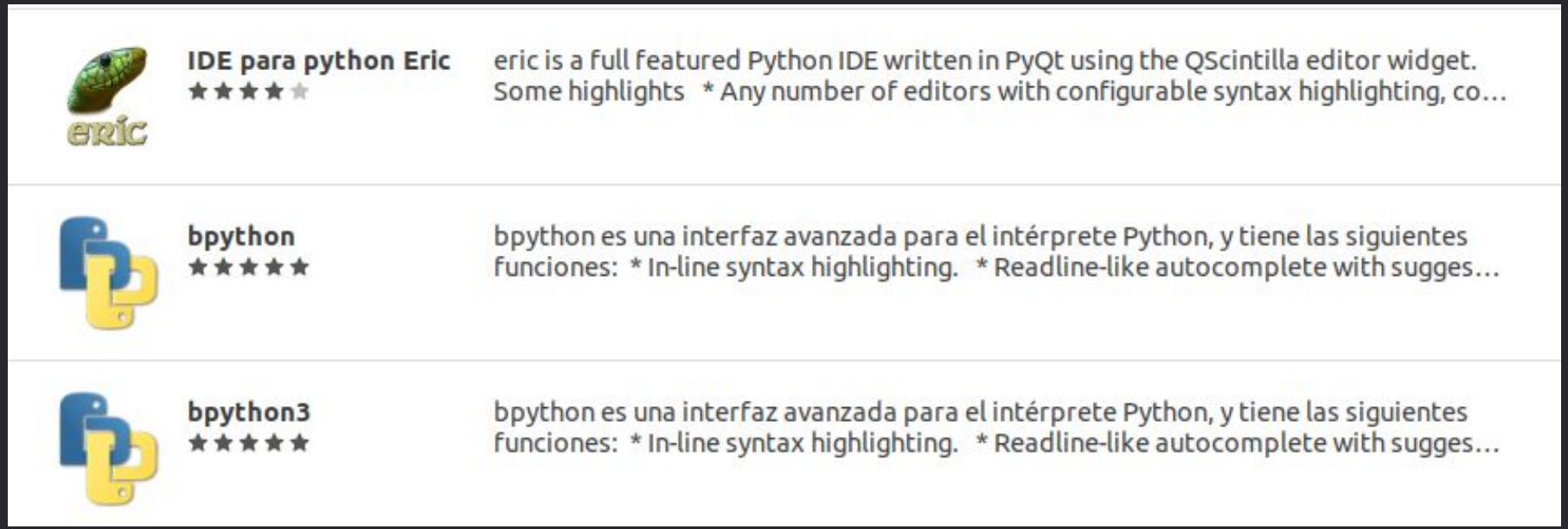
DOWNLOAD

Free, open-source


PyCharm Pro
PyCharm Professional Edition is an IDE for professional Python development. It is designed by programmers, for programmers, to provide all the tools you need for…
PyCharm CE
PyCharm Community Edition is a free and open-source IDE which is perfect for pure Python coding. For professional Web and Scientific development see PyCharm Pr…
PyCharm EDU
PyCharm Edu combines interactive learning with a powerful real-world professional development tool to provide a platform for the most effective learning and teachi…


Welcome to PyCharm
PyCharm Edu
Version 2018.1.2
Browse Courses
Create New Course
Create New Project
Open
Check out from Version Control
Configure
Get Help
Select Course
Introduction to Python
Adaptive Python
Logging in Python
Introduction to Classic Ciphers
Python Unit-Testing Course
Django Tutorial
Curso de Python Utec
Gridworld
Idioms
Introduction to Python
Python
inglés
Featured
Instructor: JetBrains
Introduction course to Python
Advanced Settings
Log in to Stepik to see more courses
Join
Canc

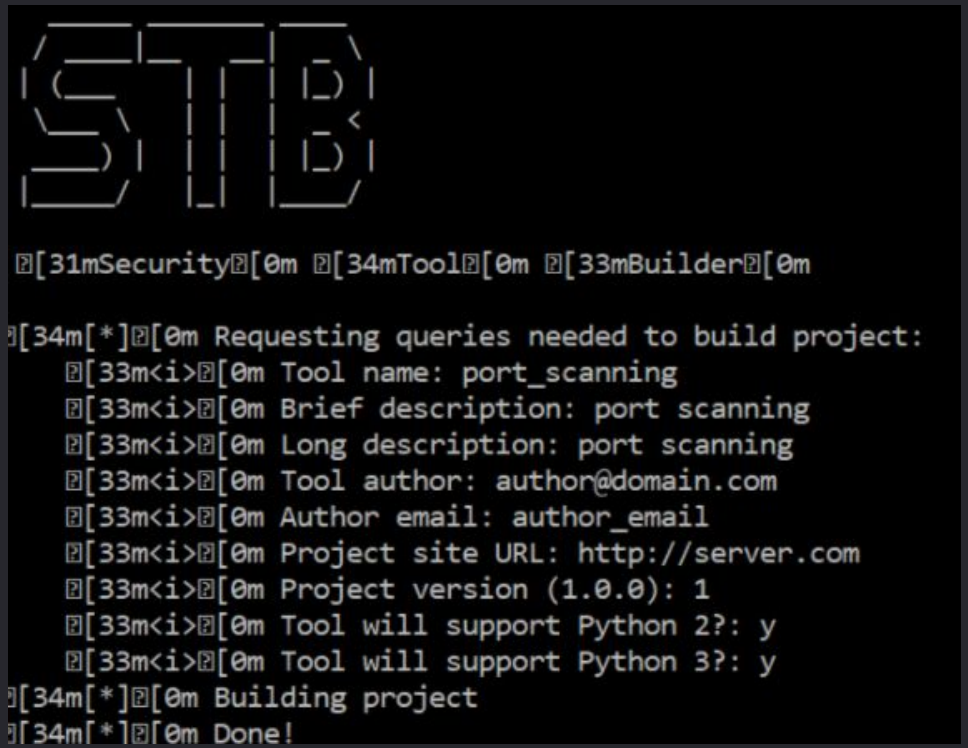

Introduction to Python1 [~/PycharmProjects/Introduction to Python1] - .../Introduction/Our first program/hello_world.py [Introduction to Python1] - PyCharm
File Edit View Navigate Code Help
Course
Introduction to
Introduction
Our first p
hello_w
Comment
Variables
Strings
Data structu
Condition ex
Loops
Functions
Classes and
Modules and
File input/ou
Python Console:
1: Project
Task Description
any programming
the context menu.
Settings
Appearance & Behavior
Keymap
Editor
Plugins
Version Control
Project: Introduction to Pytho...
Project Interpreter
Project Structure
Build, Execution, Deployment
Languages & Frameworks
Tools
Project: Introduction to Pytho... › Project Interpreter
For current project
Reset
Project Interpreter: Python 3.6 /usr/bin/python3.6
Package
Version
Latest
Brlapi 0.6.6
Mako 1.0.7 1.0.7
MarkupSafe 1.0 1.0
Pillow 5.1.0 5.1.0
PyNaCl 1.1.2 1.2.1
PyYAML 3.12 3.12
Pygments 2.2.0 2.2.0
SecretStorage 2.3.1 3.0.1
apturl 0.5.2
argcomplete 1.8.1 1.9.4
argh 0.26.2 0.26.2
asn1crypto 0.24.0 0.24.0
blessings 1.6 1.6.1
bpython 0.17.1 0.17.1
certifi 2018.1.18 2018.4.16
chardet 3.0.4 3.0.4
command-not-found 0.3
cryptography 2.1.4 2.2.2
cupshelpers 1.0
curtsies 0.2.12 0.3.0
defer 1.0.6 1.0.4
distro-info 0.18 0.10
greenlet 0.4.12 0.4.13
httplib2 0.9.2 0.11.3
Python packaging tools not found. Install packaging tools
OK
Cancel
Apply
Help

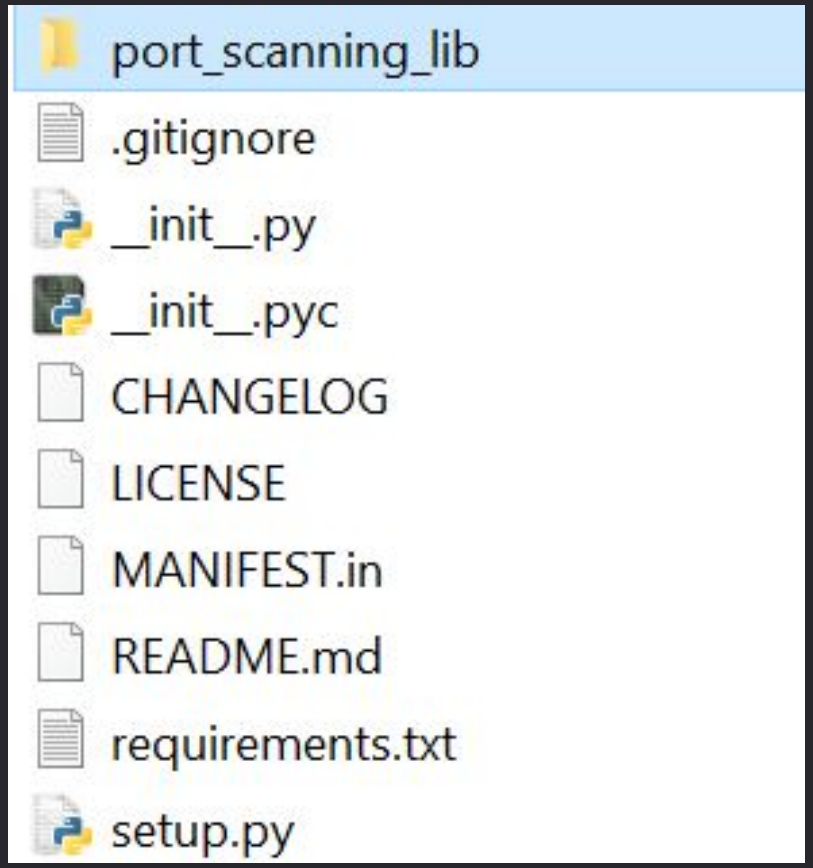

IDE para python Eric
★★★★☆
eric is a full featured Python IDE written in PyQt using the QScintilla editor widget. Some highlights * Any number of editors with configurable syntax highlighting, co…
bpython
★★★★★
bpython es una interfaz avanzada para el intérprete Python, y tiene las siguientes funciones: * In-line syntax highlighting. * Readline-like autocomplete with sugges…
bpython3
★★★★★
bpython es una interfaz avanzada para el intérprete Python, y tiene las siguientes funciones: * In-line syntax highlighting. * Readline-like autocomplete with sugges…

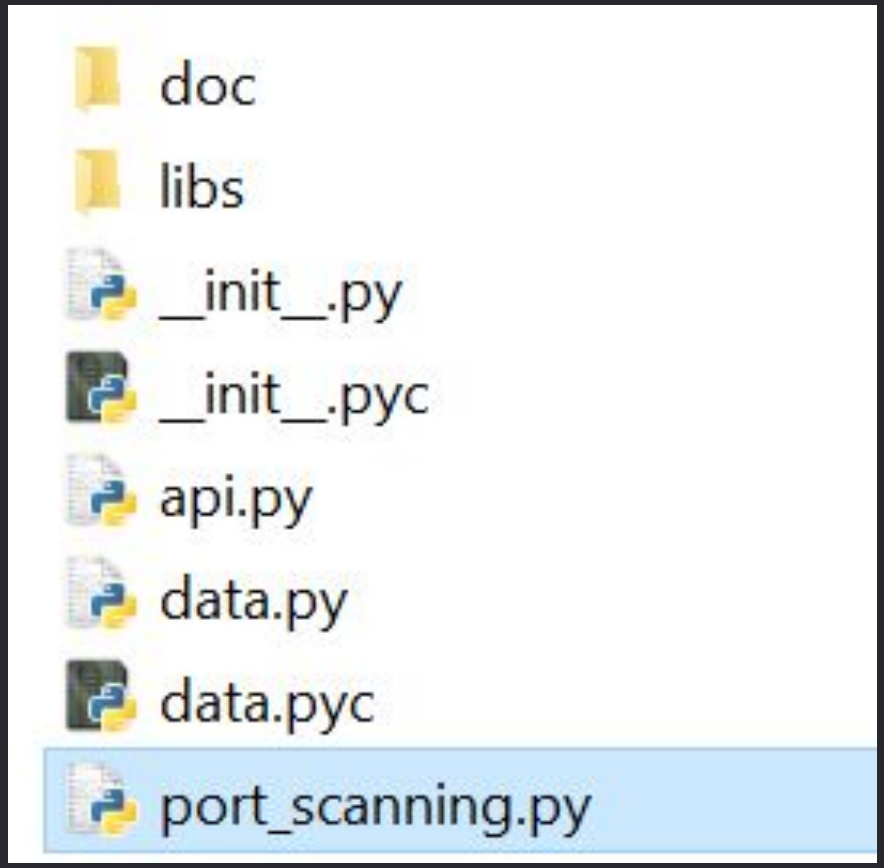
Herramientas

- **Web Scraping**(BeautifulSoup,Scrapy)
- **Desarrollo web**(Django,Flask,API REST)
- **Machine Learning**(Sklearn,Tensorflow)
- **Desarrollo de bots**(Telegram,Slack)
- **Herramientas de seguridad**(sqlmap,sparta)

STB(Security Tools Builder)

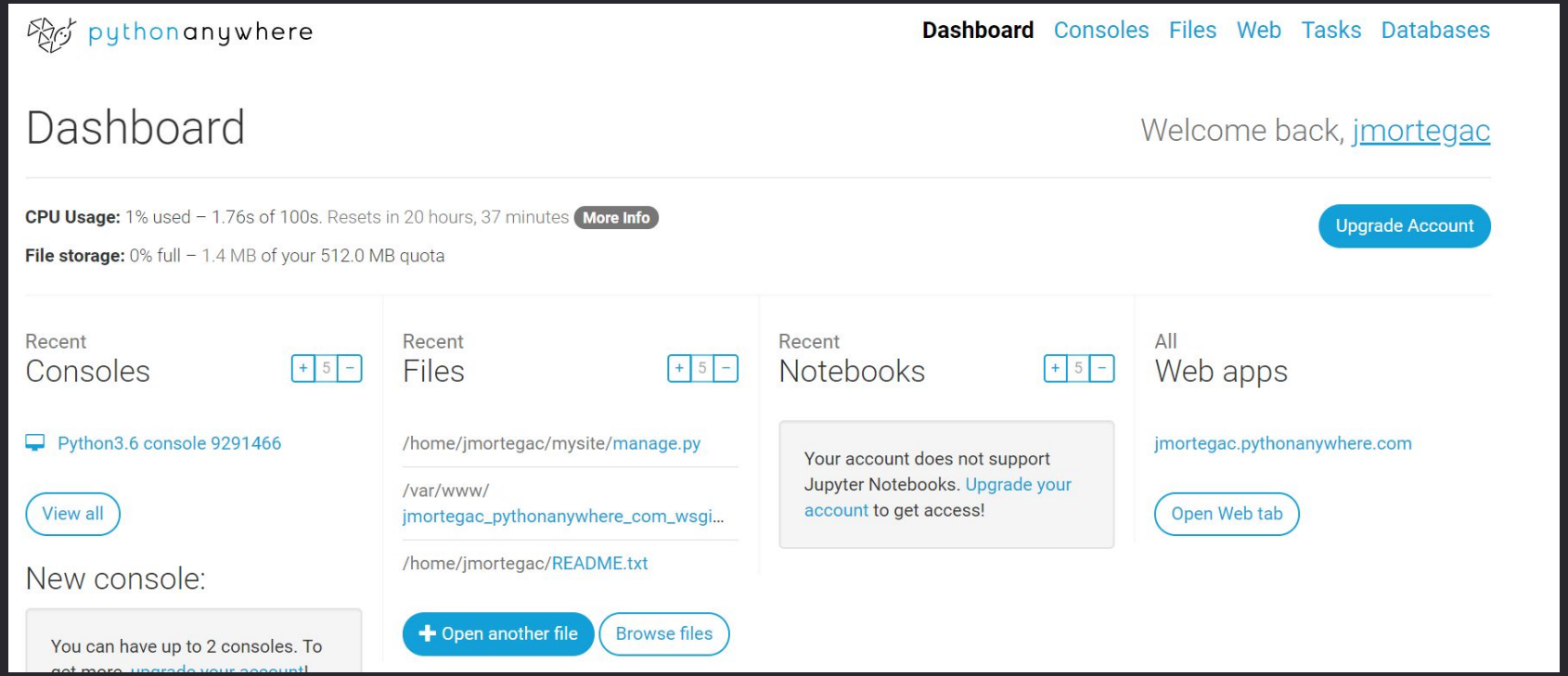



port_scanning_lib
.gitignore
__init__.py
__init__.pyc
CHANGELOG
LICENSE
MANIFEST.in
README.md
requirements.txt
setup.py


doc
libs
__init__.py
__init__.pyc
api.py
data.py
data.pyc
port_scanning.py

```
usage: port_scanning.py [-h] [-v] [TARGET [TARGET ...]]

Port_scanning security tool

positional arguments:
  TARGET

optional arguments:
  -h, --help       show this help message and exit
  -v, --verbosity  verbosity level: -v, -vv, -vvv.

Examples:

    * Scan target using default 50 most common plugins:
        port_scanning TARGET
```

Python in the cloud



Python modules

```
In : help('modules')

Please wait a moment while I gather a list of all available modules...

/usr/local/lib/python3.6/dist-packages/flask_admin/contrib/peeweemodel/__init__.py:3: UserWarning:

Flask-Admin peewee integration module was renamed as flask_admin.contrib.peewee, please use it instead.

BTrees              code                mailbox             secrets
CoolProp            codecs              mailcap             sekizai
Crypto              codeop              mako                select
Cython              collections         marisa_trie         selectors
IPython             colorsys            markdown            selenium
Levenshtein         compileall          markupsafe          serial
MySQLdb             concurrent          marshal             setproctitle
OleFileIO_PL        configobj           martian             setuptools
OpenSSL             configparser        math                shapely
PIL                 contextlib          matplotlib          shelve
SPARQLWrapper       contextlib2         mechanize           shlex
Stemmer             copy                menus               shutil
ZConfig             copyreg             mezzanine           signal
ZEO                 coverage            mimeparse           simplegeneric
ZODB                crypt               mimerender          simplejson
__future__          cryptography        mimetypes           simpy
_ast                cssselect           mistune             singledispatch
_asyncio            csv                 mlpy                singledispatch_helpers
_bisect             ctypes              mmap                site
_blake2             curl                mock                sitecustomize
_bootlocale         curses              modulefinder        six
_bz2                cvxopt              more_itertools      skimage
_cffi_backend       cycler              morepath            sklearn
_codecs             cyordereddict       mots_vides          slugify
_codecs_cn          cython              mpmath              smart_open
```

Python para proyectos de seguridad

Python se ha convertido en el lenguaje más usado para desarrollar herramientas dentro del ámbito de la seguridad.

Muchas de las herramientas que se pueden encontrar hoy en día como escáner de puertos, análisis de vulnerabilidades, ataques por fuerza bruta y hacking de passwords, se han escrito en este lenguaje ,además de ofrecer un ecosistema de herramientas para realizar pruebas de seguridad y de pentesting de aplicaciones.

Entre los puntos a tratar se pueden destacar:

- Herramientas de seguridad que se pueden encontrar realizadas en python(sqlmap,theharvester,sparta)
- Introducir librerías para obtener información del objetivo como Shodan,pygeocoder,pygeoip
- Análisis y extracción de metadatos en Python en imágenes y documentos
- Análisis de puertos con herramientas como python-nmap
- Conexión con servidores FTP,SSH