REWRITING AND RE-RELEASING

VIRTUALENV

😊 LESSONS FROM THE TRENCHES 😊

by Bernát Gábor / @gjbernat / Bloomberg

slides @ https://bit.ly/virtualenv-eupy20

Bloomberg

WHAT IS VIRTUALENV?

WHAT IS VIRTUALENV?

- a tool to create Python *virtual environments*

WHAT IS VIRTUALENV?

- a tool to create Python *virtual environments*
- Python executables that behave as it would be a separate Python installation (from the system one)

WHAT IS VIRTUALENV?

- a tool to create Python *virtual environments*
- Python executables that behave as it would be a separate Python installation (from the system one)
  - packages installed do not affect the system Python

WHAT IS VIRTUALENV

```
$ virtualenv env --python python3.8
```

WHAT IS VIRTUALENV

```
$ virtualenv env --python python3.8
```

```
$ env/bin/python -m pip install httpie
```

WHAT IS VIRTUALENV

```
$ virtualenv env --python python3.8
```

```
$ env/bin/python -m pip install httpie
```

```
$ env/bin/http https://httpie.org/hello
HTTP/1.1 200 OK
Connection: keep-alive
Content-Length: 116
Content-Type: text/x-rst;charset=utf-8
Date: Thu, 02 Jul 2020 17:00:46 GMT

Hello, World! 👋
~~~~~~~~~~~~~~~

Thank you for trying out HTTPie 🥳

I hope this will become a friendship.
```

WHAT IS A VIRTUAL ENVIRONMENT?

- While isolating package installs from the system, it mirrors the system with regards to:
  - implementation, version, build, etc

WHAT IS A VIRTUAL ENVIRONMENT?

- While isolating package installs from the system, it mirrors the system with regards to:
  - implementation, version, build, etc

```
$ /usr/bin/python -c 'import sys; import os; print(sys.executable); print(sys.version); prin
/System/Library/Frameworks/Python.framework/Versions/2.7/Resources/Python.app/Contents/MacOS
2.7.16 (default, Apr 17 2020, 18:29:03)
[GCC 4.2.1 Compatible Apple LLVM 11.0.3 (clang-1103.0.29.20) (-macos10.15-objc-
/System/Library/Frameworks/Python.framework/Versions/2.7/lib/python2.7/os.pyc
```

WHAT IS A VIRTUAL ENVIRONMENT?

- While isolating package installs from the system, it mirrors the system with regards to:
  - implementation, version, build, etc

```
$ /usr/bin/python -c 'import sys; import os; print(sys.executable); print(sys.version); prin
/System/Library/Frameworks/Python.framework/Versions/2.7/Resources/Python.app/Contents/MacOS
2.7.16 (default, Apr 17 2020, 18:29:03)
[GCC 4.2.1 Compatible Apple LLVM 11.0.3 (clang-1103.0.29.20) (-macos10.15-objc-
/System/Library/Frameworks/Python.framework/Versions/2.7/lib/python2.7/os.pyc
```

```
$ virtualenv env --python /usr/bin/python
```

WHAT IS A VIRTUAL ENVIRONMENT?

- While isolating package installs from the system, it mirrors the system with regards to:
  - implementation, version, build, etc

```
$ /usr/bin/python -c 'import sys; import os; print(sys.executable); print(sys.version); prin
/System/Library/Frameworks/Python.framework/Versions/2.7/Resources/Python.app/Contents/MacOS
2.7.16 (default, Apr 17 2020, 18:29:03)
[GCC 4.2.1 Compatible Apple LLVM 11.0.3 (clang-1103.0.29.20) (-macos10.15-objc-
/System/Library/Frameworks/Python.framework/Versions/2.7/lib/python2.7/os.pyc
```

```
$ virtualenv env --python /usr/bin/python
```

```
$ env/python -c 'import sys; import os; print(sys.executable); print(sys.version); print(os.
/tmp/env/bin/python
2.7.16 (default, Apr 17 2020, 18:29:03)
[GCC 4.2.1 Compatible Apple LLVM 11.0.3 (clang-1103.0.29.20) (-macos10.15-objc-
/System/Library/Frameworks/Python.framework/Versions/2.7/lib/python2.7/os.pyc
```

HOW POPULAR IS VIRTUALENV?

- statistics via pypinfo processes pypi.org logs stored in Google BigQuery

```
pypinfo --all -l 365 --json --start-date 2019-02-10 --end-date 2020-02-10 virtualenv
```

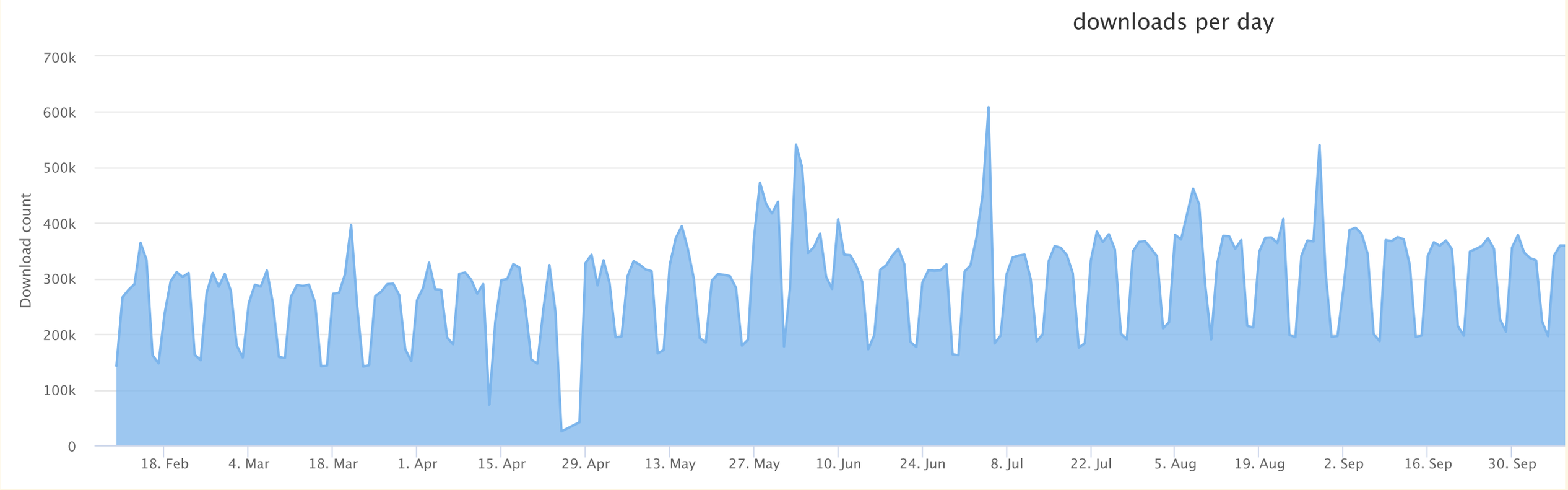
HOW POPULAR IS VIRTUALENV?

- statistics via pypinfo processes pypi.org logs stored in Google BigQuery

```
pypinfo --all -l 365 --json --start-date 2019-02-10 --end-date 2020-02-10 virtualenv
```

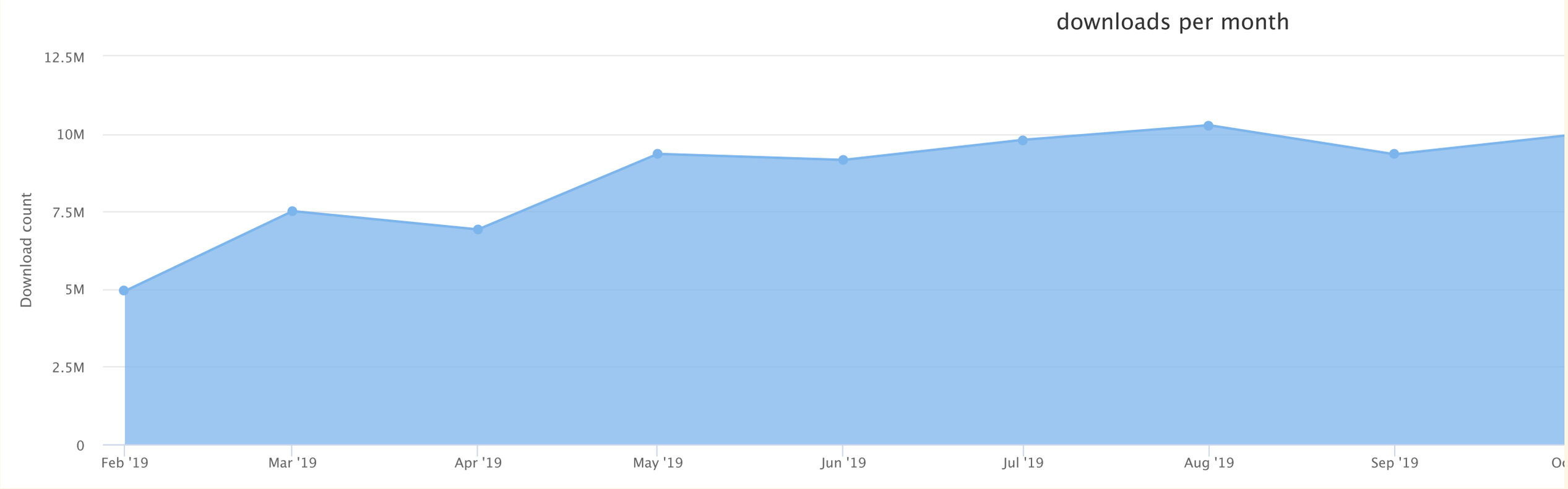
- first 1TB free (virtualenv 12 months ~0.8TB)

HOW POPULAR IS VIRTUALENV?



HOW POPULAR IS VIRTUALENV?

HOW POPULAR IS VIRTUALENV?



TOP PACKAGES CHART VIA HUGOVK.GITHUB.IO

Showing 100 packages over 365 days.

Show 100 | Show 1,000 | Show 4,000 | 30 days | 365 days

1.	urllib3	1,045,223,100
2.	six	887,178,051
3.	botocore	775,690,465
4.	requests	741,920,834
5.	python-dateutil	730,035,765
	...	
61.	psutil	132,029,527
62.	itsdangerous	131,496,848
63.	scikit-learn	131,037,830
64.	pyjwt	127,723,499
65.	wrapt	124,600,488
66.	virtualenv	121,074,344
67.	asn1crypto	117,898,115

VIRTUAL ENVIRONMENTS IN DEPTH

Bernat Gabor - status quo of virtual environments

VIRTUAL ENVIRONMENT CREATION TOOLS

1. virtualenv

```
$ virtualenv env --python python3.8
```

VIRTUAL ENVIRONMENT CREATION TOOLS

1. virtualenv

```
$ virtualenv env --python python3.8
```

2. venv as per PEP-405

```
$ python3.8 -m venv env
```

DIFFERENCES BETWEEN THE TWO TOOLS

	virtualenv	*venv*

DIFFERENCES BETWEEN THE TWO TOOLS

	virtualenv	*venv*
Installation	yes, third-party package	no, standard library

DIFFERENCES BETWEEN THE TWO TOOLS

	virtualenv	*venv*
Installation	yes, third-party package	no, standard library
Update	anytime via pip	with the interpreter

DIFFERENCES BETWEEN THE TWO TOOLS

	virtualenv	*venv*
Installation	yes, third-party package	no, standard library
Update	anytime via pip	with the interpreter
Python version	2.7 and 3.4 or later	3.3 or later

DIFFERENCES BETWEEN THE TWO TOOLS

	virtualenv	*venv*
Installation	yes, third-party package	no, standard library
Update	anytime via pip	with the interpreter
Python version	2.7 and 3.4 or later	3.3 or later
Python flavour	PyPy, CPython	self-support for 3.3+

DIFFERENCES BETWEEN THE TWO TOOLS

	virtualenv	*venv*
Installation	yes, third-party package	no, standard library
Update	anytime via pip	with the interpreter
Python version	2.7 and 3.4 or later	3.3 or later
Python flavour	PyPy, CPython	self-support for 3.3+
Configurability	CLI + environment variables + per user config file	CLI

DIFFERENCES BETWEEN THE TWO TOOLS

	virtualenv	*venv*
Installation	yes, third-party package	no, standard library
Update	anytime via pip	with the interpreter
Python version	2.7 and 3.4 or later	3.3 or later
Python flavour	PyPy, CPython	self-support for 3.3+
Configurability	CLI + environment variables + per user config file	CLI
Extensibility	via plugins that can be installed alongside	by extending the base class

DIFFERENCES BETWEEN THE TWO TOOLS

	virtualenv	*venv*
Installation	yes, third-party package	no, standard library
Update	anytime via pip	with the interpreter
Python version	2.7 and 3.4 or later	3.3 or later
Python flavour	PyPy, CPython	self-support for 3.3+
Configurability	CLI + environment variables + per user config file	CLI
Extensibility	via plugins that can be installed alongside	by extending the base class
Performance	by using the cached seed mechanism <500ms	2 seconds+ per invocation

DIFFERENCES BETWEEN THE TWO TOOLS

	virtualenv	*venv*
Installation	yes, third-party package	no, standard library
Update	anytime via pip	with the interpreter
Python version	2.7 and 3.4 or later	3.3 or later
Python flavour	PyPy, CPython	self-support for 3.3+
Configurability	CLI + environment variables + per user config file	CLI
Extensibility	via plugins that can be installed alongside	by extending the base class
Performance	by using the cached seed mechanism <500ms	2 seconds+ per invocation
API	rich: e.g., lookup executable, site package dir	limited as per PEP-405

IN A WORLD WITH VENV IS VIRTUALENV NEEDED?

IN A WORLD WITH VENV IS VIRTUALENV NEEDED?

- yes, virtualenv is a place to innovate and improve virtual environments

IN A WORLD WITH VENV IS VIRTUALENV NEEDED?

- yes, virtualenv is a place to innovate and improve virtual environments
- as of today virtualenv (compared to venv) has better
  - creation time (performance)
  - API
  - extensibility

THE MOTIVATION (WHY)

OLD PAINS

OLD PAINS

- single file - of spurious if/else branches

OLD PAINS

- single file - of spurious if/else branches
- test suite ~ 60% line coverage

OLD PAINS

- single file - of spurious if/else branches
- test suite ~ 60% line coverage
- rudimentary plugin system makes it hard to extend

OLD PAINS

- single file - of spurious if/else branches
- test suite ~ 60% line coverage
- rudimentary plugin system makes it hard to extend
- things that made sense 12 years ago are no longer true

EARLIER REWRITE EFFORTS

At least 2 attempts, both abandoned halfway through because:

EARLIER REWRITE EFFORTS

At least 2 attempts, both abandoned halfway through because:

- deceptively simple

EARLIER REWRITE EFFORTS

At least 2 attempts, both abandoned halfway through because:

- deceptively simple, but lots of nuances and edge cases

EARLIER REWRITE EFFORTS

At least 2 attempts, both abandoned halfway through because:

- deceptively simple, but lots of nuances and edge cases
- hard to test - a myriad of Python distributions/version/platforms

HOW I JUMPED IN

HOW I JUMPED IN

- tox maintainer since June 2017 (virtualenv dependency)

HOW I JUMPED IN

- tox maintainer since June 2017 (virtualenv dependency)
- virtualenv maintainer since October 2018

HOW I JUMPED IN

- tox maintainer since June 2017 (virtualenv dependency)
- virtualenv maintainer since October 2018
- tox is slow, April 2019 started tox rewrite to improve on it, discoveries:

HOW I JUMPED IN

- tox maintainer since June 2017 (virtualenv dependency)
- virtualenv maintainer since October 2018
- tox is slow, April 2019 started tox rewrite to improve on it, discoveries:
  - virtual environment creation is slow in virtualenv

HOW I JUMPED IN

- tox maintainer since June 2017 (virtualenv dependency)
- virtualenv maintainer since October 2018
- tox is slow, April 2019 started tox rewrite to improve on it, discoveries:
  - virtual environment creation is slow in virtualenv
  - virtualenv API lacks ability to provide path to site packages

HOW I JUMPED IN

- tox maintainer since June 2017 (virtualenv dependency)
- virtualenv maintainer since October 2018
- tox is slow, April 2019 started tox rewrite to improve on it, discoveries:
  - virtual environment creation is slow in virtualenv
  - virtualenv API lacks ability to provide path to site packages
  - interpreter discovery happens both in tox and virtualenv

HOW I JUMPED IN

- tox maintainer since June 2017 (virtualenv dependency)
- virtualenv maintainer since October 2018
- tox is slow, April 2019 started tox rewrite to improve on it, discoveries:
  - virtual environment creation is slow in virtualenv
  - virtualenv API lacks ability to provide path to site packages
  - interpreter discovery happens both in tox and virtualenv
- 🥺 pause on tox rewrite, start virtualenv rewrite 😱

THE PLAN

THE PLAN - IDENTIFY COMPONENTS

- creators
- seeders
- activators
- interpreter discovery

CREATOR

```
$ virtualenv env --without-pip --activators ""
created virtual environment CPython3.8.3.final.0-64 in 46ms
  creator CPython3Posix(dest=/tmp/env, clear=True, global=False)
```

```
env
├── bin
│   ├── python ⇒ /Users/bgabor8/.pyenv/versions/3.8.3/bin/python3.8
│   ├── python3 ⇒ python
│   └── python3.8 ⇒ python
├── lib
│   └── python3.8
│       └── site-packages
│           ├── _virtualenv.pth
│           └── _virtualenv.py
└── pyvenv.cfg
```

SEED MECHANISM

```
$ virtualenv env --activators ""
created virtual environment CPython3.8.3.final.0-64 in 493ms
  creator CPython3Posix(dest=/tmp/env, clear=True, global=False)
  seeder FromAppData(download=False, pip=bundle, setuptools=bundle, wheel=bundle, via=copy, app_data_
    added seed packages: pip==20.1.1, setuptools==47.3.1, wheel==0.34.2
```

```
env
└── lib
    └── python3.8
        └── site-packages
            ├── easy_install.py
            ├── pip
            ├── pip-20.1.1.dist-info
            ├── pip-20.1.1.virtualenv
            ├── pkg_resources
            ├── setuptools
            ├── setuptools-47.3.1.dist-info
            ├── setuptools-47.3.1.virtualenv
            ├── wheel
            ├── wheel-0.34.2.dist-info
            └── wheel-0.34.2.virtualenv
```

ACTIVATORS

```
$ virtualenv env --without-pip
created virtual environment CPython3.8.3.final.0-64 in 124ms
 creator CPython3Posix(dest=/tmp/env, clear=True, global=False)
 activators BashActivator,CShellActivator,FishActivator,PowerShellActivator,PythonActivat
```

```
env
└── bin
    ├── activate
    ├── activate.csh
    ├── activate.fish
    ├── activate.ps1
    ├── activate.xsh
    ├── activate_this.py
    ├── python ⇒ /Users/bgabor8/.pyenv/versions/3.8.3/bin/python3.8
    ├── python3 ⇒ python
    └── python3.8 ⇒ python
```

INTERPRETER DISCOVERY

```
$ virtualenv env --python 2

created virtual environment CPython2.7.18.final.0-64 in 1008ms
  creator CPython2Posix(dest=/tmp/venv, clear=True, global=False)
```

INTERPRETER DISCOVERY

```
$ virtualenv env --python 2

created virtual environment CPython2.7.18.final.0-64 in 1008ms
  creator CPython2Posix(dest=/tmp/venv, clear=True, global=False)
```

```
$ virtualenv env --python PyPy3.6.9-64

created virtual environment PyPy3.6.9.final.0-64 in 373ms
  creator PyPy3Posix(dest=/tmp/venv, clear=True, global=False)
```

WHAT DO WE CONTINUE TO SUPPORT?

- cross version -> if installed on 3.6, can create in 3.8 (if available on host)

WHAT DO WE CONTINUE TO SUPPORT?

- cross version -> if installed on 3.6, can create in 3.8 (if available on host)
- Python 2 - CPython for 2 years, PyPy until upstream PyPy does

WHAT DO WE CONTINUE TO SUPPORT?

- cross version -> if installed on 3.6, can create in 3.8 (if available on host)
- Python 2 - CPython for 2 years, PyPy until upstream PyPy does
- no install use (download and run mode)

WHAT DO WE NEED TO ADD?

- extensible - allow Jython/IronPython/RustPython to be external plugin

WHAT DO WE NEED TO ADD?

- extensible - allow Jython/IronPython/RustPython to be external plugin
- rich API to get data about the created virtual environments

WHAT DO WE NEED TO ADD?

- extensible - allow Jython/IronPython/RustPython to be external plugin
- rich API to get data about the created virtual environments
- unify virtual environment mechanism - adopt venv style - PEP-405

THE CHOPPING BOARD

- relocatable virtual environments

THE CHOPPING BOARD

- relocatable virtual environments, provide alternative path

THE CHOPPING BOARD

- relocatable virtual environments, provide alternative path
- drop deprecated flags, e.g., --no-site-packages

THE CHOPPING BOARD

- relocatable virtual environments, provide alternative path
- drop deprecated flags, e.g., --no-site-packages

THE CHOPPING BOARD

- relocatable virtual environments, provide alternative path
- drop deprecated flags, e.g., --no-site-packages

- entire project in one file

THE CHOPPING BOARD

- relocatable virtual environments, provide alternative path
- drop deprecated flags, e.g., --no-site-packages

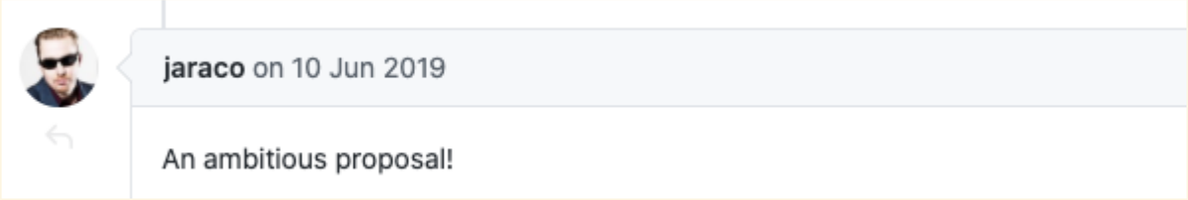
- entire project in one file
- stdlib only dependencies (and no vendored packages)

LET PEOPLE KNOW

- RFC posted on GitHub issue - June 10th 2019

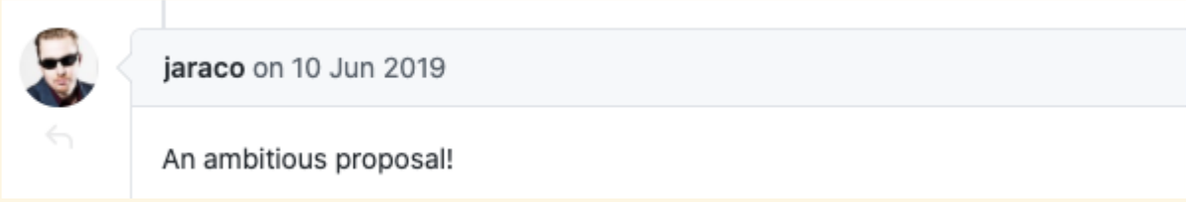
LET PEOPLE KNOW

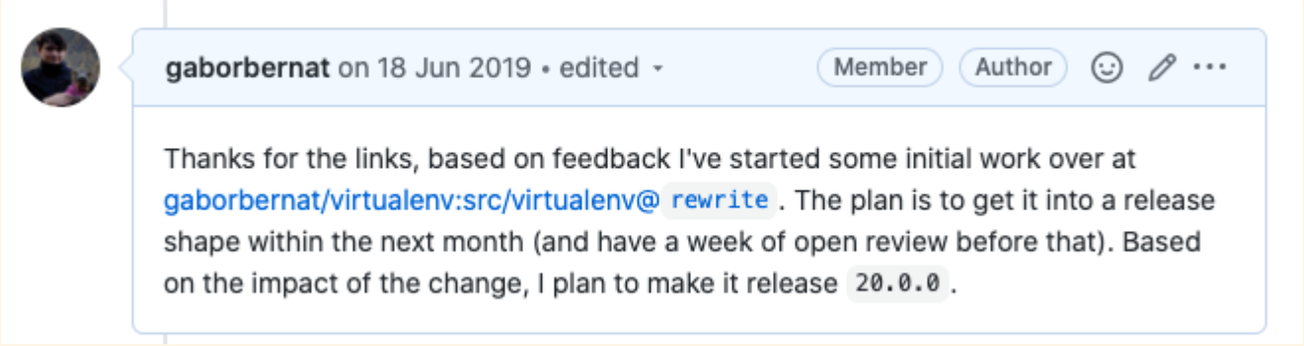
- RFC posted on GitHub issue - June 10th 2019



LET PEOPLE KNOW

- RFC posted on GitHub issue - June 10th 2019





UNDER CONSTRUCTION - UNTIL JANUARY 2020

DO RESTRICTED/BETA RELEASES - 2020

- January 6th - first review request

DO RESTRICTED/BETA RELEASES - 2020

- January 6th - first review request
- January 28th - first beta

DO RESTRICTED/BETA RELEASES - 2020

- January 6th - first review request
- January 28th - first beta
- February 4th - second beta

DO RESTRICTED/BETA RELEASES - 2020

- January 6th - first review request
- January 28th - first beta
- February 4th - second beta
- February 10th - first public release

THE CREW - FIRST RELEASE

- Siddhant Kumar - migrate activation scripts

THE CREW - FIRST RELEASE

- Siddhant Kumar - migrate activation scripts
- Bernát Gábor - everything else

THE CREW - COMMUNITY ADOPTION +27 IN 3 MONTHS

```
12907 author Bernát Gábor
  900 author Siddhant Kumar
  286 author Seungmin Ryu
  169 author Sorin Sbarnea
  113 author Julien Danjou
   52 author David Tucker
   43 author Anthony Sottile
   14 author PRAJWAL M
   11 author spetafree
   11 author Vincent Philippon
   11 author Pradyun Gedam
   11 author Nicola Soranzo
    9 author Claudio Jolowicz
    8 author Ian Wienand
    5 author Lumír 'Frenzy' Balhar
    3 author Sviatoslav Sydorenko
    3 author Michał Górny
    3 author Johannes Altmanninger
    2 author Xavier Fernandez
    2 author Jannis Leidel
    1 author txp314
    1 author shaido987
    1 author mondeja
    1 author Thomas Grainger
    1 author Theodor Dimitriou
    1 author Shantanu
    1 author Rowdy Howell
    1 author Philippe Ombredanne
    1 author Matthew StClair
    1 author João Vale
```

WIDELY PUBLICIZE THE RE-RELEASE

- forum post Python Discourse - Packaging section - explain:

WIDELY PUBLICIZE THE RE-RELEASE

- forum post Python Discourse - Packaging section - explain:
  - why the rewrite
  - what are the new features

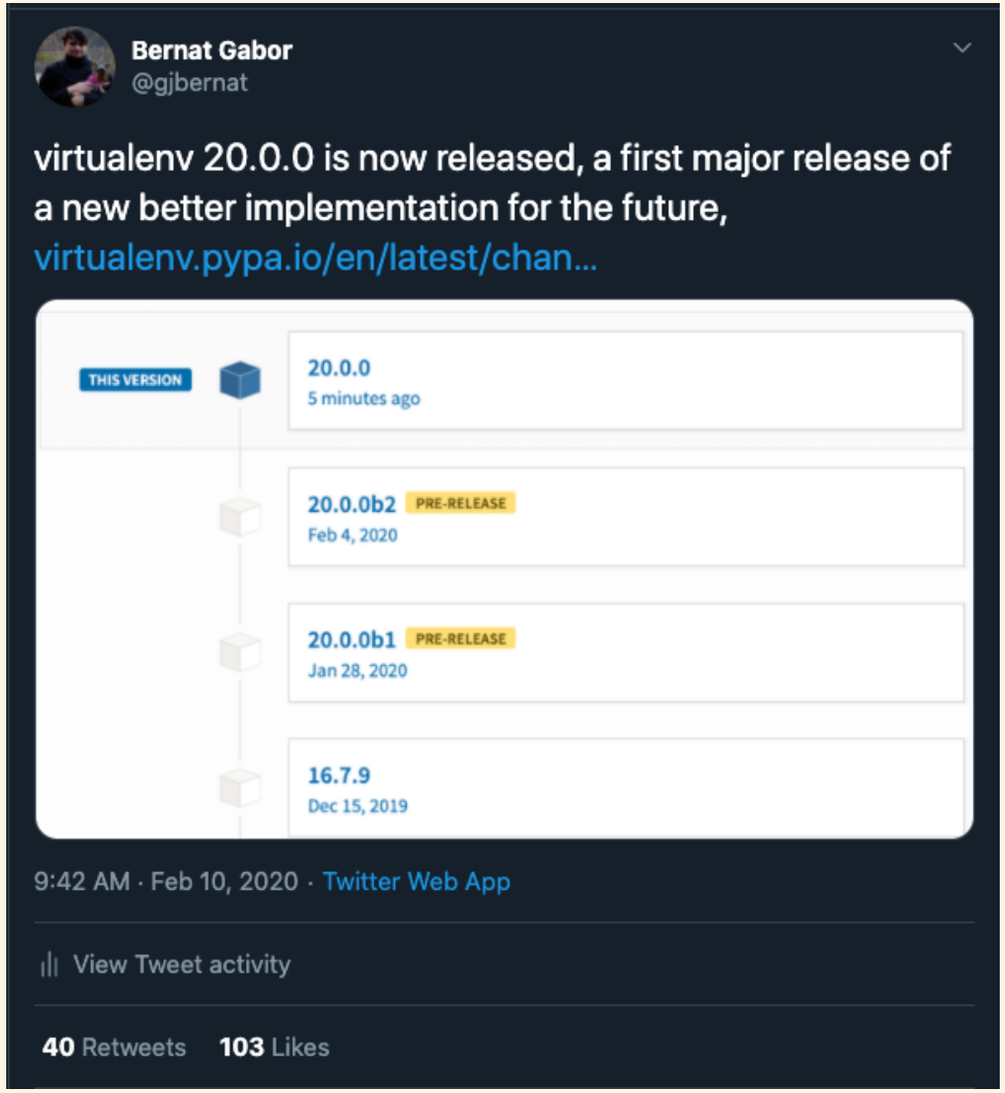
WIDELY PUBLICIZE THE RE-RELEASE

- forum post Python Discourse - Packaging section - explain:
  - why the rewrite
  - what are the new features
- distutils mail list

WIDELY PUBLICIZE THE RE-RELEASE

- forum post Python Discourse - Packaging section - explain:
  - why the rewrite
  - what are the new features
- distutils mail list
- PyPi - version 20.0.0

TWITTER ANNOUNCEMENT - 15K IMPRESSIONS



TECHNICAL GOTCHAS

CONSIDER PEOPLE STUCK ON OLD VERSION

CONSIDER PEOPLE STUCK ON OLD VERSION

- keep code on a legacy branch

CONSIDER PEOPLE STUCK ON OLD VERSION

- keep code on a legacy branch
- add a legacy docs tag on readthedocs

CONSIDER PEOPLE STUCK ON OLD VERSION

- keep code on a legacy branch
- add a legacy docs tag on readthedocs
- new changelog at the end points to legacy changelog

CONSIDER PEOPLE STUCK ON OLD VERSION

- keep code on a legacy branch
- add a legacy docs tag on readthedocs
- new changelog at the end points to legacy changelog
- keep releasing community contributions - 16.7.10 - Feb 24, 2020 (February 10th first new)

CPYTHON IS A VERY DIVERSE SNAKE PIT

sysconfig	value
scripts	`{base}/bin`
data	`{base}`
include	`{base}/include/python{py_version_short}`
stdlib	`{base}/lib64/python{py_version_short}`
platstdlib	`{platbase}/lib64/python{py_version_short}`
purelib	`{base}/lib/python{py_version_short}/site-packages`
platlib	`{platbase}/lib64/python{py_version_short}/site-packages`

LINUX DISTRIBUTIONS - FEDORA

- On Fedora, platlib is under `lib64`, while purelib is `lib`

LINUX DISTRIBUTIONS - DEBIAN

- Debian does not install `venv` by default, `python3-venv`

LINUX DISTRIBUTIONS - DEBIAN

- Debian does not install `venv` by default, `python3-venv`
- pip uses `distutils` paths to install, separate from `sysconfig`

LINUX DISTRIBUTIONS - DEBIAN

- Debian does not install `venv` by default, `python3-venv`
- pip uses `distutils` paths to install, separate from `sysconfig`
- they patch distutils paths to separate `apt` and `pip` installs

LINUX DISTRIBUTIONS - DEBIAN

- Debian does not install `venv` by default, `python3-venv`
- pip uses `distutils` paths to install, separate from `sysconfig`
- they patch distutils paths to separate `apt` and `pip` installs
- use distutils paths for virtual environment, sysconfig for host Python

LINUX DISTRIBUTIONS - CENTOS

- ships with `pip 9`

LINUX DISTRIBUTIONS - CENTOS

- ships with `pip 9` - does not understand `python-requires` on wheels

LINUX DISTRIBUTIONS - CENTOS

- ships with `pip 9` - does not understand `python-requires` on wheels
- had to vendor `six.ensure_path/ensure_text`, relax `six` requirement

LINUX DISTRIBUTIONS - CENTOS

- ships with `pip 9` - does not understand `python-requires` on wheels
- had to vendor `six.ensure_path/ensure_text`, relax `six` requirement
- Better error message in `setup.py`

MACOS - SO MANY PYTHONS

- install from `python.org/pyenv`

MACOS - SO MANY PYTHONS

- install from `python.org/pyenv`
- part of the OS `/usr/bin/python` -> `/System/Library/Frameworks/Python.framework/Versions/2.7`

MACOS - SO MANY PYTHONS

- install from `python.org/pyenv`
- part of the OS `/usr/bin/python` -> `/System/Library/Frameworks/Python.framework/Versions/2.7`
- brew installation via `python@2` and `python@3`

MACOS - SO MANY PYTHONS

- install from `python.org/pyenv`
- part of the OS `/usr/bin/python` -> `/System/Library/Frameworks/Python.framework/Versions/2.7`
- brew installation via `python@2` and `python@3`
- XCode - Developer Tools v2 `/Library/Frameworks/Python.framework/`

MACOS - SO MANY PYTHONS

- install from `python.org/pyenv`
- part of the OS `/usr/bin/python` -> `/System/Library/Frameworks/Python.framework/Versions/2.7`
- brew installation via `python@2` and `python@3`
- XCode - Developer Tools v2 `/Library/Frameworks/Python.framework/`
- XCode - Developer Tools v3 `/Library/Frameworks/Python3.framework/` a

MACOS - SO MANY PYTHONS

- install from `python.org/pyenv`
- part of the OS `/usr/bin/python ->` `/System/Library/Frameworks/Python.framework/Versions/2.7`
- brew installation via `python@2` and `python@3`
- XCode - Developer Tools v2 `/Library/Frameworks/Python.framework/`
- XCode - Developer Tools v3 `/Library/Frameworks/Python3.framework/` a
- Apple shipped Python are hard coded to static path (Mach-o)

MACOS - FIXING DISTUTILS

- pip uses distutils to determine where to install

MACOS - FIXING DISTUTILS

- pip uses distutils to determine where to install
- distutils can be configured via files

Type of file	Location and filename
system	*prefix*/lib/python*ver*/distutils/distutils.cfg
personal	$HOME/.pydistutils.cfg
local	setup.cfg

MACOS - FIXING DISTUTILS

- pip uses distutils to determine where to install
- distutils can be configured via files

Type of file	Location and filename
system	*prefix*`/lib/python`*ver*`/distutils/distutils.cfg`
personal	`$HOME/.pydistutils.cfg`
local	`setup.cfg`

- can set `prefix` and `install_{purelib,platlib,headers,scripts,data}`

MACOS - FIXING DISTUTILS

- pip uses distutils to determine where to install
- distutils can be configured via files

Type of file	Location and filename
system	*prefix*`/lib/python`*ver*`/distutils/distutils.cfg`
personal	`$HOME/.pydistutils.cfg`
local	`setup.cfg`

- can set `prefix` and `install_{purelib,platlib,headers,scripts,data}`
- closed 5 times, be tenacious about fixing bugs

WINDOWS STORE PYTHON

- delegate creation to env (needs implementation for inline run)

WINDOWS STORE PYTHON

- delegate creation to env (needs implementation for inline run)
- surprising behaviour, cannot check if exists (no read), but can run

WINDOWS STORE PYTHON

- delegate creation to env (needs implementation for inline run)
- surprising behaviour, cannot check if exists (no read), but can run

```
PS C:\Users\traveler\git\virtualenv> python -c 'import sys; from pathlib import Path; import os;
    print(sys.executable); \
    print(os.path.exists(sys.executable)); \
    print(Path(sys.executable).exists())'

C:\Users\traveler\AppData\Local\Microsoft\WindowsApps\PythonSoftwareFoundation.Python.3.7_qbz5n2k
False
Traceback (most recent call last):
  File "string", line 1, in module
  File "C:\Program Files\WindowsApps\PythonSoftwareFoundation.Python.3.7_3.7.1776.0_x64__qbz5n2kf
    self.stat()
  File "C:\Program Files\WindowsApps\PythonSoftwareFoundation.Python.3.7_3.7.1776.0_x64__qbz5n2kf
    return self._accessor.stat(self)
OSError: [WinError 1920] The file cannot be accessed by the system: 'C:\\Users\\traveler\\AppData
```

PYPY MATURING - QUICK TO ANSWER/FIX

- pypy3.6-7.3.0 venv fails with --copies on Linux
- pypy on Windows uses non Windows install path
- venv non-ASCII support - Windows

EXPECT THE UNEXPECTED - PYC ONLY MODE

- Python 2 uses the host os.py as landmark file to detect stdlib

```
env
├── bin
│   ├── python
│   ├── python2 ⇒ python
│   └── python2.7 ⇒ python
├── include
│   └── python2.7 ⇒ /2.7.18/include/python2.7
├── lib
│   └── python2.7
│       ├── config
│       │   └── Makefile ⇒ /2.7.18/lib/python2.7/config/Makefile
│       ├── lib-dynload ⇒ /2.7.18/lib/python2.7/lib-dynload
│       ├── os.py ⇒ /2.7.18/lib/python2.7/os.py
│       ├── os.pyc ⇒ /2.7.18/lib/python2.7/os.pyc
│       ├── site-packages
│       └── site.py
└── pyvenv.cfg
```

EXPECT THE UNEXPECTED - PYC ONLY MODE

- Python 2 uses the host os.py as landmark file to detect stdlib

```
env
├── bin
│   ├── python
│   ├── python2 ⇒ python
│   └── python2.7 ⇒ python
├── include
│   └── python2.7 ⇒ /2.7.18/include/python2.7
├── lib
│   └── python2.7
│       ├── config
│       │   └── Makefile ⇒ /2.7.18/lib/python2.7/config/Makefile
│       ├── lib-dynload ⇒ /2.7.18/lib/python2.7/lib-dynload
│       ├── os.py ⇒ /2.7.18/lib/python2.7/os.py
│       ├── os.pyc ⇒ /2.7.18/lib/python2.7/os.pyc
│       ├── site-packages
│       └── site.py
└── pyvenv.cfg
```

- Python can still function with only pyc files

EXPECT THE UNEXPECTED - PYC ONLY MODE

- Python 2 uses the host os.py as landmark file to detect stdlib

```
□ env
├── □ bin
│   ├── □ python
│   ├── □ python2 ⇒ python
│   └── □ python2.7 ⇒ python
├── □ include
│   └── □ python2.7 ⇒ /2.7.18/include/python2.7
├── □ lib
│   └── □ python2.7
│       ├── □ config
│       │   └── □ Makefile ⇒ /2.7.18/lib/python2.7/config/Makefile
│       ├── □ lib-dynload ⇒ /2.7.18/lib/python2.7/lib-dynload
│       ├── □ os.py ⇒ /2.7.18/lib/python2.7/os.py
│       ├── □ os.pyc ⇒ /2.7.18/lib/python2.7/os.pyc
│       ├── □ site-packages
│       └── □ site.py
└── □ pyvenv.cfg
```

- Python can still function with only pyc files
- benefits of doing this: obfuscate source code or maximize storage

EXPECT THE UNEXPECTED - PYC ONLY MODE

- Python 2 uses the host os.py as landmark file to detect stdlib

```
env
├── bin
│   ├── python
│   ├── python2 ⇒ python
│   └── python2.7 ⇒ python
├── include
│   └── python2.7 ⇒ /2.7.18/include/python2.7
├── lib
│   └── python2.7
│       ├── config
│       │   └── Makefile ⇒ /2.7.18/lib/python2.7/config/Makefile
│       ├── lib-dynload ⇒ /2.7.18/lib/python2.7/lib-dynload
│       ├── os.py ⇒ /2.7.18/lib/python2.7/os.py
│       ├── os.pyc ⇒ /2.7.18/lib/python2.7/os.pyc
│       ├── site-packages
│       └── site.py
└── pyvenv.cfg
```

- Python can still function with only pyc files
- benefits of doing this: obfuscate source code or maximize storage
- solution: require only one of the py/pyc files

EXPECT THE UNEXPECTED - PYC ONLY MODE

- Python 2 uses the host os.py as landmark file to detect stdlib

```
env
├── bin
│   ├── python
│   ├── python2 ⇒ python
│   └── python2.7 ⇒ python
├── include
│   └── python2.7 ⇒ /2.7.18/include/python2.7
├── lib
│   └── python2.7
│       ├── config
│       │   └── Makefile ⇒ /2.7.18/lib/python2.7/config/Makefile
│       ├── lib-dynload ⇒ /2.7.18/lib/python2.7/lib-dynload
│
│       ├── os.pyc ⇒ /2.7.18/lib/python2.7/os.pyc
│       ├── site-packages
│       └── site.py
└── pyvenv.cfg
```

- Python operates can work with pyc only
- benefits of doing this: obfuscate source code or maximize storage
- solution: require only one of the py/pyc files

SEEDERS - PIP

SEEDERS - PIP

- existing seed mechanism (venv does similar) - 2 seconds + pip

```
export PYTHONPATH=pip-20.1.1-py2.py3-none-any.whl
env/bin/python -m pip install pip-20.1.1-py2.py3-none-any.whl
```

SEEDERS - PIP

- existing seed mechanism (venv does similar) - 2 seconds + pip

```
export PYTHONPATH=pip-20.1.1-py2.py3-none-any.whl
env/bin/python -m pip install pip-20.1.1-py2.py3-none-any.whl
```

- pip is a general purpose install tool, some of the (avoidable) overheads:

SEEDERS - PIP

- existing seed mechanism (venv does similar) - 2 seconds + pip

```
export PYTHONPATH=pip-20.1.1-py2.py3-none-any.whl
env/bin/python -m pip install pip-20.1.1-py2.py3-none-any.whl
```

- pip is a general purpose install tool, some of the (avoidable) overheads:
  - wheel validation

SEEDERS - PIP

- existing seed mechanism (venv does similar) - 2 seconds + pip

```
export PYTHONPATH=pip-20.1.1-py2.py3-none-any.whl
env/bin/python -m pip install pip-20.1.1-py2.py3-none-any.whl
```

- pip is a general purpose install tool, some of the (avoidable) overheads:
  - wheel validation
  - startup time

SEEDERS - PIP

- existing seed mechanism (venv does similar) - 2 seconds + pip

```
export PYTHONPATH=pip-20.1.1-py2.py3-none-any.whl
env/bin/python -m pip install pip-20.1.1-py2.py3-none-any.whl
```

- pip is a general purpose install tool, some of the (avoidable) overheads:
  - wheel validation
  - startup time
  - self upgrade check

SEEDERS - PIP

- existing seed mechanism (venv does similar) - 2 seconds + pip

```
export PYTHONPATH=pip-20.1.1-py2.py3-none-any.whl
env/bin/python -m pip install pip-20.1.1-py2.py3-none-any.whl
```

- pip is a general purpose install tool, some of the (avoidable) overheads:
  - wheel validation
  - startup time
  - self upgrade check
  - extracting files from a zip usually triggers the antivirus

SEEDERS - APP-DATA - CACHE

cache what we can cache to avoid performing the same operation

SEEDERS - APP-DATA - CACHE

cache what we can cache to avoid performing the same operation

- `appdirs` - for platform specific application data location

SEEDERS - APP-DATA - CACHE

cache what we can cache to avoid performing the same operation

- `appdirs` - for platform specific application data location
- create base image into app data folder:
  - validate wheel
  - extract wheel
  - generate pyc files, `INSTALLER`
  - fix `RECORDS` - add metadata + console script files

SEEDERS - APP-DATA - CACHE

cache what we can cache to avoid performing the same operation

- `appdirs` - for platform specific application data location
- create base image into app data folder:
  - validate wheel
  - extract wheel
  - generate pyc files, `INSTALLER`
  - fix `RECORDS` - add metadata + console script files
- install operation
  - generate console scripts (`pip`, `pip3`, `pip3.8`)
  - copy/symlink from app data to `purelib`

SEED PACKAGES - AUTO UPDATE?

- should we
  - install latest pip/setuptools/wheel
  - install embedded pip/setuptools/wheel

SEED PACKAGES - AUTO UPDATE?

- should we
  - install latest pip/setuptools/wheel ⇒ **convenience and ecosystem**
  - install embedded pip/setuptools/wheel ⇒ **speed and stability**

SEED PACKAGES - AUTO UPDATE?

- should we
  - install latest pip/setuptools/wheel ⇒ **convenience and ecosystem**
  - install embedded pip/setuptools/wheel ⇒ **speed and stability**
- middle-ground update every two weeks, use if at least 28 days old

SEED PACKAGES - RESILIENCE

new approaches/solutions will expose new failure types

SEED PACKAGES - RESILIENCE

new approaches/solutions will expose new failure types

- handle if app data is in read only mode

SEED PACKAGES - RESILIENCE

new approaches/solutions will expose new failure types

- handle if app data is in read only mode
- always default to safer methods - copy over symlink

BE FRIENDLY AND PLAN FOR REDISTRIBUTION

Fedora, Debian, etc. often apply patches on top of our source

BE FRIENDLY AND PLAN FOR REDISTRIBUTION

- Fedora, Debian, etc. often apply patches on top of our source
- understand why

BE FRIENDLY AND PLAN FOR REDISTRIBUTION

Fedora, Debian, etc. often apply patches on top of our source

- understand why
- make it easy and straight forward what and where to patch

BE FRIENDLY AND PLAN FOR REDISTRIBUTION

Fedora, Debian, etc. often apply patches on top of our source

- understand why
- make it easy and straight forward what and where to patch
- document what and where should be patched

DOWNLOAD ONLY (NO INSTALL) MODE

- with old version, you could just download repository and then run `virtualenv.py`

DOWNLOAD ONLY (NO INSTALL) MODE

- with old version, you could just download repository and then run `virtualenv.py`
- follow prior art with pip

```
curl https://bootstrap.pypa.io/get-pip.py -o get-pip.py
python get-pip.py
```

DOWNLOAD ONLY (NO INSTALL) MODE

- with old version, you could just download repository and then run `virtualenv.py`
- follow prior art with pip

```
curl https://bootstrap.pypa.io/get-pip.py -o get-pip.py
python get-pip.py
```

- take advantage of newer Python features:

DOWNLOAD ONLY (NO INSTALL) MODE

- with old version, you could just download repository and then run `virtualenv.py`
- follow prior art with pip

```
curl https://bootstrap.pypa.io/get-pip.py -o get-pip.py
python get-pip.py
```

- take advantage of newer Python features:
  create a zipapp package requiring no install

```
curl https://bootstrap.pypa.io/virtualenv.pyz -o virtualenv.pyz
python3.8 virtualenv.pyz env
```

DOWNLOAD ONLY (NO INSTALL) MODE

- with old version, you could just download repository and then run `virtualenv.py`
- follow prior art with pip

```
curl https://bootstrap.pypa.io/get-pip.py -o get-pip.py
python get-pip.py
```

- take advantage of newer Python features:
  create a zipapp package requiring no install

```
curl https://bootstrap.pypa.io/virtualenv.pyz -o virtualenv.pyz
python3.8 virtualenv.pyz env
```

- Make it easy to update, https://bootstrap.pypa.io/virtualenv.pyz
  automatically feeds from https://github.com/pypa/get-virtualenv

ZIPAPP CONTENT STRUCTURE

```
.
├── __main__.py
├── virtualenv
│   └  ...
├── virtualenv-20.0.25.dist-info
│   ├── entry_points.txt
│   ├── LICENSE
│   ├── METADATA
│   ├── RECORD
│   ├── top_level.txt
│   ├── WHEEL
│   └── zip-safe
├── __virtualenv__
│   ├── appdirs-1.4.4-py2.py3-none-any
│   ├── configparser-4.0.2-py2.py3-none-any
│   ├── contextlib2-0.6.0.post1-py2.py3-none-any
│   ├── distlib-0.3.0-py3-none-any
│   ├── filelock-3.0.12-py3-none-any
│   ├── importlib_metadata-1.1.3-py2.py3-none-any
│   ├── importlib_metadata-1.6.1-py2.py3-none-any
│   ├── importlib_resources-1.0.2-py2.py3-none-any
│   ├── importlib_resources-2.0.1-py2.py3-none-any
│   ├── pathlib2-2.3.5-py2.py3-none-any
│   ├── scandir-1.10.0-cp38-cp38-macosx_10_14_x86_64
│   ├── six-1.15.0-py2.py3-none-any
│   ├── typing-3.7.4.1-py2-none-any
│   ├── typing-3.7.4.1-py3-none-any
│   ├── zipp-1.2.0-py2.py3-none-any
│   └── zipp-3.1.0-py3-none-any
├── distributions.json
└── modules.json
```

ZIPAPP WITH DYNAMIC DEPENDENCY

- some dependencies are OS specific
- some dependencies are Python version specific

```
install_requires =
   appdirs>=1.4.3,<2
   distlib>=0.3.0,<1
   filelock>=3.0.0,<4
   six>=1.9.0,<2   # keep it >=1.9.0 as it may cause problems on LTS platforms
   importlib-metadata>=0.12,<2;python_version<"3.8"
   importlib-resources>=1.0;python_version<"3.7"
   pathlib2>=2.3.3,<3;python_version < '3.4' and sys.platform != 'win32'
```

ZIPAPP WITH DYNAMIC DEPENDENCIES

	3.9	3.8	3.7	3.6	3.5	3.4	2.7
appdirs	1.4.4	1.4.4	1.4.4	1.4.4	1.4.4	1.4.4	1.4.4
distlib	0.3.1	0.3.1	0.3.1	0.3.1	0.3.1	0.3.1	0.3.1
filelock	3.0.12	3.0.12	3.0.12	3.0.12	3.0.12	3.0.12	3.0.12
six	1.15.0	1.15.0	1.15.0	1.15.0	1.15.0	1.15.0	1.15.0
importlib-metadata			1.7.0	1.7.0	1.7.0	1.1.3	1.1.3
importlib-resources				2.0.1	2.0.1	1.0.2	1.0.2
pathlib2						2.3.5	2.3.5

ZIPAPP CONTENT STRUCTURE

```
.
├── __main__.py
├── virtualenv
│   └  ...
├── virtualenv-20.0.25.dist-info
│   ├── entry_points.txt
│   ├── LICENSE
│   ├── METADATA
│   ├── RECORD
│   ├── top_level.txt
│   ├── WHEEL
│   └── zip-safe
├── __virtualenv__
│   ├── appdirs-1.4.4-py2.py3-none-any
│   ├── configparser-4.0.2-py2.py3-none-any
│   ├── contextlib2-0.6.0.post1-py2.py3-none-any
│   ├── distlib-0.3.0-py3-none-any
│   ├── filelock-3.0.12-py3-none-any
│   ├── importlib_metadata-1.1.3-py2.py3-none-any
│   ├── importlib_metadata-1.6.1-py2.py3-none-any
│   ├── importlib_resources-1.0.2-py2.py3-none-any
│   ├── importlib_resources-2.0.1-py2.py3-none-any
│   ├── pathlib2-2.3.5-py2.py3-none-any
│   ├── scandir-1.10.0-cp38-cp38-macosx_10_14_x86_64
│   ├── six-1.15.0-py2.py3-none-any
│   ├── typing-3.7.4.1-py2-none-any
│   ├── typing-3.7.4.1-py3-none-any
│   ├── zipp-1.2.0-py2.py3-none-any
│   └── zipp-3.1.0-py3-none-any
├── distributions.json
└── modules.json
```

USE IMPORT HOOKS WITHIN ZIPAPP

- Python 2: PEP-302 - New import hooks
- Python 3: PEP-451 - A ModuleSpec Type for the Import System

ZIPAPP WITH DYNAMIC DEPENDENCIES

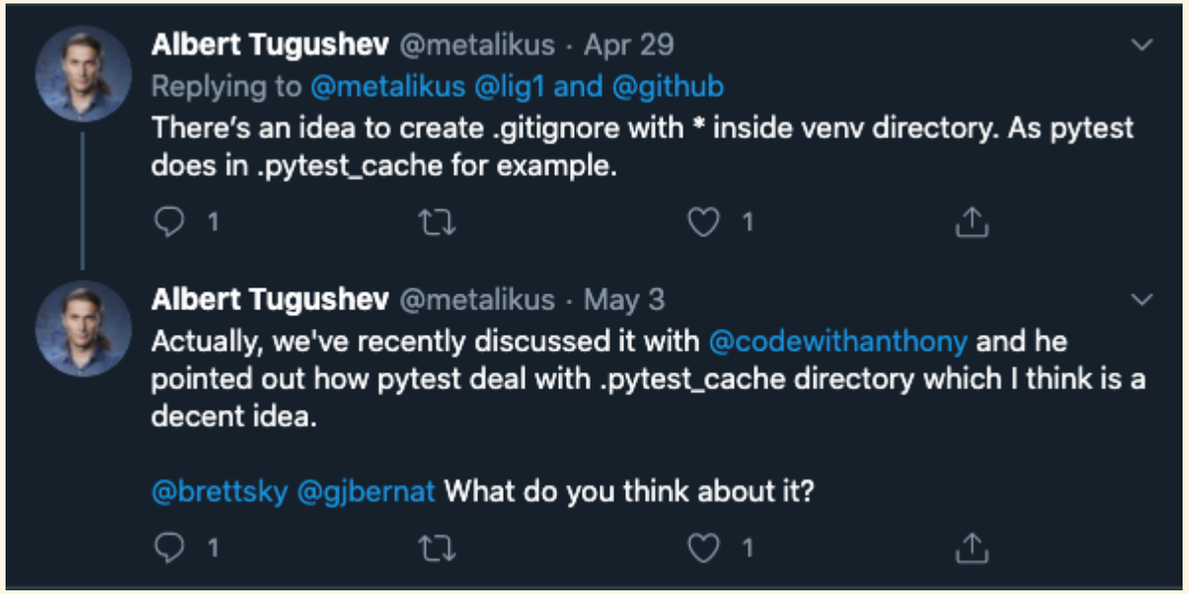
	3.9	3.8	3.7	3.6	3.5	3.4	2.7
appdirs	1.4.4	1.4.4	1.4.4	1.4.4	1.4.4	1.4.4	1.4.4
distlib	0.3.1	0.3.1	0.3.1	0.3.1	0.3.1	0.3.1	0.3.1
filelock	3.0.12	3.0.12	3.0.12	3.0.12	3.0.12	3.0.12	3.0.12
six	1.15.0	1.15.0	1.15.0	1.15.0	1.15.0	1.15.0	1.15.0
importlib-metadata			1.7.0	1.7.0	1.7.0	1.1.3	1.1.3
importlib-resources				2.0.1	2.0.1	1.0.2	1.0.2
pathlib2						2.3.5	2.3.5

THE CHERRY ON TOP

- Fix upstream bugs before upstream
  - strip `__PYVENV_LAUNCHER__` on MacOs

THE CHERRY ON TOP

- Fix upstream bugs before upstream
  - strip `__PYVENV_LAUNCHER__` on MacOs
- Do not version track virtual environments - add `.gitignore` May 20th

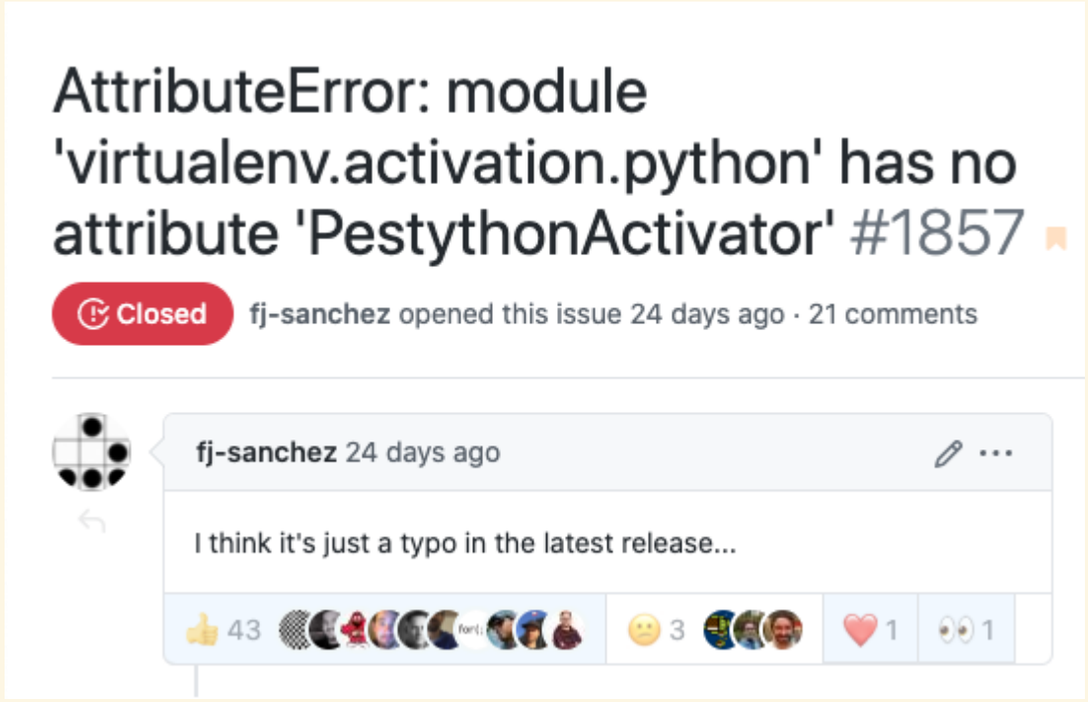


BREAKING THE WORLD

- since 2018 October - 41 successful releases

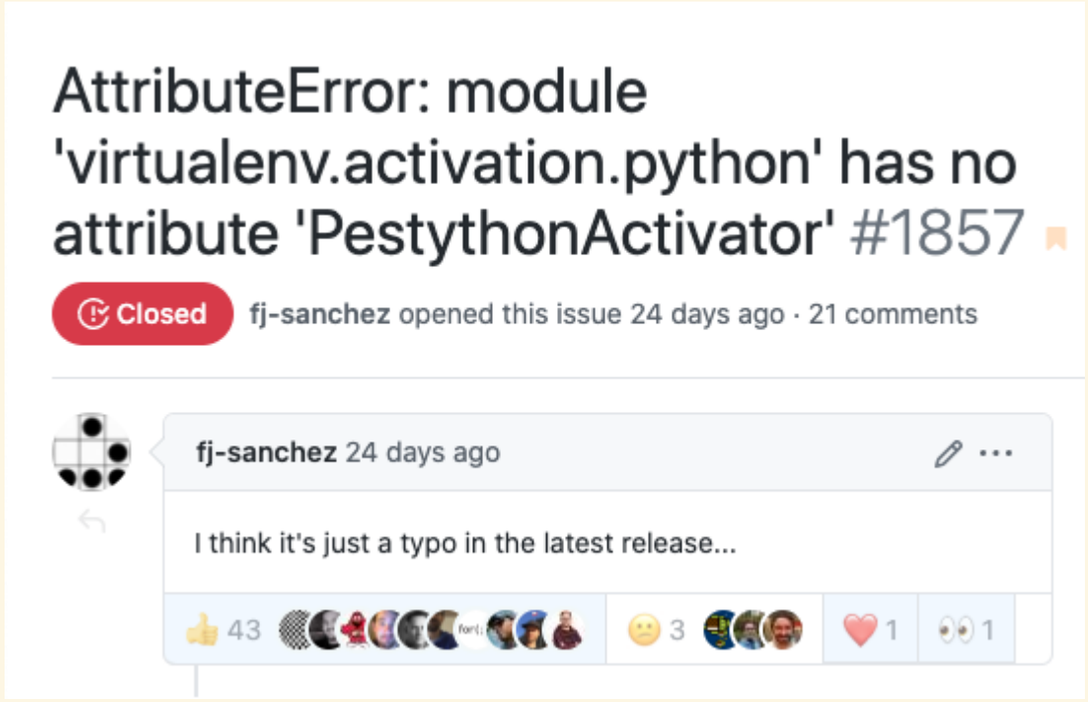
BREAKING THE WORLD

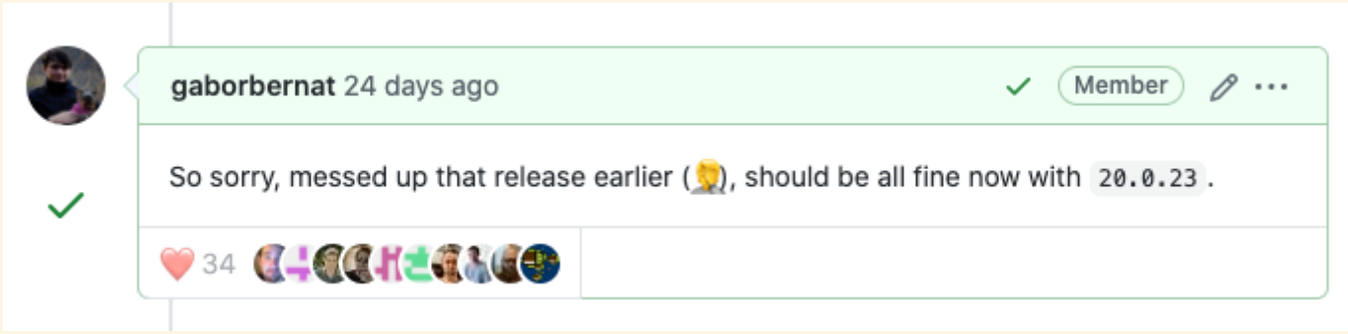
- since 2018 October - 41 successful releases
- Friday evening release - typo in `setup.cfg`



BREAKING THE WORLD

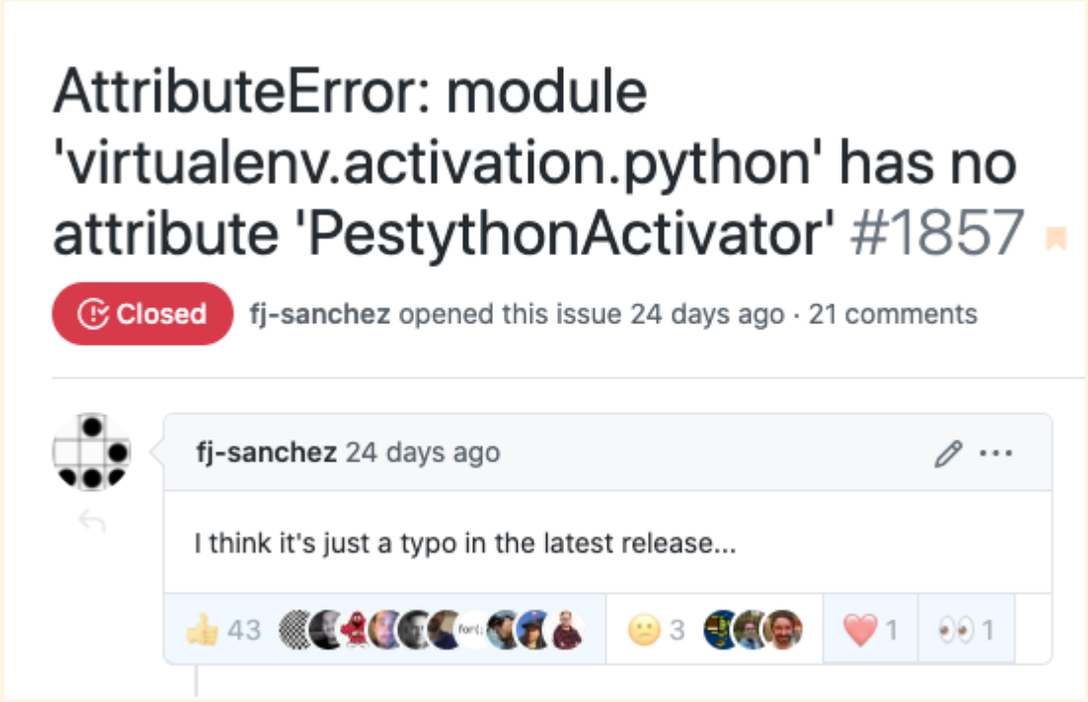
- since 2018 October - 41 successful releases
- Friday evening release - typo in `setup.cfg`

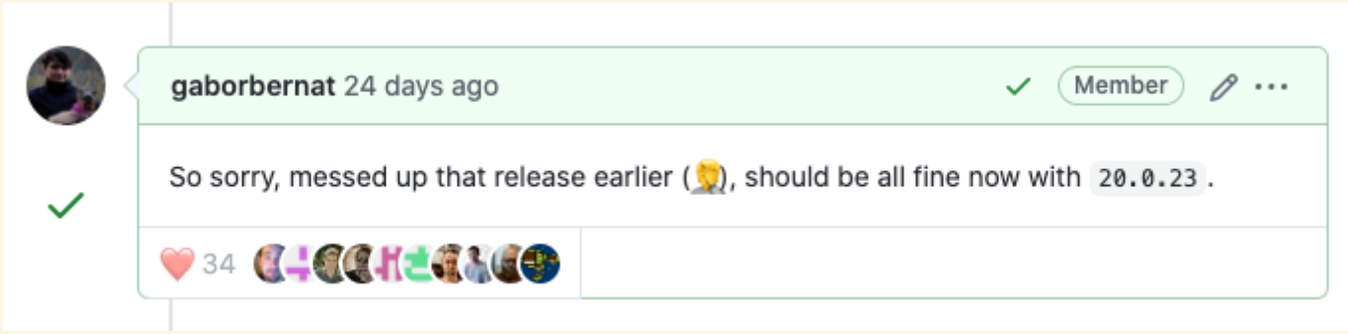




BREAKING THE WORLD

- since 2018 October - 41 successful releases
- Friday evening release - typo in `setup.cfg`





- Always be available in the next 1-2 hours post release

CONCLUSION

LESSONS LEARNED

LESSONS LEARNED

- Have a well defined delivery plan

LESSONS LEARNED

- Have a well defined delivery plan
- Test a lot, automate via CIs

LESSONS LEARNED

- Have a well defined delivery plan
- Test a lot, automate via CIs
- Prepare for a lot of bug fixes (on short turnaround - 27 releases in 5 months)

LESSONS LEARNED

- Have a well defined delivery plan
- Test a lot, automate via CIs
- Prepare for a lot of bug fixes (on short turnaround - 27 releases in 5 months)
- Always be available post release

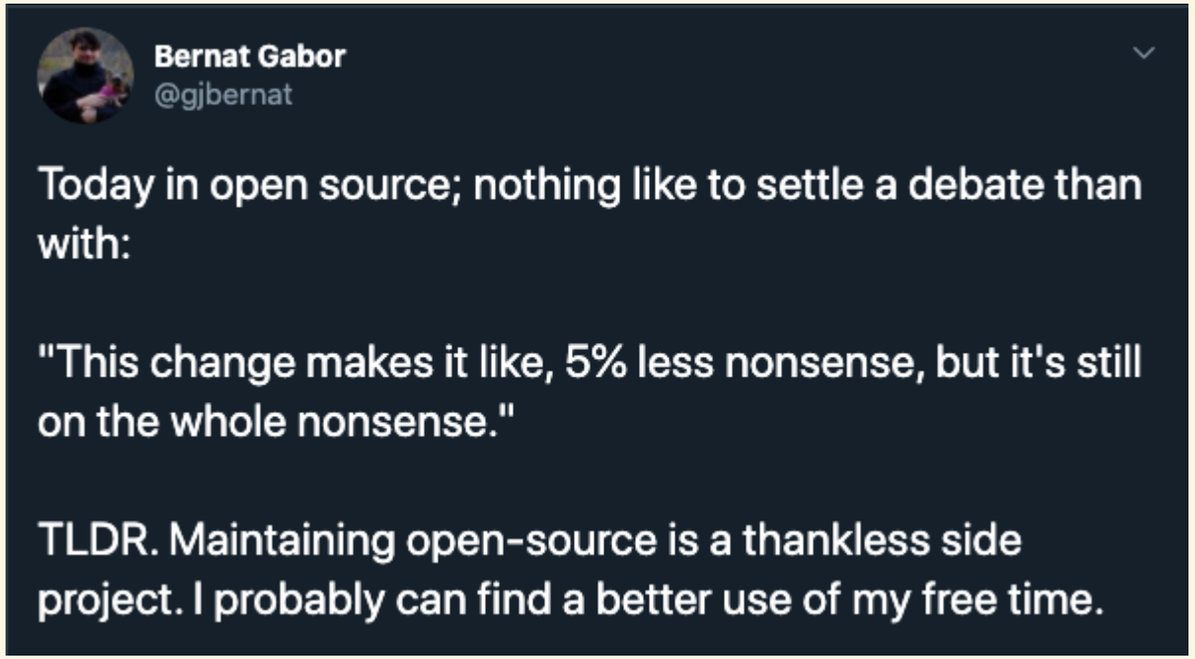
LESSONS LEARNED

- Have a well defined delivery plan
- Test a lot, automate via CIs
- Prepare for a lot of bug fixes (on short turnaround - 27 releases in 5 months)
- Always be available post release
- Be ready to periodically re-evaluate things from the ground up

LESSONS LEARNED

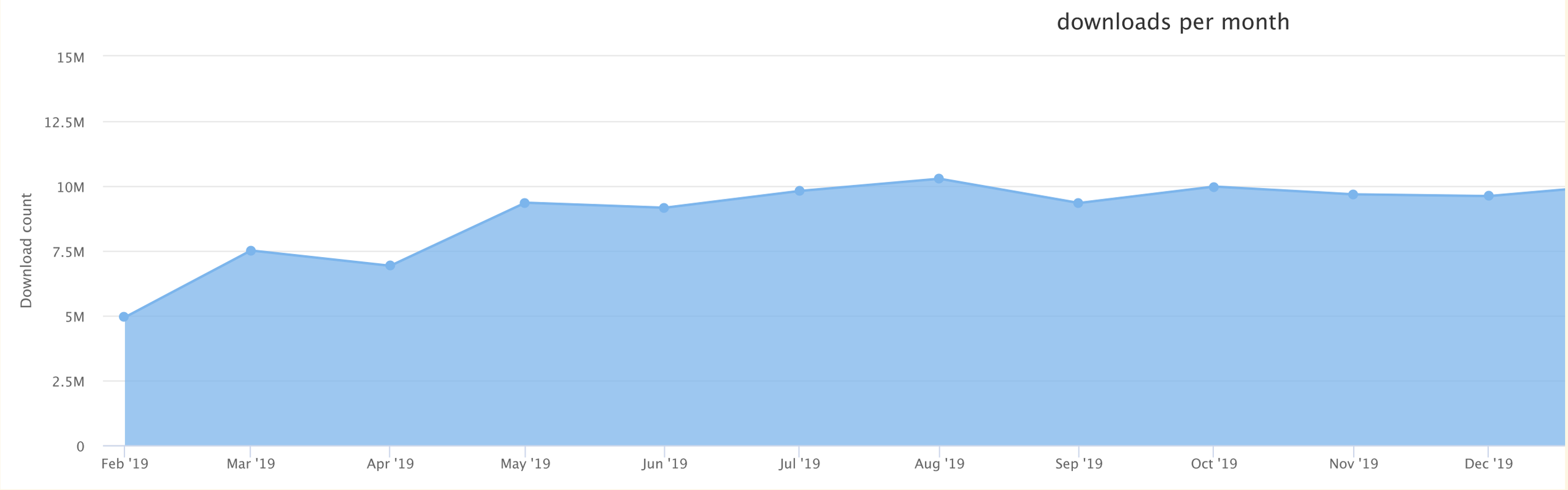
- Have a well defined delivery plan
- Test a lot, automate via CIs
- Prepare for a lot of bug fixes (on short turnaround - 27 releases in 5 months)
- Always be available post release
- Be ready to periodically re-evaluate things from the ground up
- In general (but especially in opensource), be nice

THE STRAW THAT ALMOST BROKE THE CAMEL'S BACK



Be nice but have/use code of conducts when needed

NOW, HOW POPULAR IS VIRTUALENV?



THANK YOU

AND

QUESTIONS