REWWRITING AND RE-RELEASING

VIRTUALENV

LESSONS FROM THE TRENCHES 😊

by Bernát Gábor / @gjbernat / Bloomberg

slides @ https://bit.ly/virtualenv-eupy20
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 if it were a separate Python installation (from the system one)
WHAT IS VIRTUALENV?

- a tool to create Python virtual environments
- Python executables that behave as if 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
$ 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
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); print(os.path.dirname(sys.executable));'

/System/Library/Frameworks/Python.framework/Versions/2.7/Resources/Python.app/Contents/MacOS/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

$ 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); print(os)'
/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)'
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

  ```bash
  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.

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

<table>
<thead>
<tr>
<th>Installation</th>
<th><code>virtualenv</code></th>
<th><code>venv</code></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>yes, third-party package</td>
<td>no, standard library</td>
</tr>
</tbody>
</table>
## Differences Between the Two Tools

<table>
<thead>
<tr>
<th></th>
<th>virtualenv</th>
<th>venv</th>
</tr>
</thead>
<tbody>
<tr>
<td>Installation</td>
<td>yes, third-party package</td>
<td>no, standard library</td>
</tr>
<tr>
<td>Update</td>
<td>anytime via pip</td>
<td>with the interpreter</td>
</tr>
</tbody>
</table>
## Differences Between the Two Tools

<table>
<thead>
<tr>
<th></th>
<th>virtualenv</th>
<th>venv</th>
</tr>
</thead>
<tbody>
<tr>
<td>Installation</td>
<td>yes, third-party package</td>
<td>no, standard library</td>
</tr>
<tr>
<td>Update</td>
<td>anytime via pip</td>
<td>with the interpreter</td>
</tr>
<tr>
<td>Python version</td>
<td>2.7 and 3.4 or later</td>
<td>3.3 or later</td>
</tr>
</tbody>
</table>
## Differences Between the Two Tools

<table>
<thead>
<tr>
<th></th>
<th>virtualenv</th>
<th>venv</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Installation</strong></td>
<td>yes, third-party package</td>
<td>no, standard library</td>
</tr>
<tr>
<td><strong>Update</strong></td>
<td>anytime via pip</td>
<td>with the interpreter</td>
</tr>
<tr>
<td><strong>Python version</strong></td>
<td>2.7 and 3.4 or later</td>
<td>3.3 or later</td>
</tr>
<tr>
<td><strong>Python flavour</strong></td>
<td>PyPy, CPython</td>
<td>self-support for 3.3+</td>
</tr>
</tbody>
</table>
## Differences Between the Two Tools

<table>
<thead>
<tr>
<th></th>
<th>virtualenv</th>
<th>venv</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Installation</strong></td>
<td>yes, third-party package</td>
<td>no, standard library</td>
</tr>
<tr>
<td><strong>Update</strong></td>
<td>anytime via pip</td>
<td>with the interpreter</td>
</tr>
<tr>
<td><strong>Python version</strong></td>
<td>2.7 and 3.4 or later</td>
<td>3.3 or later</td>
</tr>
<tr>
<td><strong>Python flavour</strong></td>
<td>PyPy, CPython</td>
<td>self-support for 3.3+</td>
</tr>
<tr>
<td><strong>Configurability</strong></td>
<td>CLI + environment variables + per user config file</td>
<td>CLI</td>
</tr>
</tbody>
</table>
## DIFFERENCES BETWEEN THE TWO TOOLS

<table>
<thead>
<tr>
<th></th>
<th><strong>virtualenv</strong></th>
<th><strong>venv</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Installation</strong></td>
<td>yes, third-party package</td>
<td>no, standard library</td>
</tr>
<tr>
<td><strong>Update</strong></td>
<td>anytime via pip</td>
<td>with the interpreter</td>
</tr>
<tr>
<td><strong>Python version</strong></td>
<td>2.7 and 3.4 or later</td>
<td>3.3 or later</td>
</tr>
<tr>
<td><strong>Python flavour</strong></td>
<td>PyPy, CPython</td>
<td>self-support for 3.3+</td>
</tr>
<tr>
<td><strong>Configurability</strong></td>
<td>CLI + environment variables + per user config file</td>
<td>CLI</td>
</tr>
<tr>
<td><strong>Extensibility</strong></td>
<td>via plugins that can be installed alongside</td>
<td>by extending the base class</td>
</tr>
</tbody>
</table>
### Differences Between The Two Tools

<table>
<thead>
<tr>
<th></th>
<th>virtualenv</th>
<th>venv</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Installation</strong></td>
<td>yes, third-party package</td>
<td>no, standard library</td>
</tr>
<tr>
<td><strong>Update</strong></td>
<td>anytime via pip</td>
<td>with the interpreter</td>
</tr>
<tr>
<td><strong>Python version</strong></td>
<td>2.7 and 3.4 or later</td>
<td>3.3 or later</td>
</tr>
<tr>
<td><strong>Python flavour</strong></td>
<td>PyPy, CPython</td>
<td>self-support for 3.3+</td>
</tr>
<tr>
<td><strong>Configurability</strong></td>
<td>CLI + environment variables + per user config file</td>
<td>CLI</td>
</tr>
<tr>
<td><strong>Extensibility</strong></td>
<td>via plugins that can be installed alongside</td>
<td>by extending the base class</td>
</tr>
<tr>
<td><strong>Performance</strong></td>
<td>by using the cached seed mechanism &lt;500ms</td>
<td>2 seconds+ per invocation</td>
</tr>
</tbody>
</table>
## Differences Between the Two Tools

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

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

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

├── 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`

Did the `--no-site-packages` flag recently disappear? #1681

Closed  gvanrossum opened this issue on 29 Feb · 5 comments

```
gvanrossum on 29 Feb

Some readthedocs jobs recently started failing with a complaint from virtualenv about `--no-site-packages` (readthedocs)
```
THE CHOPPING BOARD

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

- entire project in one file
• relocatable virtual environments, provide alternative path
• drop deprecated flags, e.g., --no-site-packages

Did the '--no-site-packages' flag recently disappear? #1681

- Closed
  gvanrossum opened this issue on 29 Feb · 5 comments

  gvanrossum on 29 Feb
  Some readthedocs jobs recently started failing with a complaint from virtualenv about --no-site-packages (readthedocs)

• 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

jaraco on 10 Jun 2019

An ambitious proposal!
LET PEOPLE KNOW

• RFC posted on GitHub issue - June 10th 2019

jaraco on 10 Jun 2019

An ambitious proposal!

gaborbernat on 18 Jun 2019 • edited •

Thanks for the links, based on feedback I've started some initial work over at
:gaborbernat/virtualenv:src/virtualenv@rewrite . The plan is to get it into a release
shape within the next month (and have a week of open review before that). Based
on the impact of the change, I plan to make it release 20.0.0 .
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
<table>
<thead>
<tr>
<th>Author</th>
<th>Contributions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bernát Gábor</td>
<td>12907</td>
</tr>
<tr>
<td>Siddhant Kumar</td>
<td>900</td>
</tr>
<tr>
<td>Seungmin Ryu</td>
<td>286</td>
</tr>
<tr>
<td>Sorin Sbarnea</td>
<td>169</td>
</tr>
<tr>
<td>Julien Danjou</td>
<td>113</td>
</tr>
<tr>
<td>David Tucker</td>
<td>52</td>
</tr>
<tr>
<td>Anthony Sottile</td>
<td>43</td>
</tr>
<tr>
<td>Prajwal M</td>
<td>14</td>
</tr>
<tr>
<td>spetafree</td>
<td>11</td>
</tr>
<tr>
<td>Vincent Philippon</td>
<td>11</td>
</tr>
<tr>
<td>Pradyun Gedam</td>
<td>11</td>
</tr>
<tr>
<td>Nicola Soranzo</td>
<td>9</td>
</tr>
<tr>
<td>Claudio Jolowicz</td>
<td>8</td>
</tr>
<tr>
<td>Ian Wienand</td>
<td>5</td>
</tr>
<tr>
<td>Lumír 'Frenzy' Balhar</td>
<td>5</td>
</tr>
<tr>
<td>Sviatoslav Sydorenko</td>
<td>3</td>
</tr>
<tr>
<td>Michal Górny</td>
<td>3</td>
</tr>
<tr>
<td>Johannes Altmanninger</td>
<td>3</td>
</tr>
<tr>
<td>Xavier Fernandez</td>
<td>2</td>
</tr>
<tr>
<td>Jannis Leidel</td>
<td>2</td>
</tr>
<tr>
<td>txp314</td>
<td>1</td>
</tr>
<tr>
<td>shaido987</td>
<td>1</td>
</tr>
<tr>
<td>mondeja</td>
<td>1</td>
</tr>
<tr>
<td>Thomas Grainger</td>
<td>1</td>
</tr>
<tr>
<td>Theodor Dimitriou</td>
<td>1</td>
</tr>
<tr>
<td>Shantanu</td>
<td>1</td>
</tr>
<tr>
<td>Rowdy Howell</td>
<td>1</td>
</tr>
<tr>
<td>Philippe Ombredanne</td>
<td>1</td>
</tr>
<tr>
<td>Matthew StClair</td>
<td>1</td>
</tr>
<tr>
<td>João Vale</td>
<td>1</td>
</tr>
</tbody>
</table>
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
virtualenv 20.0.0 is now released, a first major release of a new better implementation for the future, virtualenv.pypa.io/en/latest/chan...
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**

<table>
<thead>
<tr>
<th>sysconfig</th>
<th>value</th>
</tr>
</thead>
<tbody>
<tr>
<td>scripts</td>
<td><code>{base}/bin</code></td>
</tr>
<tr>
<td>data</td>
<td><code>{base}</code></td>
</tr>
<tr>
<td>include</td>
<td><code>{base}/include/python{py_version_short}</code></td>
</tr>
<tr>
<td>stdlib</td>
<td><code>{base}/lib64/python{py_version_short}</code></td>
</tr>
<tr>
<td>platstdlib</td>
<td><code>{platbase}/lib64/python{py_version_short}</code></td>
</tr>
<tr>
<td>purelib</td>
<td><code>{base}/lib/python{py_version_short}/site-packages</code></td>
</tr>
<tr>
<td>platlib</td>
<td><code>{platbase}/lib64/python{py_version_short}/site-packages</code></td>
</tr>
</tbody>
</table>
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
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/
- 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**

<table>
<thead>
<tr>
<th>Type of file</th>
<th>Location and filename</th>
</tr>
</thead>
<tbody>
<tr>
<td>system</td>
<td><code>prefix/lib/pythonver/distutils/distutils.cfg</code></td>
</tr>
<tr>
<td>personal</td>
<td><code>$HOME/.pydistutils.cfg</code></td>
</tr>
<tr>
<td>local</td>
<td><code>setup.cfg</code></td>
</tr>
</tbody>
</table>
**MACOS - FIXING DISTUTILS**

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

<table>
<thead>
<tr>
<th>Type of file</th>
<th>Location and filename</th>
</tr>
</thead>
<tbody>
<tr>
<td>system</td>
<td><em>prefix/lib/python</em>&lt;sup&gt;ver&lt;/sup&gt;/distutils/distutils.cfg</td>
</tr>
<tr>
<td>personal</td>
<td><em>$HOME/.pydistutils.cfg</em></td>
</tr>
<tr>
<td>local</td>
<td><em>setup.ccg</em></td>
</tr>
</tbody>
</table>

- 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

<table>
<thead>
<tr>
<th>Type of file</th>
<th>Location and filename</th>
</tr>
</thead>
<tbody>
<tr>
<td>system</td>
<td>prefix/lib/pythonver/distutils/distutils.cfg</td>
</tr>
<tr>
<td>personal</td>
<td>$HOME/.pydistutils.cfg</td>
</tr>
<tr>
<td>local</td>
<td>setup.cfg</td>
</tr>
</tbody>
</table>

- can set prefix and install_{purelib,platlib,headers,scripts,data}
- closed 5 times, be tenacious about fixing bugs
• delegate creation to env (needs implementation for inline run)
• delegate creation to env (needs implementation for inline run)
• surprising behaviour, cannot check if exists (no read), but can run
delegate creation to env (needs implementation for inline run)
surprising behaviour, cannot check if exists (no read), but can run

```bash
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__qbz5n2k
self.stat()
    File "C:\Program Files\WindowsApps\PythonSoftwareFoundation.Python.3.7_3.7.1776.0_x64__qbz5n2k
return self._accessor.stat(self)
OSError: [WinError 1920] The file cannot be accessed by the system: 'C:\Users\traveler\AppData\Local\Microsoft\WindowsApps\PythonSoftwareFoundation.Python.3.7_qbz5n2k'
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
• Python 2 uses the host os.py as landmark file to detect stdlib
Python 2 uses the host os.py as landmark file to detect stdlib

Python can still function with only pyc files
Python 2 uses the host os.py as landmark file to detect stdlib

- Python can still function with only pyc files
- benefits of doing this: obfuscate source code or maximize storage
Python 2 uses the host os.py as landmark file to detect stdlib.

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/pc files.
EXPECT THE UNEXPECTED - PYC ONLY MODE

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

```bash
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

```bash
export PYTHONPATH=\*pip-2.1.1-py2.py3-none-any.whl
env/bin/python -m pip install \*pip-2.1.1-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

```bash
export PYTHONPATH=\$env\pipe\20.1.1-py2-py3-none-any.whl
env/bin/python -m pip install \$env/pipe\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
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
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/setup tools/wheel
  - install embedded pip/setup tools/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 straightforward 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`

```bash
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
  ```bash
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
  ```bash
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](https://bootstrap.pypa.io_virtualenv.pyz)
  automatically feeds from [https://github.com/pypa/get-virtualenv](https://github.com/pypa/get-virtualenv)
ZIPAPP WITH DYNAMIC DEPENDENCY

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

```python
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

<table>
<thead>
<tr>
<th>Library</th>
<th>3.9</th>
<th>3.8</th>
<th>3.7</th>
<th>3.6</th>
<th>3.5</th>
<th>3.4</th>
<th>2.7</th>
</tr>
</thead>
<tbody>
<tr>
<td>appdirs</td>
<td>1.4.4</td>
<td>1.4.4</td>
<td>1.4.4</td>
<td>1.4.4</td>
<td>1.4.4</td>
<td>1.4.4</td>
<td>1.4.4</td>
</tr>
<tr>
<td>distlib</td>
<td>0.3.1</td>
<td>0.3.1</td>
<td>0.3.1</td>
<td>0.3.1</td>
<td>0.3.1</td>
<td>0.3.1</td>
<td>0.3.1</td>
</tr>
<tr>
<td>filelock</td>
<td>3.0.12</td>
<td>3.0.12</td>
<td>3.0.12</td>
<td>3.0.12</td>
<td>3.0.12</td>
<td>3.0.12</td>
<td>3.0.12</td>
</tr>
<tr>
<td>six</td>
<td>1.15.0</td>
<td>1.15.0</td>
<td>1.15.0</td>
<td>1.15.0</td>
<td>1.15.0</td>
<td>1.15.0</td>
<td>1.15.0</td>
</tr>
<tr>
<td>importlib-metadata</td>
<td>1.7.0</td>
<td>1.7.0</td>
<td>1.7.0</td>
<td>1.7.0</td>
<td>1.1.3</td>
<td>1.1.3</td>
<td></td>
</tr>
<tr>
<td>importlib-resources</td>
<td>2.0.1</td>
<td>2.0.1</td>
<td>1.0.2</td>
<td>1.0.2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>pathlib2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>2.3.5</td>
</tr>
</tbody>
</table>
ZIPAPP CONTENT STRUCTURE

```
1 □ .
2    □ __main__.py
3    □ virtualenv
4    □ ...
5    □ virtualenv-20.0.25.dist-info
6       □ entry_points.txt
7       □ LICENSE
8       □ METADATA
9       □ RECORD
10      □ top_level.txt
11      □ WHEEL
12      □ zip-safe
13 □ virtualenv
14    □ appdirs-1.4.4-py2.py3-none-any
15    □ configparser-4.0.2-py2.py3-none-any
16    □ contextlib2-0.6.0.post1-py2.py3-none-any
17    □ distlib-0.3.0-py3-none-any
18    □ filelock-3.0.12-py3-none-any
19    □ importlib_metadata-1.1.3-py2.py3-none-any
20    □ importlib_metadata-1.6.1-py2.py3-none-any
21    □ importlib_resources-1.0.2-py2.py3-none-any
22    □ importlib_resources-2.0.1-py2.py3-none-any
23    □ pathlib2-2.3.5-py2.py3-none-any
24    □ scandir-1.10.0-cp38-cp38-macosx_10_14_x86_64
25    □ six-1.15.0-py2.py3-none-any
26    □ typing-3.7.4.1-py2-none-any
27    □ typing-3.7.4.1-py3-none-any
28    □ zipp-1.2.0-py2.py3-none-any
29    □ zipp-3.1.0-py3-none-any
30 □ distributions.json
31 □ 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

<table>
<thead>
<tr>
<th>Package</th>
<th>3.9</th>
<th>3.8</th>
<th>3.7</th>
<th>3.6</th>
<th>3.5</th>
<th>3.4</th>
<th>2.7</th>
</tr>
</thead>
<tbody>
<tr>
<td>appdirs</td>
<td>1.4.4</td>
<td>1.4.4</td>
<td>1.4.4</td>
<td>1.4.4</td>
<td>1.4.4</td>
<td>1.4.4</td>
<td>1.4.4</td>
</tr>
<tr>
<td>distlib</td>
<td>0.3.1</td>
<td>0.3.1</td>
<td>0.3.1</td>
<td>0.3.1</td>
<td>0.3.1</td>
<td>0.3.1</td>
<td>0.3.1</td>
</tr>
<tr>
<td>filelock</td>
<td>3.0.12</td>
<td>3.0.12</td>
<td>3.0.12</td>
<td>3.0.12</td>
<td>3.0.12</td>
<td>3.0.12</td>
<td>3.0.12</td>
</tr>
<tr>
<td>six</td>
<td>1.15.0</td>
<td>1.15.0</td>
<td>1.15.0</td>
<td>1.15.0</td>
<td>1.15.0</td>
<td>1.15.0</td>
<td>1.15.0</td>
</tr>
<tr>
<td>importlib-metadata</td>
<td>1.7.0</td>
<td>1.7.0</td>
<td>1.7.0</td>
<td>1.1.3</td>
<td>1.1.3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>importlib-resources</td>
<td>2.0.1</td>
<td>2.0.1</td>
<td>1.0.2</td>
<td>1.0.2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>pathlib2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>2.3.5</td>
<td>2.3.5</td>
</tr>
</tbody>
</table>
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

```
AttributeError: module 'virtualenv.activation.python' has no attribute 'PestythonActivator' #1857
```

fj-sanchez opened this issue 24 days ago · 21 comments

fj-sanchez 24 days ago

I think it's just a typo in the latest release...
BREAKING THE WORLD

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

AttributeError: module 'virtualenv.activation.python' has no attribute 'PestythonActivator' #1857

fj-sanchez opened this issue 24 days ago - 21 comments

fj-sanchez 24 days ago

I think it's just a typo in the latest release...

gaborbernat 24 days ago

So sorry, messed up that release earlier 😕, should be all fine now with 20.0.23.
Breaking The World

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

```
AttributeError: module 'virtualenv.activation.python' has no attribute 'PestyPythonActivator' #1857
```

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

Today in open source; nothing like to settle a debate than with:

"This change makes it like, 5% less nonsense, but it's still on the whole nonsense."

TLDR. Maintaining open-source is a thankless side project. I probably can find a better use of my free time.

Be nice but have/use code of conducts when needed
NOW, HOW POPULAR IS VIRTUALENV?
THANK YOU AND QUESTIONS