

What's in your pip toolbox?

Jon Banafato

@jonafato

PyCon 2017

2017-05-19

Overview

- Introduction to `pip`
- Improved dependency management workflow
- Other related tools

A framing device

```
"""A microservice solely to ``unicode`` strings."""
```

```
from flask import abort, Flask, request  
from unicode import unicode
```

```
app = Flask(__name__)
```

```
@app.route('/')
```

```
def slash():
```

```
    """Unicode an arg and return it."""
```

```
    arg = request.args.get('arg')
```

```
    if not arg:
```

```
        abort(400)
```

```
    decoded = unicode(arg)
```

```
    return decoded
```

Running the app

```
$ export FLASK_APP=app  
$ flask run
```

```
$ curl -G http://example.com:5000 \  
      --data-urlencode arg=Motorhead  
Motorhead
```

A First attempt

```
$ ssh jonafato@example.com  
$ sudo apt-get install python-flask  
$ sudo pip install unidecode
```

The End.

Questions?

A note on environments

Always use virtualenv

Installing dependencies

```
$ pip install flask unidecode
Collecting flask
  Using cached Flask-0.12-py2.py3-none-any.whl
Collecting unidecode
  Using cached Unidecode-0.04.20-py2.py3-none-any.whl
Collecting click>=2.0 (from flask)
  Using cached click-6.7-py2.py3-none-any.whl
Collecting Jinja2>=2.4 (from flask)
  Using cached Jinja2-2.9.4-py2.py3-none-any.whl
Collecting itsdangerous>=0.21 (from flask)
Collecting Werkzeug>=0.7 (from flask)
  Using cached Werkzeug-0.11.15-py2.py3-none-any.whl
Collecting MarkupSafe>=0.23 (from Jinja2>=2.4->flask)
Installing collected packages: click, MarkupSafe, Jinja2, Werkzeug, flask, unidecode
Successfully installed Jinja2-2.9.4 MarkupSafe-0.23
click-6.7 flask-0.12 itsdangerous-0.24 unidecode
```


Managing dependencies

```
$ pip freeze > requirements.txt
```

```
# requirements.txt  
click==6.7  
Flask==0.12  
itsdangerous==0.24  
Jinja2==2.9.4  
MarkupSafe==0.23  
Unidecode==0.4.20  
Werkzeug==0.11.15
```

Please don't do this

Why use requirements.txt?

`requirements.txt` communicates
application dependencies to:

1. Machines (build and deployment environments)
2. People (developers, ops teams, etc.)

My real requirements.txt

```
# requirements.txt
blessings==1.6
bpython==0.16
click==6.7
curtsies==0.2.11
first==2.0.1
flake8==3.2.1
Flask==0.12
greenlet==0.4.11
itsdangerous==0.24
jedi==0.9.0
Jinja2==2.9.4
MarkupSafe==0.23
mccabe==0.5.3
msgpack-python==0.4.8
neovim==0.1.13
pep257==0.7.0
...
```

A better way

Manage two sets of dependencies: one for people, one for machines.

```
# requirements-for-people.txt  
Flask  
Unidecode
```

Jumpstarting your requirements files

`pipreqs` automatically generates requirements based on application imports.

```
$ pipreqs .
```

```
# requirements.txt
Unidecode==0.04.20
Flask==0.12
```

Managing two sets of dependencies

```
# requirements-for-people.txt
Unidecode==0.04.20
Flask==0.12
```

```
# requirements-for-machines.txt
click==6.7
Flask==0.12
itsdangerous==0.24
Jinja2==2.9.4
MarkupSafe==0.23
Unidecode==0.4.20
Werkzeug==0.11.15
```

A better pip freeze

`pip-compile` **takes a** `requirements.in`
generates a frozen `requirements.txt`.

```
$ pipreqs . --savepath requirements.in  
$ pip-compile
```


A better pip freeze

```
# Requirements.txt
#
# This file is autogenerated by pip-compile
# To update, run:
#
#     pip-compile --output-file requirements.txt req
#
click==6.7                # via flask
Flask==0.12
itsdangerous==0.24       # via flask
Jinja2==2.9.4            # via flask
MarkupSafe==0.23         # via jinja2
Unidecode==0.04.20
Werkzeug==0.11.15        # via flask
```

Pipfile: the future?

A PyPA alternative to `pip-compile`.

Staying updated

From the CLI:

```
$ pip list --outdated  
Flask (0.10) - Latest: 0.12 [wheel]
```

On the web:

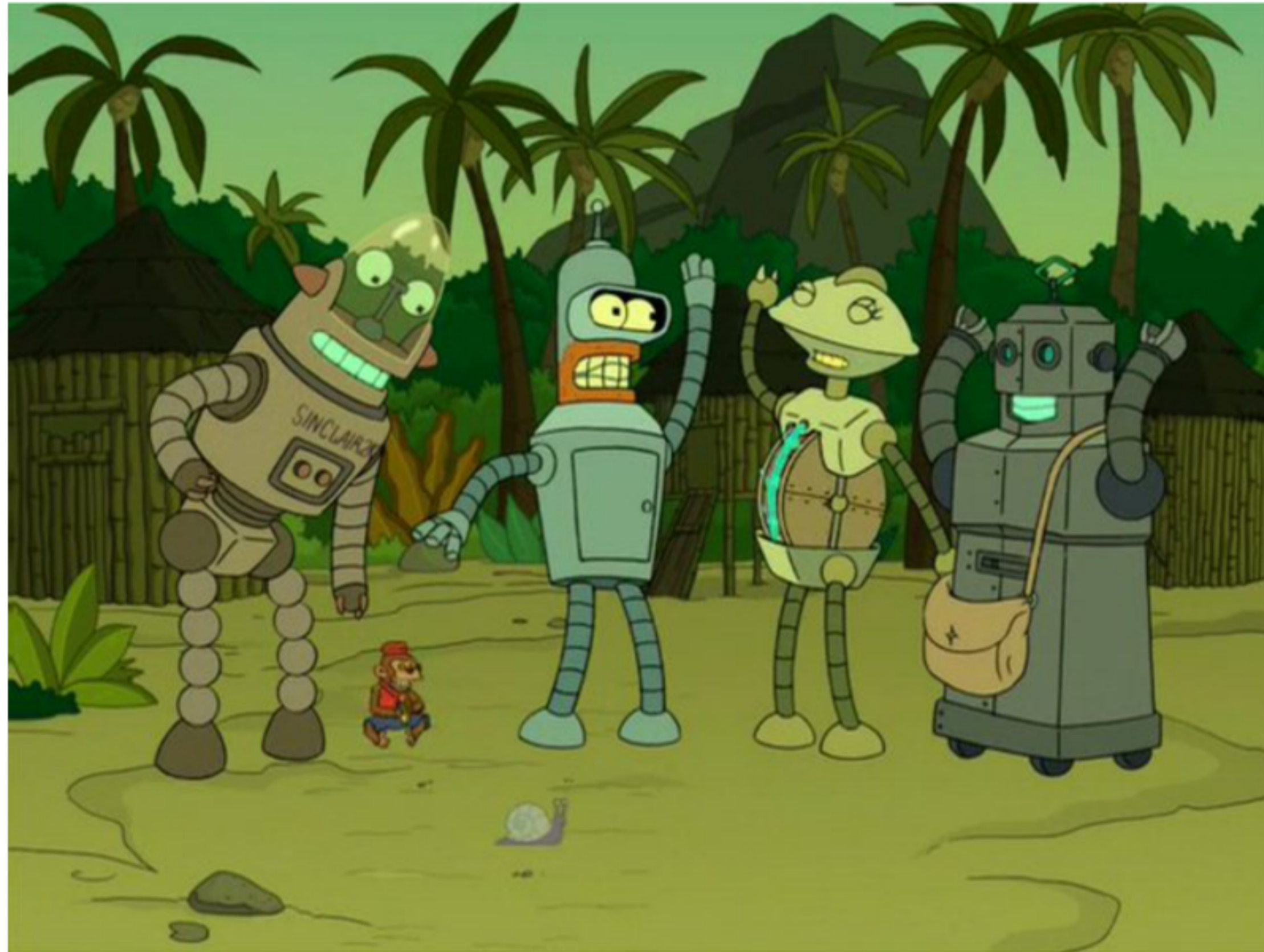
<http://therealkatie.net/version-check/>

If you use GitHub or Bitbucket:

<https://requires.io/>

<https://pyup.io/>

We did it!



But wait...



Hash checking

Making sure you install what you expect to install.

`hashin` **and** `pip-compile --generate-hashes` produce hashes. `pip` will validate against these hashes when installing packages.

Hash checking

```
$ pip-compile --generate-hashes
#
# This file is autogenerated by pip-compile
# To update, run:
#
#     pip-compile --generate-hashes --output-file re
#
click==6.7 \
    --hash=sha256:29f99fc6125fbc931b758dc053b3114e5!
    --hash=sha256:f15516df478d5a56180fbf80e68f20601!
Flask==0.12 \
    --hash=sha256:7f03bb2c255452444f7265eddb5160180!
    --hash=sha256:93e803cdbe326a61ebd5c5d353959397c!
itsdangerous==0.24 \
    --hash=sha256:cbb3fcf8d3e33df861709ecaf89d9e662!
...
```

Potpourri

Other tools to work with Python packages.

(Also the best Jeopardy! category.)

Exploring an environment

`pipdeptree` generates a dependency tree for everything currently installed.

```
$ pipdeptree
Flask==0.12
  - click [required: >=2.0, installed: 6.7]
  - itsdangerous [required: >=0.21, installed: 0.24]
  - Jinja2 [required: >=2.4, installed: 2.9.4]
    - MarkupSafe [required: >=0.23, installed: 0.23]
  - Werkzeug [required: >=0.7, installed: 0.11.15]
pipdeptree==0.9.0
  - pip [required: >=6.0.0, installed: 8.1.1]
setuptools==20.10.1
Unidecode==0.4.20
```

Why is this thing installed?

```
$ pipdeptree --reverse
blessings==1.6
  - curtsies==0.2.11 [requires: blessings>=1.5]
    - bpython==0.16 [requires: curtsies>=0.1.18]
click==6.7
  - Flask==0.12 [requires: click>=2.0]
  - pip-tools==1.8.0 [requires: click>=6]
first==2.0.1
  - pip-tools==1.8.0 [requires: first]
greenlet==0.4.11
  - bpython==0.16 [requires: greenlet]
  - neovim==0.1.13 [requires: greenlet]
```

Installing Python CLI utilities

```
$ pipsi install bpython
```

```
...
```

```
$ pipsi list
```

```
Packages and scripts installed through pipsi:
```

```
  Package "bpython":
```

```
    bpdb
```

```
    bpython
```

```
    bpython-curses
```

```
    bpython-urwid
```

```
  Package "pipsi":
```

```
    pipsi
```

Package info from PyPI

```
$ leak flask
```

```
=====  
Flask
```

```
A microframework based on Werkzeug, Jinja2 and good
```

```
-----  
Author:                Armin Ronacher
```

```
Available versions:    21
```

```
Home page:             http://github.com/pallets/flas:
```

```
=====  
(0.12) Most recent. 21/12/2016 20:22 release date
```

```
(0.11.1)
```

```
(0.11)
```

```
(0.10.1) Most popular. 10129912 downloads
```

```
(0.10)
```

```
(0.9)
```

```
...
```

Closing remarks

What are your favorite pip tools?

Questions?

Links

pipreqs:

<https://github.com/bndr/pipreqs>

pip-tools:

<https://github.com/nvie/pip-tools>

Pipfile:

<https://github.com/pypa/pipfile>

hashin:

<https://github.com/peterbe/hashin>

Links

pipdeptree:

<https://github.com/naiquevin/pipdeptree>

pipsi:

<https://github.com/mitsuhiko/pipsi>

pipenv:

<https://github.com/kennethreitz/pipenv>

leak: <https://github.com/bmwant/leak>

Links

requires.io: <https://requires.io>

pyup.io: <https://pyup.io/>

version-check:

<http://therealkatie.net/version-check/>

Fin.