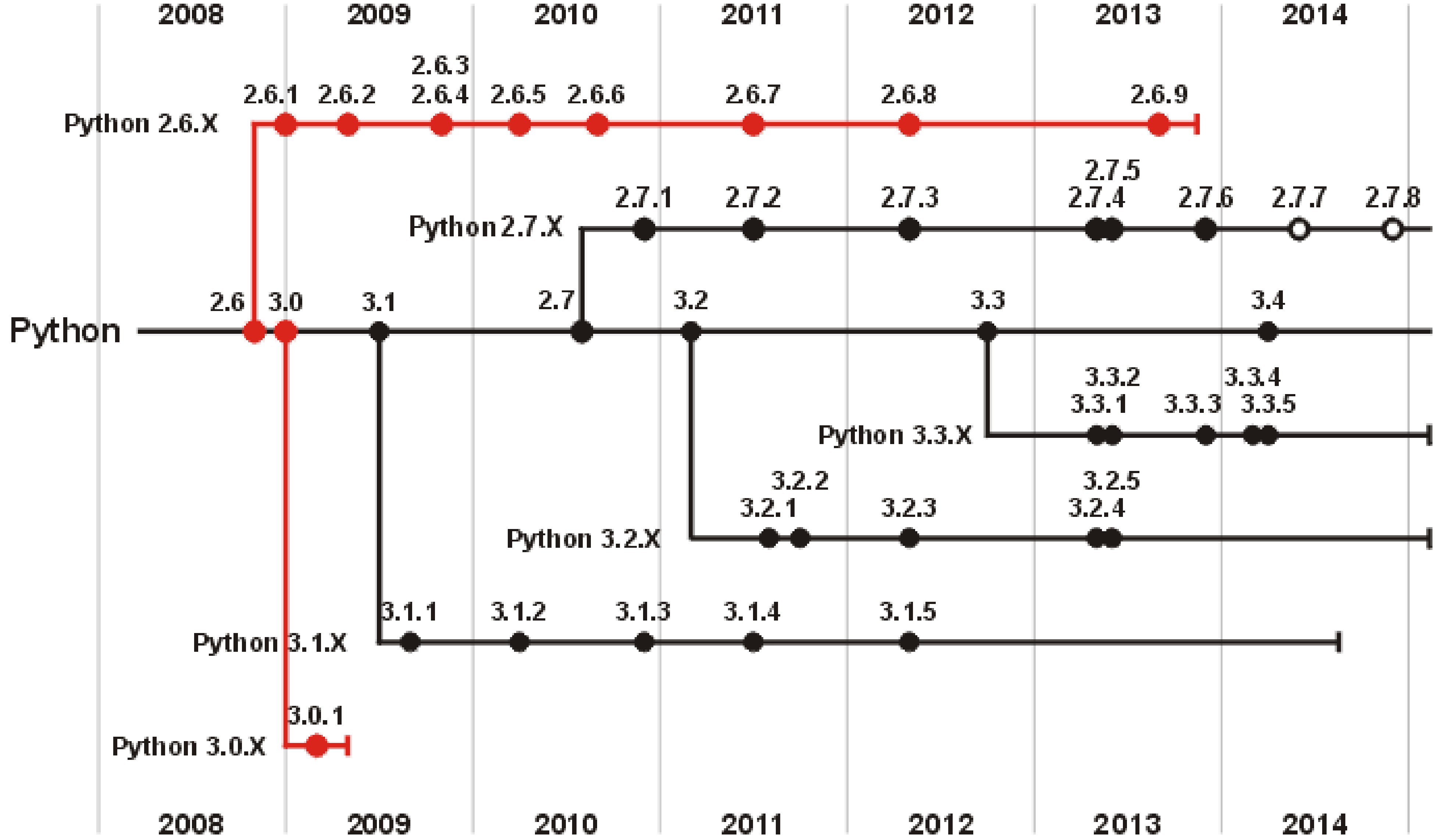
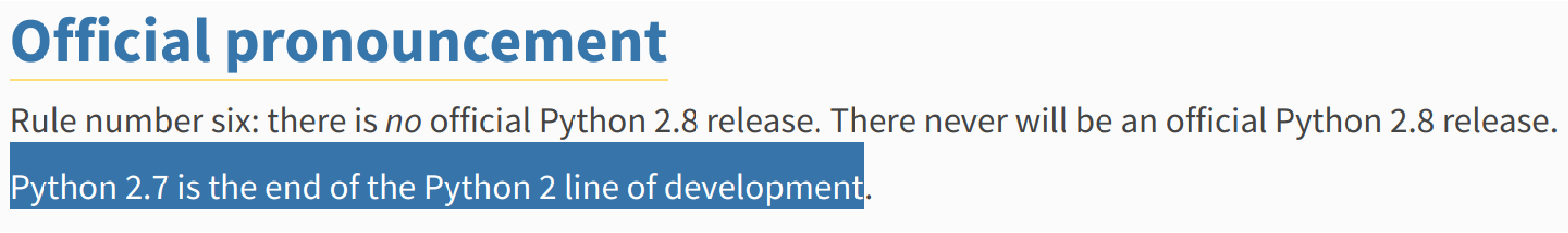
Python 3

1. HISTORY (what Python 2 is)


2008
2009
2010
2011
2012
2013
2014
Python 2.6.X
2.6.1
2.6.2
2.6.3
2.6.4
2.6.5
2.6.6
2.6.7
2.6.8
2.6.9
Python 2.7.X
2.7.1
2.7.2
2.7.3
2.7.4
2.7.5
2.7.6
2.7.7
2.7.8
Python
2.6
3.0
3.1
2.7
3.2
3.3
3.4
Python 3.3.X
3.3.1
3.3.2
3.3.3
3.3.4
3.3.5
Python 3.2.X
3.2.1
3.2.2
3.2.3
3.2.4
3.2.5
Python 3.1.X
3.1.1
3.1.2
3.1.3
3.1.4
3.1.5
Python 3.0.X
3.0.1

Python 2 is Windows XP



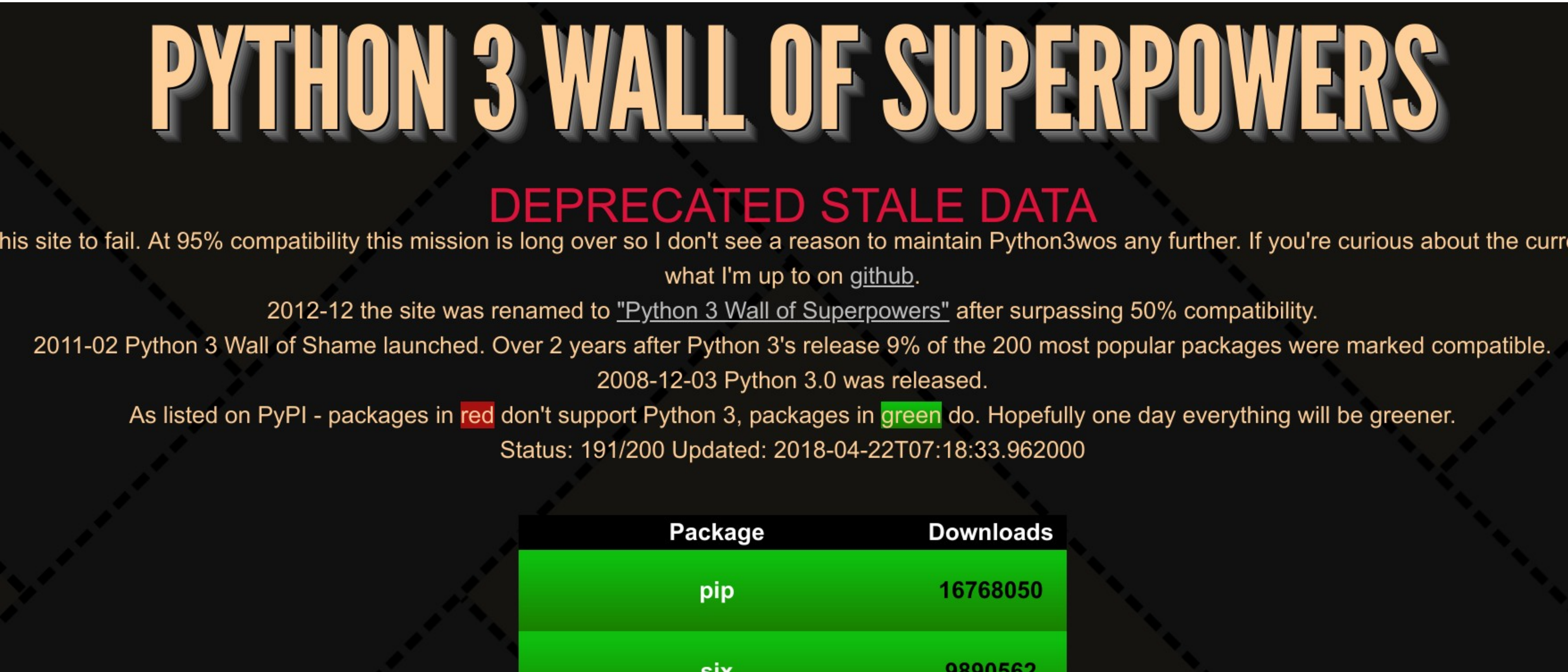
Official pronouncement

Rule number six: there is *no* official Python 2.8 release. There never will be an official Python 2.8 release. Python 2.7 is the end of the Python 2 line of development.

- only security and bug fixes until 2020

  https://www.python.org/dev/peps/pep-0404/

support is a deprecated excuse since ages ago



PYTHON 3 WALL OF SUPERPOWERS

DEPRECATED STALE DATA

his site to fail. At 95% compatibility this mission is long over so I don't see a reason to maintain Python3wos any further. If you're curious about the curr

what I'm up to on github.

2012-12 the site was renamed to "Python 3 Wall of Superpowers" after surpassing 50% compatibility.

2011-02 Python 3 Wall of Shame launched. Over 2 years after Python 3's release 9% of the 200 most popular packages were marked compatible.

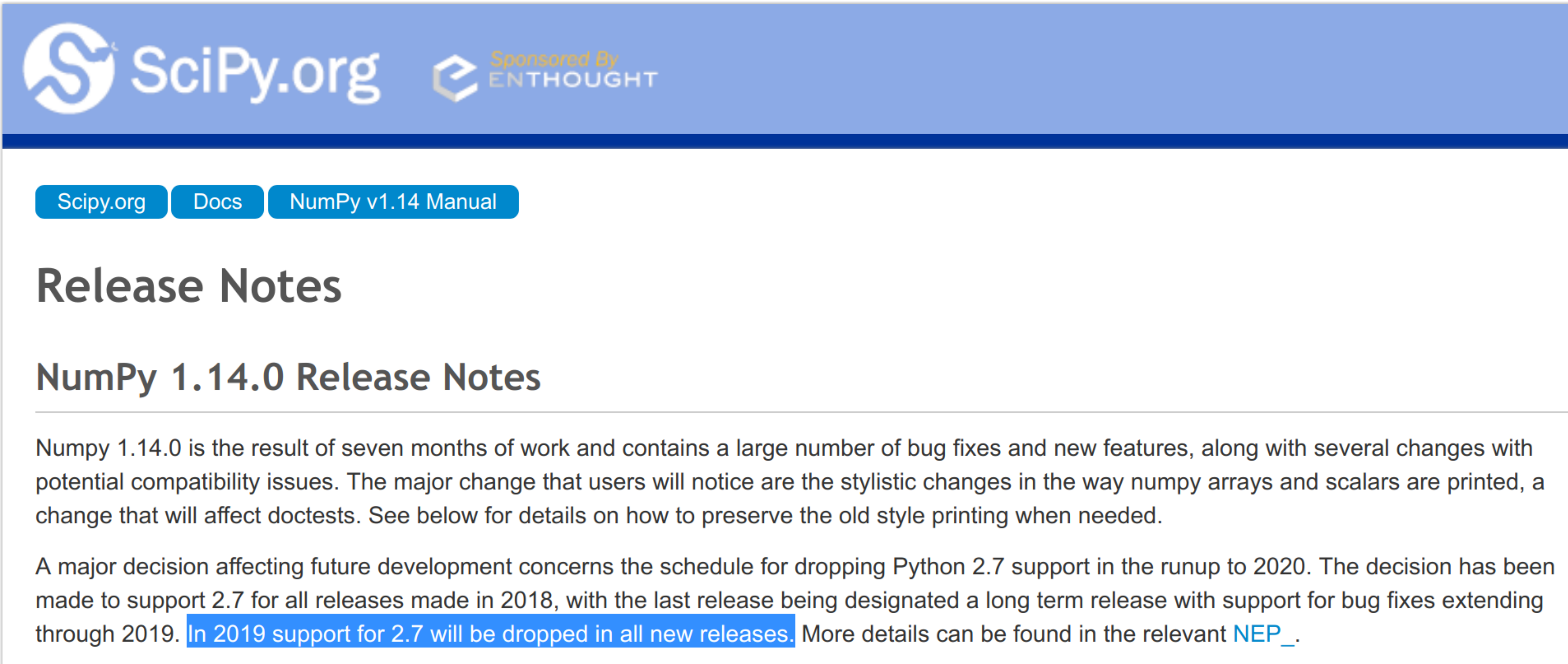
2008-12-03 Python 3.0 was released.

As listed on PyPI - packages in red don't support Python 3, packages in green do. Hopefully one day everything will be greener.

Status: 191/200 Updated: 2018-04-22T07:18:33.962000

Package	Downloads
pip	16768050
six	9890562

No more NumPy for Python 2



Release Notes

NumPy 1.14.0 Release Notes

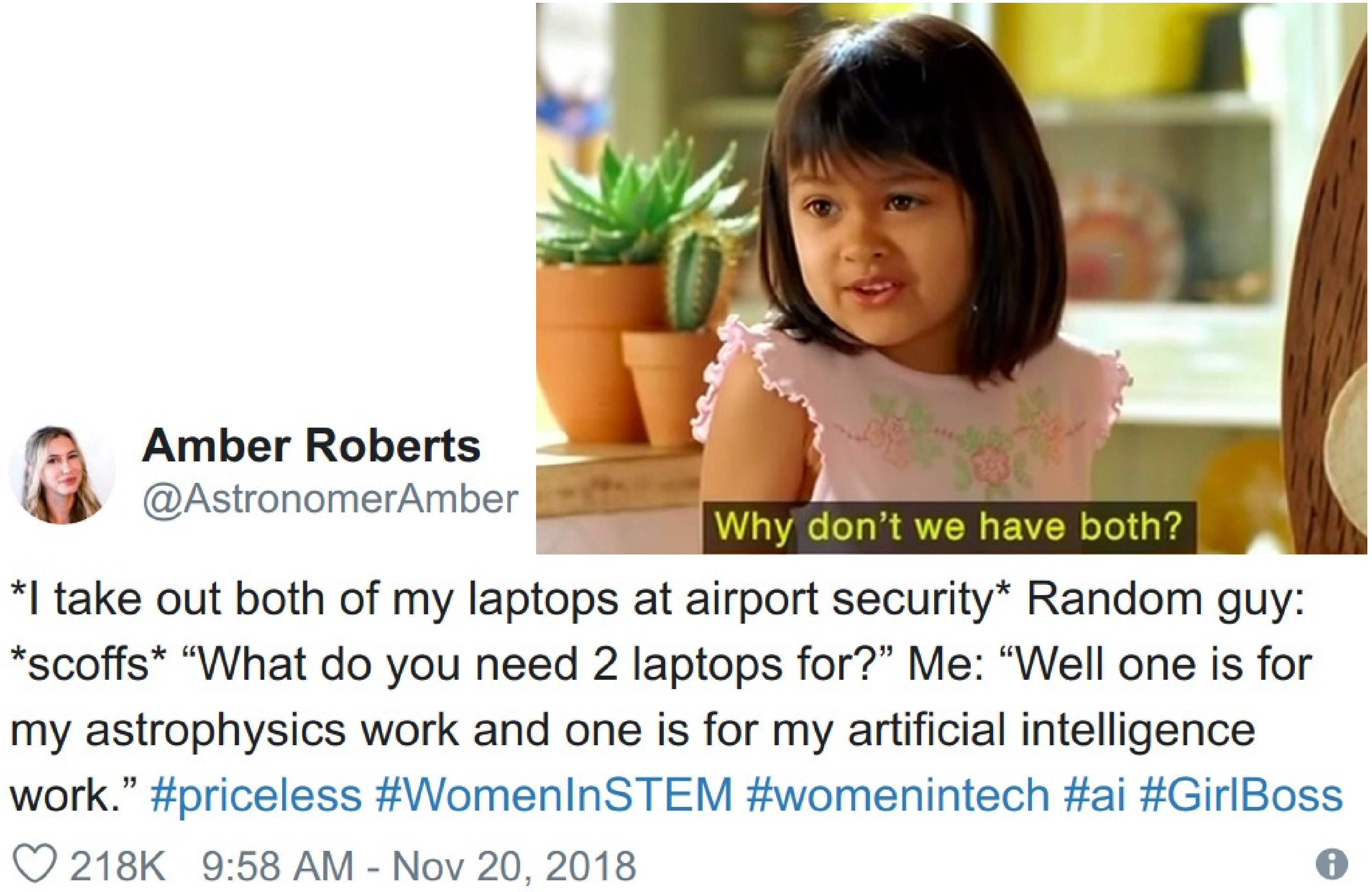
Numpy 1.14.0 is the result of seven months of work and contains a large number of bug fixes and new features, along with several changes with potential compatibility issues. The major change that users will notice are the stylistic changes in the way numpy arrays and scalars are printed, a change that will affect doctests. See below for details on how to preserve the old style printing when needed.

A major decision affecting future development concerns the schedule for dropping Python 2.7 support in the runup to 2020. The decision has been made to support 2.7 for all releases made in 2018, with the last release being designated a long term release with support for bug fixes extending through 2019. In 2019 support for 2.7 will be dropped in all new releases. More details can be found in the relevant NEP_.

Python 3

2. HOW TO (it’s easy)

Still more excuses??



Amber Roberts
@AstronomerAmber

I take out both of my laptops at airport security Random guy: *scoffs* “What do you need 2 laptops for?” Me: “Well one is for my astrophysics work and one is for my artificial intelligence work.” #priceless #WomenInSTEM #womenintech #ai #GirlBoss

218K 9:58 AM - Nov 20, 2018

Summary (pretty much everything)

`#!/usr/bin/env python2`	`#!/usr/bin/env python`
`print “stuff”`	`print(“stuff”)`

`map(mod, list)`
`dict.keys()`
	`list(map(mod, list))` *
	`dict` *

`def f((x, y)):`	`def fp(xy): return f(*xy)`
	`def f(x, y):`

`xrange(3)`
`range(3)`
	`range(3)`
	`list(range(3))`

`1 / 2 = 0`, `1 // 2 = 0`	`1 // 2 = 0`, `1 / 2 = 0.5`
.	* optional

Automated Conversion (not quite so easy)

- `2to3 python2.py` prints diff
  `2to3 python2.py -w -n` writes changes, no backup

Print (biggest offender)

- Cannot have `“print “`, only `“print(“`
- `print(“%s %d” % (a, b))` instead of `print(a, b)` for compatibility without future (below)
- If using `print(“no LF”, end=””)` or
  `print(“#!/usr/bin/env bash”, file=f)` then add
  `from __future__ import print_function` **before** any other imports

.

Automated Checkers

:(

```
% black python2.py

error: cannot format python2.py: cannot use --safe with this file; failed to parse source file with Python 3.6's builtin AST. Re-run with --fast or stop using deprecated Python 2 syntax. AST error message: Missing parentheses in call to 'print'. Did you mean print("stuff")? (<unknown>, line 11)

All done! 💥 💔 💥

1 file failed to reformat.

123 %
```

```
% black python3.py

reformatted python3.py

All done! ✨ 🍰 ✨

1 file reformatted.

%
```

...or just run it with python3.

features that you can't use because you refuse to upgrade

- `a, b, *rest = range(10)`
- `def f(a, b, *, option=True):`
- `None > 123` (no longer valid)
- `π = math.pi`
- `np.dot(a, b)` -> `a @ b` (__matmul__)

-
```
from pathlib import Path
directory = Path("/etc")
filepath = directory / "test_file.txt"
if filepath.exists():
    stuff
```

https://www.asmeurer.com/python3-presentation/python3-presentation.pdf

How to make it easy

- (now) No pull request accepted if non-compliant code found in diffed area (green on github)
- After significant changes to a script, run compliance test
- (later) Default Python on hypocentre reverted to 3
- Continuous integration / tests will test compliance