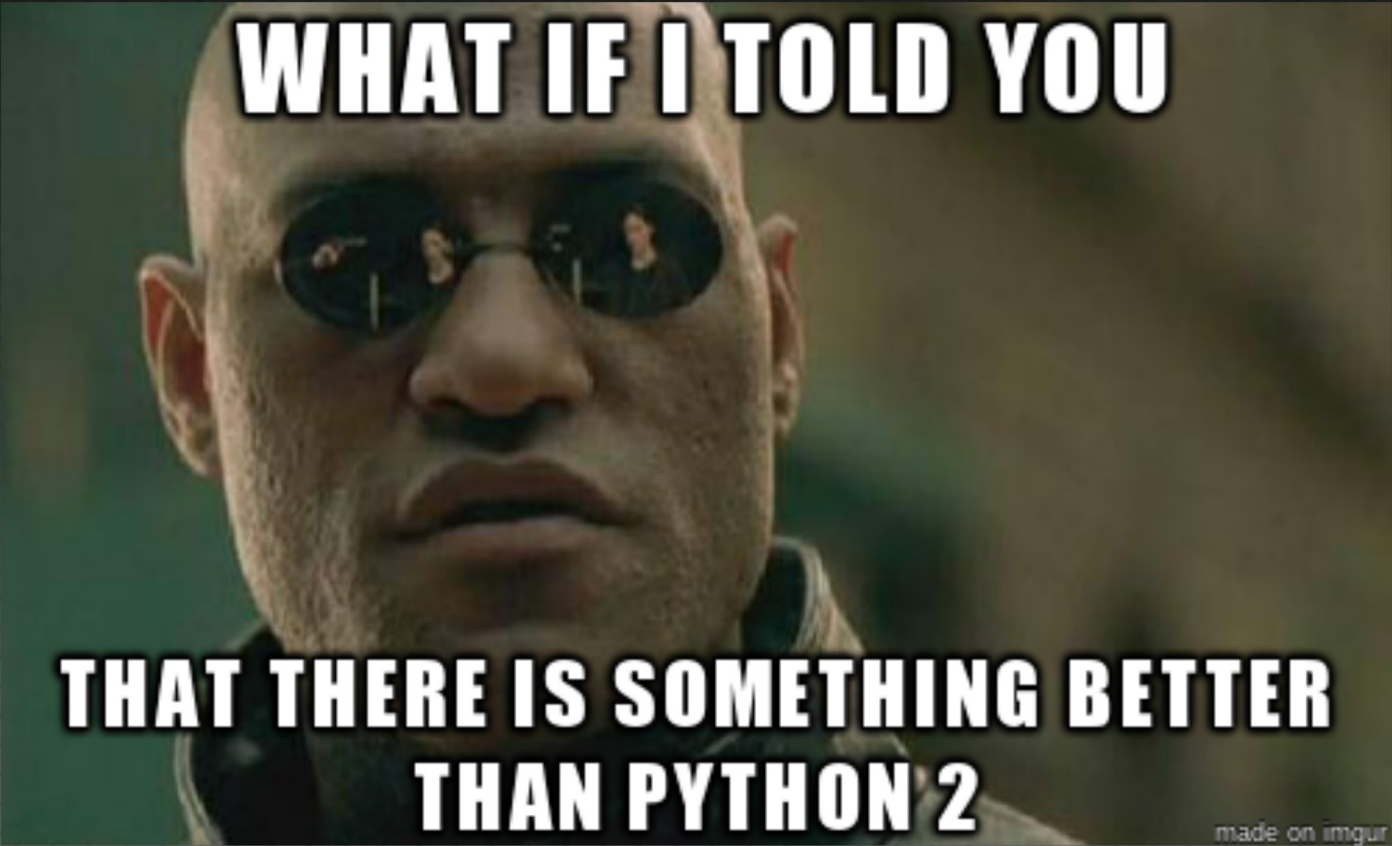




Python 3 is coming. Are you ready?

Slavek Kabrda

Senior Software Engineer, Red Hat

A close-up of Morpheus from The Matrix, wearing his signature sunglasses. The reflection in the lenses shows a scene from the movie. The image is framed by a dark background with a repeating circular pattern.

WHAT IF I TOLD YOU

**THAT THERE IS SOMETHING BETTER
THAN PYTHON 2**

made on imgur

Agenda

- What? Python 3?
- What's new?
- Porting
- Fedora 23: Python 3 as Default
- Q & A, discussion

What? Python 3?

Python 3: the programming language
the creators of Python 2 prefer to use :)

Nick Coghlan @ twitter

https://twitter.com/ncoghlan_dev/status/487250116433039360

Python 3 is...

- Also known as Python 3000 or Py3k
- Not fully compatible with Python 2.x
 - But you *can* write code that runs on both
- More Pythonic
- The present and future of the Python language

What's Different? (1/6)

- Print is no longer statement
- `from __future__ import print_function`
- `print 'foo' => print('foo')`

What's Different? (2/6)

- `dict.has_key('foo')` \Rightarrow `'foo' in dict`
- `dict.iter<what>()` \Rightarrow `dict.<what>()`
- `dict.<what>()` \Rightarrow `list(dict.<what>())`
- For `<what>` in `['keys', 'values', 'items']`

What's Different? (3/6)

- `basestring` => no longer exists
- `str` => `bytes`
- `unicode` => `str`
- `long/int` => `int`

What's Different? (4/6)

- `map()`, `filter()` and `zip()` return iterators
- `range()/xrange()` => `range()`

What's Different? (5/6)

- `except MyException, foo`
- `=>`
- `except MyException as foo`

What's Different? (6/6)

- `import foo` => always absolute
- To import from current package, use relative import:
 - `from . import foo`

And more...

- Renamed/split libraries (`configparser`, `urllib`, ...)
- New IO library
- No `cmp()`
- No ``12``
- Changes in C bindings
- Iterators ...

NOT SURE IF PY3K PORTING

IS WORTH THE EFFORT

made on imgur

What's new?
(or Python 3 Goodies)

Keyword Only Arguments (1/2)

- PY2: `def foo(a, b, beep=False, explode=False)`
 - `foo(1, 2, True)`
- PY3: `def foo(a, b, *, beep=False, explode=False)`
 - `foo(1, 2, True) => TypeError`
 - `foo(1, 2, beep=True) => correct`

Keyword Only Arguments (2/2)

- Prevents:
 - accidental passing of more arguments
 - errors when reordering arguments
- Helps get rid of `**kwargs` in some situations

Chained Exceptions

- `raise SomeException from e`
- Allows chaining tracebacks => preserves the original error

Advanced Unpacking

- `first, *x, last = file.readlines()`
- `*zero_to_eight, nine = range(10)`

Matrix Multiplication

- New in Python 3.5
- `c = a @ b`
- `a @= b`
- Not actually implemented for any of standard types

Type Hinting

- Optional, never to be made mandatory
- New in Python 3.5 (arbitrary annotations were added in 3.0)
- For use by linters/IDEs

```
def foo(x: int, y: str) -> str:  
    return y * x
```

Subgenerators

- `yield from other_generator()`
- `yield from range(10)`
- `yield from [1, 2, 3]`
- `yield from afile.readlines()`

Asyncio/coroutines

- Asyncio added to Python 3.4
- New syntax (and more support) for asynchronous programming added to Python 3.5
- Coroutines, awaitable objects, asynchronous loops, asynchronous context managers, ...

“Trying to decide if I have enough spare brain cells to understand the Python3.4 asyncio module.”

David Beazley, <https://twitter.com/dabeaz/status/440214755764994048>

And more...

- `faulthandler`
- `ipaddress`, `pathlib`
- `enum (!)`
- Fine grained `OSError` subclasses
- (Almost) Everything is an iterator
- No more funny comparisons
 - `'abc' > 42 ==> TypeError`



Porting

How to Port?

- Decide if you want to keep backwards compatibility
 - If yes, don't go below Python 2.6 (RHEL 5 anyone?)
 - Ideally, port to 3.3+
- Good test coverage is a must
- Useful tools:
 - 2to3
 - python-modernize
 - python-six
 - Other tools like pyflakes, pep8, ...

Resources

- <https://docs.python.org/3/howto/pyporting.html>
- <https://docs.python.org/3/howto/cporting.html>
- <http://lucumr.pocoo.org/2013/5/21/porting-to-python-3-redux/>
- <http://python3porting.com/>
- <http://www.wefearchange.org/2011/12/lessons-in-porting-to-python-3.html>
- <https://pythonhosted.org/six/>
- <https://wiki.gnome.org/action/show/Projects/PyGObject/IntrospectionPorting>
- <http://overtag.dk/wordpress/2013/01/first-impressions-of-gtk3-migration-in-python>
- <https://bkabrda.fedorapeople.org/python3-are-you-ready.pdf>
- slavek@redhat.com

Fedora 23 – Python 3 as Default

**ONE DOES NOT
SIMPLY**

**SWITCH A LINUX DISTRO TO
PYTHON 3**

memegenerator.net

It means ...

- We now prefer Python 3 for applications
- Workstation LiveCD only has Python 3
- Everything on Cloud Image and Atomic Host uses Python 3, but Python 2 is still there

It doesn't mean ...

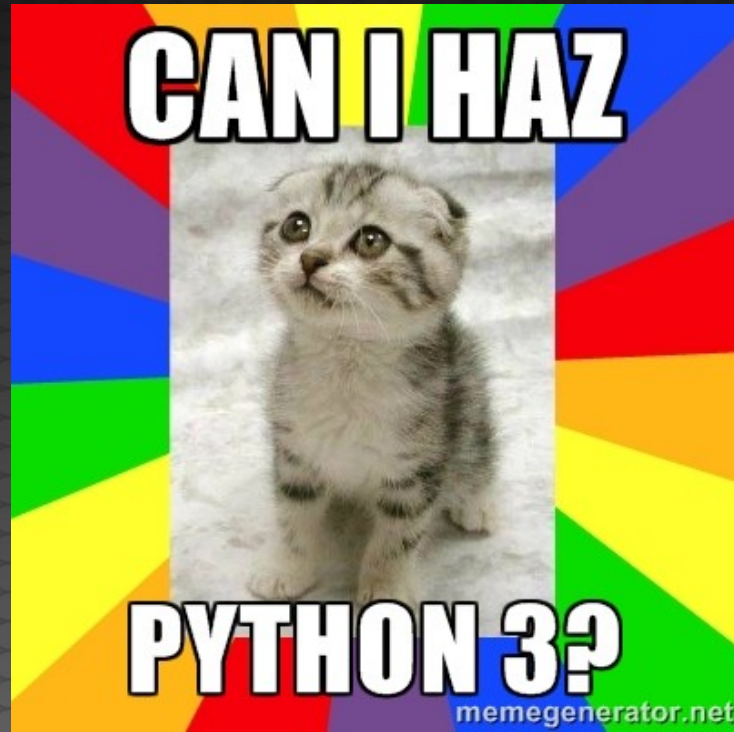
- Dropping Python 2 from the distribution
- Forcing people to write downstream-only patches
- Pointing `/usr/bin/python` to `python3`

Helping out

We Need Your Help!

- Join Fedora Activity Day on 14-15 November
 - https://fedoraproject.org/wiki/FAD_Python_3_Porting_2015
- We're now concentrating on Server LiveDVD
 - But also everything else that's in Fedora and it's Python 2 only
- Remember, upstream is first

Questions?



slavek@redhat.com