PYTHON 2 IS DEAD!

DRAG YOUR OLD CODE INTO THE MODERN AGE

BECKY SMITH

PyData Edinburgh 6 Dec 2018

ABOUT ME

Becky Smith

brainwave

ecometrica

@rebkwok

rebkwok@gmail.com

joined the choir invisible

shuffled off its mortal coil

passed on

ceased to exist

it is a stiff

expired

it is no more

it is an ex-python

run down the curtain

python2

gone

this python is no more

deceased

dead

expired

pushing up the daisies

kicked the bucket

gone to meet its maker

WHAT THIS TALK IS

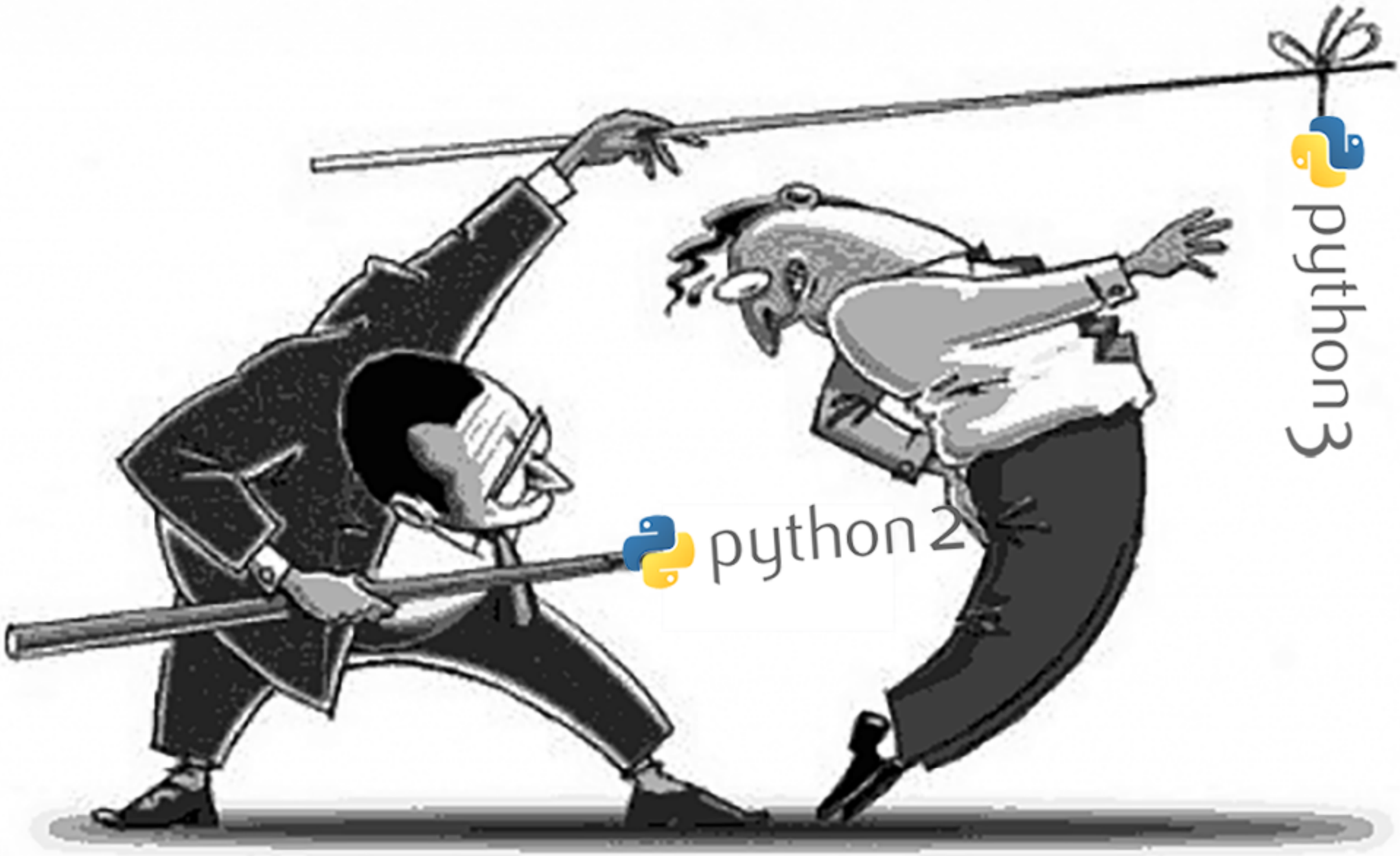
- ☑ A case study: upgrading a large Python/Django codebase
- ☑ Introduction to available tools
- ☑ Our experience, strategies and gotchas encountered on the way

WHAT THIS TALK IS NOT

- ☐ How-to instruction
- ☐ A single “best” way to deal with python 3 upgrading

WHY UPGRADE?

- Python 2.7 support dropped January 2020
- Major projects dropping or pledged to drop Python 2 support
  - pandas
  - NumPy
  - Django
  - SciPy


python 3
python 2

WHY UPGRADE?

AKA WHY YOU WANT PYTHON 3

- That unicode thing (https://snarky.ca/why-python-3-exists)
- Better iterations; everything's an iterator
  - `range/~~xrange~~`
  - `dict.keys/~~dict.iterkeys~~`
  - `map/~~itertools.imap~~`
- Restrictions on comparators - no more **`'foo' > 4`**
- Advanced unpacking

```
>>> a, *b, c = range(5)
>>> a
0
>>> c
4
>>> b
[1, 2, 3]
```

```
>>> d1 = {'a': 1, 'b': 2}
>>> d2 = {'c': 3, 'd': 4}
>>> d3 = {**d, **d1, 'e': 5}
>>> d3
{'b': 2, 'e': 5, 'd': 4, 'a': 1, 'c': 3}
>>> d4 = {**d3, 'b': 10}
>>> d4
{'a': 1, 'b': 10, 'd': 4, 'c': 3, 'e': 5}
```

WHY UPGRADE?

AKA WHY YOU WANT PYTHON 3

- Keyword only arguments

```
def myfunction(a, b, *, c=False):
    pass
```

WHY UPGRADE?

AKA WHY YOU WANT PYTHON 3

- f-strings
  - `"Hello {}".format(name)` → `f"Hello {name}"`
  - `"Hello {}".format(name.upper())` → `f"Hello {name.upper()}`
- asyncio

THE PROJECT

- Ecometrica's Mapping platform
- Django project
- ~70,000 lines of python code in the core project
- 8 years old
- Multiple dependencies

Select Map Style | Display Groups | Area Groups

Filter areas by name...

Panama
Medellín
Venezuela
Guyana
Suriname
French Guiana
Colombia
Ecuador
Belém
Peru
Lima
Brazil
Bolivia
Brasília
Paraguay
São Paulo
Rio de

500km
-0.26772959, -58.452497

Select Map Style | ESA CCI 2010 Land C... | Area Groups

Filter areas by name...

Legend | Description

ESA CCI 2010 Land Cover 300m

Filter legend...

- Rainfed Cropland
- Rainfed Cropland (Herbaceous cover)
- Rainfed Cropland (Tree or shrub cover)
- Cropland, irrigated or post-flooding
- Mosaic cropland (>50%) / natural vegetation (tree, shrub, herbaceous cover) (<50%)
- Mosaic natural vegetation (tree, shrub, herbaceous cover) (>50%) / cropland (<50%)
- Tree cover, broadleaved, evergreen, closed to open (>15%)
- Tree cover, broadleaved, deciduous, closed to open (>15%)
- Tree cover, broadleaved, deciduous, closed (>40%)

500km

5.1818852, -64.503485

Transparency: 100%

Select Map Style | Display Groups | Area Groups

Filter areas by name...

Parque Nacional Serrania de La Neblina

Amazonas National Forest

Terra Indígena Balaio

Pico da Neblina National Park

Parque Nacional do Pico da Neblina

Results | Description | Details

São Gabriel da Cachoeira

Terra Indígena Médio Rio Negro II

Terra Indígena Médio Rio Negro I

Santa Isabel do Rio Negro

Terra Indígena Rio Téa

30km

1.2190119, -67.291401

Filter areas by

Results | Description | Details

Query results for Parque Nacional do Pico da Neblina

▸ Pantropical Carbon Estimates

▾ **Total Biomass & Carbon (~2003)**

Above-Ground Biomass
556 374 602 t

Below-Ground Biomass
148 216 854 t

Carbon Stock
352 295 691 ± 110 654 207 tC

▸ Average Biomass & Carbon Density (where >0 tC/ha)

▸ I. Carbon in Vegetation Types (ESA Globcover 2009)

▸ III. Carbon in Vegetation Types (ESA CCI 600m)

▸ II. Carbon in Vegetation Types (ESA CCI 300m)

▸ Past Deforestation - Prodes Annual Deforestation

▸ Past Deforestation - Hansen et al (2013) v1.2 (Staff Only)

Select Map Style | Display Groups | Area Groups

Filter areas by

Parque Nacional Serrania de La Neblina

Amazonas National Forest

Pico da Neblina

Parque Nacional do Pico da Neblina

Results | Description | Details

Terra Indígena Médio Rio Negro I

Terra Indígena Rio Téa

Results | Description | Details

Deforestation Risk

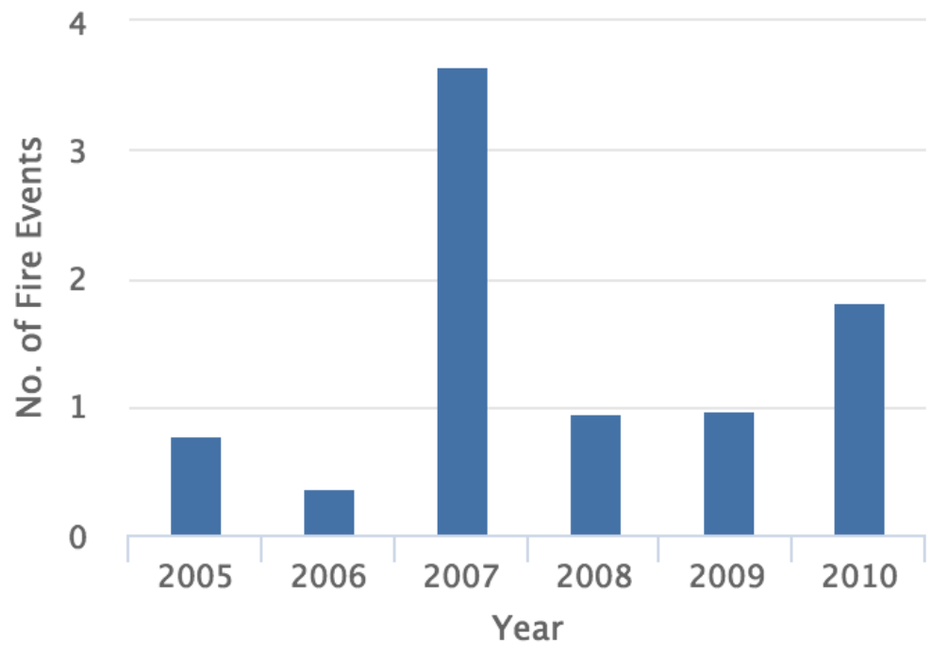
Approximate Biodiversity Metric Score

Approximate Population Count & Density

Human Water Security

Past Fire Events

Total number of fire events, derived from density maps of fire incidences recorded by MODIS.



TOOLS

- caniusepython3
- 2to3
- modernize
- Python docs
- Django’s guide (<1.11)
- Pylint

CANIUSEPYTHON3

- pip install it
- run it on your dependencies

```
>>> caniusepython3 -r requirements.txt

>>> caniusepython3 -m PKG-INFO

>>> caniusepython3 -p django pillow
```

2TO3

```
def welcome_to_pydata(name, month):
    print "Welcome to the {} PyData meetup, {}!".format(month, name)


if __name__ == '__main__':
    print "What's your name?"
    name = raw_input()

    print "What month is it?"
    month = raw_input()

    try:
        month = int(month)
    except ValueError, e:
        print "Provide a valid numeric month ({})".format(e.message)
    else:
        month = calendar.month_name(month)
        welcome_to_pydata(name, month)
```

2TO3

>>> 2to3 welcome.py

```
RefactoringTool: Skipping optional fixer: idioms
RefactoringTool: Skipping optional fixer: set_literal
RefactoringTool: Skipping optional fixer: ws_comma
RefactoringTool: Refactored welcome.py
--- welcome.py  (original)
+++ welcome.py  (refactored)
@@ -3,20 +3,20 @@


 def welcome_to_pydata(name, month):
-    print "Welcome to the {} PyData meetup, {}!".format(month, name)
+    print("Welcome to the {} PyData meetup, {}!".format(month, name))


 if __name__ == '__main__':
-    print "What's your name?"
-    name = raw_input()
+    print("What's your name?")
```

2TO3

```
print "What month is it?"
month = raw_input()
print("What month is it?")
month = input()

try:
    month = int(month)
except ValueError, e:
    print "Provide a valid numeric month ({})".format(e.message)
except ValueError as e:
    print("Provide a valid numeric month ({})".format(e.message))
else:
    month = calendar.month_name(month)
    welcome_to_pydata(name, month)
actoringTool: Files that need to be modified:
actoringTool: welcome.py
```

LINTING

```
✨  pylint --py3k welcome.py
No config file found, using default configuration
************* Module welcome
E:  5, 4: print statement used (print-statement)
E:  9, 4: print statement used (print-statement)
W: 10,11: raw_input built-in referenced (raw_input-builtin)
E: 12, 4: print statement used (print-statement)
W: 13,11: raw_input built-in referenced (raw_input-builtin)
E: 18, 8: print statement used (print-statement)
W: 18,49: Exception.message removed in Python 3 (exception-message-attribute)

--------------------------------------
Your code has been rated at -9.17/10
```

CASE STUDY:
WHAT WE DID

ARCHITECTURE

- Ubuntu 16.04
- Python 3.6
  - requires additional system packages

```
sudo add-apt-repository ppa:deadsnakes/ppa
sudo apt-get update
sudo apt-get install python3.6
sudo apt-get install python3.6-dev
```

  - virtualenv

```
mkvirtualenv --python=$(which python3.6) venv
```

#1: LEARN ABOUT DIFFERENCES BETWEEN PYTHON 2 AND PYTHON 3

- python3porting.com
- python-future's cheat sheet
- python-future.org/compatible_idioms.html

#2: TESTS

#3: DEPENDENCIES

```
$ caniusepython3 -r requirements.txt

Finding and checking dependencies ...

You need 11 projects to transition to Python 3.
Of those 11 projects, 10 have no direct dependencies blocking their
transition:

    django-colorfield
    django-hashedfilenamestorage
    django-migration-testcase
    django-paintstore
    django-uuidfield
    geographiclib
    pyproj (which is blocking geopandas)
    grandfatherson
    numexpr
    scrubber
    yas3fs
```

#4: FIX THE CODE

- App by app
  - 2to3 on the entire app (keeping backup files)
  - Run and fix tests
  - Commit changes app by app
  - Run django

#5: REVIEW / REFACTOR

- App by app
- 2to3 is conservative
  - `list()` around new iterators e.g. `list(foo.keys())`
  - extra parentheses around `print()` statements
  - `is_callable(x)` → `isinstance(x, collections.Callable)`
- Refactor
  - beware of `from __future__ import unicode_literals`
  - python-future.org/unicode_literals.html

#6: LINTING

- `pylint --py3k`

(Disclaimer: we didn't do this, but I wish we had)

#8: USER TESTING

Everything was going so well…

- gdal2mbtiles
- check all your dependencies

GOTCHAS

- rounding
  - Python 2: round exact half away from zero

```
round(2.5) == 3
round(3.5) == 4
```

  - Python 3: round exact half to nearest even (Bankers Rounding)

```
round(2.5) == 2
round(3.5) == 4
```

- exceptions
  - `Exeption.message` no longer exists; use `str(e)`

GOTCHAS

- `hash()`
  - randomisation is on by default
- pickling
  - pickle in python 2, unpickle in python 3 results in UnicodeDecodeError
  - new pickle protocols in python 3
  - **django-redis** defaults to latest protocol

GOTCHAS

- Sorting/comparing things

Python 2

```
>>> foo = {'a': 1}
>>> foo > 0
True
>>> foo = {}
>>> foo > 0
True
>>> foo = None
>>> foo > 0
False


>>> sorted(['1', 2, '4', '3'])
[2, '1', '3', '4']
```

Python 3

```
>>> foo = {'a': 1}
>>> foo > 0
Traceback (most recent call last):
  File "<stdin>", line 1, in
<module>
TypeError: unorderable types:
dict() > int()


>>> sorted(['1', 2, '4', '3'])
Traceback (most recent call last):
  File "<stdin>", line 1, in
<module>
TypeError: unorderable types: str()
< int()
```

RESULTS

- 325 files changed
- 2269 lines added
- 2561 lines deleted

LESSONS LEARNED

- Doesn’t have to be (too) painful
- Make python 2 code python 3 compatible in advance
- Be familiar with python 3 changes
- 2to3 is great, but only up to a point
- Good test coverage
- Check all your dependencies
- Be prepared to upgrade 3rd party libraries

THANK YOU!

https://speakerdeck.com/rebkwok/pydata-edinburgh-2018

RESOURCES

- Projects committed to moving to python 3 (including timelines)
  - python3statement.org
- Porting to python 3
  - docs.python.org/3/howto/pyporting.html
  - python3porting.com
  - docs.djangoproject.com/en/1.11/topics/python3/
  - python-future.org/compatible_idioms.html
- Description of the unicode/str issue and why python 3 largely exists to fix it
  - snarky.ca/why-python-3-exists/
- Python 3 features
  - asmeurer.com/python3-presentation/slides.html
  - eev.ee/blog/2016/07/31/python-faq-why-should-i-use-python-3/