```python
import bacon
```

PyDays 2019

self.intro()

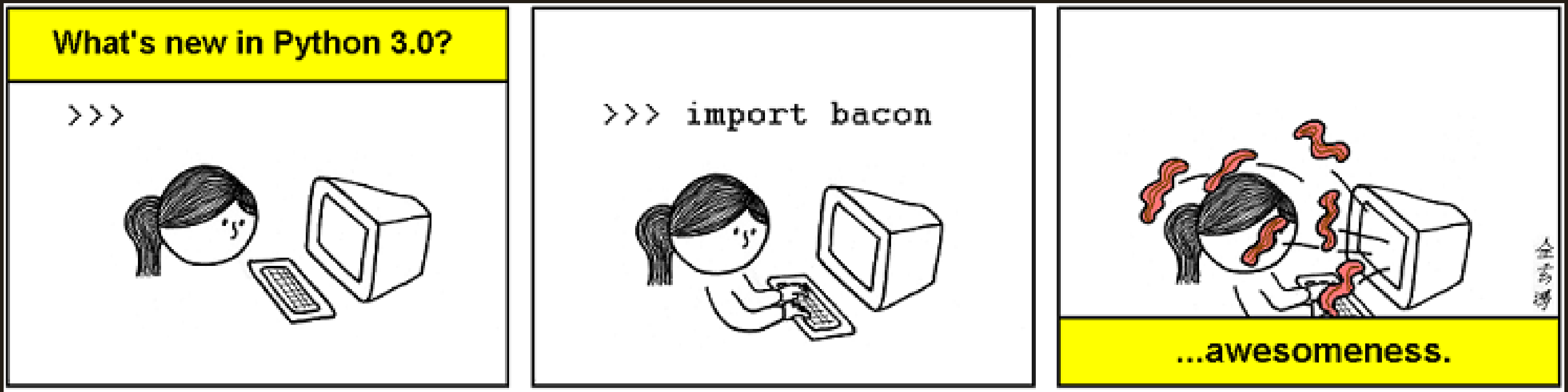
Ivana Kellyerova

@aerinthenniel   gitlab.com/jenx

- backend developer at iMobility
- dabbling in Python for 5+ years
- tea drinker
- game player
- prog metal listener

icons from icofont.com

The Python standard library
can do almost anything.



abstrusegoose.com/81

Python 3.7.3

https://docs.python.org/3/library/

the standard library

800+

contributors

10 000+

commits

also

the standard library

639

XXXs

```
# XXX obviously wrong, see #3232
```

cpython/Lib/encodings/idna.py, line 194

```
# XXX: shouldn't this code be removed, not commented out?
```

cpython/Lib/imaplib.py, line 423

```
# XXX yuck
```

cpython/Lib/importlib/_bootstrap_external, line 411

639

XXXs

80

hacks

```
# Can't think of a clean solution, so we hack away
```

cpython/Lib/idlelib/paragraph.py, line 164

```
# Now for some serious hackery
```

cpython/Lib/email/_header_value_parser.py, line 2283

639

XXXs

80

hacks

74

TODOs

```
# TODO: replace the frame hack if a blessed way
# to know the calling module is ever developed
```

cpython/Lib/enum.py, line 426

```
self.assertEqual(binop.col_offset, 3)  # FIXME: this is wrong
self.assertEqual(binop.left.col_offset, 3)  # FIXME: this is wrong
self.assertEqual(binop.right.col_offset, 7)  # FIXME: this is wrong
self.assertEqual(binop.col_offset, 3)  # FIXME: this is wrong
self.assertEqual(binop.left.col_offset, 3)  # FIXME: this is wrong
self.assertEqual(binop.right.col_offset, 7) # FIXME: this is wrong
```

cpython/Lib/test/test_fstring.py, lines 250, 251, 252, 262, 263, 264

61

FIXMEs

```
#######    vererbung!!!!!!!!!!!!!!!!!!!!!!
```

cpython/Lib/turtle.py, line 3188

61

FIXMEs

8

!!!s

639

XXXs

80

hacks

74

TODOs

61

FIXMEs

8

!!!s

the noteworthy

(for all sorts of reasons)

the builtins

Built-ins

Constants, functions, types living in the global namespace.

- `True`, `False`, `None`, `Ellipsis`

```python
def function():
    ...
```

Built-ins

Constants, functions, types living in the global namespace.

- `True`, `False`, `None`, `Ellipsis`
- standard types & their methods
- built-in functions

```
>>> divmod(44, 10)
(4, 4)
```

Built-ins

Constants, functions, types living in the global namespace.

- `True`, `False`, `None`, `Ellipsis`
- standard types & their methods
- built-in functions

```
>>> all(x > 5 for x in [6, 9, 100])
True

>>> any(x < 5 for x in [6, 9, 100])
False
```

the modules

turtle

Turtle graphics.

```
if len(self.currentLine) > 42: # 42! answer to the ultimate question
                               # of life, the universe and everything
    self._newLine()
```

cpython/Lib/turtle.py, line 3192

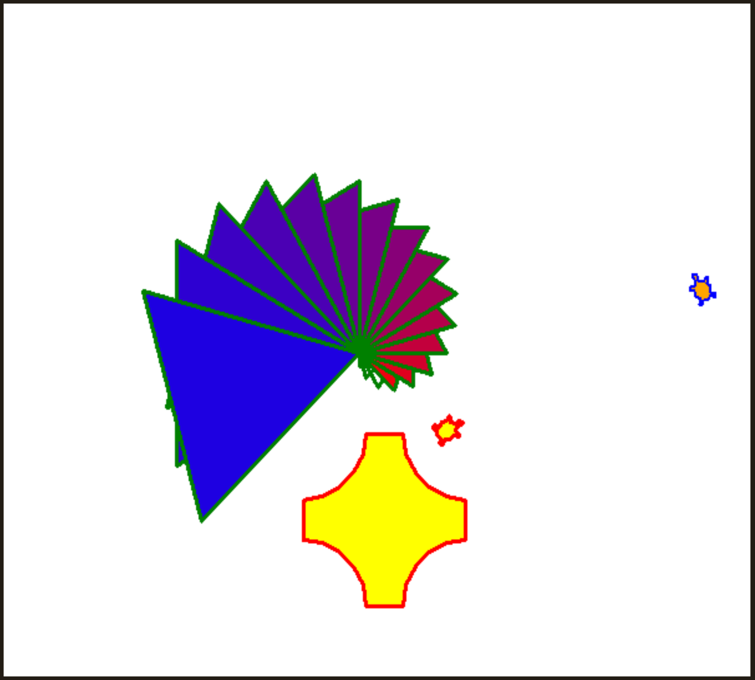
```
class TurtleGraphicsError(Exception):
    """Some TurtleGraphics Error
    """
```

cpython/Lib/turtle.py, line 866

```
if abs(self._position - new) > 0.5:
    print ("undogoto: HALLO-DA-STIMMT-WAS-NICHT!")
```

cpython/Lib/turtle.py, line 3204

```
$ python3 -m turtle
```



string

⋆ ~~ just string things ~~ ⋆

```
>>> import string

>>> string.ascii_lowercase
'abcdefghijklmnopqrstuvwxyz'

>>> string.printable
'0123456789abcdefghijklmnopqrstuvwxyzABCDEFGHIJK
LMNOPQRSTUVWXYZ!"#$%&\'()*+,-./:;<=>?
@[\\]^_`{|}~ \t\n\r\x0b\x0c'
```

2to3

Automatic Python 2 to 3 code translation.

```
name = raw_input()
print "I am {} the Mighty, a knight of great renown.
My weapon, 'Life Ender' has struck down hundreds of
foes. Begone, or you'll be next!".format(name)
```

```
$ 2to3 example.py
```

```
name = input()
print("I am {} the Mighty, a knight of great renown.
My weapon, 'Life Ender' has struck down hundreds of
foes. Begone, or you'll be next!".format(name))
```

textwrap

Paragraph & indentation helpers.

```
>>> intro = """\
...     A Slug captain hails and invites himself aboard
...     your ship to present a flask of something slimy.
...     "Now, most gracioussss captain, you must join me
...     please in a drink to our alliance!"
... """
```

```
>>> intro
'    A Slug captain hails and invites himself aboard\n
your ship to present a flask of something slimy.\n
"Now, most gracioussss captain, you must join me\n
please in a drink to our alliance!"\n'
```

```
>>> intro
```

```
    A Slug captain hails and invites himself aboard\n
    your ship to present a flask of something slimy.\n
    "Now, most gracioussss captain, you must join me\n
    please in a drink to our alliance!"\n
```

```
>>> import textwrap
>>> # Remove leading whitespace
>>> intro = textwrap.dedent(intro)
>>> intro
```

```
A Slug captain hails and invites himself aboard\n
your ship to present a flask of something slimy.\n
"Now, most gracioussss captain, you must join me\n
please in a drink to our alliance!"\n
```

```
>>> # Wrap to a specific number of characters
>>> textwrap.fill(intro, width=40)
```

```
A Slug captain hails and invites himself\n
aboard your ship to present a flask of\n
something slimy. Now, most gracioussss\n
captain, you must join me please in a\n
drink to our alliance!\n
```

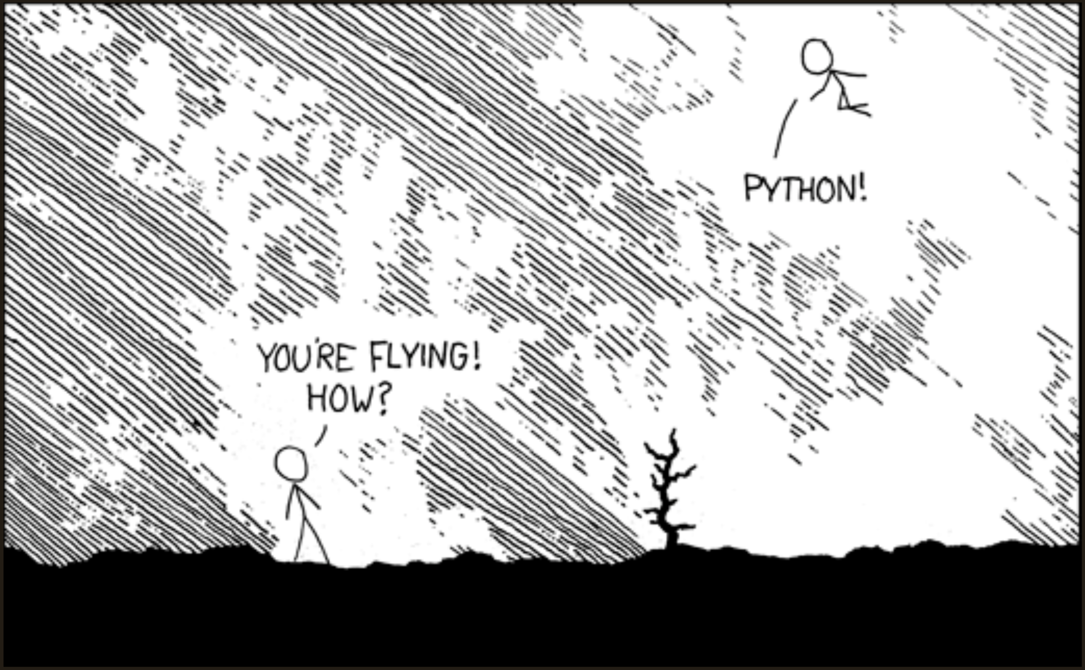
https://ftl.fandom.com/wiki/Races/Crew_Members

webbrowser

Your partner for all your browser-opening needs.

```
webbrowser.open("https://xkcd.com/353/")
```

cpython/Lib/antigravity.py, line 5



xkcd.com/353

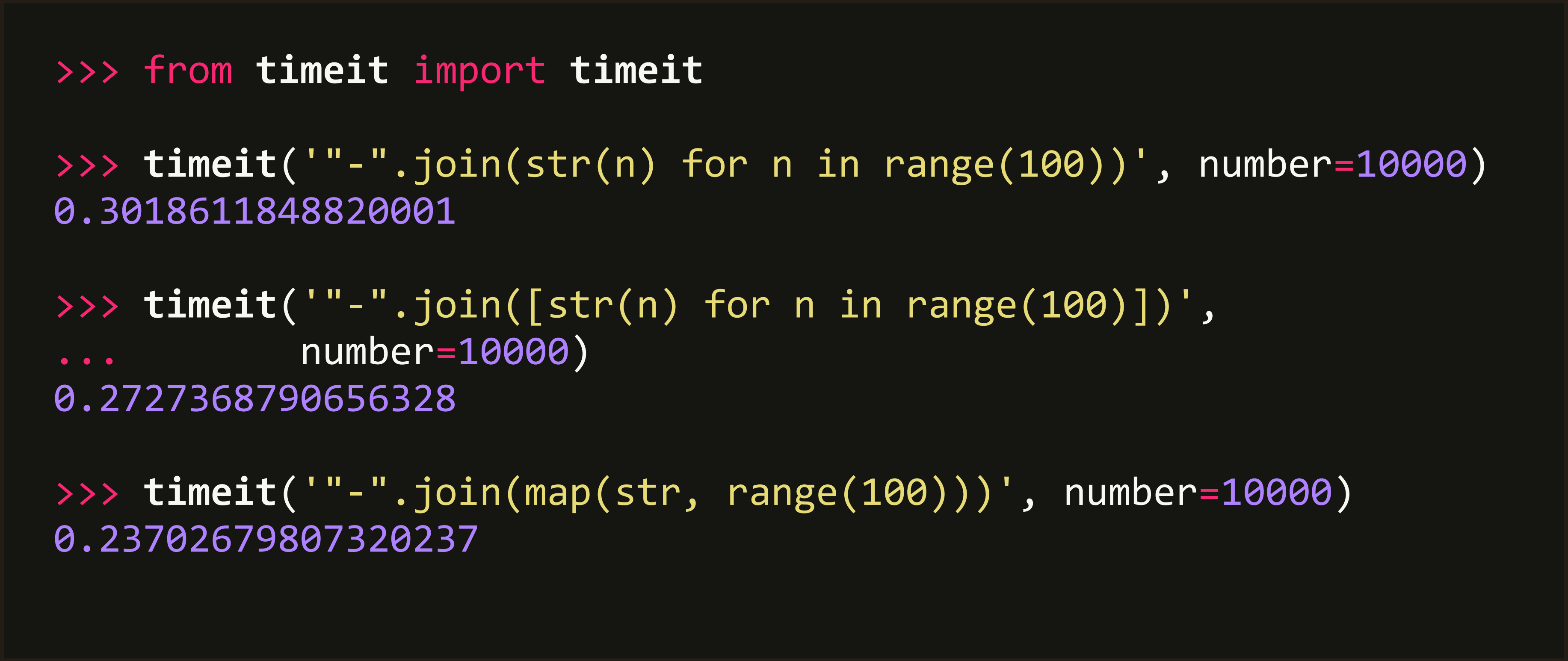
```
>>> import antigravity
```

```
>>> from antigravity import ???
```

timeit

...duh.

```
>>> # "0-1-2-3-4-...-99"

>>> "-".join(str(n) for n in range(100))

>>> "-".join([str(n) for n in range(100)])

>>> "-".join(map(str, range(100))
$ python3 -m timeit '"-".join(str(n) for n in range(100))'
10000 loops, best of 3: 30.2 usec per loop

$ python3 -m timeit '"-".join([str(n) for n in
range(100)])'
10000 loops, best of 3: 27.5 usec per loop

$ python3 -m timeit '"-".join(map(str, range(100)))'
10000 loops, best of 3: 23.2 usec per loop
```



```
>>> from timeit import timeit

>>> timeit('"-".join(str(n) for n in range(100))', number=10000)
0.3018611848820001

>>> timeit('"-".join([str(n) for n in range(100)])',
...        number=10000)
0.2727368790656328

>>> timeit('"-".join(map(str, range(100)))', number=10000)
0.23702679807320237
```

urllib

Working with URLs.

```
>>> from urllib.request import urlopen

>>> request = urlopen(
...     'https://pokeapi.co/api/v2/pokemon/umbreon/'
... )
```

Make sure to follow PokeApi's usage policy: https://pokeapi.co/docs/v2.html/

urllib

Working with URLs.

```
>>> from urllib.request import urlopen

>>> headers = {
...     'User-Agent': 'Python/urllib @ PyDays'
... }
>>> request = urlopen(
...     'https://pokeapi.co/api/v2/pokemon/umbreon/'
... )
>>> # ...I can has sends headers?
```

urllib

Working with URLs.

```
>>> from urllib.request import Request, urlopen

>>> headers = {
...     'User-Agent': 'Python/urllib @ PyDays'
... }
>>> request = Request(
...     'https://pokeapi.co/api/v2/pokemon/umbreon/',
...     headers=headers
... )
>>> umbreon = urlopen(request).read()
>>> # ...I can has JSON?
```

urllib

Working with URLs.

```
>>> import json
>>> from urllib.request import Request, urlopen

>>> headers = {
...     'User-Agent': 'Python/urllib @ PyDays'
... }
>>> request = Request(
...     'https://pokeapi.co/api/v2/pokemon/umbreon/',
...     headers=headers
... )
>>> umbreon = json.load(urlopen(request))
```

https://sonicpokemon.fandom.com/wiki/File:Glaceon.png

Maybe, just this once...

```
$ pip install requests
```

pprint

String representation for humans.

```
>>> import requests   # pip install requests
>>> umbreon = requests.get(
...    'https://pokeapi.co/api/v2/pokemon/umbreon/',
...    headers={'User-Agent': 'Python/urllib @ PyDays'}
... ).json()
```

pprint

String representation for humans.

```
>>> import requests   # pip install requests
>>> umbreon = requests.get(
...     'https://pokeapi.co/api/v2/pokemon/umbreon/',
...     headers={'User-Agent': 'Python/urllib @ PyDays'}
... ).json()
```

```
{'abilities': [{'ability': {'name': 'inner-focus', 'url': 'https://pokeapi.co/api/v2/abili
ty/39/'}, 'is_hidden': True, 'slot': 3}, {'ability': {'name': 'synchronize', 'url': 'https
://pokeapi.co/api/v2/ability/28/'}, 'is_hidden': False, 'slot': 1}], 'base_experience': 18
4, 'forms': [{'name': 'umbreon', 'url': 'https://pokeapi.co/api/v2/pokemon-form/197/'}], '
game_indices': [{'game_index': 197, 'version': {'name': 'white-2', 'url': 'https://pokeapi
.co/api/v2/version/22/'}}, {'game_index': 197, 'version': {'name': 'black-2', 'url': 'htt
ps://pokeapi.co/api/v2/version/21/'}}, {'game_index': 197, 'version': {'name': 'white', '
url': 'https://pokeapi.co/api/v2/version/18/'}}, {'game_index': 197, 'version': {'name': '
black', 'url': 'https://pokeapi.co/api/v2/version/17/'}}, {'game_index': 197, 'version': {
'name': 'soulsilver', 'url': 'https://pokeapi.co/api/v2/version/16/'}},

# ...and much more (261+ kB)
```

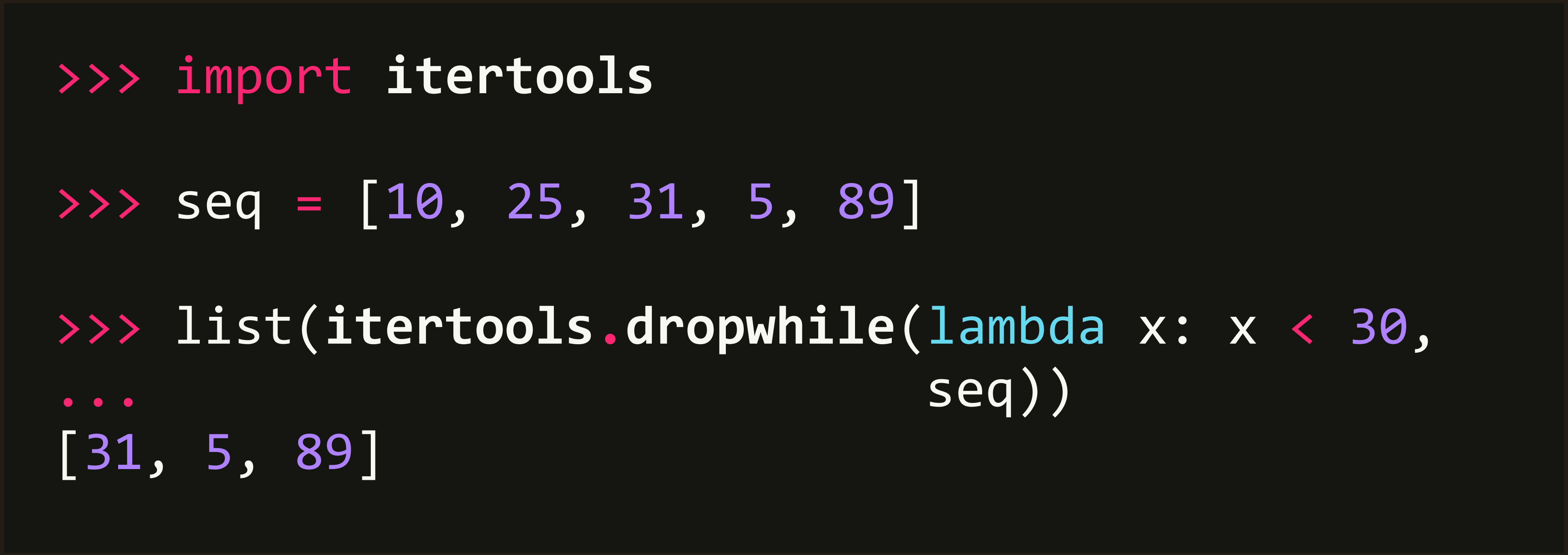
```
>>> from pprint import pprint
>>> pprint(umbreon)
```

```
{'abilities': [{'ability': {'name': 'inner-focus',
                            'url': 'https://pokeapi.co/api/v2/ability/39/'},
                'is_hidden': True,
                'slot': 3},
               {'ability': {'name': 'synchronize',
                            'url': 'https://pokeapi.co/api/v2/ability/28/'},
                'is_hidden': False,
                'slot': 1}],
 'base_experience': 184,
 'forms': [{'name': 'umbreon',
            'url': 'https://pokeapi.co/api/v2/pokemon-form/197/'}],
 'game_indices': [{'game_index': 197,
                   'version': {'name': 'white-2',
                               'url': 'https://pokeapi.co/api/v2/version/22/'}},

# ... and much more
```

itertools

Advanced iterators.

```
>>> import itertools

>>> seq = [[10], [25, 31, 5], [], [89]]

>>> list(itertools.chain.from_iterable(seq))
[10, 25, 31, 5, 89]
```



```
>>> import itertools

>>> seq = [10, 25, 31, 5, 89]

>>> list(itertools.dropwhile(lambda x: x < 30,
...                          seq))
[31, 5, 89]
```

```
>>> [''.join(p)
...  for p in itertools.permutations('abc', 2)]
['ab', 'ac', 'ba', 'bc', 'ca', 'cb']
```

functools

Higher order functions.

```
>>> import functools
>>> import math

>>> functools.reduce(
...     math.gcd,
...     [105, 21, 99, 10, 256]
... )
1
```

```
>>> print_word = functools.partial(print,
                                   end='')

>>> for i in range(3):
...     print_word('NYEH HEH HEH ')
NYEH HEH HEH NYEH HEH HEH NYEH HEH HEH
```

```
>>> import functools
>>> # Memoize all the things!
>>> @functools.lru_cache(maxsize=1024)
>>> def address(latitude, longitude):
...     address = expensive_geocoder(latitude,
...                                  longitude)
...     return address
```

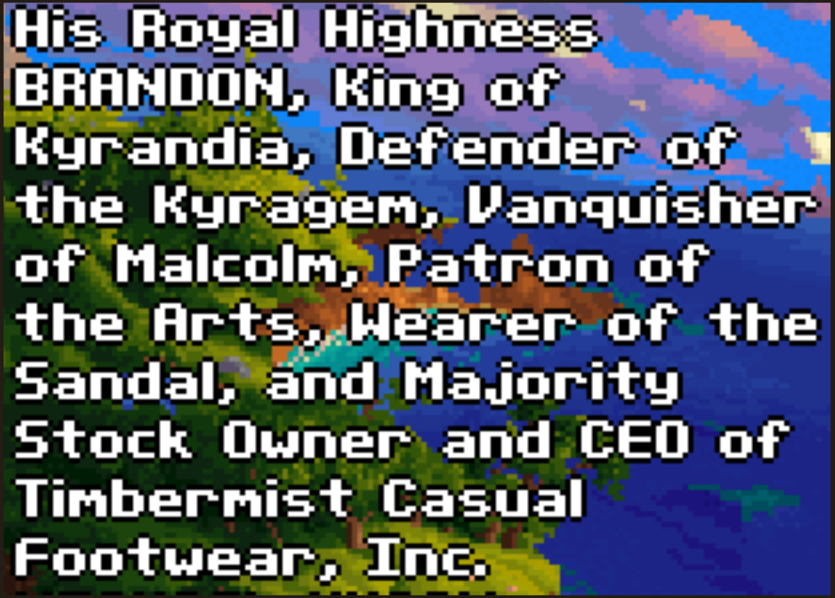
collections

Additional useful data structures.

```
>>> import collections
>>> inventory = ['aloe', 'salve', 'ginseng',
...              'ginseng', 'aloe', 'aloe']

>>> counter = collections.Counter(inventory)
>>> counter
Counter({'aloe': 3, 'ginseng': 2, 'salve': 1})

>>> counter['ginseng']
2
>>> counter['bezel cup']
0
```

```
>>> occupations = {'Zanthia': 'mystic',
...                'Malcolm': 'jester'}
>>> occupations['Brandon']
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
KeyError: 'Brandon'
```

```
>>> occupations = \
...     collections.defaultdict(lambda: 'n/a')
>>> occupations['Zanthia'] = 'mystic'
>>> occupations['Malcolm'] = 'jester'

>>> occupations['Brandon']
'n/a'
```



His Royal Highness
BRANDON, King of
Kyrandia, Defender of
the Kyragem, Vanquisher
of Malcolm, Patron of
the Arts, Wearer of the
Sandal, and Majority
Stock Owner and CEO of
Timbermist Casual
Footwear, Inc.

https://lparchive.org/Legend-of-Kyrandia-2-Hand-of-Fate/Update%2016/72-kyrandia2_update15_037.png

```
>>> occupations = \
...     collections.defaultdict(lambda: 'n/a')
>>> occupations['Zanthia'] = 'mystic'
>>> occupations['Malcolm'] = 'jester'

>>> occupations['Brandon']
'n/a'
```

bisect

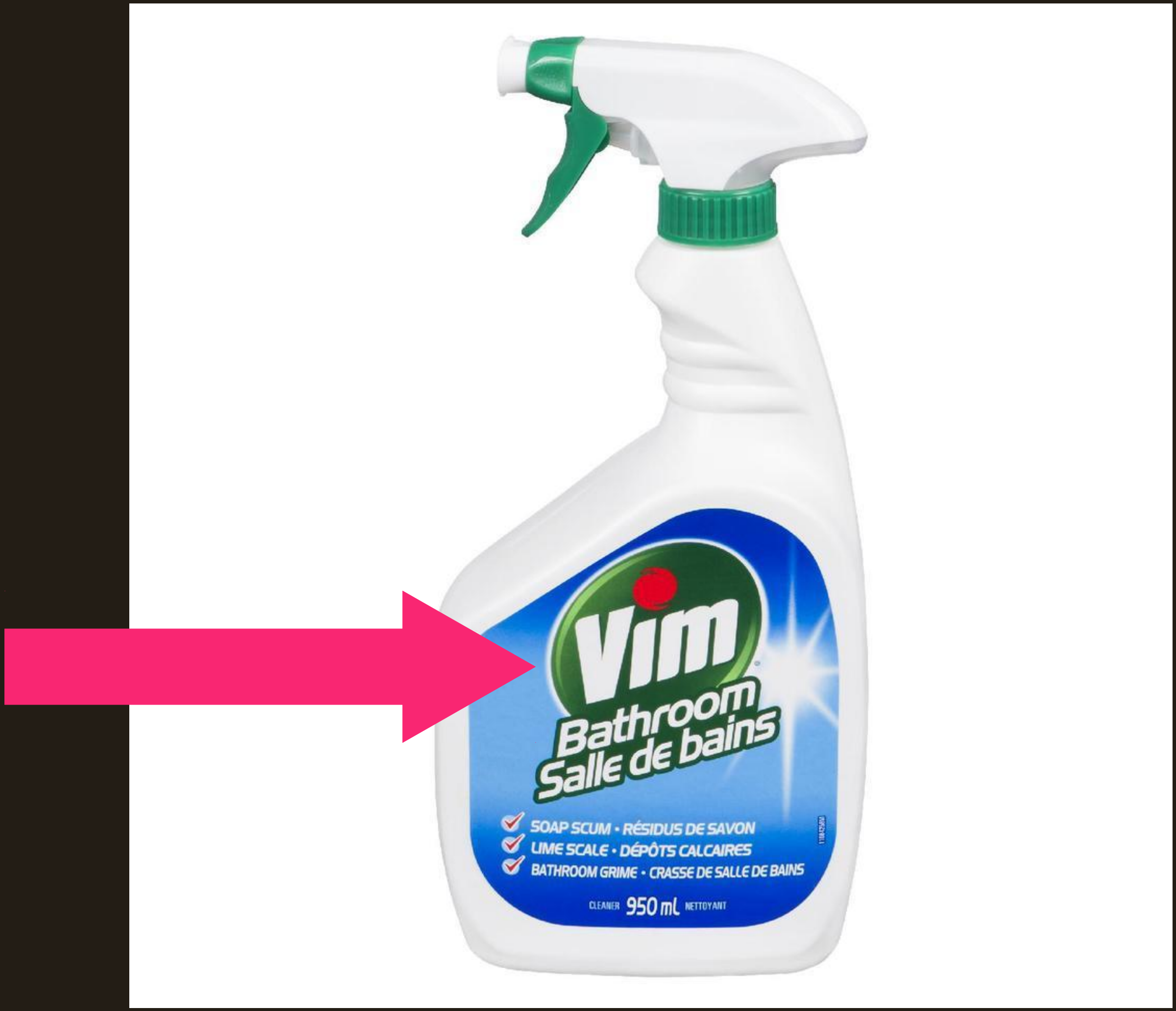
Efficient search and maintenance of sorted lists.

```
>>> import bisect

>>> seq = [1, 4, 7, 89, 123, 200, 345, 678]

>>> # Use binary search to find out where an
element is located:
>>> bisect.bisect_left(seq, 200)
5
>>> # Insert an element, keeping seq sorted:
>>> bisect.insort(seq, 470)
>>> seq
[1, 4, 7, 89, 123, 200, 345, 470, 678]
```

traceback

Stack traces you *like* to see.

```
>>> def frobnicate():
...     foobarify()

>>> def foobarify():
...     bazificate()

>>> def bazificate():
...     import traceback
...     traceback.print_stack()

>>> frobnicate()
File "<stdin>", line 1, in <module>
File "<stdin>", line 2, in frobnicate
File "<stdin>", line 2, in foobarify
File "<stdin>", line 3, in bazificate
```


Vim
Bathroom
Salle de bains
SOAP SCUM • RÉSIDUS DE SAVON
LIME SCALE • DÉPÔTS CALCAIRES
BATHROOM GRIME • CRASSE DE SALLE DE BAINS
CLEANER 950 mL NETTOYANT

https://images.chickadvisor.com/item/47673/original/285af65a5ddf144a4c59a224d69e9f6c.jpg

tabnanny

Check Python files for whitespace issues.

```
"""The Tab Nanny despises ambiguous indentation.
She knows no mercy.
```

cpython/Lib/tabnanny.py, line 3

```
>>> import tabnanny

>>> tabnanny.check('~/my_perfectly_indented_file.py')
```

```
thisguy = Whitespace(token)
if not indents[-1].less(thisguy):
    witness = indents[-1].not_less_witness(thisguy)
    msg = "indent not greater e.g. " + format_witnesses(witness)
    raise NannyNag(start[0], msg, line)
```

cpython/Lib/tabnanny.py, line 296

...and the list goes on.

difflib weakref pickle enum

asyncio email

argparse wave

multiprocessing array

os ipaddress colorsys

inspect csv

re

threading logging

...and the list goes on.

curses binhex

sys

tempfile copy

struct ast marshal base64

smtpd

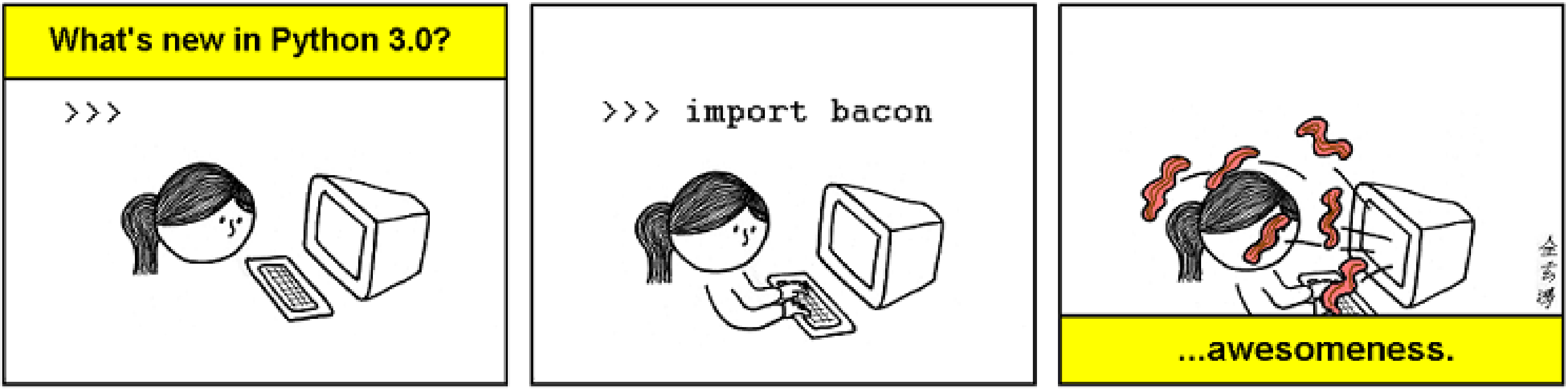
time signal

html dbm subprocess

numbers platform

gettext

The Python standard library
can do almost anything.



abstrusegoose.com/81

```
import bacon     # ???
```

For the rest, there is the Python Package Index.

```
pip install baconify
```

Thank you!

Ivana Kellyerova

🐦 @aerinthenniel