Lesser known gems of the standard library

The No. 6 and A23s of "batteries included"

Kenneth Nielsen[1]

[1] Center for Individual Nanoparticle Functionality (CINF)
Institute of Physics
Technical University of Denmark (DTU)

Python and beers presentation,
Aprils 29, 2015

About me

- My name is Kenneth Nielsen and I work at CINF
- Free and open source software (FOSS) enthusiast
- Use way too much time on python conference presentations on youtube

Slides and examples

```
git clone https://github.com/KennethNielsen/presentations.git
```

https://github.com/KennethNielsen/presentations
http://bit.ly/1FclzDR

Outline

A peak into some of the (maybe) lesser known gems of the standard library

- Convenient containers (... and more)
  - defaultdict, Counter
- Special operators for key functions (get rid of lambdas)
  - itemgetter, attrgetter
- Function tools
  - partial
- Part way between anonymous data and full classes
  - namedtuple
- Random stuff ;)
  - choice

defaultdict

```
from collections import defaultdict
groups = defaultdict(default_callable)
```

> A dictionary that will fill in values for unknown keys with defaults supplied by a callable

- Useful e.g. for grouping, where you would otherwise have to write annoying first-item-init code
- Supply a callable when creating it, that will be called to ask for a default value

LIVE! Example ex1_default_dict.py

Counter

```python
from collections import Counter
counter = Counter()
```

A default dict with default value 0 (and some special counter relevant math)

- Useful for counting!
- Can be added and subtracted
- Flattens out at 0 when subtracted

`LIVE! Example ex2_counter.py`

itemgetter and attrgetter

```
from operator import itemgetter, attrgetter
```

Gives you a function that returns an element from an iterable or dict or a specific attribute from an object

- EXTREMELY usefull as `key` functions for `sort`, `sorted`, `min` and `max` ...
- ... especially because they make it possible to get rid of those pesky unreadable lambda expressions

```
LIVE! Example ex3_itemgetter_attrgetter.py
```

partial

```
from functools import partial
```

From original functions create new ones, where certain values in the original functions calls are already filled out

- To call the same function a lot of times with one spcific argument
- To create custom functions that calls others with an argument filled out

`LIVE! Example ex4_partial.py`

`LIVE! Example ex5_partial_valve_control.py`

namedtuple

```
from collections import namedtuple
```

Named tuples can be used to give your tuple like data structures a name and make it possible to access its values by names instead of index WITHOUT writing a class by hand for it

- Usefull for naming data structures
- Usefull for getting . access instead of []

```
LIVE! Example ex6_namedtuple.py
```

choise

```
from random import choice
```

Pick an element from an iterable at random.

LIVE! Example ex7_choice.py

Summary

- The standard library has lots of little tools
- It is worth it to look for them, when needing them instead of rolling your own