Imperative programming with Python

January 2012 project: Class #9

Facundo Carreiro

ILLC, University of Amsterdam

January 25th, 2012

Today's agenda

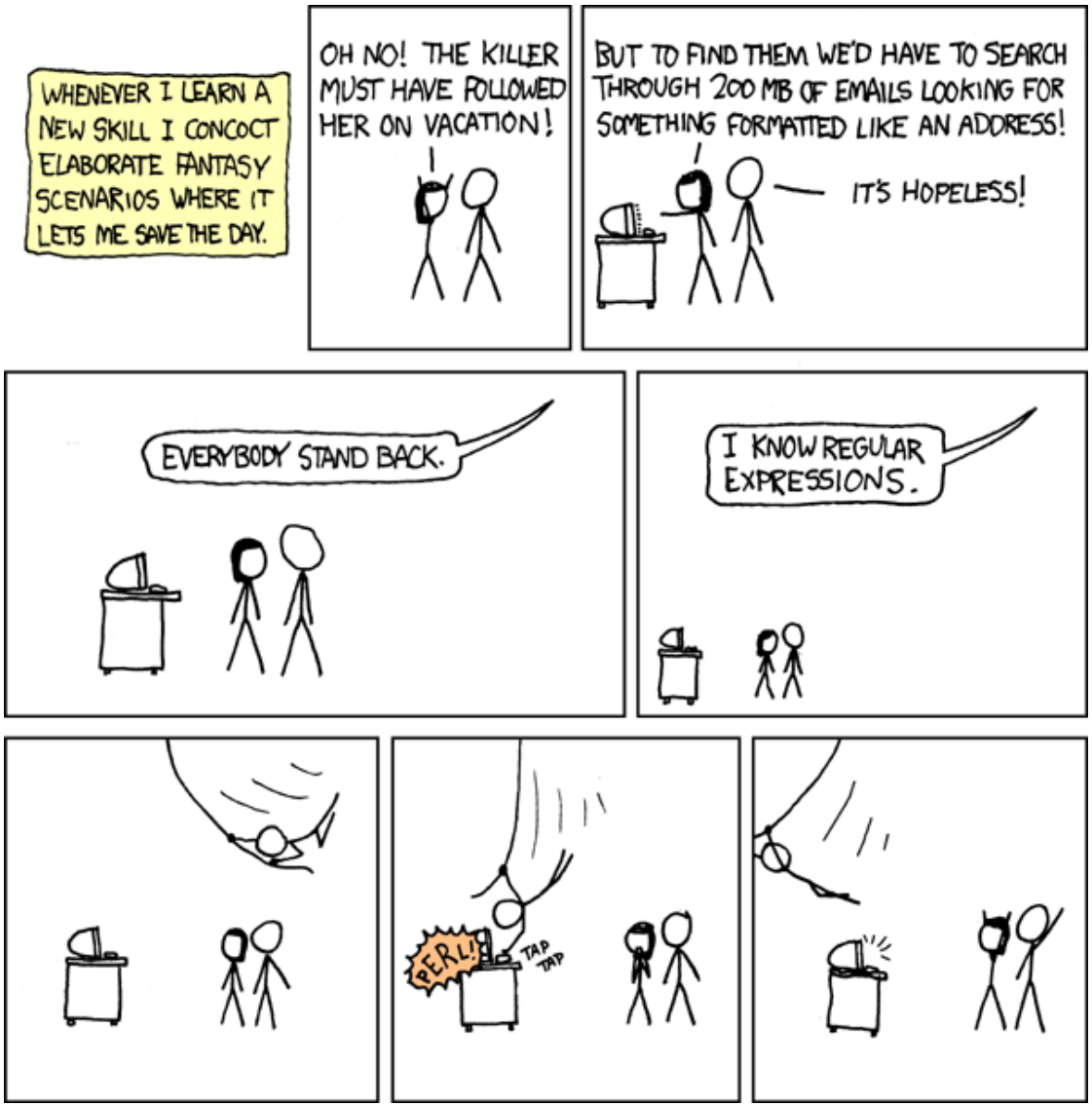
1. Regular expressions
2. Evaluation forms
3. How to write modules
4. Groups and final projects

Ready, set. . .

(read Beckles & Downling's slides or the subset I picked)

What to do with what we've learnt?

By the XKCD webcomic



Evaluation forms

Please fill them

(while you take a break)

Creating modules

In the (small) AH example we had a lot of classes

- `System` → `system.py`
- `Product` → `product.py`
- `Discount` → `discount.py`
- `TwoForOneDiscount` → `discount.py`
- `PercentageDiscount` → `discount.py`
- `Reader` → `reader.py`
- `BuyingContext` → `system.py/bc.py`? depends on size
- `...`

it would be better organized in *Modules* (files!).

Using our modules

We save all files in the same directory and use them as follows

```
# in product.py
class Product(object):
    def __init__(self,...):
        ...
```

```
# in discount.py
class Discount(object):
    ...
class TwoForOneDiscount(Discount):
    ...
class PercentageDiscount(Discount):
    ...
```

```
# in system.py
import product, discount          # among others
from reader import Reader         # among others

class System(object):
    def loadProductsFromFile(self):
        ...
        self.product_list[barcode] = product.Product(datafromfile)
```

Groups and final project

Time to talk with your classmates!

(come on, don't be afraid)

(I mean it...)

References

- UCS course on Python Regular Expressions
  http://www.ucs.cam.ac.uk/docs/course-notes/unix-courses/PythonRE/
- Python `re` module
  http://docs.python.org/library/re.html
- Python regular expressions HOWTO
  http://docs.python.org/howto/regex.html#regex-howto
- `regexp.info`
  http://regexp.info
- Python RegEx Tool
  http://www.pythonregex.com/