PV248 Python

Petr Ročkai

Disclaimer

- I am not a Python programmer
- please don't ask sneaky language-lawyer questions

Goals

- let's learn to use Python in practical situations
- have a look at existing packages and what they can do for us
- code up some cool stuff & have fun

Organisation

- I'm in India next Monday, Mr. Kaplan will come instead
- starting 9th of Oct, we can start at 8:30 (let's have a vote)

Stuff We Could Try

- working with text, regular expressions
- using the `pdb` debugger
- plotting stuff with bokeh (`https://bokeh.pydata.org`)
- talking to SQL databases
- talking to HTTP servers
- being an HTTP server
- implementing a JSON-based REST API
- parsing YAML and/or JSON data
- … (suggestions welcome)

Some Resources

- https://docs.python.org/3/ (obviously)
- https://github.com/VerosK/python-pv248
- https://msivak.fedorapeople.org/python/
- study materials in IS
- ...

Part 1: Text & Regular Expressions

Reading Input

- opening files: `open('scorelib.txt', 'r')`
- files can be iterated

```
f = open( 'scorelib.txt', 'r' )
for line in f:
    print line
```

Regular Expressions

- compiling: `r = re.compile( r"Composer: (.*)" )`
- matching: `m = r.match( "Composer: Bach, J. S." )`
- extracting captures: `print m.group(1)`
  - prints `Bach, J. S.`
- substitutions: `s2 = re.sub( r"\s*$", '', s1 )`
  - strips all trailing whitespace in `s1`

Other String Operations

- better whitespace stripping: `s2 = s1.strip()`
- splitting: `str.split(';')`

Dictionaries

- associative arrays: map (e.g.) strings to numbers
- nice syntax: `dict = { 'foo': 1, 'bar': 3 }`
- nice & easy to work with
- can be iterated: `for k, v in dict.items()`

Counters

- `from collections import Counter`
- like a dictionary, but the default value is 0
- `ctr = Counter()`
- compare `ctr['baz'] += 1` with `dict`

Exercise 1: Input

- get yourself a git/mercurial/darcs repository
- grab input data (`scorelib.txt`) from study materials
- read and process the text file
- use regular expressions to extract data
- use dictionaries to collect stats
- beware! hand-written, somewhat irregular data

Exercise 1: Output

- print some interesting statistics
  - how many pieces by each composer?
  - how many pieces composed in a given century?
  - how many in the key of c minor?
- bonus if you are bored: searching
  - list all pieces in a given key
  - list pieces featuring a given instrument (say, bassoon)

Exercise 1: Example Output

- Telemann, G. P.: 68
- Bach, J. S.: 79
- Bach, J. C.: 6
- …

For centuries:

- 16th century: 10
- 17th century: 33
- 18th century: 4

Cheat Sheet

`for line in open('file', 'r')`	read lines
`dict = {}`	an empty dictionary
`dict[key] = value`	set a value in a dictionary
`r = re.compile(r"(.*):")`	compile a regexp
`m = r.match("foo: bar")`	match a string
`if m is None: continue`	match failed, loop again
`print m.group(1)`	extract a capture
`for k, v in dict.items()`	iterate a dictionary
`print "%d, %d" % (12, 1337)`	print some numbers