Python programming — course introduction

Finn Årup Nielsen

Department of Informatics and Mathematical Modelling
Technical University of Denmark
Lundbeck Foundation Center for Integrated Molecular Brain Imaging
Neurobiology Research Unit,
Copenhagen University Hospital Rigshospitalet

August 31, 2009
Python programming

DTU course 02820 Python programming

Project course with a few introductory lectures, but mostly self-taught.

Deliverables: A report, a poster and an oral presentation at the poster about a Python program you write in a group.

Autumn 2009 first run for the course: The course is somewhat experimental.

Teachers: Bart Wilkowski, Marcin Szewczyk, Finn Årup Nielsen
Schedule for autumn 2009

31. August. Introduction to Python. (Finn)

7. September. Installation party. (Marcin)

14. September. “Python as Matlab”: NumPy, SciPy, MatPlot (Finn, Marcin)

21. September. Web and text processing, “natural language processing” (Finn, Marcin, Bart)

5. October. Mobile Python. Use on Nokia smartphones (Bart)

(19. October Machine learning with Python)
Other courses

Introductory programming and mathematical modelling (linear algebra, statistics, machine learning)

DTU 02827: Mobile Application Prototype Development. Also with Python.

Poster

Construct a poster.

“Defend” the poster, i.e., give a relatively short oral presentation of the poster and answer questions.

Inspired from DTU course 02459 Machine Learning for Signal Processing
Why Python?

Interpreted, readable (usually clearer than Perl), interactive, many libraries, runs on many platforms, e.g., Nokia smartphones and Apache Web servers.

With Python one can construct numerical programs, though with a bit more boilerplate than Matlab.

Google and Yahoo! is (has been?) using it. 2.73% of Open Source code written in Python (Black Duck Software, 2009).

“Without [Python] a project the size of Star Wars: Episode II would have been very difficult to pull off.” — http://python.org/about/quotes/

XKCD 353: “I wrote 20 short programs in Python yesterday. It was wonderful. Perl I’m leaving you.”
Why Python? Interactive language!

Interactive session

$ python
Python 2.4.4 (#2, Oct 22 2008, 19:52:44)
[GCC 4.1.2 20061115 (prerelease) (Debian 4.1.1-21)] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>> 1+1
2

However, Matlab-like computation is not straightforward, e.g., what is the result of

>>> 1/2
Why Python? Interactive language!

Interactive session

$ python
Python 2.4.4 (#2, Oct 22 2008, 19:52:44)
[GCC 4.1.2 20061115 (prerelease) (Debian 4.1.1-21)] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>> 1+1
2

However, Matlab-like computation is not straightforward, e.g., what is the result of

>>> 1/2
0  # Integer division!
Example projects for inspiration

1. Characterize external links from DTU’s Web-site.

2. Characterize internal link structure on DTU.


4. Sentiment analysis of Tweets.

5. Sentiment analysis of blogs.

6. A wiki-based database for brain activations

8. Sound segmentation
More information

Google, Bing or Yahoo. http://www.python.org/

Google for error messages, “Python tutorial”

Books

*Python Essential References* (Beazley, 2000): Introduction and list of Python functions with small examples. Somewhat old

*Python cookbook* (Martelli et al., 2005): Short program examples for somewhat specific problems.

*Practical Programming. An introduction to computer science using Python*, (Campbell et al., 2009) Introductory programming.
Specialized books relevant for the course

*Python scripting for computational science* (Langtangen, 2005): Python book with many examples especially for numerical processing. Not fully up to date on numerical Python.


*Natural language processing with Python* (Bird et al., 2009): Text mining with Python.

*Mobile Python* (Scheible and Tuulos, 2007): Mobile Python.

Other O’Reilly titles: Python in a Nutshell, Python Pocket Reference, Learning Python, Programming Python?
Example: A fielded wiki . . .

Web script in Python implementing a fielded wiki for personality genetics.

Persistence with a small SQLite database.

Some of the Python libraries used: cgi, Cookie, math, pysqlite2, scipy, sha.

One Python script with 2269 lines of code.
Computation of effect sizes (a statistical value) and comparison to statistical distributions.

Generation of interactive and hyperlinked plots in SVG (an XML format)
Structured information from Wikipedia

Get Wikipedia pages that contain a specific template, download the page, extract information from the templates and render the result on an HTML page.

Python libraries: json, re, urlib2

Around 25 Python lines to get the data, and around 120 to render the result.
Web script for Twitter annotation

CGI program that searches Twitter with a user-defined query, obtain tweets and present them in a Web form for manual annotation and stores the result in a SQL database.

Python libraries: codecs, json, re, cgi, urllib2, pysqlite2, xml.

500 Python lines.
Searching brain coordinates

CGI script to search the “Brede Database” for coordinates.

Reading of XML files, adding data to SQLite database, search database given a user query and generate an HTML Web page.

700 lines of Perl code, but could have been written in Python.
Brede Database.

http://muzeeker.com/
References


