Basic Programming in Python: Course Outline

Sebastian Höffner Aline Vilks

Tue, 13 June 2017

Note that the schedule presented below is subject to change, depending on how far we get. We will try to keep it up to date so that you can always take a quick look at what we covered.

Week of	Name	Topics
2017-04-03	`print('Hello World!')`	Organization; Motivation; Installation of Python; Getting started with Hello World programs
2017-04-10	Variables, Assignments, Functions	Basic data types: strings, integers, floats; Arithmetics; Concept of variables and assignments (in comparison to mathematical equality); Functions
2017-04-17	`True` or `False`?	Boolean variables; Comparisons; `If/Elif/Else`; Control Flow: `While/For`
2017-04-24	Collections and File I/O	Reading and writing files; Lists/Dictionaries/Tuples/Sets
2017-05-01	Errors and Debugging	try/except; Error types; Spyder's debugging functions; Documentation and style; Tests
2017-05-08	Python Packages	Using other packages; Complex programs; Python modules
2017-05-15	Sorting and OOP	Sorting; Classes and Instances
2017-05-22	Practical Python	Built-ins; Python Standard Library; string formats
2017-05-29	Errors and Finite State Machines	try/except/raise, finite state machines
2017-06-05	Regular Expressions / HTTP	`re` module, short intro to HTTP verbs, `requests.get`
2017-06-12	Dates, Times, Documentation	datetime, timezones, leapseconds, sphinx

Week of	Name	Topics
2017-06-19	Project week 1	Projects can be e.g. vocabulary trainers, simple games, esoteric language interpreters, image processing, applications from other subjects, ...
2017-06-26	Project week 2	Continuation of previous week
2017-07-03	What next?	Project presentations; future applications

Organizational issues

- There are no grades for this class, only a Fail/Pass option.
- To pass, work successful on nine (out of 13) homework sheets.
- The final projects will span approximately three of the homework sheets (that means they are included, not extra!)

Additional Material and Advice to learn Python

We did not learn Python with a book or a specific online tutorial, so it would be foolish to recommend one. But there are many out there when you search the web for "learning python", "python tutorials", etc. One word of advice though: Many tutorials are a little bit older and use Python 2, while we teach Python 3. There are not too many differences you should be concerned about, but these are important:

- `/` works slightly different on integers
- `print` needs no parentheses and has different arguments
- `import ... from __future__` is often used to get new language features of Python 3 in Python 2

The best way to learn Python (and programming in general) is to just do something. Try to solve a problem of your own, and when you can not solve it, try to figure out what blocks you and search the web for a solution. Websites like the Python documentation or StackOverflow[1] will usually have an answer. If not: Why don't you ask a question in class (or post one on the internet)?

[1] https://stackoverflow.com/

References

These are all references used throughout the class. The list will grow from time to time.

Cham, Jorge. 2007. “Master of the Universe.” *PhD – Piled Higher and Deeper*, November.

Climo, Liz. 2014. “Happy Father’s Day.” *Hi, I’m Liz*, June.

Darwin, Charles. 1859. *On the Origin of Species.*

Evans, Julia. 2016. “How I Got Better at Debugging.” *Julia’s Drawings*, November.

Fielding, R., J. Gettys, J. Mogul, H. Frystyk, L. Masinter, P. Leach, and T. Berners-Lee. 1999. “Hypertext Transfer Protocol – Http/1.1.” *Request for Comments*, June.

Goose, Abstruse. 2017. “How to Teach Yourself Programming.” *Abstruse Goose*, no. 249 (April).

Marciuliano, Francesco, and Jim Keefe. 2013. “Maze.” *Sally Forth*, October.

Munroe, Randall. 2009. “Tech Support Cheat Sheet.” *Xkcd. A Webcomic of Romance, Sarcasm, Math, and Language.*, no. 627 (August).

———. 2011. “Turtles.” *Xkcd. A Webcomic of Romance, Sarcasm, Math, and Language.*, no. 889 (April).

———. 2013. “ISO 8601.” *Xkcd. A Webcomic of Romance, Sarcasm, Math, and Language.*, no. 1179 (February).

———. 2014. “Future Self.” *Xkcd. A Webcomic of Romance, Sarcasm, Math, and Language.*, no. 1421 (September).

Python Software Foundation. 2017. *Python 3.6.0 Documentation.* 3.6.0 ed. Beaverton, Oregon, USA: Python Software Foundation.

Sadasivam, Krishna M. 2012. “Loopy de Loop.” *The PC Weenies*, January.

Sawyer, Ryan. 2014. “A Comic I Did for @Webs Is up!” *Twitter @EightballArt*, September.

Shakespeare, William. 1623. *Romeo and Juliet.*

The Behemoth. 2012. “Screenshot 7.” *Castle Crashers.*

Wikipedia contributors. 2017. “Regular Language.” *Wikipedia, the Free Encyclopedia*, May.

Wirth, Niklaus. 2006. “Good Ideas, Through the Looking Glass.” *IEEE Computer* 39 (1): 28–39.

Zauron. 2008. “Character Guide.” *GameFAQs: Castle Crashers*, September.