

Introduction to Python for Scientific Computing

Money, Banking and Payment Systems
Chair of Economic Theory

Resources

- ▶ QuantEcon lecture site:
<http://lectures.quantecon.org/>
- ▶ Scipy lecture notes:
<http://www.scipy-lectures.org/>
- ▶ Scipy cookbook:
<http://scipy-cookbook.readthedocs.io/>
- ▶ <https://www.reddit.com/r/Python/>
- ▶ <http://stackoverflow.com/questions/tagged/python>

3 tutorial sessions planned:

1. Introduction to Python
2. Simple economic applications in Python
3. Basic OLG model in Python

Tutorial notebooks will be available on the course web page before class.

Graded group assignment:

- ▶ Program one of the extensions of the OLG model seen in the lectures;
- ▶ The submitted notebook should be self-contained (code, math, explanation, results, plots).

Assignment details TBA on the course web page.

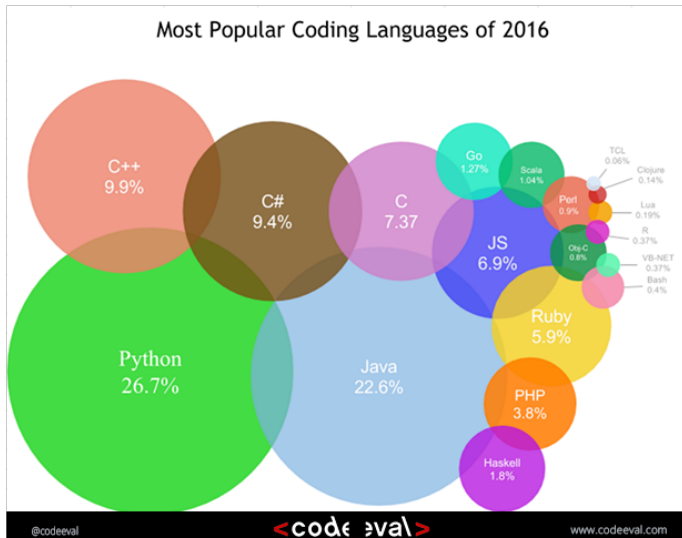
About Python

Modern, high level, free and open source, general purpose programming language.

Used extensively by:

- ▶ Tech firms (e.g. YouTube, Dropbox, Reddit);
- ▶ Finance industry (e.g. hedge funds);
- ▶ Research agencies (e.g. NASA, CERN);
- ▶ Academia

Why Python?



Why Python?

Python is a free and open source programming language:

- ▶ **Free** as in freedom (libre);
- ▶ **Free** as in "free beer" (gratis).

This means:

- ▶ Free to install and use;
- ▶ No license issues;
- ▶ Source code can be freely read, modified and shared.

Why Python?

- ▶ Simple to learn;
- ▶ Clean, elegant and very readable syntax;
- ▶ High productivity;
- ▶ Vast collection of libraries for almost everything;
- ▶ Powerful enough for scientific computing;
- ▶ Relatively simple tweaks offer performance comparable to compiled languages (C, Fortran).

Major Scientific Libraries

- ▶ **numpy**: basic data types, array operations.
- ▶ **scipy**: high-level numerical routines (e.g. integration, interpolation, optimization).
- ▶ **matplotlib**: plotting 2D and 3D figures.
- ▶ **sympy**: symbolic math computations (similar to Maple/Mathematica).
- ▶ **pandas**: data analysis.
- ▶ **statsmodels**: statistics and econometrics.
- ▶ **scikitlearn**: machine learning.
- ▶ **numba**: just-in-time compilation for higher performance.

Getting started

It is strongly recommended to install one of the many Python distributions (e.g. Anaconda, Canopy, WinPython) and to choose a good programming interface (e.g. Jupyter, Vim, Spyder, PyCharm).

For this class, we will be using:

- ▶ Anaconda with Python 3.x;
- ▶ Jupyter (iPython) notebook.

Most popular scientific Python distribution!

Installation:

- ▶ Available at: <https://www.continuum.io/downloads>
- ▶ Choose Python 3.6 version;
- ▶ For details, please refer to installation guide on course website.

Jupyter Notebooks

For the tutorials, we will use Jupyter notebooks:

- ▶ Browser based front-end for over 40 programming languages (e.g. Python, R, Julia, C++);
- ▶ Allows for live code, equations, visualizations and explanatory text.

Jupyter is included in Anaconda:

- ▶ Command line: `jupyter notebook`