PYTHON FOR NUMERICAL COMPUTING

R06945037 曾文緯

LET'S INSTALL PYTHON

- https://www.anaconda.com/download/
  - Choose the Python 3.6 64-bit one
- ???
- Profit!

THANK YOU

WELL, NOT YET!

ABOUT TA-01

林祐德

qwerty239qwe@gmail.com

禮拜二 下午 1:30~4:30

明達705

ABOUT TA-02

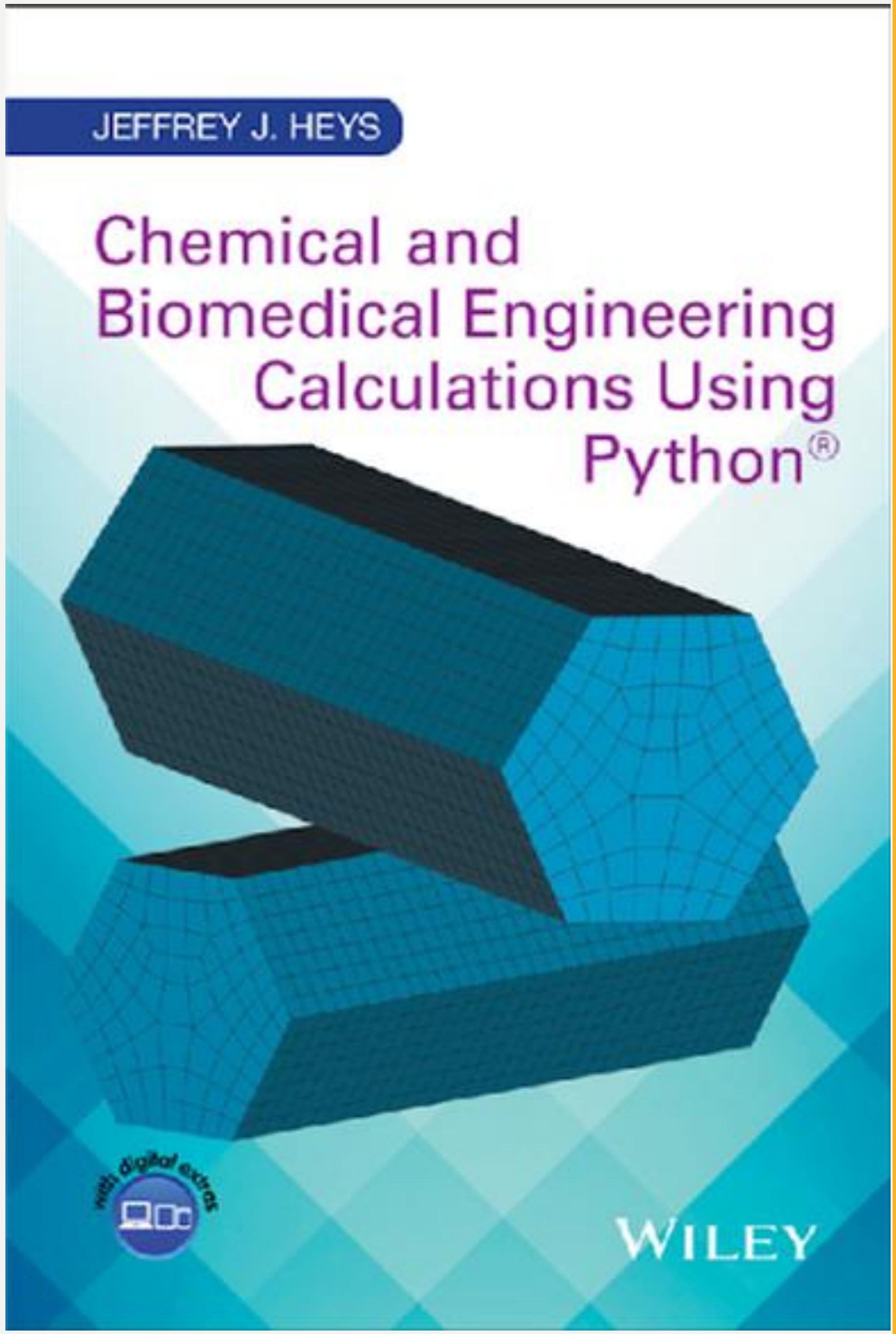
- 曾文緯 (Tseng Wen-Wei)
- First grade MS student in BEBI
- Previously graduated from the department of medicine
- A nerd who wants to change himself and pursues virtue rather than merit
- Doing numerical simulations of cardiomyocytes
  - And trying to make it faster
- Writes in C++ initially; then in Python

WHY PYTHON ?

- Because 3 of my colleagues are using it!
- Because it’s free and open-source
- ~~Because it’s the first class programming language for machine learning~~
- Development speed is much faster than in C++
  - Rich library (modules) support
  - Responsiveness and quick feedback
  - Less boilerplate code

MY REFERENCES

- https://www.python-course.eu/
- http://www.scipy-lectures.org/
- https://barbagroup.github.io/essential_skills_RRC/
- https://automatetheboringstuff.com/
- https://jakevdp.github.io/PythonDataScienceHandbook/
- https://www.machinelearningplus.com/101-numpy-exercises-python/
- There are tons of online courses about python
  - Edx, Udacity, Coursera, This 10-hour one


JEFFREY J. HEYS

Chemical and Biomedical Engineering Calculations Using Python®

with digital extras

WILEY

SOME SELF-ANSWERED QUESTIONS

- Isn't Python slower than language X?
  - Yes, thus it calls for help (eg. Numpy & Scipy: C/FORTRAN code inside)
  - You can even 'compile' Python code to other languages
- Are you going to cover some popular things in Python like web-scrapping, automation, data science, and/or machine learning?
  - Sorry, I'll focus on numerical computations only. Some important aspects may not be covered:
    - Defining a class, inheritance (OOP), doing the animal example
    - Inputs from keyboard, building a GUI, creating a game
    - Making a website, downloading from the internet, auto-pressing the like button
    - String/Voice/Image processing/classification, regular expression
    - Advanced topics such as generators, decorators, coroutines
- You're not going to be an MD, are you mad?

SOME NOTES

- Google before DIY
  - DRY (Don’t repeat yourself)
- Fail early, fail fast
  - Test your code little by little
- Read the FINE manual
  - And make clear ones in your code
- Name-Reference-Object rule
  - Less surprises later if you know this

SELECTIONS OF EDITORS / IDES

- Jupyter Notebook: both of us
  - Great for demos
- If you have questions about the IDEs below,
  - Spyder (unfortunately unfunded): Wen-Wei
  - Pycharm: Yuter
  - VS Code: Wen-Wei

LET’S START CODING (I)

- Variable assignment & printing
  - Data type
- Simple arithmetic
  - Floor division (//) vs true division (/)
  - Exponential
  - Long integer
  - Logic
- If /else statements
  - Indentation

LET'S START CODING (II)

YOU'RE MY SON AND NO MATTER WHAT, YOUR MOTHER AND I WILL ALWAYS LOVE YOU.
ARRAYS START AT 1
FREE DISAPPOINTMENT

- Sequential objects
  - Tuple, string (immutable)
  - List (mutable)
  - Selecting elements: Indexing and slicing
  - Functions for them
  - Concatenate, length, is inside (in)
- Associative container: dictionary
- For loops
- While loops
- List comprehension (9*9)

LET'S START CODING (III)

- Defining functions
  - Parameters
    - Positional
    - Keyword
    - Variable
- Side effects
- Functions are also objects
- Lambda: miniature function

NUMPY INTRO(I)

- Convention : import numpy as np
- Array-based operations, **element-wise**
- Array properties
  - Shape
  - Data type
- Array creation
  - From existing sequence
  - Routines

NUMPY INTRO(II)

- Element selection
  - Single
  - Slices
  - Boolean array
  - Index number
- Array manipulation
  - Reshape
  - Concat
  - Transpose
  - Reduction (sum, cumsum, min, max)

NUMPY INTRO(III)

- Linear algebra
  - Dot product
  - Matrix product

SCIPY INTRO: IVP