## Introduction to (or review of) Python

**Include a picture of yourself on CSNS.** Please be sure it’s one that lets me recognize you. Your GET picture is probably a good one to use. To add a picture: go to your profile, click edit (at the right on the gray bar with your name), scroll to the bottom, and add the picture.

**Install Python** on your personal computer. Use either [Anaconda](https://www.anaconda.com/download/) (recommended) or the [Python download](https://www.python.org/downloads/).

**Examine various IDE’s and select one you like.** (See the recommendations under resources. For now, I’ve settled on PyCharm. But it has quite a few frustrating features.) The simplest option—and many people think it’s the best—is to use an editor that supports Python (such as [Sublime Text](https://www.sublimetext.com/) or [Notepad++](https://notepad-plus-plus.org/)) and execute your code from the command line.

**Read**:

[Object-oriented programming in Python](http://python-textbok.readthedocs.io/en/1.0/index.html) (recommended), especially [Collections](http://python-textbok.readthedocs.io/en/1.0/Collections.html), [Classes](http://python-textbok.readthedocs.io/en/1.0/Classes.html), and [Object-oriented programming](http://python-textbok.readthedocs.io/en/1.0/Object_Oriented_Programming.html). Do the exercises in these sections and review the answers.

*or*

[Python tutorial](https://docs.python.org/3.5/tutorial/index.html), especially [Sec 5. Data Structures](https://docs.python.org/3/tutorial/datastructures.html) and [Sec 9. Classes](https://docs.python.org/3/tutorial/classes.html).

*or*

[Python Numpy tutorial](http://cs231n.github.io/python-numpy-tutorial/)

**Do the following programming exercises:** (from two sources): [List comprehensions](https://www.practicepython.org/exercise/2014/03/19/07-list-comprehensions.html), [Cows and bulls](https://www.practicepython.org/exercise/2014/07/05/18-cows-and-bulls.html) (The website has a number of mistakes, including getting the names *bulls* and *cows* backward.), [Tic-tac-toe game](https://www.practicepython.org/exercise/2014/12/27/24-draw-a-game-board.html), (Generate the board below rather than the one in the exercise. The dots (.) mean that the cell is not occupied.)

. | . | .

---+---+---

. | . | .

---+---+---

. | . | .

[Invert dictionary (Ex 14)](http://wasabiapp.org/python_course/exercises_4.pdf), and [Word frequencies (Ex 15)](http://wasabiapp.org/python_course/exercises_4.pdf).

**Think about:** [Class discussion notes](https://drive.google.com/file/d/1TzG9fqr3uFsJ7qRpyvBiWMy-AwzTTlv2/view?usp=sharing).