SMU | Lyle School of Engineering

CS 1340 Introduction to Computing Concepts

Instructor: Xinyi Ding

Aug 26 2019, Lecture 1

About Me

- About:
  - Me : Xinyi Ding, PhD student, Computer Science Department
  - Email : xding@smu.edu
  - Home page : xding.me
  - Research Interests: Ubiquitous Computing, Machine Learning, Educational Data Mining

Agenda

- Agenda:
  - Logistics and syllabus
  - About this course
  - Install python
  - First python program
  - Variables and Data types

Syllabus

- Syllabus details: http://xding.me/cs1340/
- Githhub page: https://github.com/dxywill/cs1340
- Assessment:
  - Home Assignments: weekly based (20%)
  - Quizzes: one week notification (10%)
  - Lab projects: bi-weekly (30%)
  - Mid exam: (20%)
  - Final exam: (20%)
- Use canvas for course management

Syllabus

- Lab Sessions:
  - M 10:00AM - 11:50 AM or Tu 1:00PM - 2:50 PM
  - Start from the 3rd week
- Office hours: TBD, check canvas for updates
- Help Desk caruth 484.
- Altshuler Learning Enhancement Center (A-LEC)

Syllabus

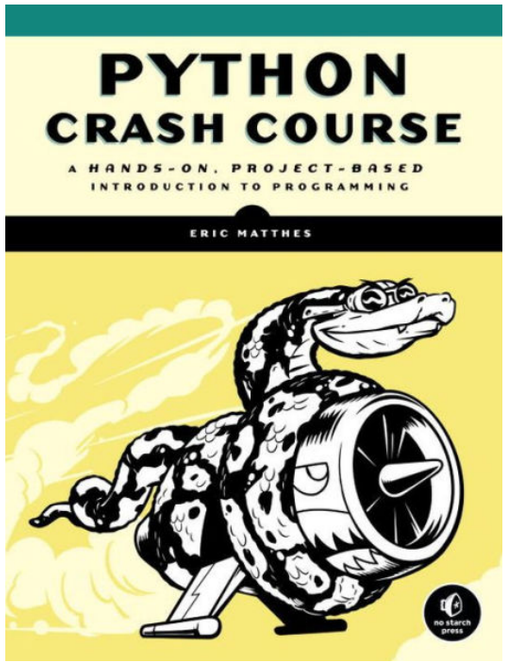
- Participation: required, missing three or more will lead to points loss (Rarely are these measures needed!)
- Late Submission:
  - Home assignments
    - Up to 3 times without credit reduction (use wisely)
    - Must be submitted within one week after deadline
  - Lab Projects:
    - No late submission will be accepted

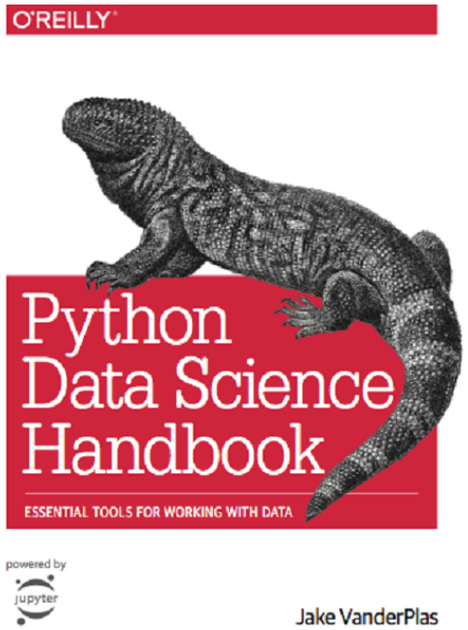
Syllabus

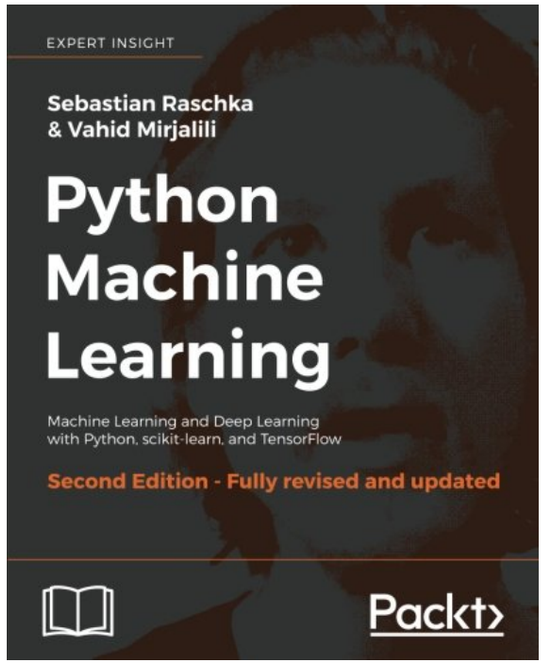
- Disability Accommodations
  - Students needing academic accommodations for a disability must first register with Disability Accommodations & Success Strategies (DASS).
- Religious Observance
  - Religiously observant students wishing to be absent on holidays that require missing class should notify their professors in writing at the beginning of the semester.
- Find more: http://xding.me/cs1340/

Introduction

- Textbooks:
  - **Not required**, but some recommended Books…





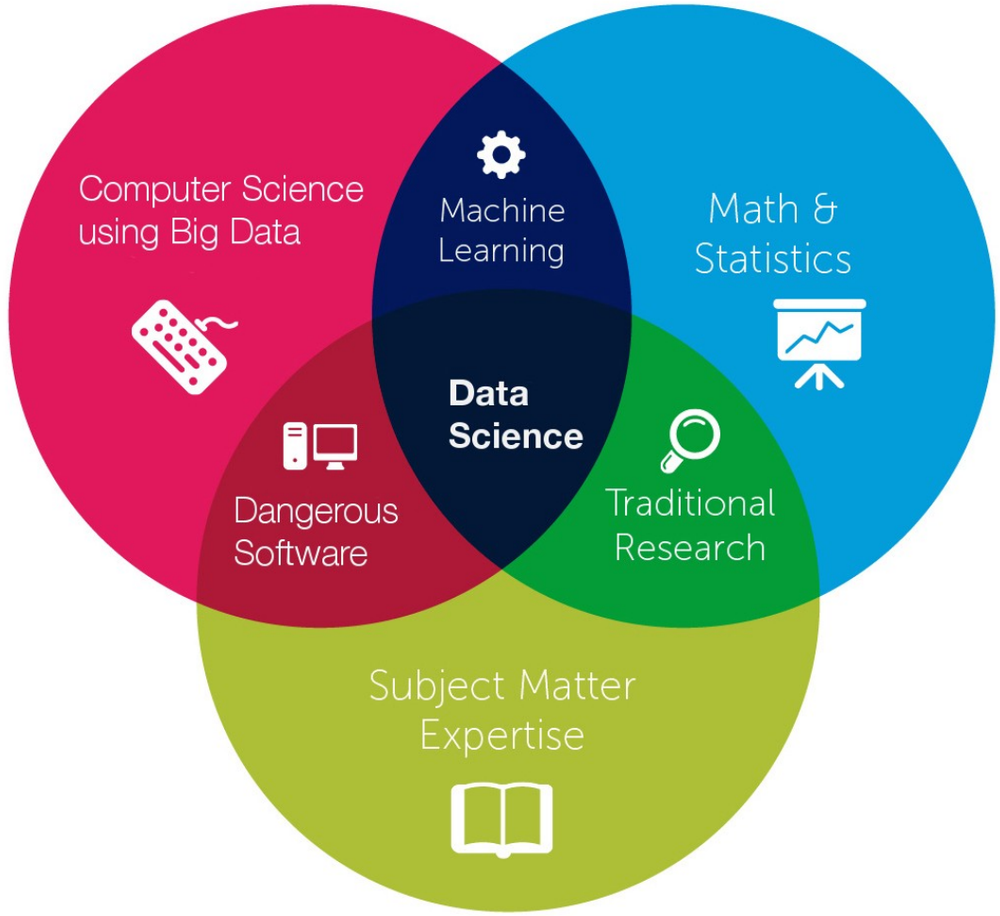


- If you have a specific question, try Google first
- stackoverflow.com
- PEP-8 style guide for python

  https://www.python.org/dev/peps/pep-0008/

Introduction

- Why Python?
  - Easy to learn
  - Efficient and Powerful
  - Big Community support
  - …
- Why Data Science?
  - Data fuels the future
  - Many career options
  - Make the world a better place
  - …



Introduction

- Prerequisite: ***None***
- Course Overview:
  - Part 1:
    - Variables, Data types, Common data structures (List, dict..)
    - Control statement, if/else, loops
    - Functions
    - Object Oriented Programming (OOP)
  - Part 2:
    - IPython, Jupyter Notebook
    - Numpy, Pandas, Matplotlib
    - Scikit-Learn

How to do well in this course?

- Do not fall asleep
- Practice, Practice and Practice
- Have Fun!

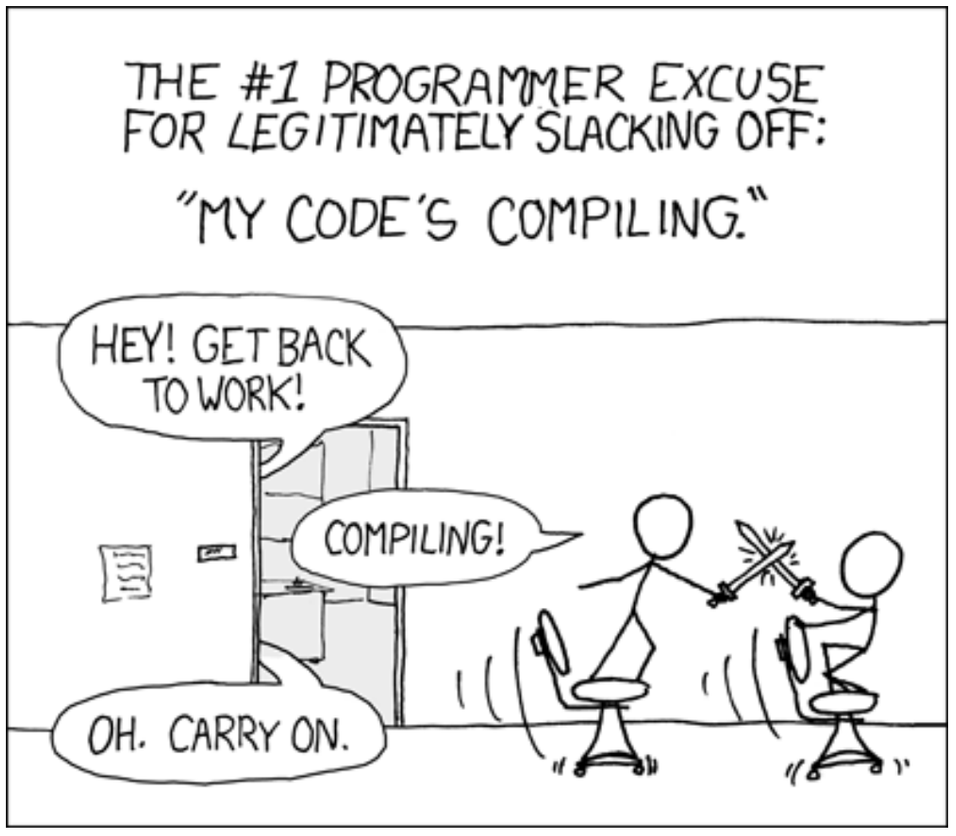
Four stages of morning class

Python programming language

"Python is an interpreted, high-level, general-purpose programming language. Created by Guido van Rossum and first released in 1991, *Python's design philosophy emphasizes code readability with its notable use of significant whitespace."*

—Wikipedia

- Compiled language:
  - C/C++
  - Delphi
- Interpreted language:
  - Python
  - PHP
  - Ruby



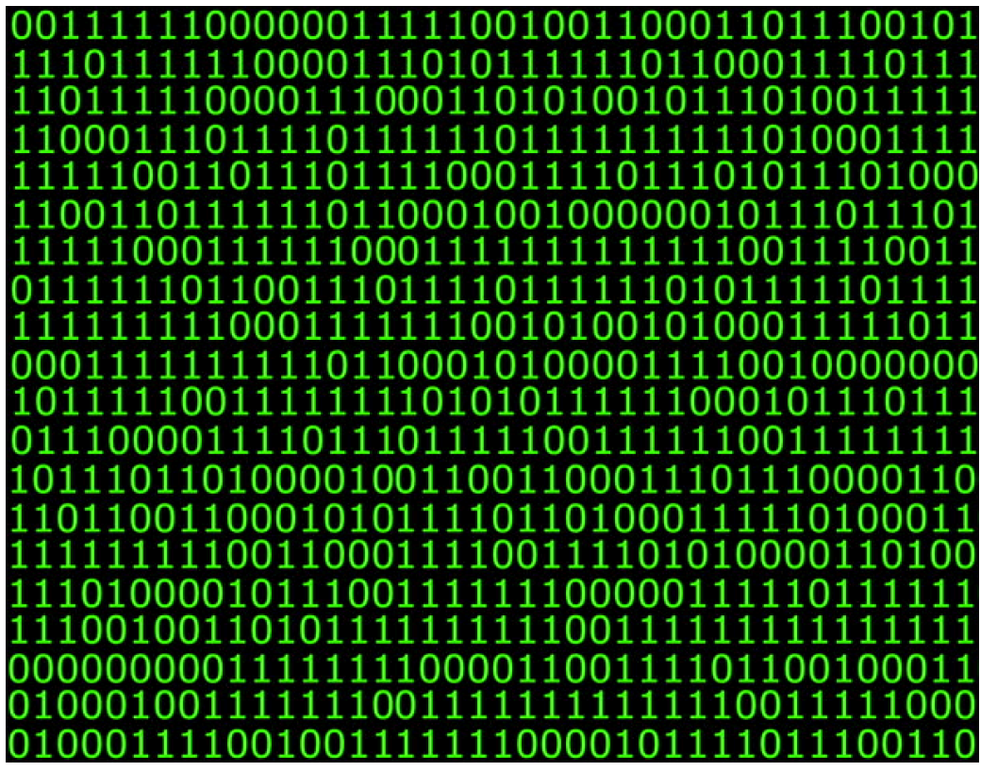
The fact is that the interpreted/compiled distinction is actually an arbitrary one

Python programming language

- Language features
  - Interpreted language
  - Indentation instead of braces
  - Dynamically typed: variables do not have a predefined type
  - Simple Object system
  - Rich, built in collection types:
    - Lists
    - Tuples
    - Dictionaries (maps)
    - Sets

Python programming language

```python
def load_data(fileName):
    rows = []
    max_skill_num = 0
    max_num_problems = 0
    with open(fileName, "r") as csvfile:
        reader = csv.reader(csvfile, delimiter=',')
        for row in reader:
            rows.append(row)
    index = 0
    print("the number of rows is " + str(len(rows)))
    tuple_rows = []
    #turn list to tuple
    while(index < len(rows)-1):
        problems_num = int(rows[index][0])
        tmp_max_skill = max(map(int, rows[index+1]))
        if(tmp_max_skill > max_skill_num):
            max_skill_num = tmp_max_skill
        if(problems_num <= 2):
            index += 3
```



Python Code

Interpreter

Binary Code

Python programming language

- Versions of Python
  - Python 2.X is dying…
  - Python 3.X
    - install through anaconda
    - use conda environment

- Check python version: *python - - version*

```
XinyideMacBook-Pro-4:~ xinyi$ python --version
Python 3.5.2 :: Anaconda custom (x86_64)
XinyideMacBook-Pro-4:~ xinyi$
```

- Check which python in use: *which python*

```
XinyideMacBook-Pro-4:~ xinyi$ which python
/Users/xinyi/anaconda/bin/python
XinyideMacBook-Pro-4:~ xinyi$
```

Anaconda

- Anaconda Python distribution include:
  - Python
  - A lot other useful tools
- Anaconda can help:
  - Installing Python on multiple platforms
  - Separating out different environments
  - Dealing with not having correct privileges and
  - Getting up and running with specific packages and libraries

Anaconda Environment

- conda environment
  - A conda environment is a directory that contains a specific collection of conda packages that you have installed.
    - *For example, you may have one environment with NumPy 1.7 and its dependencies, and another environment with NumPy 1.6 for legacy testing*
- Create conda environment: *conda create --name cs1340*
- Activate conda environment: *conda activate cs1340*
- Other useful commands*:* https://docs.conda.io/projects/conda/en/latest/user-guide/getting-started.html#managing-environments

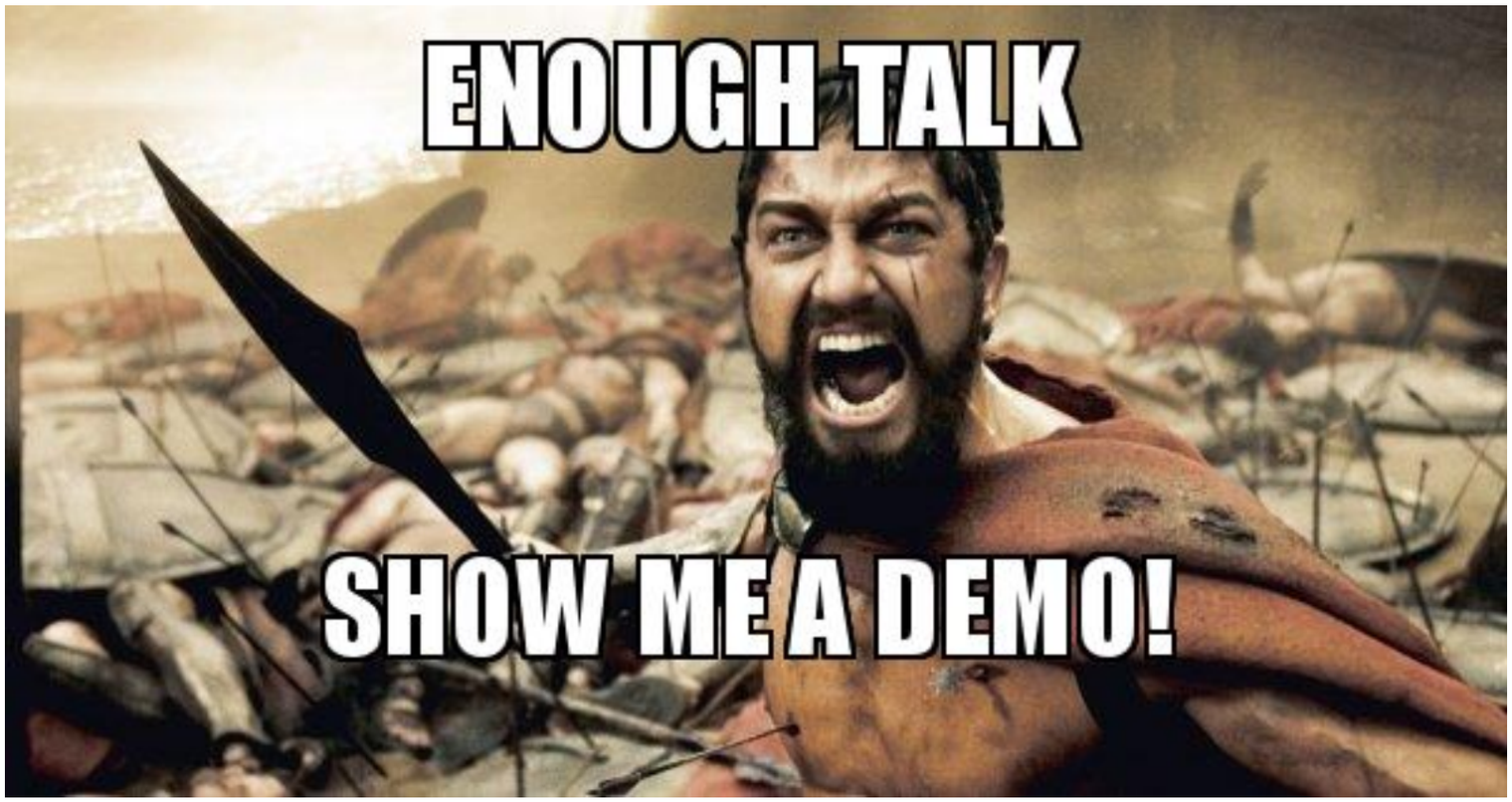
Anaconda Environment

```
(cs1340) XinyideMacBook-Pro-4:~ xinyi$ which python
/Users/xinyi/anaconda/bin/python
(cs1340) XinyideMacBook-Pro-4:~ xinyi$
```

```
(torch) XinyideMacBook-Pro-4:~ xinyi$ which python
/Users/xinyi/anaconda/envs/torch/bin/python
(torch) XinyideMacBook-Pro-4:~ xinyi$
```

```
(mlearn) XinyideMacBook-Pro-4:~ xinyi$ which python
/Users/xinyi/anaconda/envs/mlearn/bin/python
(mlearn) XinyideMacBook-Pro-4:~ xinyi$
```

Demo



Python programming language

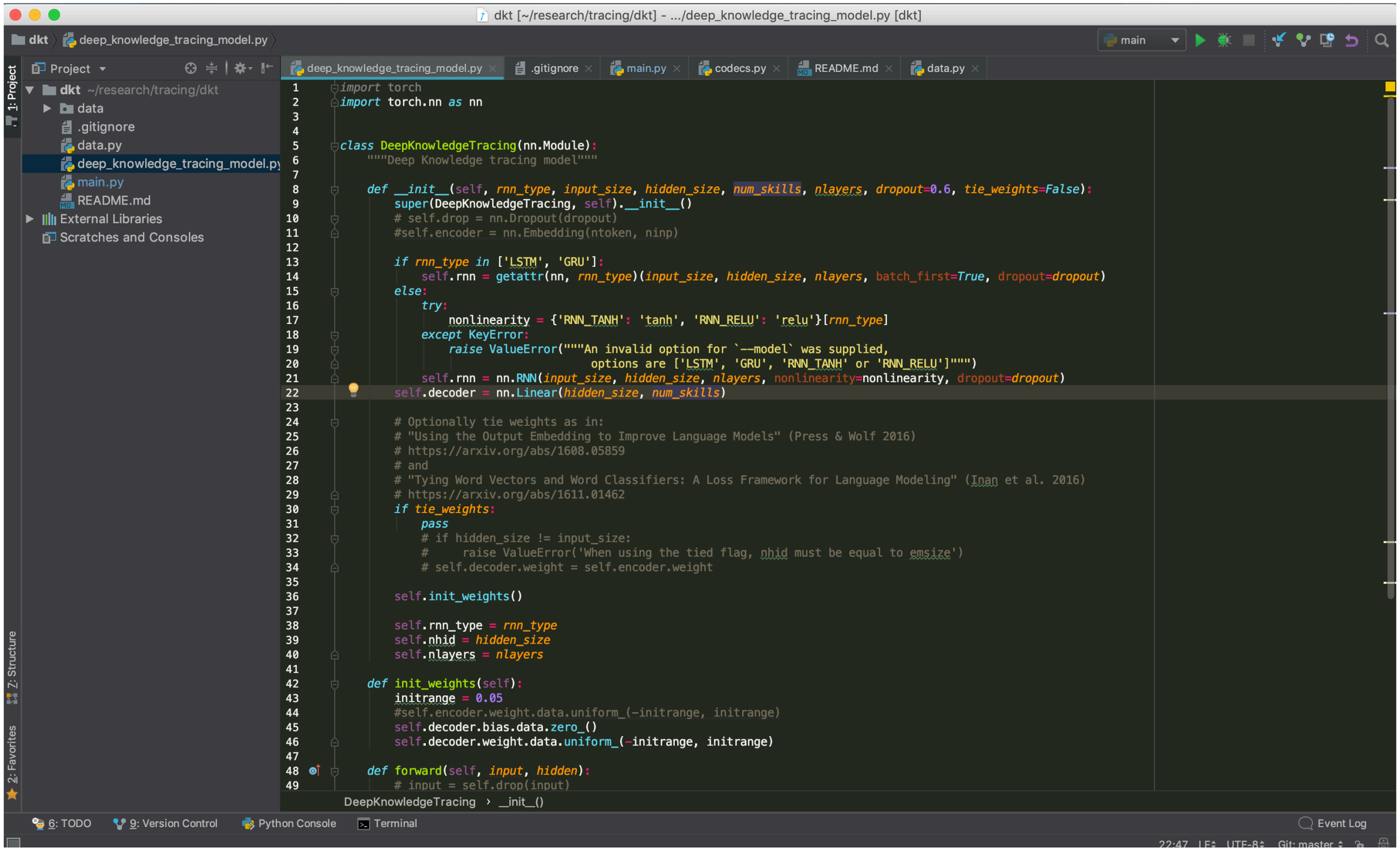
- Ways to use python
  - Interactive shell

```
xinyi — python — 130×39
~ — python
(cs1340) XinyideMacBook-Pro-4:~ xinyi$ which python
/Users/xinyi/anaconda/bin/python
(cs1340) XinyideMacBook-Pro-4:~ xinyi$ python
Python 3.5.2 |Anaconda custom (x86_64)| (default, Jul  2 2016, 17:52:12)
[GCC 4.2.1 Compatible Apple LLVM 4.2 (clang-425.0.28)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
>>> print("Hello world")
Hello world
>>>
```

  - Python script: *python hello_world.py*

```
week1 — -bash — 130×39
~/Courses/cs1340/week1 — -bash
(cs1340) XinyideMacBook-Pro-4:week1 xinyi$ ls
hello_wolrd.py
(cs1340) XinyideMacBook-Pro-4:week1 xinyi$ python hello_wolrd.py
Hello wolrd
(cs1340) XinyideMacBook-Pro-4:week1 xinyi$
```

Pycharm: Integrated development environment (IDE)



More resources: https://www.jetbrains.com/pycharm/