Python Programming

Start your python scripts in PyCharm

Harry Lee

1/10/2018

CEE 696 & Stanford CEE 268

Automation

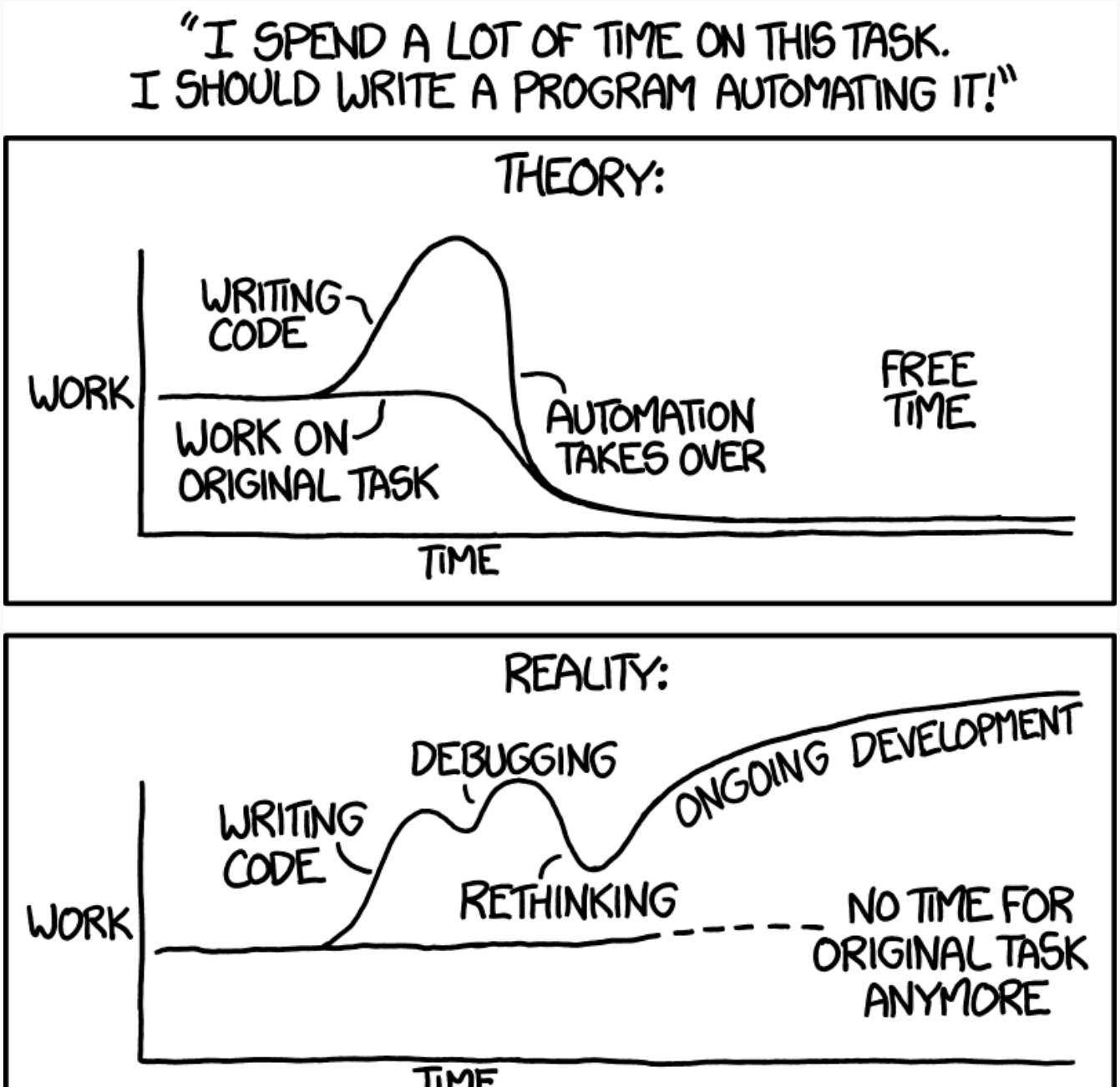

"I SPEND A LOT OF TIME ON THIS TASK.
I SHOULD WRITE A PROGRAM AUTOMATING IT!"
THEORY:
WORK
WRITING CODE
WORK ON ORIGINAL TASK
AUTOMATION TAKES OVER
FREE TIME
TIME
REALITY:
WORK
WRITING CODE
DEBUGGING
RETHINKING
ONGOING DEVELOPMENT
NO TIME FOR ORIGINAL TASK ANYMORE
TIME

Table of Contents

PyCharm Setup

Python execution

How can you execute your python scripts?

- go to terminal (“Command Prompt” in Windows) - run python
- go to terminal - run ipython
- Jupyter Notebook

Terminal - Python

Command Prompt - Python

You can run a python script by typing “python your_script _name.py” or open interactive shell by simply typing “python”

Terminal - IPython

Ipython is an interactive command-line terminal with extra features like tab-completion, testing, debugging, system calls and other features.

type “ipython” in the terminal

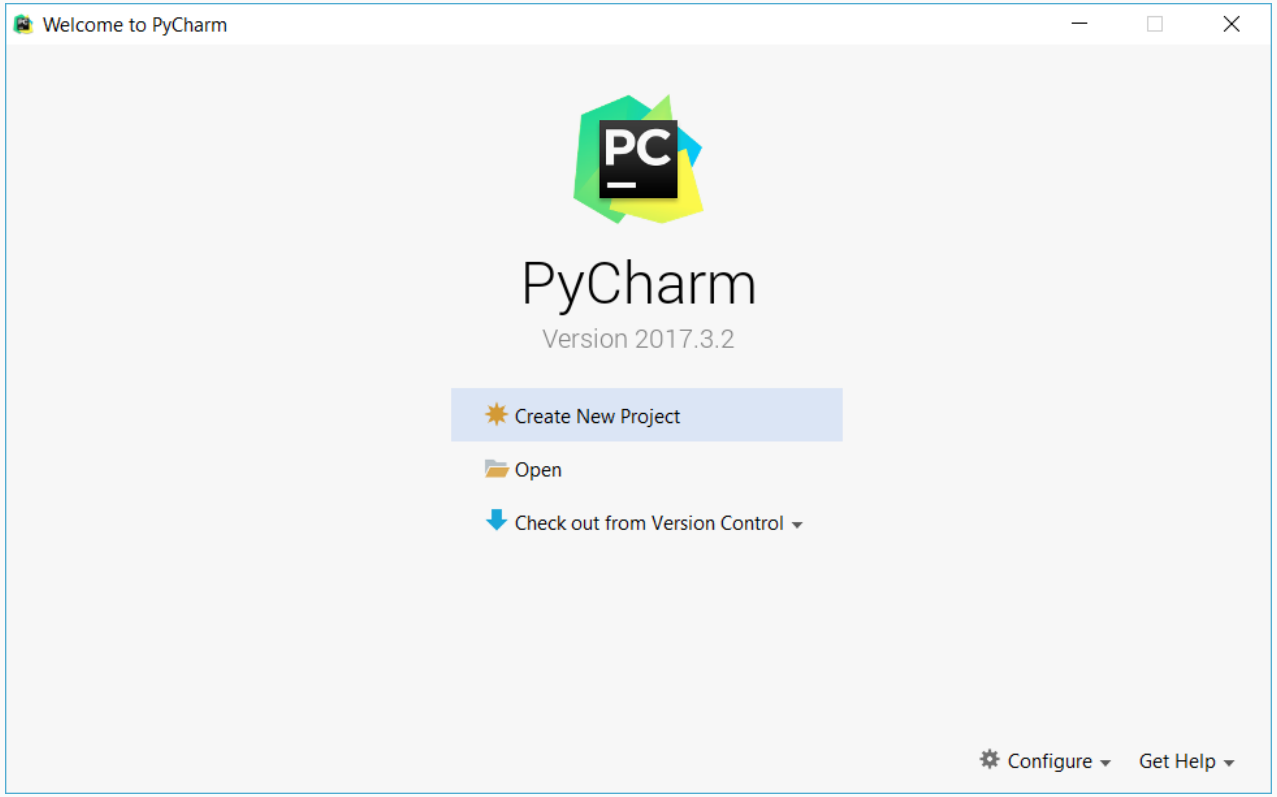
Python execution

How will we write and execute python scripts in this class?

- ~~go to terminal - python, ipython~~
- PyCharm (with ipython) - we will use this
- We can use Jupyter Notebook using PyCharm as well

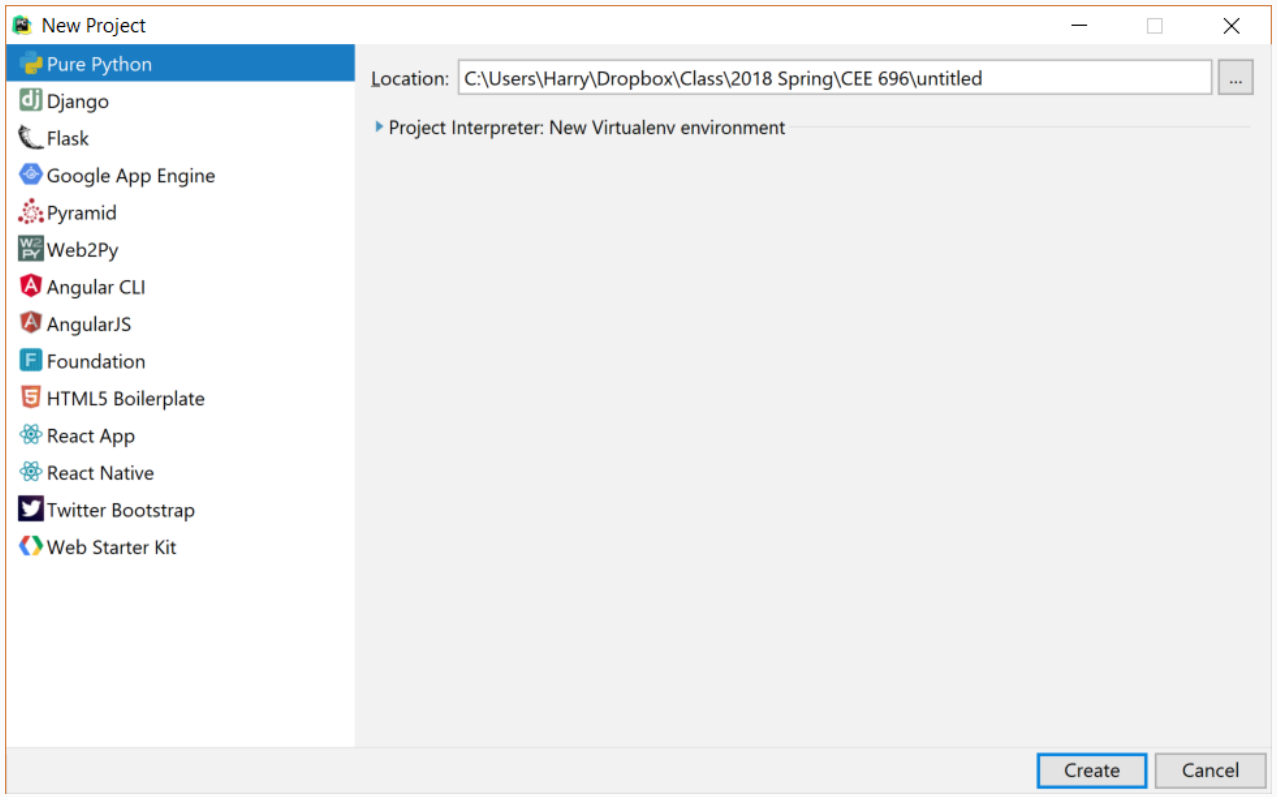
Start PyCharm (1)

1. Start PyCharm



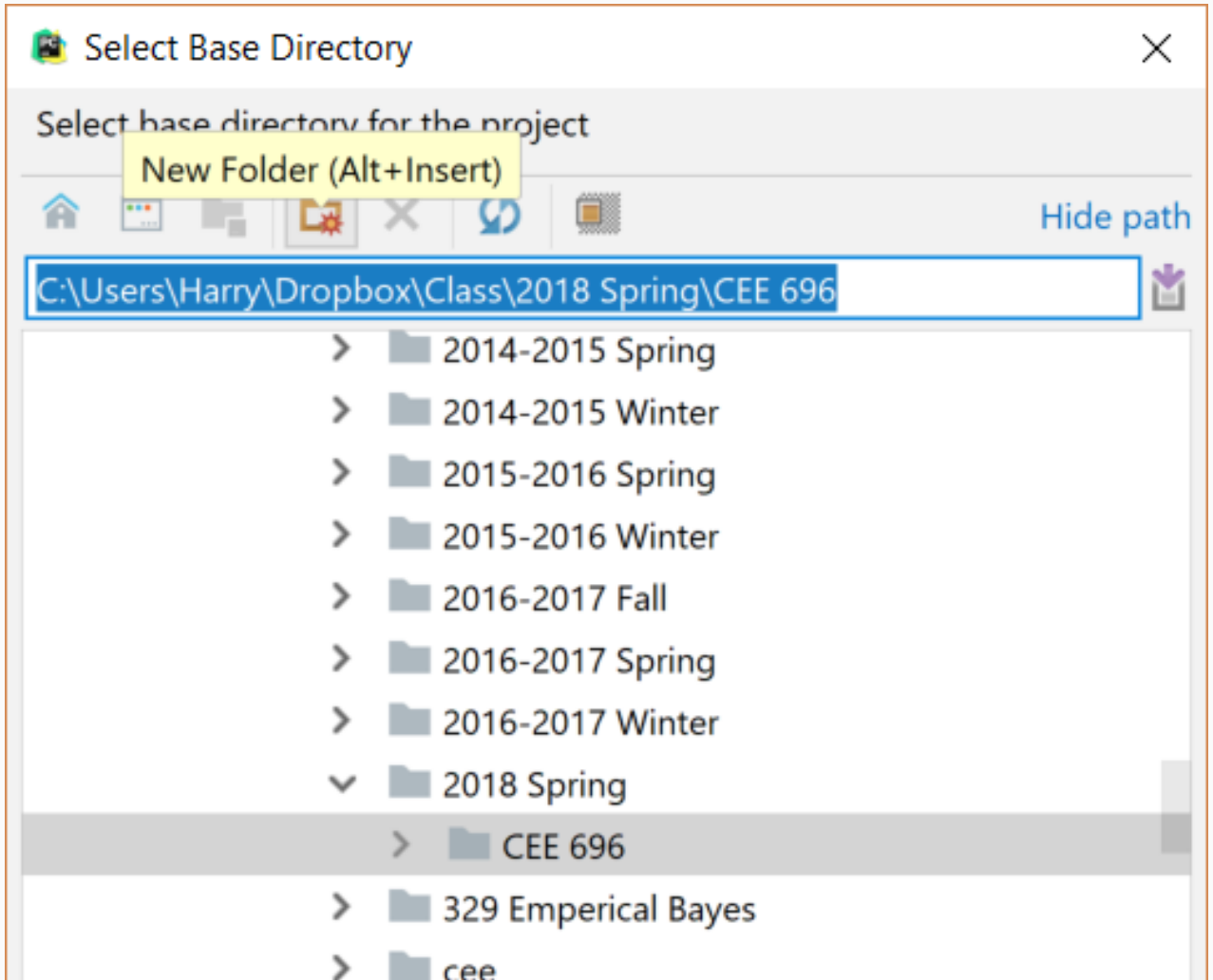
Start PyCharm (2)

1. Start PyCharm
2. Create new project



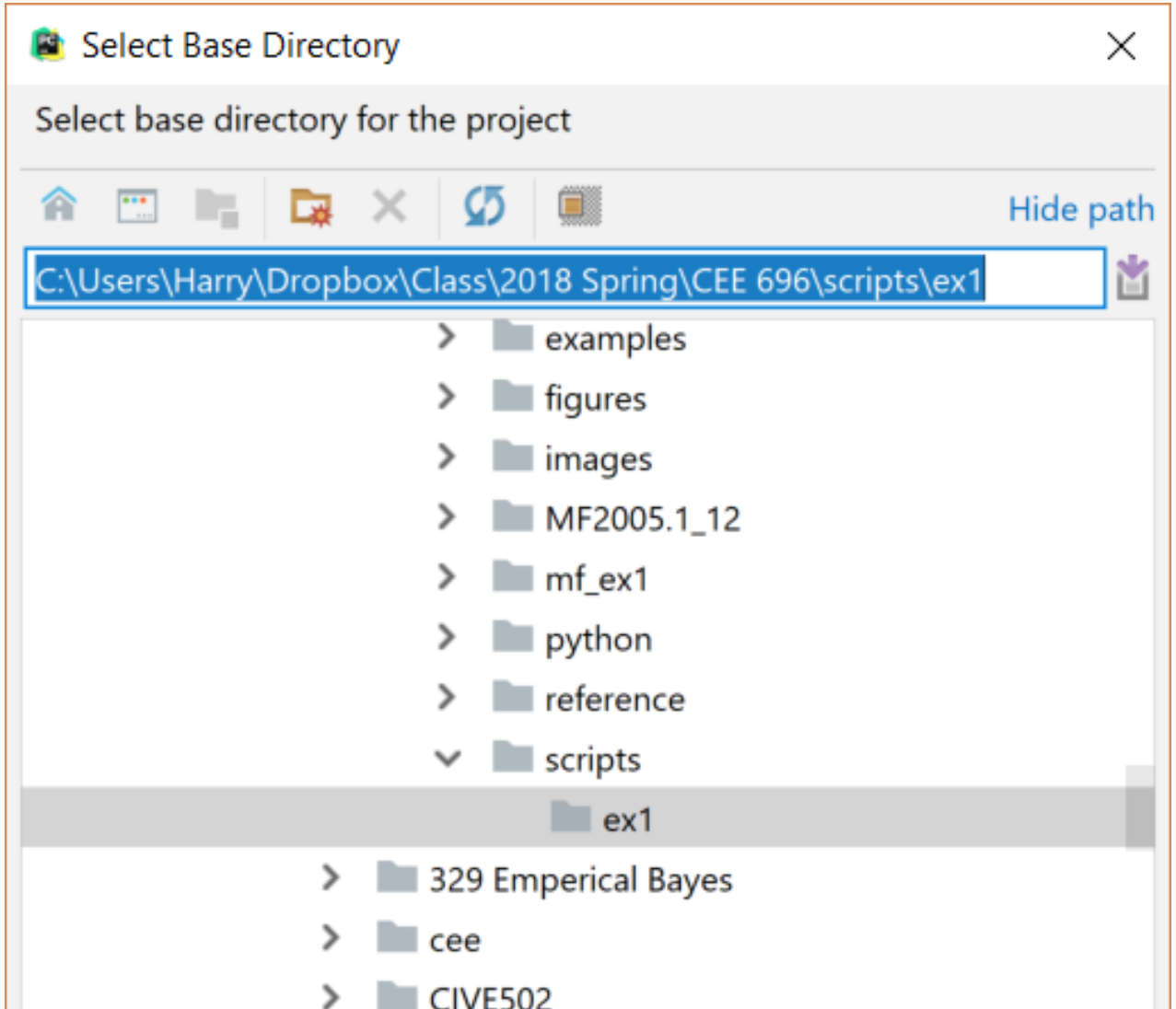
Start PyCharm (3)

1. Start PyCharm
2. Create new project - choose your directory - create a new folder if needed



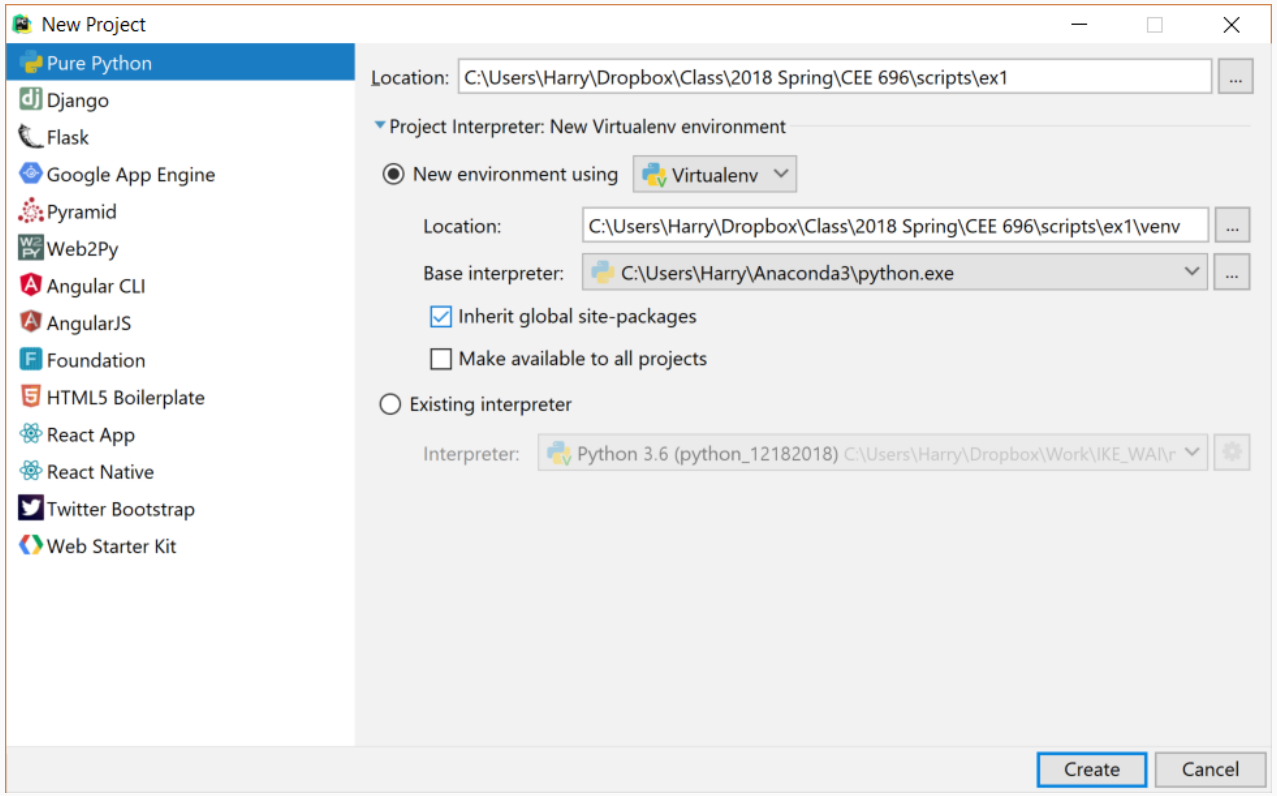
Start PyCharm (4)

1. Start PyCharm
2. Create new project - choose your directory



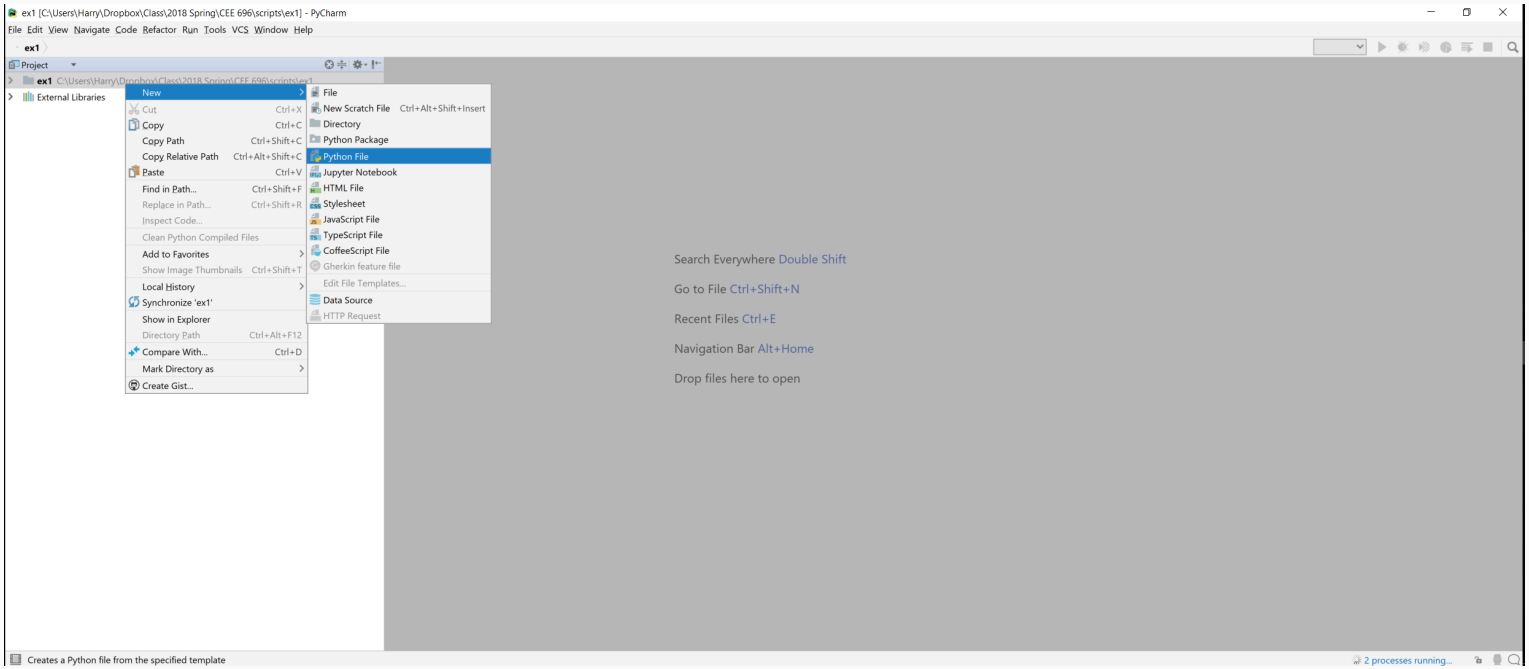
Start PyCharm (5)

1. Start PyCharm
2. Create new project - choose your directory
3. Project Interpreter- New environment using Virtualenv - Inherit global site-packages unless you want to build everything from scratch

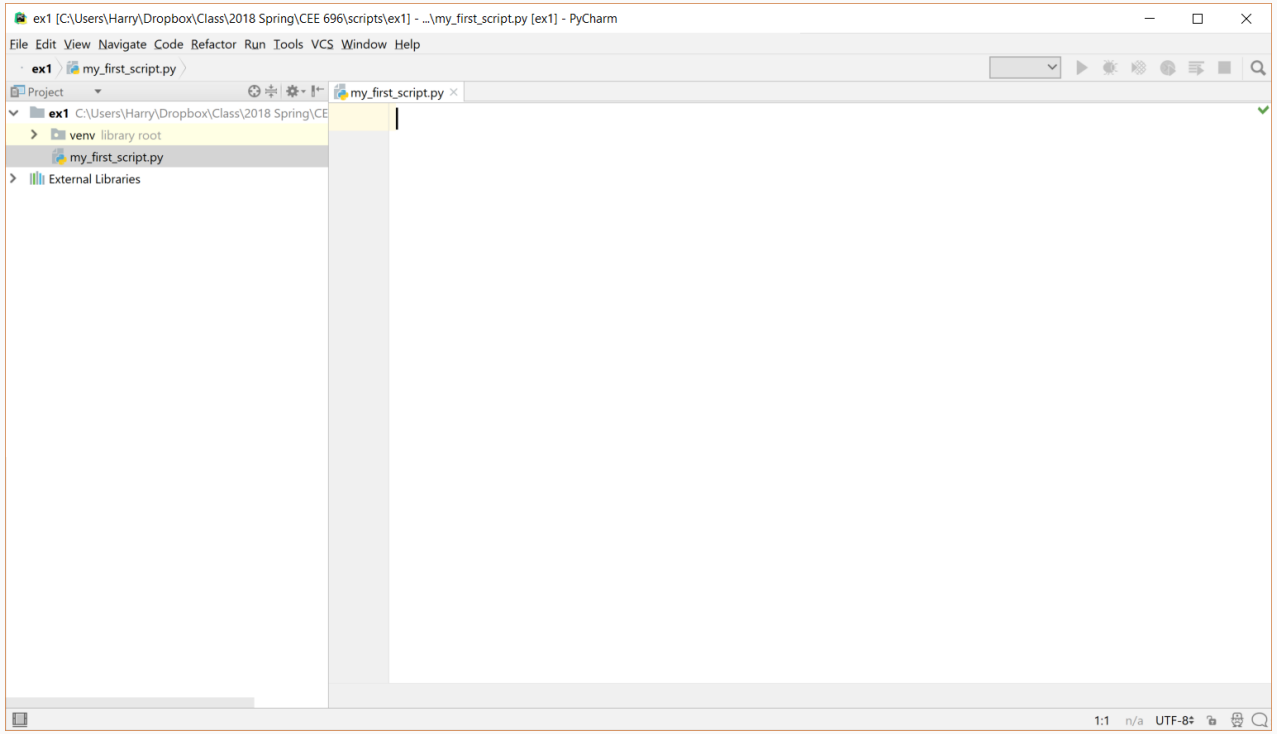


Start PyCharm (6)

We are almost there! Create your first python script



Start Pycharm



Now you can write your own script.

Installation of Python packages

1. NumPy : N-dimensional array package
2. SciPy : Scientific computing library package
3. Matplotlib : Plotting package
4. FloPy : MODFLOW python interface package
5. Jupytor : Jupyter Notebook (not required)

Package installation:

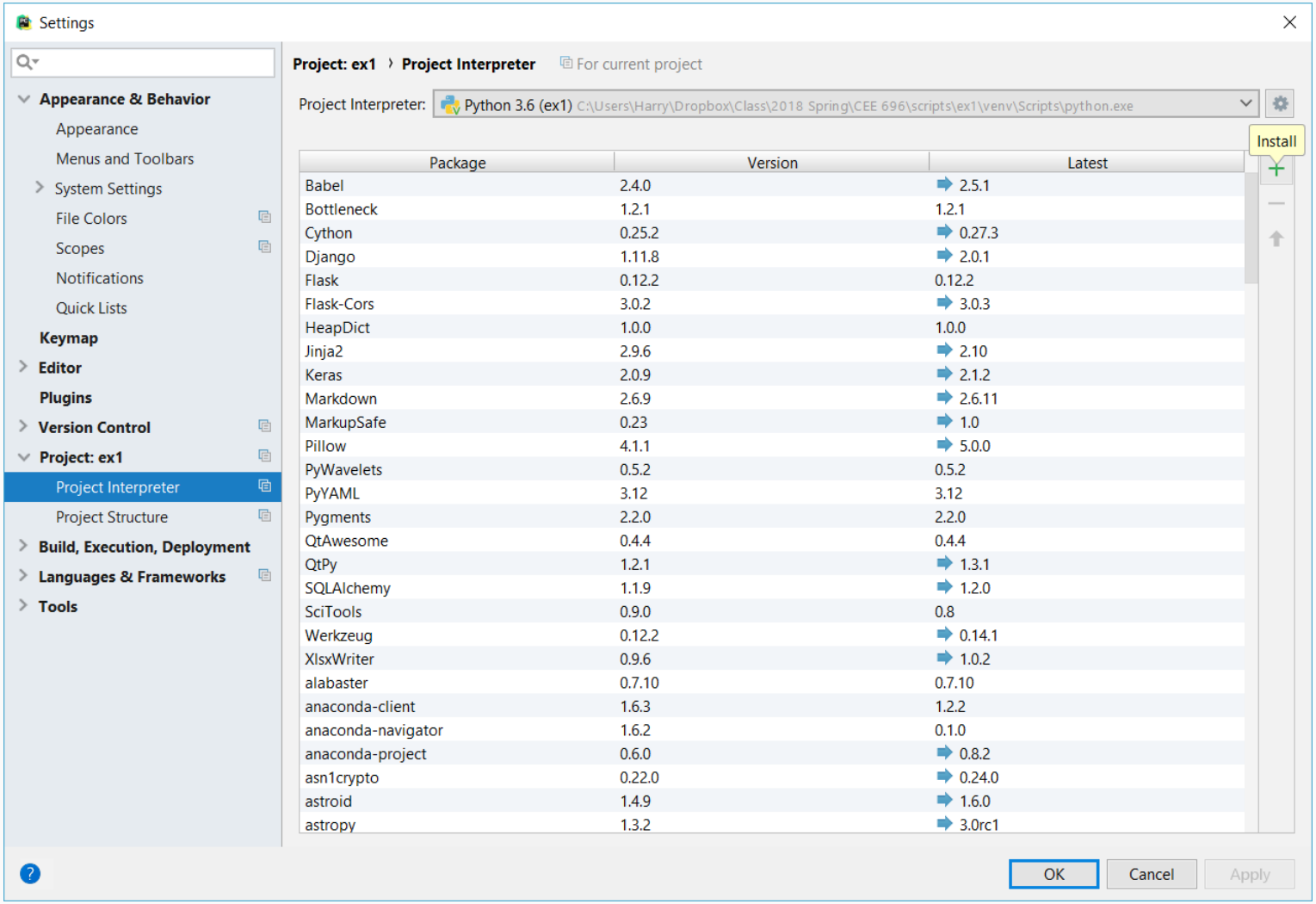
- You can install them in our Anaconda Python (“Anaconda Navigator” - Environments - root - change “installed“” to “not installed” - and “search Packages”: this might require root or administrator permission)
- Or, we can install them separately within your virtual environment

Virtual Environment (virtualenv)

What is Virtual Environment?

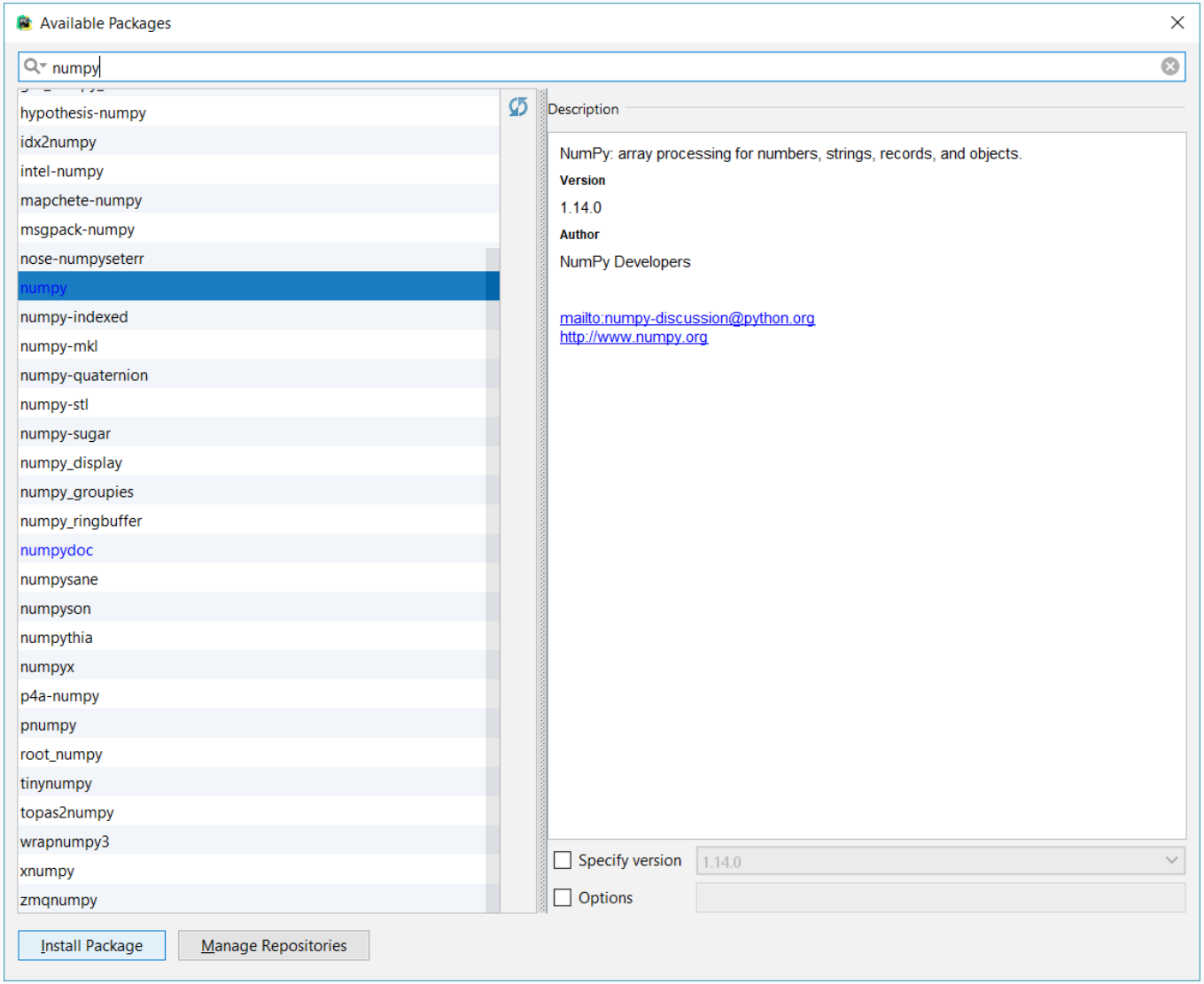
- It is a tool to create isolated Python development environments
- This will address the issue of version dependencies and permissions.
- For example, your current application requires version 1.0 of program A and another application needs version 1.57 of program A, how can you maintain both applications?
- Or, you accidentally upgrade program A to its version 2.0 and your applications stop working. How can you deal with this.
- virtualenv basically creates a development environment that has its own installation directory under your current workspace.

Package installation (1)



File - Settings - Project - Project Interpretator + click ”+” button

Package installation (2)



Install numpy, scipy, matplotlib and flopy (and jupyter if you want)

Package installation (3)

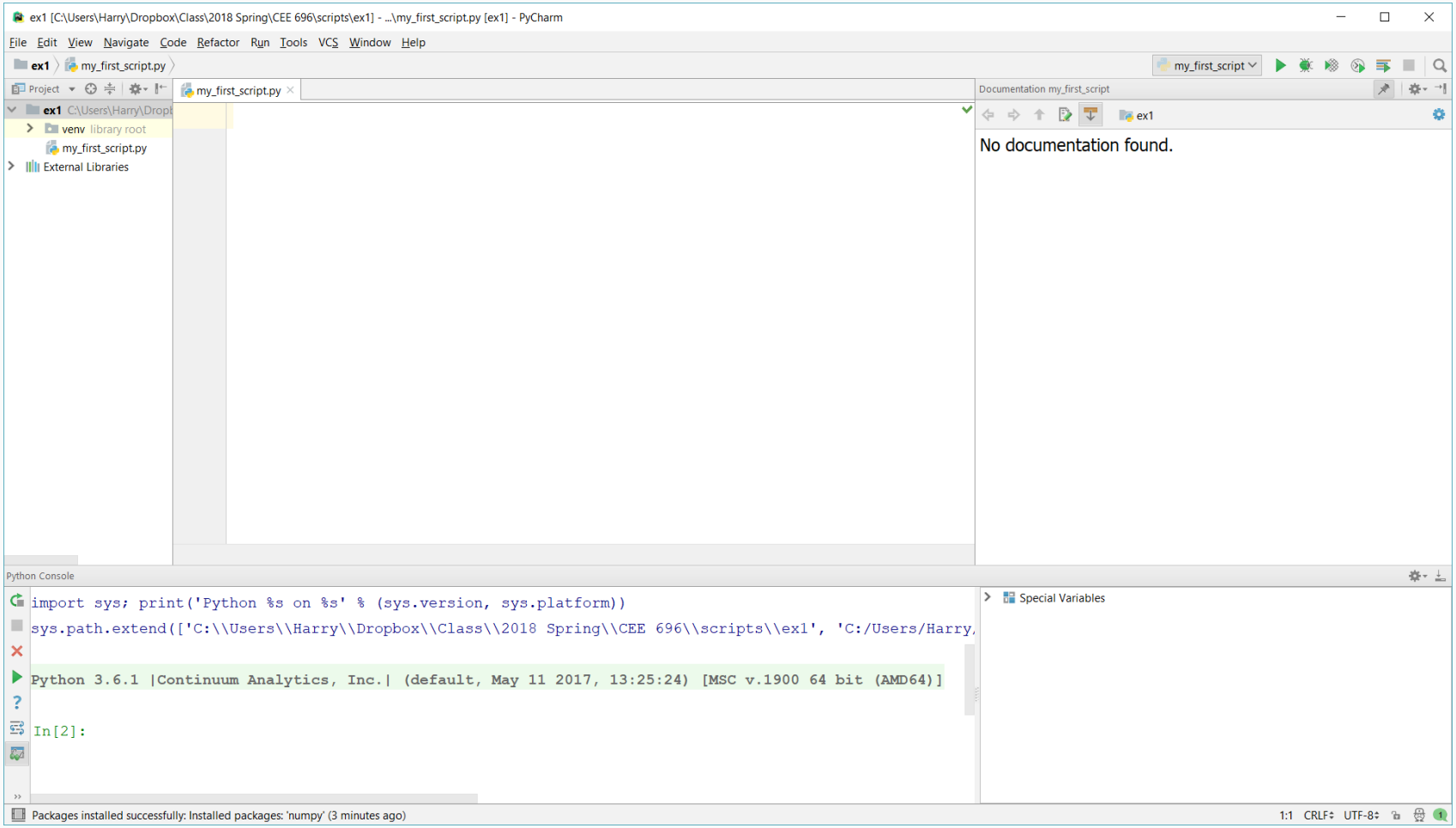
If you have issues with installation, you can go to ”Command Prompt” (Windows) or Terminal (Linux or Mac) and type

“pip install your_package”

for example, “pip install flopy”

Write your first script

Then turn on ”Scientific mode” View - Scientific Mode



my_first_python_script.py

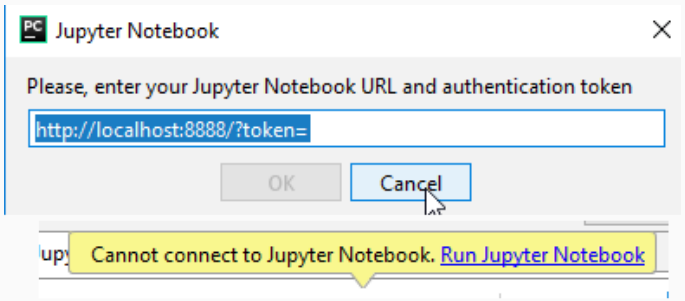
```
nlay = 5 # number of layers
nrow = 10 # number of rows
ncol = 20 # number of columns

nlay,nrow,ncol = 5,10,20 # more compact way
# spacing along a row (ncol) and column (nrow), respecti
delr, delc = 1.5, 2.5

print("nrow: %d, delc: %f" % (nrow, delc))
```

- run you script by Run - Run or Alt+Shift+F10
- or you can select lines and mouse right-click - Execute Selection in Console or Alt+Shift+E

Jupyter Notebook (not required!)

Jupyter Notebook is another interactive computational environment on your web browser supported by Python community. While we stick to PyCharm in this class, you can use the Notebook for interactive python scripting.

1. Create your notebook file : File - New - Jupyter Notebook
2. click ”green” run cell button
3. cancel the dialog box below
4. click ”Run Jupyter Notebook” in yellow error message
5. click the url ”http://127.0.0.1:8888....” below to open your default browser
6. When you finish the notebook, stop it by clicking the red stop button in Run Tool Window (or Ctrl+F2)



numpy and matplotlib

We will look into the code later in detail

```
import numpy as np # n-dimensional array
x = np.linspace(0,2*np.pi,100)
y = np.sin(x)

import matplotlib.pylab as plt
plt.figure()
plt.plot(x,y)
plt.title('my first plot!')
plt.show()
```