EML4930/EML6934: Lecture 00 - About Python

Python2 vs Python3, Hello World, IPython, Notebooks, Installation

Charles Jekel

August 24, 2017

About me

PhD Student in the MDO lab.

I look at improving techniques for selecting material parameters.

My interests:

- non-linear finite element (FE) method
- non-linear material modeling
- inverse analysis
- optimization
- regression and classification
- digital image correlation (DIC)
- high performance computing (HPC)
- Python

For more see `http://jekel.me`

About the course

Syllabus available online.

The intention of this course is to prepare you to for doing numerical work in Python.

Course expectations:

- 12 out of 14 homework — 60 %
- 2 quizzes — 10 %
- 1 final exam — 30 %

Intended audience

Course description

Python is a general purpose programming language. Course covers the basics, linear algebra, plotting, and more to prepare students for solving numerical problems. Prerequisite: COP 2271 MATLAB or equivalent.

You already know how to program in some language.
You are interested in doing numerical analysis in Python.

What is Python? - python.org

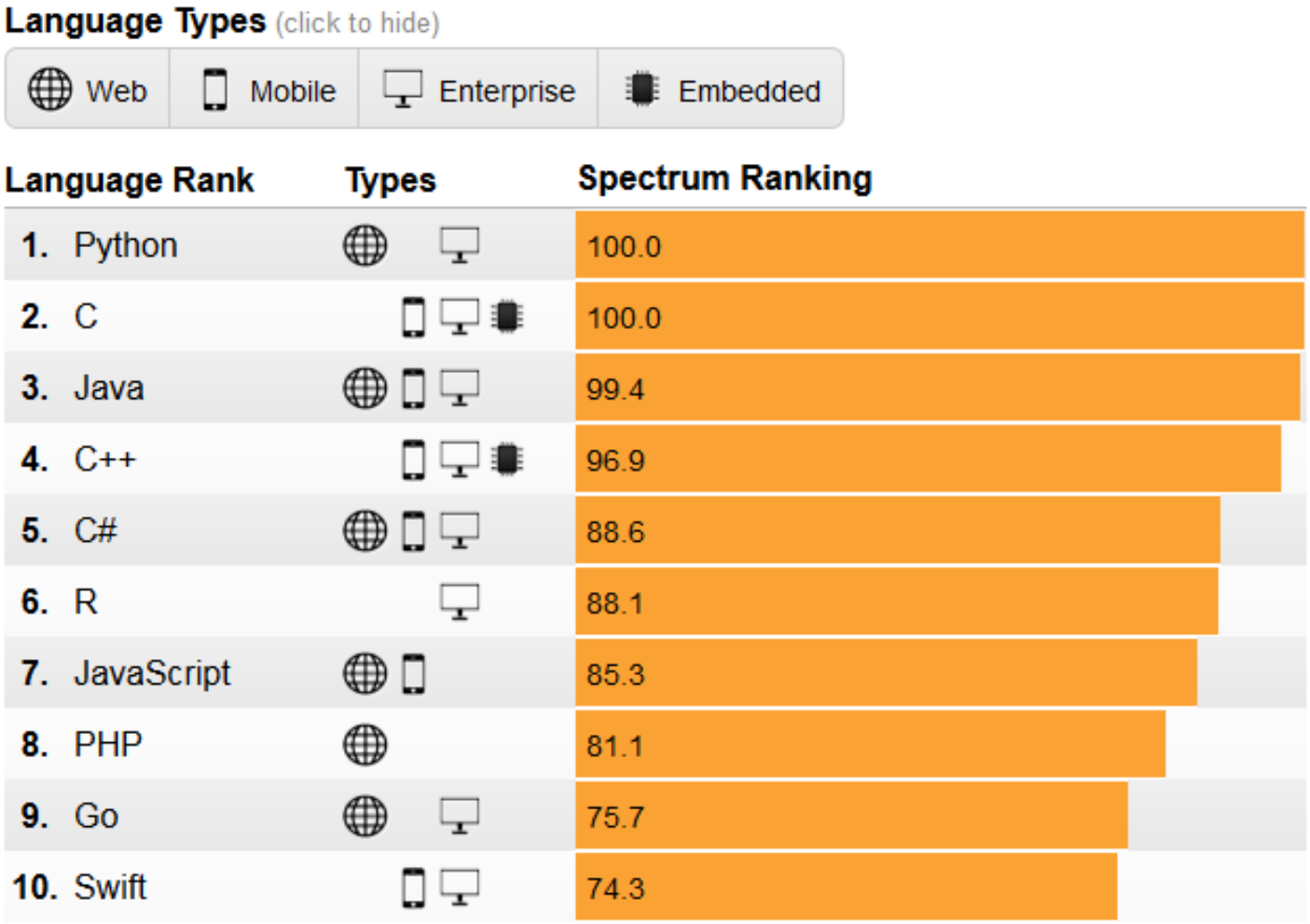
Let's see what `http://python.org` has to say about Python.

Python is

- an interpreted, object-oriented, high-level programming language with dynamic semantics
- very attractive for Rapid Application Development
- a simple and easy to learn syntax

Python is a very popular programming language

For the first time ever, IEEE Spectrum rated Python the most popular programming language in 2017. `http://spectrum.ieee.org/computing/software/the-2017-top-programming-languages`



Built using Python

Why Python? - Free, Open, and Powerful

- Python is Free and Open
- Python can be used commercially
- From research to deployment
- Libraries to do everything
- Adapted by scientist and engineers
- Cross platform support

Python2 vs Python3

There is a syntax difference between Python version 2 and Python version 3.

When Python 3.0 was released in 2008 it broke backwards compatibility with Python 2.X. This was a mistake, and resulted in a very slow adaption of Python 3.

There are likely Python libraries that have yet to be ported to Python3.

Additionally there are libraries that were only written in Python3.

All new code should probably be written in a Python3 syntax, but I won't force you to use Python3. I still primarily use Python 2.7. Choose your version based on your library needs.

For more see: `https://wiki.python.org/moin/Python2orPython3`

Python2 vs Python3 comparison

Python2 wins

- Speed - Python 2.7 will always be faster
- Legacy
- Python updates won't break your code!

Python3 wins

- Unicode identifiers
- Strings unicode by default
- Simple matrix multiplication with @

TensorFlow and Windows - you need Python 3.5

If you are intested in using **TensorFlow on Windows,** you must use a very specific version of Python.

TensorFlow is a state-of-the-art machine learning library.

See `https://www.tensorflow.org/install/install_windows`

Ways to run Python

Here is the Python2 hello world program saved as helloWorld2.py

```
print 'hello world'
```

These are some of the ways to run Python.

- from an IDE (integrated development environment)
- python
- python helloWorld2.py
- ipython
- (while in IPython) %run helloWorld2.py
- ipython qtconsole (qtconsole has added benefits)
- ipython notebook

python3 helloWorld2.py

Running this (helloWorld2.py) in Python3:

```
print 'hello world'
```

gives me the following error:

```
lloWorld2.py", line 1
    print 'hello world'
                      ^
SyntaxError: Missing parentheses in call to 'print'
```

because in Python3 *print* is now a function

python3 helloWorld3.py

With Python 3.X *print* must include an open and closed parentheses.

The following code is saved as helloWorld3.py

```
print('hello world')
```

You can run this code with both python2 and python3.

If you want to write code for Python 2.7 and Python 3.X

Your *.py script should always begin with these first three lines. The following code was saved as helloWorld.py.

```
from __future__ import absolute_import
from __future__ import division
from __future__ import print_function
print('hello world')
```

This code runs with both python2 and python3.

Using future will force you to write in the Python 3.X syntax

Executing the following in python2

```
from __future__ import absolute_import
from __future__ import division
from __future__ import print_function
print 'hello world'
```

Will give you an error!

Python 3.X allows you to use unicode identifiers

é is a unicode character which can be used in the identifier (as a variable) in Python 3.X but not Python 2.X.

Additionally strings in Python 3.X default to unicode (not the case in Python 2.X).

The following code will only run in python3 (unicode breaks my syntax highlighting...)

Fréchet = 'Fréchet is famous a French mathematician'

print(Fréchet)

Comparing strings to numbers in Python 2 is weird

The following is input and output with python2 (it is not suppose to make any sense).

```
'123' > 900
```

True

```
'123' > 0
```

True

```
'123' < 900
```

False

```
'123' < 0
```

False

Do not compare different data types in Python 2.X!

Python 3.X fixes string to number comparison

by not letting you compare data types that don't make sense

```
In [1]: '123' > 900
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
<ipython-input-1-17541aca4e39> in <module>()
----> 1 '123' > 900

TypeError: unorderable types: str() > int()
```

Python 2.7 vs Python 3.X depends on you

Syntax differences:

- Python 3.X requires print to use parentheses
- In Python 3.X print is a function
- You can access the print function of Python 3.X in Python 2.7 by importing *Future print function*

Additional resources:

- `http://python-future.org/imports.html`
- `https://wiki.python.org/moin/Python2orPython3`
- `https://learntocodewith.me/programming/python/python-2-vs-python-3/`

No more time for Python 2.7 vs Python 3.X. Pick one. Everything we do in this course will be compatible in both versions. You're allowed to use the version that best suits you.

Install python - do not install from `https://python.org`

Python builds from `https://www.python.org/downloads/` do not include pre-built libraries for numerical and scientific work. Compiling and building libraries is very complicated on Windows (and slightly less complicated on other operating systems). Think of this download as the bare basic Python with no libraries.

DO NOT DOWNLOAD, DO NOT INSTALL.

Looking for a specific release?

Python releases by version number:

Release version	Release date	
Python 3.6.2	2017-07-17	Download
Python 3.6.1	2017-03-21	Download
Python 3.4.6	2017-01-17	Download
Python 3.5.3	2017-01-17	Download
Python 3.6.0	2016-12-23	Download

Install Python - two choices

For numerical and scientific work I recommend installing Anaconda or Enthought Canopy. These installations include the most popular libraries that are pre-built for your system. (I prefer Canopy, but use both.)

Only a few months ago did Canopy start support Python 3.X, while Anaconda has supported Python 3.X for some time. I don't like Spyder IDE included with Anaconda because you can't bind crt + / to comment..

Install Anaconda if ...

You want

- to have it your way
- to run multiple Python versions
- install TensorFlow or other ML libraries on Windows
- the powerful conda console tool for installing libraries
- the latest and greatest Python libraries
- Spyder IDE
- automatic environmental variable setup

Install Enthought Canopy if ...

You want

- the easiest Python scientific environment
- a pretty GUI for library management and updates
- Canopy IDE
- fewer options

If you are coming from MATLAB ...

Useful resources

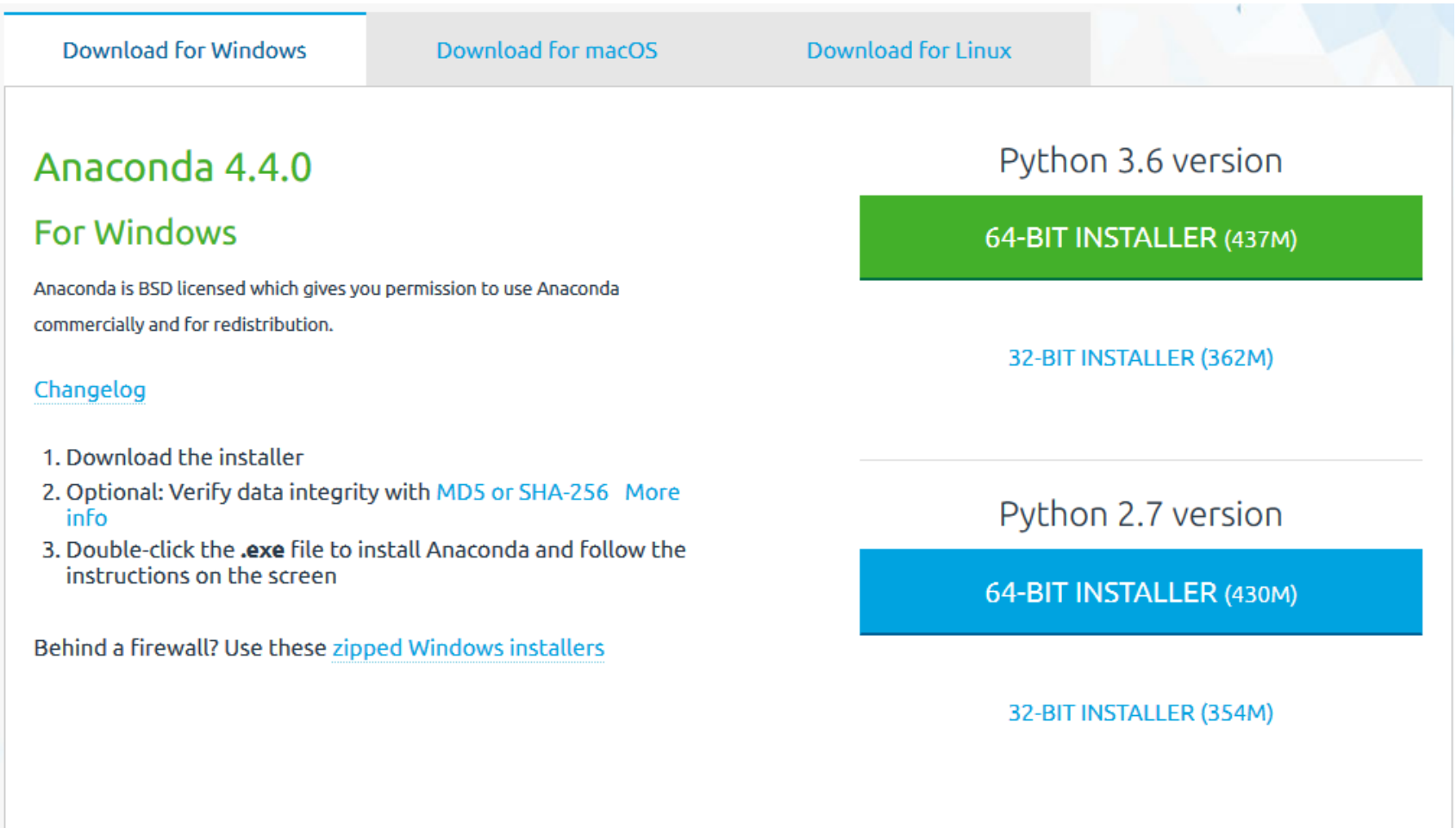
`http://mathesaurus.sourceforge.net/matlab-numpy.html`

`https://docs.scipy.org/doc/numpy-dev/user/numpy-for-matlab-users.html`

Install Python - Anaconda

Anaconda gives you more control over which Python version. Also Anaconda updates more frequently.

`https://www.continuum.io/downloads`



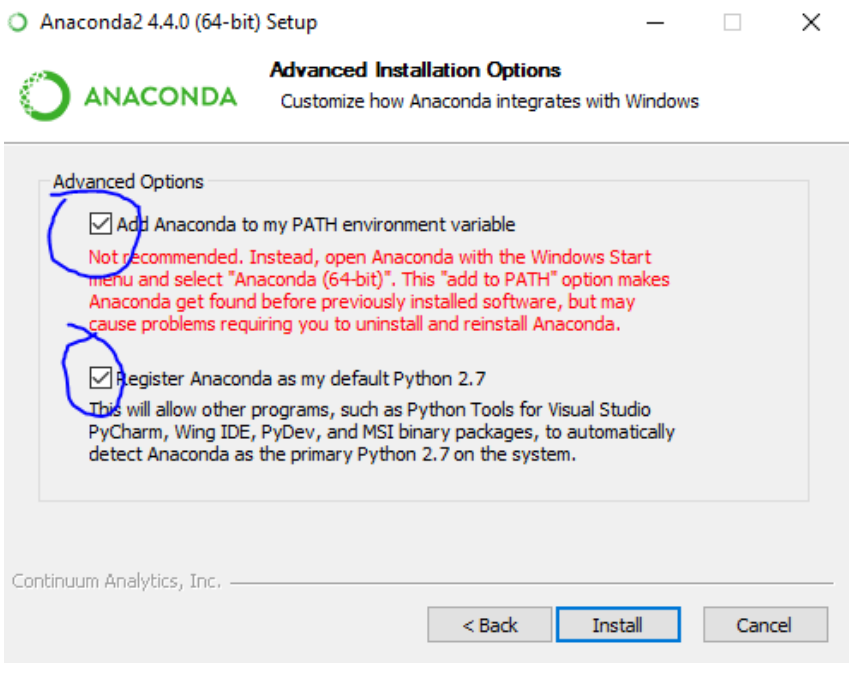
GOLDEN RULE with software...

my golden rule

If you don't know what advanced options do in software (especially when installing new software), leave the advanced options to their default setting. This will come up during the Anaconda installation.

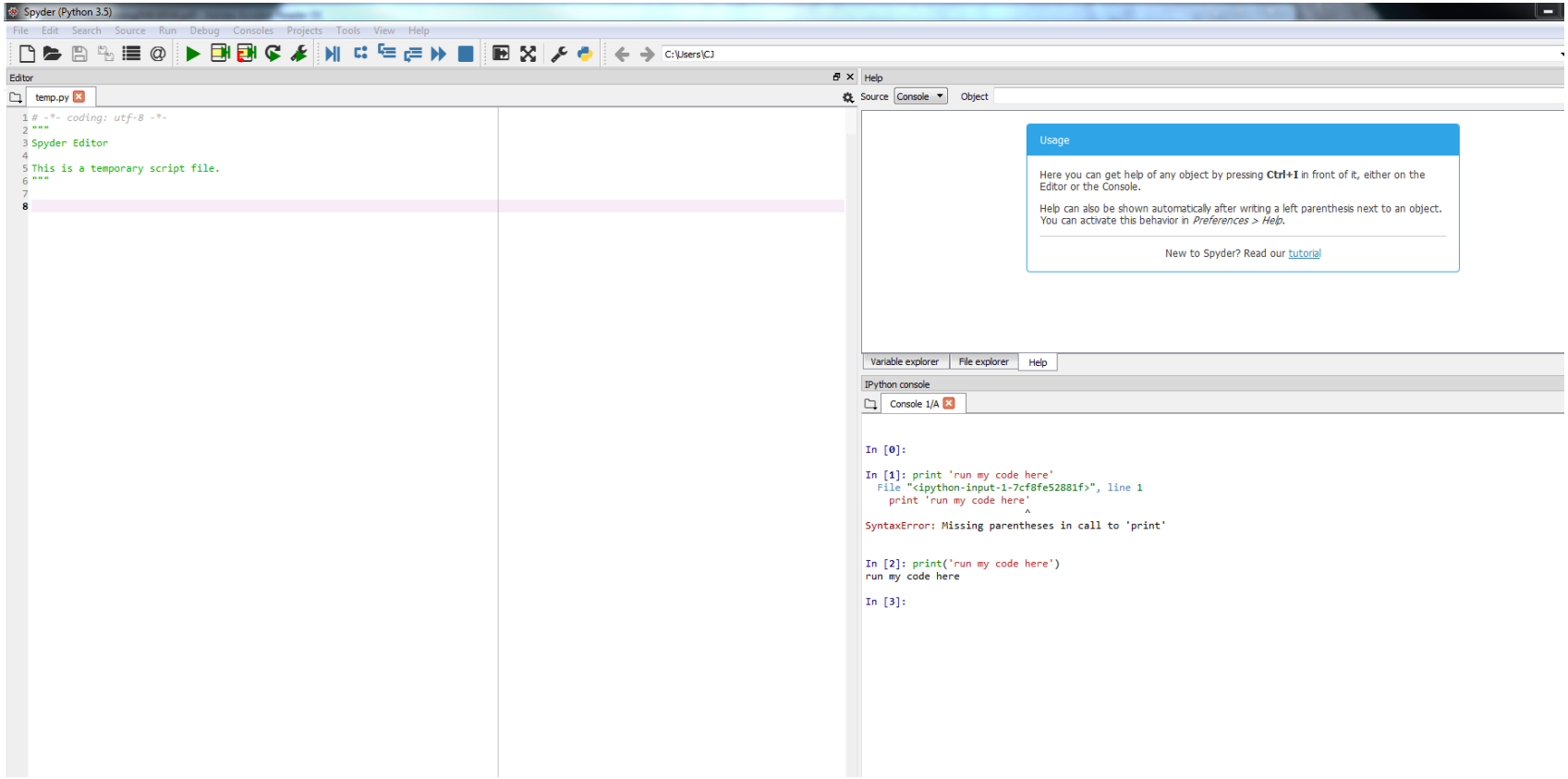
Anaconda - Advanced Options

Checking the top box will add the Anaconda installation to your PATH environment variable. Do this on Windows if you want access to python, ipython, pip, and conda from any command prompt or shell on your system for your user. If you don't check the top box, you can always access these applications from the *Anaconda Prompt*.



On Linux (and maybe macOS) do not check the top box.

Spyder IDE installed with Anaconda



Install Python - Enthought Canopy

With Enthought Canopy you may either install Python 2.7 or Python 3.5
`https://store.enthought.com/downloads/`

v2.1.3 v1.7.4 Documentation

Platform	Python		Released	Size
Linux [64-bit]	2.7	download	2017-06-16	697.8 MB
Linux [64-bit]	3.5	download	2017-06-16	574.8 MB
macOS [64-bit]	2.7	download	2017-06-16	572.1 MB
macOS [64-bit]	3.5	download	2017-06-16	464.0 MB
Windows [64-bit]	2.7	download	2017-06-16	513.8 MB
Windows [32-bit]	2.7	download	2017-06-16	420.9 MB
Windows [64-bit]	3.5	download	2017-06-16	431.3 MB
Windows [32-bit]	3.5	download	2017-06-16	350.2 MB

Canopy no longer sets up your environment variables by default...

In order to run Python from any command prompt/ shell on your Windows computer, I recommend adding the recently installed python directories to your PATH.

`C:\Users\yourUserNameHere\AppData\Local\Enthought\Canopy\edm\envs\User`

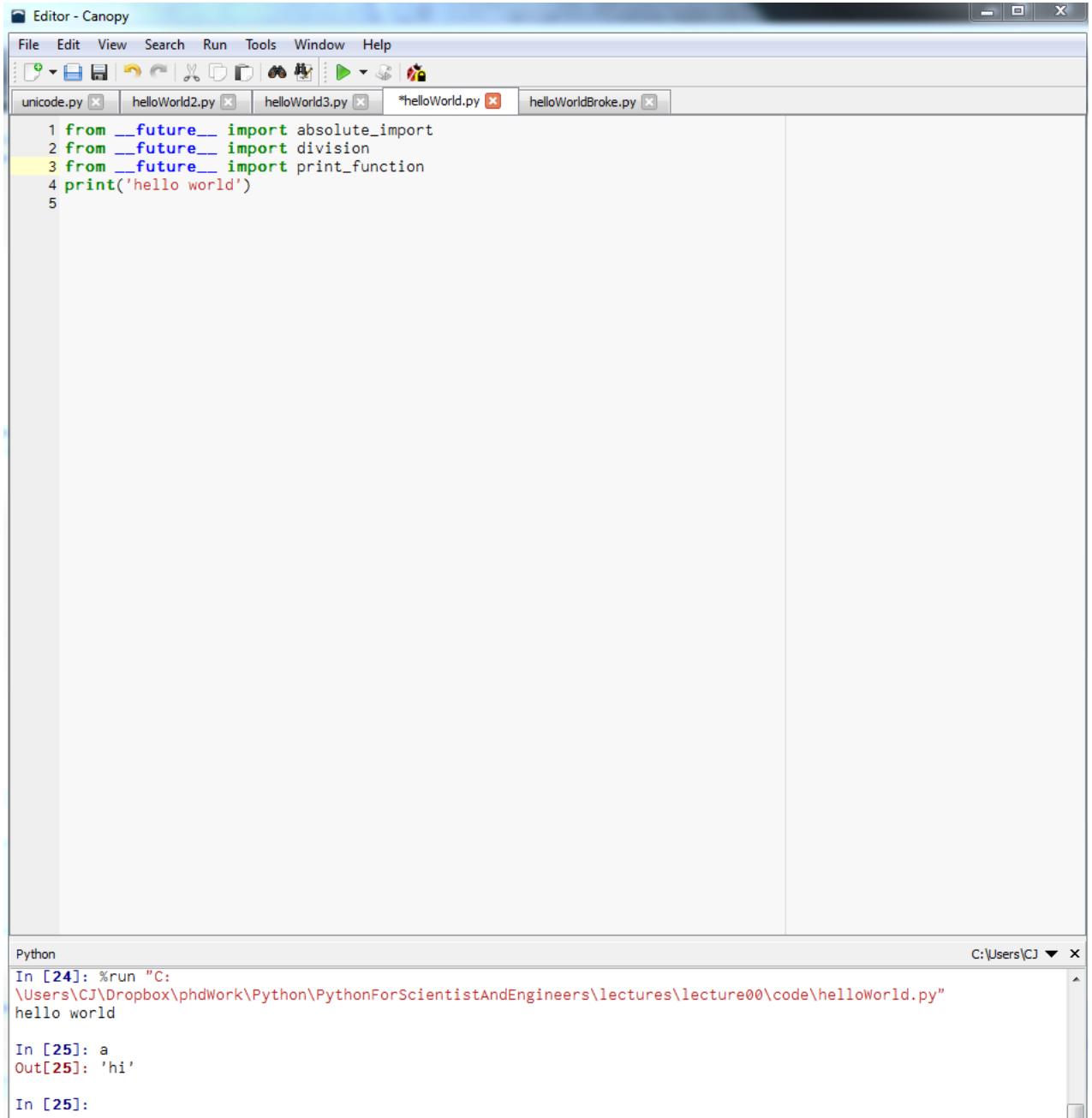
`C:\Users\yourUserNameHere\AppData\Local\Enthought\Canopy\edm\envs\User\scripts`

On Windows 7, 8, and 10 go to System > Advanced System Settings > environment variables.

In Windows environment variables are separated without spaces using ;

(If you didn't check the boxes with Anaconda, you'll need to set up the environment variables yourself)

Canopy editor IDE installed with Enthought Canopy



Anaconda vs Enthought Canopy doesn't matter for this course

... but what does matter is that you have a working Python installation with pre-built scientific libraries.

If you added Python to your PATH - Make sure it works

You can run python from the command line by typing python in your favorite shell/command prompt.

```
cj@Iris:~> python
Python 2.7.13 (default, Jan 03 2017, 17:41:54) [GCC] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>> 
```

On Windows (Open up a command prompt and type python followed by an enter). To open up a command prompt, hit the windows key and type command prompt. You should see a command prompt show up in the search.

If you did not add Python to your PATH

Open up the Anaconda Prompt, and type python followed by an enter to open the python interpreter to make sure it is working.

If you installed Enthought Canopy, open up the editor. Then go to tools on the top ribbon $>$ and open Canopy Prompt. When the command prompt opens type python hit enter.

To exit the Python interpreter, type exit() followed by an enter.

HW 00: due one week from today - turn in online before class starts

This HW will appear as a Quiz in Canvas. If you don't see the canvas Quiz please let me know!

1. Install Python. A) Tell me if you installed Anaconda or Enthought Canopy. B) Tell me which python version you installed (2.7, 3.5, 3.6?).
2. What operating system do you use? (Ex: Windows 7, OS X Yosemite, Ubuntu 14)
3. What previous programming languages are you familiar with? (Hint: only name the ones you have used the most)

Tools used to make this presentation

This presentation was made with Beamer using LaTeX. Texmaker is my favorite cross-platform LaTeX editor. I use Metropolis, a modern Beamer theme. Syntax highlighting is done using Minted and Pygmentex. I use pdfpc (PDF presenter console) to present PDF files.

Links:

- https://en.wikipedia.org/wiki/Beamer_(LaTeX)
- https://www.latex-project.org/
- http://www.xm1math.net/texmaker/
- https://github.com/matze/mtheme
- https://www.ctan.org/pkg/minted?lang=en
- https://www.ctan.org/pkg/pygmentex?lang=en
- https://pdfpc.github.io/