Winkenwerder, Laurel
Module 7. Assignment 7
05/11/2019
https://github.com/LWinkenwerder/IntrotoProg_Python
My Blog: https://www.elwinkenerder.com/2019/05/university-of-washington-intro-to.html

Document Your Knowledge

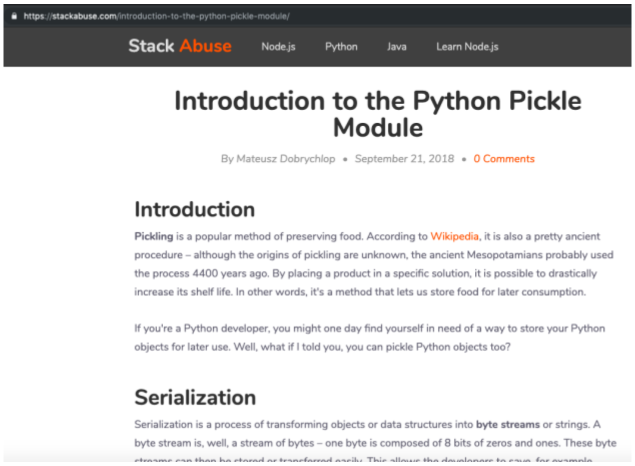
- In this module, we are demonstrating the concepts of pickling and exception handling, which we also did in a previous module.
    - In addition to our Word doc and GitHub repo, we are also publishing our work by making a blog this week.
- This is a great time to start learning about pickling, since my the team I run at work was just talking this week about utilizing this method of "serialization" or "flattening" to store and retrieve some file objects that we need for one of the new apps we are building.
    - Two-thirds of my team (my statisticians) program in R and one-third (my more general data scientists) program in Python, so it was an interesting discussion for the R-programmers to discuss it with the Python programmers. Our R programmers are learning Python and vice versa.
    - We typically read and write data for our apps from our own dedicated databases that have staging tables that hold our data and everyone on the team utilizes SQL extensively.

What is pickling and why should a Python programmer use it?

- **Pickling is a way to store and retrieve objects for future use.**
    - What is a Python object?
        - Wikipedia provided the most straightforward definition of what an object is that a I as a newbie could follow:
            - "Object-oriented programming (OOP) is a programming paradigm based on the concept of **"objects", which can contain data, in the form of fields (often known as attributes), and code, in the form of procedures (often known as methods)**. A feature of objects is an object's procedures that can access and often modify the data fields of the object with which they are associated (objects have a notion of "this" or "self"). In OOP, computer programs are designed by making them out of objects that interact with one another.[1][2] OOP languages are diverse, but the most popular ones are class-based, meaning that objects are instances of classes, which also determine their types." (source: https://en.wikipedia.org/wiki/Object-oriented_programming)

Why do we want to use pickling?

- "The purpose of pickling is to translate data into a format that can be transferred from RAM to disk." (souce: https://stackabuse.com/introduction-to-the-python-pickle-module/)
- I reviewed a number of sources beyond our class YouTube videos and our textbook. The source that I thought was most useful and clear was actually a posting in StackAbuse because it gave a very detailed explanation of why I would need to pickle and unpickle data and this explanation was more in-line with the real-world examples I see on my own team.


https://stackabuse.com/introduction-to-the-python-pickle-module/
Stack Abuse
Node.js
Python
Java
Learn Node.js

Introduction to the Python Pickle Module

By Mateusz Dobrychlop • September 21, 2018 • 0 Comments

Introduction

Pickling is a popular method of preserving food. According to Wikipedia, it is also a pretty ancient procedure – although the origins of pickling are unknown, the ancient Mesopotamians probably used the process 4400 years ago. By placing a product in a specific solution, it is possible to drastically increase its shelf life. In other words, it's a method that lets us store food for later consumption.

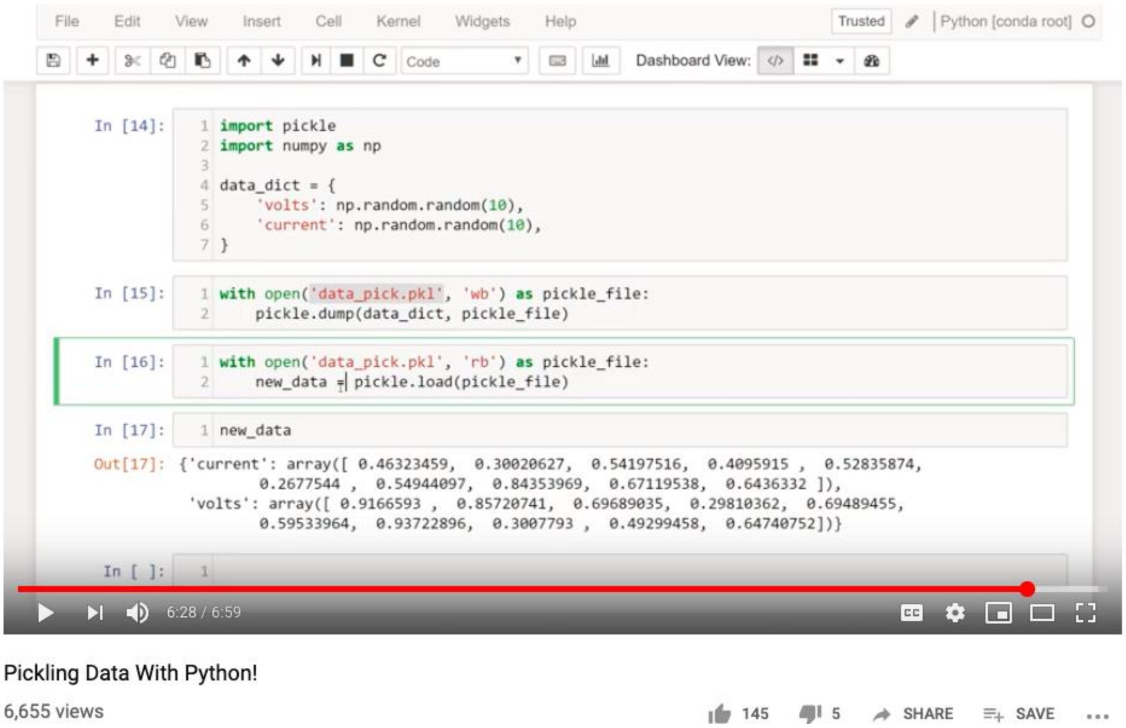
If you're a Python developer, you might one day find yourself in need of a way to store your Python objects for later use. Well, what if I told you, you can pickle Python objects too?

Serialization

Serialization is a process of transforming objects or data structures into **byte streams** or strings. A byte stream is, well, a stream of bytes – one byte is composed of 8 bits of zeros and ones. These byte streams can then be stored or transferred easily. This allows the developers to save, for example

(Source: https://stackabuse.com/introduction-to-the-python-pickle-module/)

- After figuring out what pickling is and where I would apply it, I went out in search of some data to pickle and unpickle.
    - I came across this great YouTube video that incorporated an import of the NumPy package to generate random data: https://www.youtube.com/watch?v=Pl4Hp8qwwes



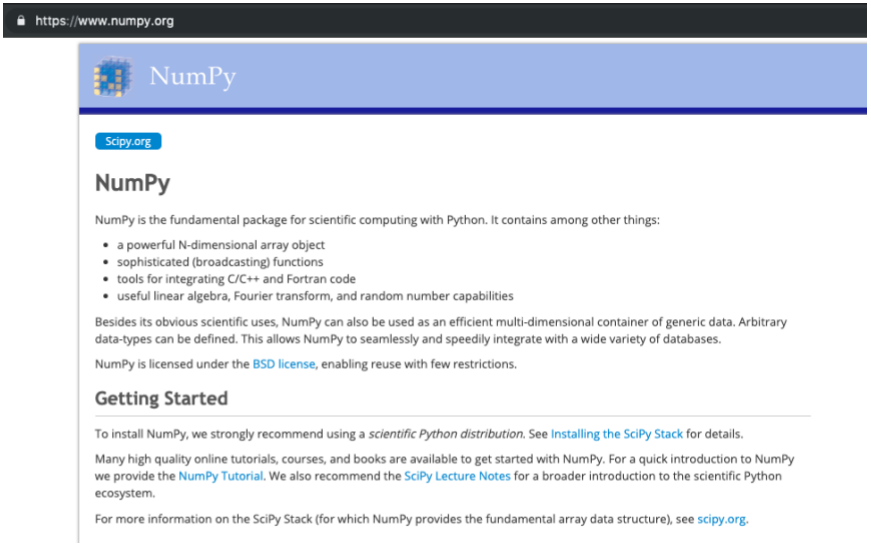
```
In [14]:  import pickle
          import numpy as np

          data_dict = {
              'volts': np.random.random(10),
              'current': np.random.random(10),
          }

In [15]:  with open('data_pick.pkl', 'wb') as pickle_file:
              pickle.dump(data_dict, pickle_file)

In [16]:  with open('data_pick.pkl', 'rb') as pickle_file:
              new_data = pickle.load(pickle_file)

In [17]:  new_data

Out[17]: {'current': array([ 0.46323459,  0.30020627,  0.54197516,  0.4095915 ,  0.52835874,
                 0.2677544 ,  0.54944097,  0.84353969,  0.67119538,  0.6436332 ]),
          'volts': array([ 0.9166593 ,  0.85720741,  0.69689035,  0.29810362,  0.69489455,
                 0.59533964,  0.93722896,  0.3007793 ,  0.49299458,  0.64740752])}

In [ ]:
```

Pickling Data With Python!

6,655 views

- My team uses the NumPy package, so I wanted to use this opportunity to learn what it is, get experience with it, as well as get experience installing my own packages.
    - So, for this assignment, I decided to attempt pickling by replicating what I saw in the above YouTube video to generate some random data into a dictionary and using the NumPy package.

- Researching NumPy:



https://www.numpy.org

NumPy

Scipy.org

NumPy

NumPy is the fundamental package for scientific computing with Python. It contains among other things:

- a powerful N-dimensional array object
- sophisticated (broadcasting) functions
- tools for integrating C/C++ and Fortran code
- useful linear algebra, Fourier transform, and random number capabilities

Besides its obvious scientific uses, NumPy can also be used as an efficient multi-dimensional container of generic data. Arbitrary data-types can be defined. This allows NumPy to seamlessly and speedily integrate with a wide variety of databases.

NumPy is licensed under the BSD license, enabling reuse with few restrictions.

Getting Started

To install NumPy, we strongly recommend using a *scientific Python distribution*. See Installing the SciPy Stack for details.

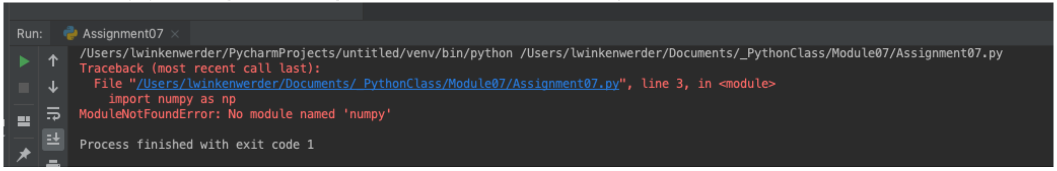
Many high quality online tutorials, courses, and books are available to get started with NumPy. For a quick introduction to NumPy we provide the NumPy Tutorial. We also recommend the SciPy Lecture Notes for a broader introduction to the scientific Python ecosystem.

For more information on the SciPy Stack (for which NumPy provides the fundamental array data structure), see scipy.org.

(Source: https://www.numpy.org/)

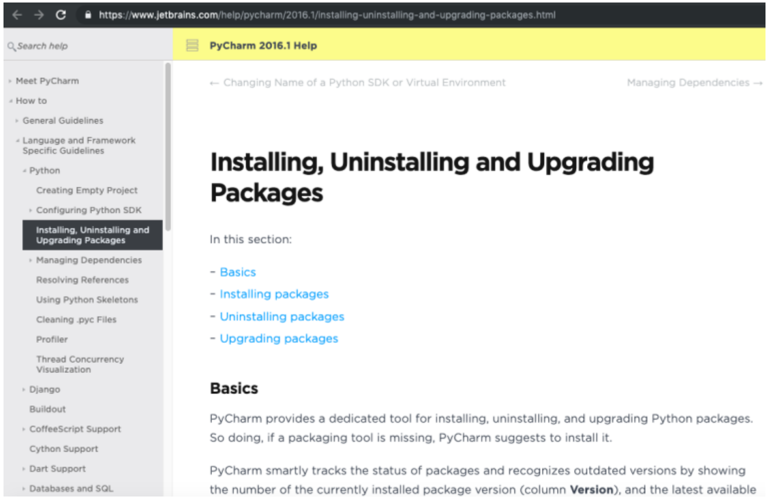
- This was a great experience because it led me down quite a rabbit path of:
    - Setting up my Python interpreter correctly (I had a number of conflicting interpreters which kept generating an error) when I tried to install NumPy,
    - Installing pip in my Terminal (I was initially unable to install NumPy and need to install pip apparently first), and
    - Finally installing the NumPy package so that I could actually import it.

- No NumPy package! This generated an error object called “ModuleNotFoundError”:



```
/Users/lwinkenwerder/PycharmProjects/untitled/venv/bin/python /Users/lwinkenwerder/Documents/_PythonClass/Module07/Assignment07.py
Traceback (most recent call last):
  File "/Users/lwinkenwerder/Documents/_PythonClass/Module07/Assignment07.py", line 3, in <module>
    import numpy as np
ModuleNotFoundError: No module named 'numpy'

Process finished with exit code 1
```

- Learning to install a package using the JetBrains' PyCharm help features:



Installing, Uninstalling and Upgrading Packages

In this section:

- Basics
- Installing packages
- Uninstalling packages
- Upgrading packages

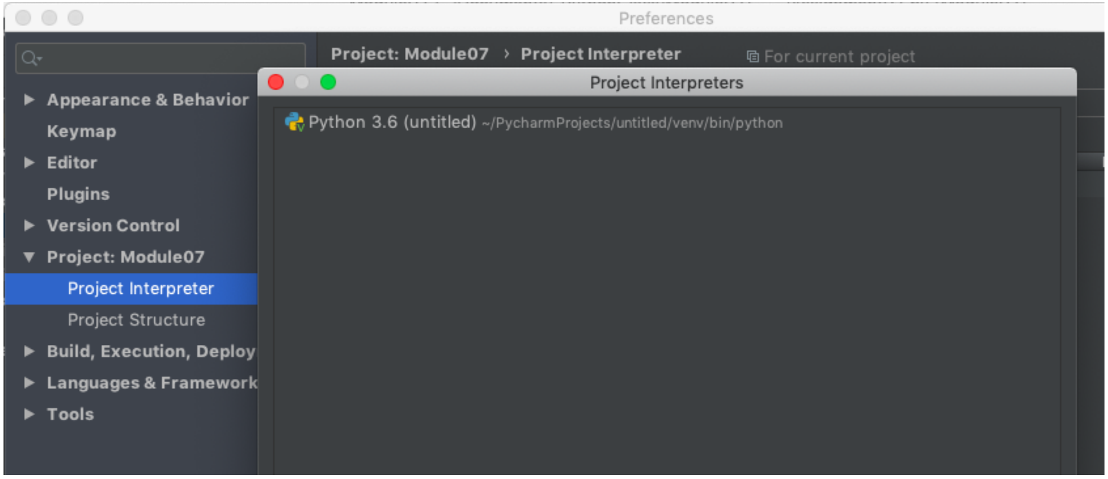
Basics

PyCharm provides a dedicated tool for installing, uninstalling, and upgrading Python packages. So doing, if a packaging tool is missing, PyCharm suggests to install it.

PyCharm smartly tracks the status of packages and recognizes outdated versions by showing the number of the currently installed package version (column **Version**), and the latest available

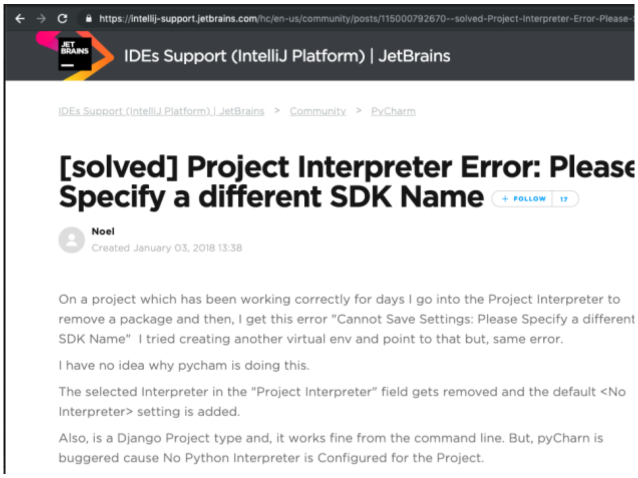
(Source: https://www.jetbrains.com/help/pycharm/2016.1/installing-uninstalling-and-upgrading-packages.html)

- At first, something was wrong with my interpreter (in this screenshot, I have finally cleaned it up):



Python 3.6 (untitled) ~/PycharmProjects/untitled/venv/bin/python

- When I first try to install the NumPy package, I keep getting a “Project Interpreter Error”:



https://intellij-support.jetbrains.com/hc/en-us/community/posts/115000792670--solved-Project-Interpreter-Error-Please-

IDEs Support (IntelliJ Platform) | JetBrains

IDEs Support (IntelliJ Platform) | JetBrains > Community > PyCharm

[solved] Project Interpreter Error: Please Specify a different SDK Name

Noel
Created January 03, 2018 13:38

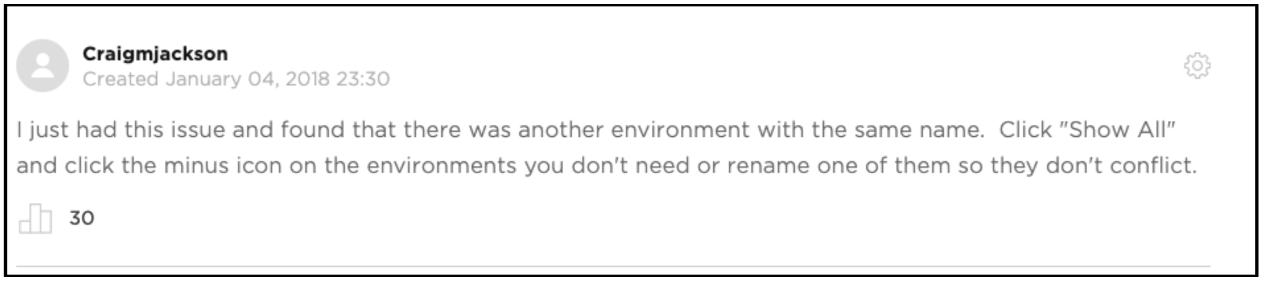
On a project which has been working correctly for days I go into the Project Interpreter to remove a package and then, I get this error "Cannot Save Settings: Please Specify a different SDK Name" I tried creating another virtual env and point to that but, same error.

I have no idea why pycham is doing this.

The selected Interpreter in the "Project Interpreter" field gets removed and the default <No Interpreter> setting is added.

Also, is a Django Project type and, it works fine from the command line. But, pyCharn is buggered cause No Python Interpreter is Configured for the Project.

- Turns out, it may be that I have too many conflicting interpreters that I need to clean up. So, I google around until I find a possible solution that I can replicate:

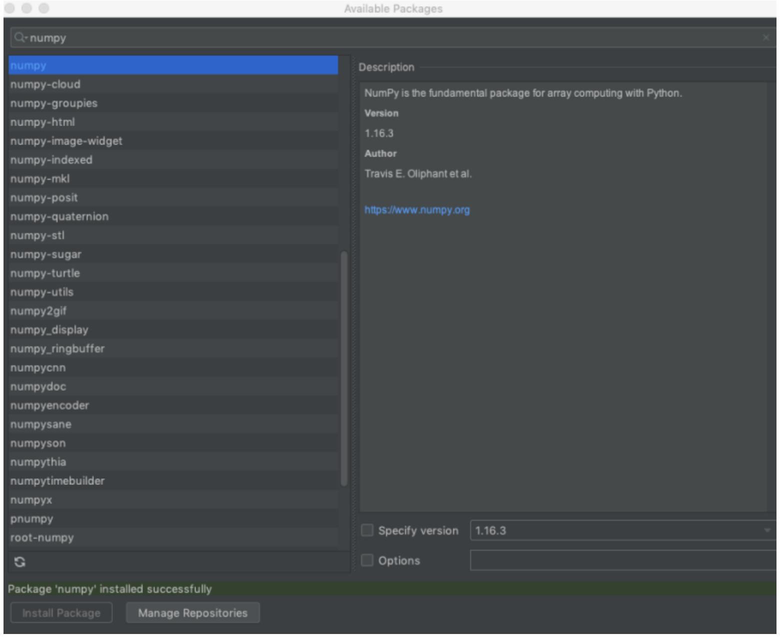


Craigmjackson
Created January 04, 2018 23:30

I just had this issue and found that there was another environment with the same name. Click "Show All" and click the minus icon on the environments you don't need or rename one of them so they don't conflict.

(Source: https://intellij-support.jetbrains.com/hc/en-us/community/posts/115000792670--solved-Project-Interpreter-Error-Please-Specify-a-different-SDK-Name)

- I finally can install NumPy and it installs successfully!



numpy

numpy
numpy-cloud
numpy-groupies
numpy-html
numpy-image-widget
numpy-indexed
numpy-mkl
numpy-posit
numpy-quaternion
numpy-stl
numpy-sugar
numpy-turtle
numpy-utils
numpy2gif
numpy_display
numpy_ringbuffer
numpycnn
numpydoc
numpyencoder
numpysane
numpyson
numpythia
numpytimebuilder
numpyx
pnumpy
root-numpy

Description

NumPy is the fundamental package for array computing with Python.

Version

1.16.3

Author

Travis E. Oliphant et al.

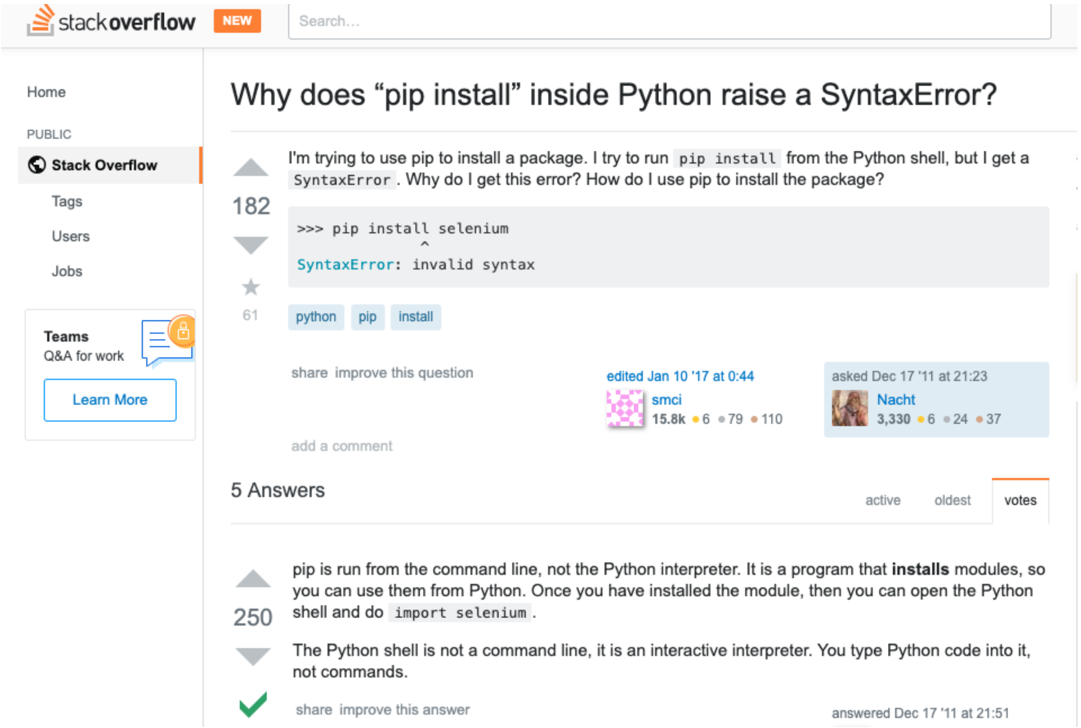
https://www.numpy.org

□ Specify version 1.16.3

□ Options

Package 'numpy' installed successfully

- But even after all of these mini-successes, I still encounter problems – every time I try to install NumPy I get the same error: “SyntaxError -invalid Syntax” exactly like the example below:



Why does “pip install” inside Python raise a SyntaxError?

182

I'm trying to use pip to install a package. I try to run `pip install` from the Python shell, but I get a `SyntaxError`. Why do I get this error? How do I use pip to install the package?

```
>>> pip install selenium
              ^
SyntaxError: invalid syntax
```

61

python pip install

edited Jan 10 '17 at 0:44
smci
15.8k ●6 ●79 ●110

asked Dec 17 '11 at 21:23
Nacht
3,330 ●6 ●24 ●37

5 Answers

250

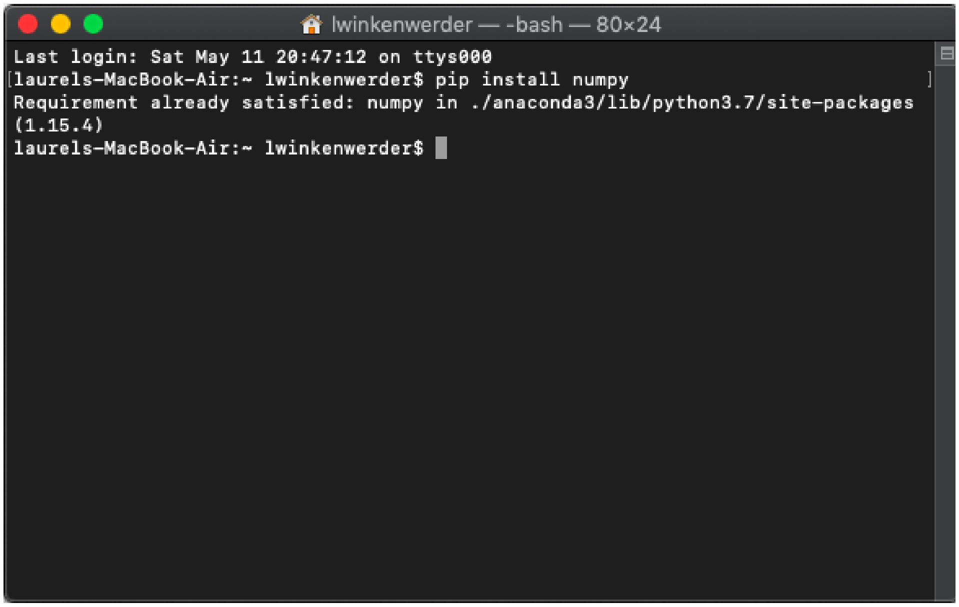
pip is run from the command line, not the Python interpreter. It is a program that **installs** modules, so you can use them from Python. Once you have installed the module, then you can open the Python shell and do `import selenium`.

The Python shell is not a command line, it is an interactive interpreter. You type Python code into it, not commands.

answered Dec 17 '11 at 21:51

- So, I go find what I think is the right command line terminal and replicate what I see in this StackOverflow thread.

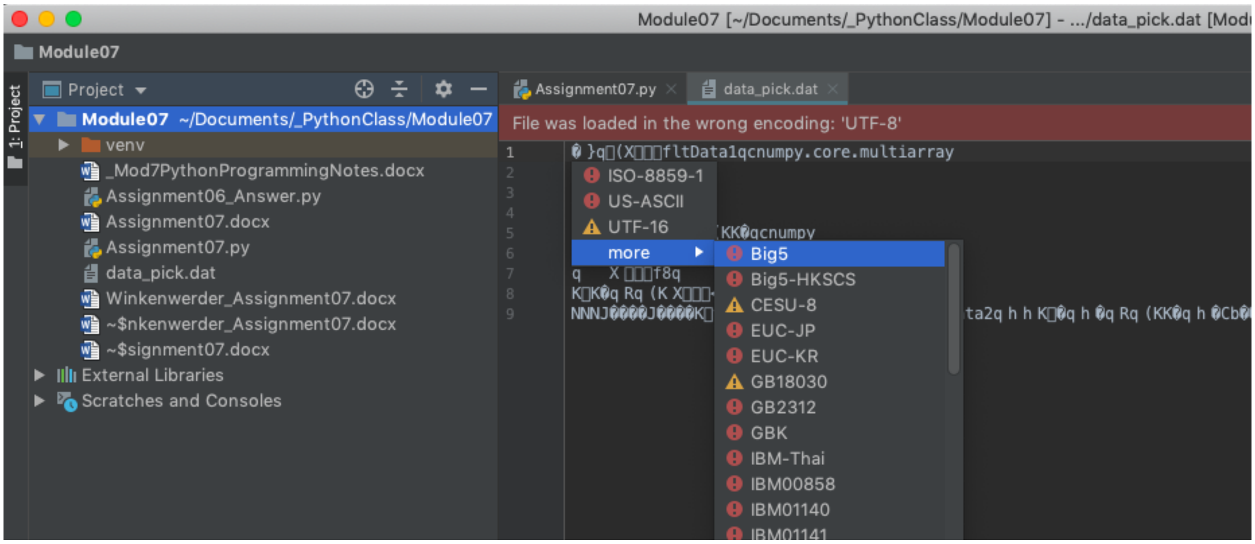
- I think I may have conflicting versions of Python and I am not 100% sure I have pip installed from the command line, and I am still not exactly sure what I am doing or why, but I get this message (below) and decide to try my import NumPy script again in PyCharm.



```
Last login: Sat May 11 20:47:12 on ttys000
[laurels-MacBook-Air:~ lwinkenwerder$ pip install numpy
Requirement already satisfied: numpy in ./anaconda3/lib/python3.7/site-packages
(1.15.4)
laurels-MacBook-Air:~ lwinkenwerder$
```

- Somehow it finally works and I am relieved to be able to move on and try out pickling. I then finish coding up my pickling and unpickling process with my randomly generated data that I learn to generate by re-watching my YouTube video mentioned above a few times, as well as reading my text book and watching Randal's YouTube videos, plus doing more searching around the internet.

- However, I quickly run into another snafu that I think is interesting:
    - My data writing to my binary .dat file generates an error: "File was loaded in the wrong encoding: 'UTF-8'"

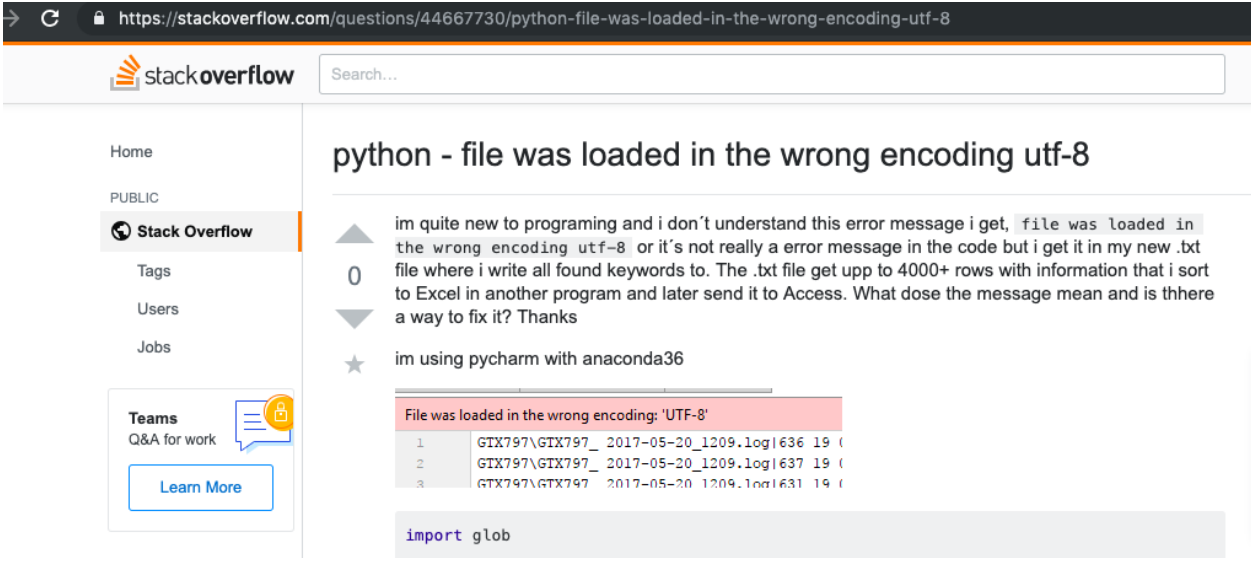


Module07 [~/Documents/_PythonClass/Module07] - .../data_pick.dat [Modu

File was loaded in the wrong encoding: 'UTF-8'

```
� }q(XfltData1qcnumpy.core.multiarray
(KK�qcnumpy
q    X f8q
KK�q Rq (K X
NNNJ����J����K                    ta2q h h K�q h �q Rq (KK�q h �Cb�
```

- So, I go investigating the internet for a solution by googling the error:



https://stackoverflow.com/questions/44667730/python-file-was-loaded-in-the-wrong-encoding-utf-8

python - file was loaded in the wrong encoding utf-8

0

im quite new to programing and i don´t understand this error message i get, `file was loaded in the wrong encoding utf-8` or it´s not really a error message in the code but i get it in my new .txt file where i write all found keywords to. The .txt file get upp to 4000+ rows with information that i sort to Excel in another program and later send it to Access. What dose the message mean and is thhere a way to fix it? Thanks

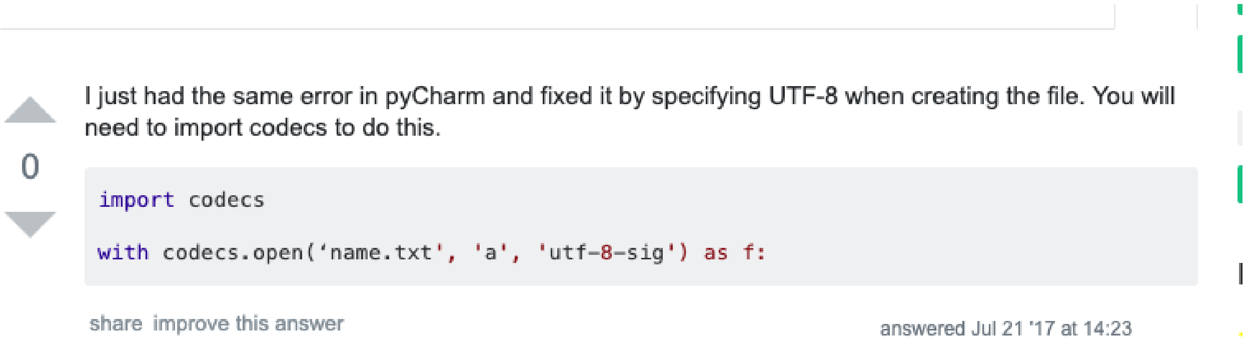
im using pycharm with anaconda36

File was loaded in the wrong encoding: 'UTF-8'

```
GTX797\GTX797_ 2017-05-20_1209.log|636 19 (
GTX797\GTX797_ 2017-05-20_1209.log|637 19 (
GTX797\GTX797_ 2017-05-20_1209.log|631 19 (
```

```
import glob
```

- I find a solution on StackOverflow and try to replicate it by adding in an import codecs package and specify that the file can be UTF-8 when I create my own file, similar to what I found here:



I just had the same error in pyCharm and fixed it by specifying UTF-8 when creating the file. You will need to import codecs to do this.

```
import codecs

with codecs.open(‘name.txt', 'a', 'utf-8-sig') as f:
```

- Here is my project in PyCharm (Assignment07.py) and my data_pick.dat file, below:

- My script is written with these steps:
    - Import pickle & Import codecs (so I can fix the error on the type of output the binary file can create)

    - With codecs, I open the file I will pickle data to and specify that the file will be in UTF-8.
        - I learned in the YouTube pickle video that I got this code idea from that the ‘with’ function is a very common function for Python programmers.
        - So far in this class and doing my research, I have seen a ton of instances where programmers use ‘with’, so I decide in this assignment to take it for a spin myself.

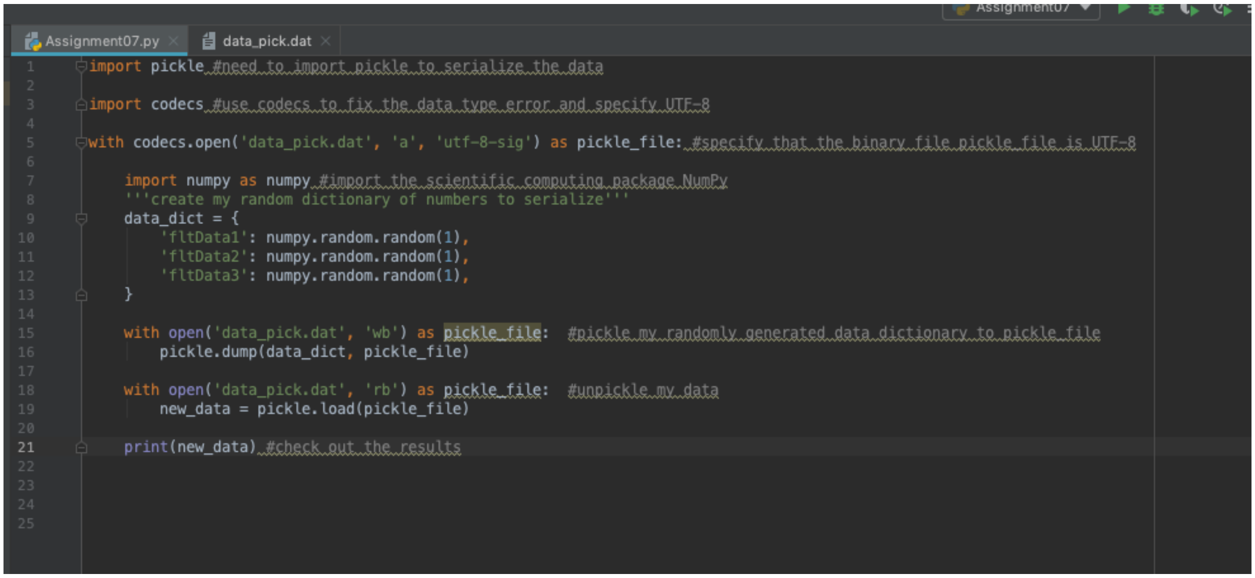
    - I then import the NumPy package, which I will later use to generate some random data into a dictionary to get some data to pickle and write to my .dat file
        - I learn that it is common to write the code as:
            - Import numpy as np
                - Using ‘np’ is just a shortened version of NumPy
                - I wrote out NumPy so I don’t get confused in the future.

    - I randomly generate three variables of data and put them into my data_dict dictionary – I generate only one data point for each of the three variables (which will be floating integers), but I can change that number to anything I want.
        - I just chose 1 data point because the output was easier to read once I print the data to check and see what I pickled.

    - Next, I pickle and store the data to pickle_file using the pickle.dump() function
        - I learned from my textbook and the YouTube videos, that my Pickle.dump() function requires two arguments – the data to pickle and the file to store it to (pickle_file).

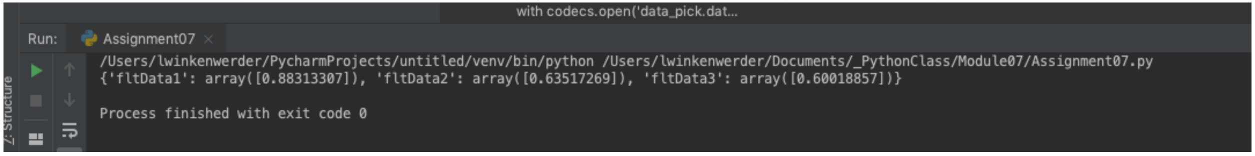
- My data is then written (‘wb’ or “write binary”) to pickle_data and writes it all as one object to the binary file, which I have named data_pick.dat.

- Now I want to read (‘rb’) the data from the binary file and unpickle it:
    - I create a new variable and retrieve the data and unpickle it using the pickle.load() function – this function needs one argument – the file I had previously stored my pickled data in (pickle_file).
    - I then print the new_data variable so I can see what I unpickled.



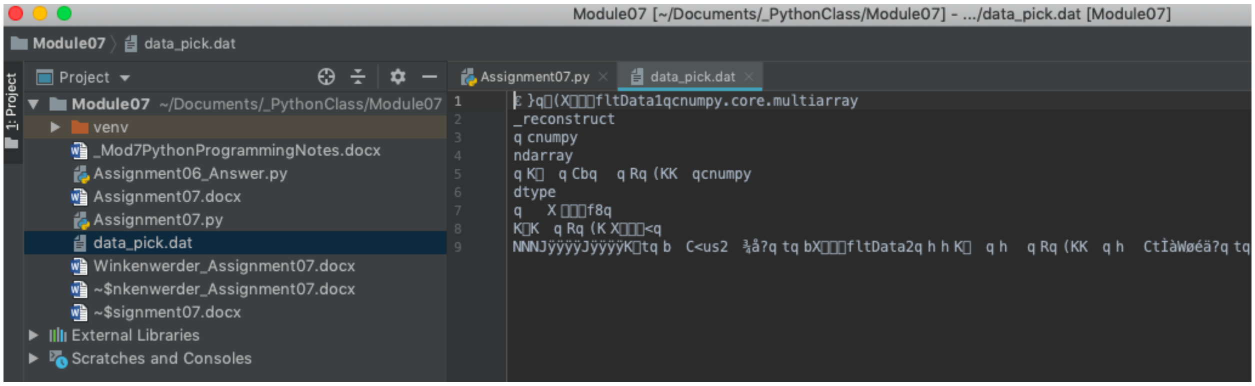
```python
import pickle #need to import pickle to serialize the data

import codecs #use codecs to fix the data type error and specify UTF-8

with codecs.open('data_pick.dat', 'a', 'utf-8-sig') as pickle_file: #specify that the binary file pickle_file is UTF-8

    import numpy as numpy #import the scientific computing package NumPy
    '''create my random dictionary of numbers to serialize'''
    data_dict = {
        'fltData1': numpy.random.random(1),
        'fltData2': numpy.random.random(1),
        'fltData3': numpy.random.random(1),
    }

    with open('data_pick.dat', 'wb') as pickle_file:  #pickle my randomly generated data dictionary to pickle_file
        pickle.dump(data_dict, pickle_file)

    with open('data_pick.dat', 'rb') as pickle_file:  #unpickle my data
        new_data = pickle.load(pickle_file)

    print(new_data) #check out the results
```

- With NumPy it looks like I randomly generated some arrays of data to pickle and unpickle!



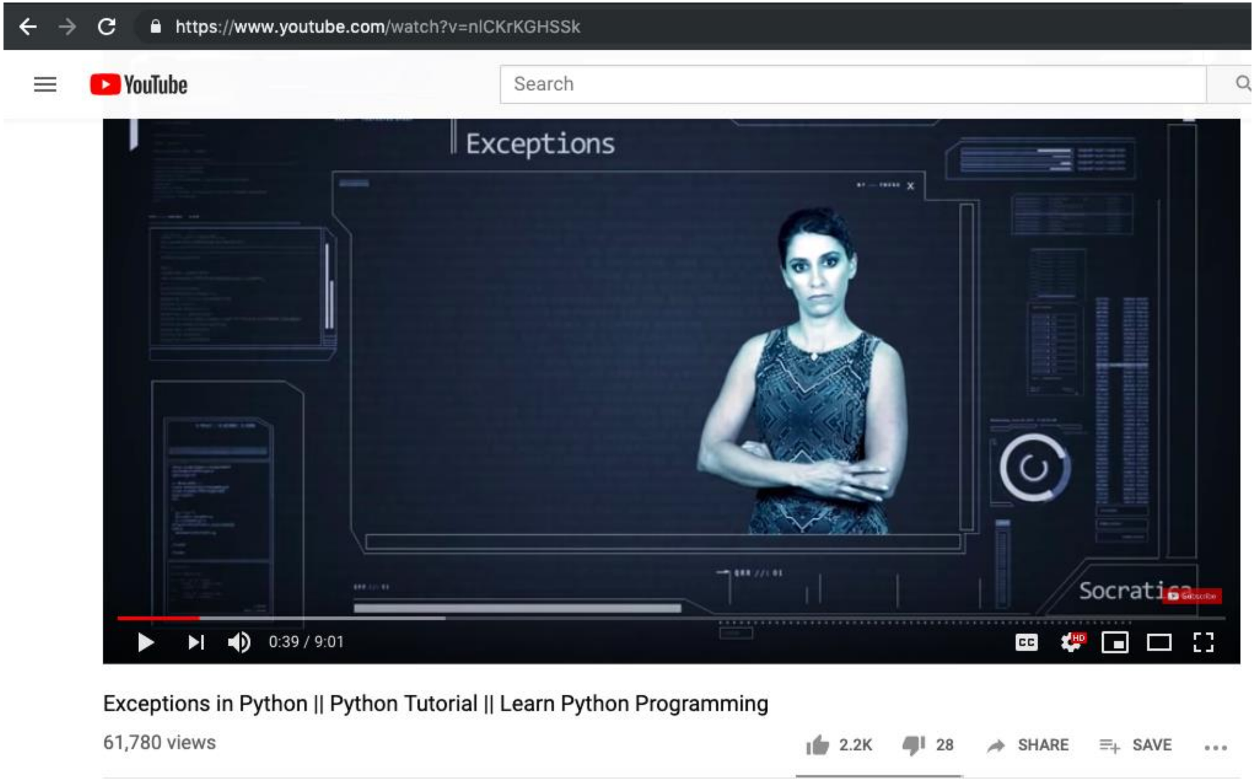
```
/Users/lwinkenwerder/PycharmProjects/untitled/venv/bin/python /Users/lwinkenwerder/Documents/_PythonClass/Module07/Assignment07.py
{'fltData1': array([0.88313307]), 'fltData2': array([0.63517269]), 'fltData3': array([0.60018857])}

Process finished with exit code 0
```

- My binary file is encrypted, just as we learned would happen from Randal’s YouTube videos and by adding in the import codec and specifying UTF-8, I no longer get that red error message pop up when I run my code – so I think it works!


Module07 [~/Documents/_PythonClass/Module07] - .../data_pick.dat [Module07]
Module07
data_pick.dat
1: Project
Project
Module07 ~/Documents/_PythonClass/Module07
venv
_Mod7PythonProgrammingNotes.docx
Assignment06_Answer.py
Assignment07.docx
Assignment07.py
data_pick.dat
Winkenwerder_Assignment07.docx
~$nkenwerder_Assignment07.docx
~$signment07.docx
External Libraries
Scratches and Consoles
Assignment07.py
data_pick.dat

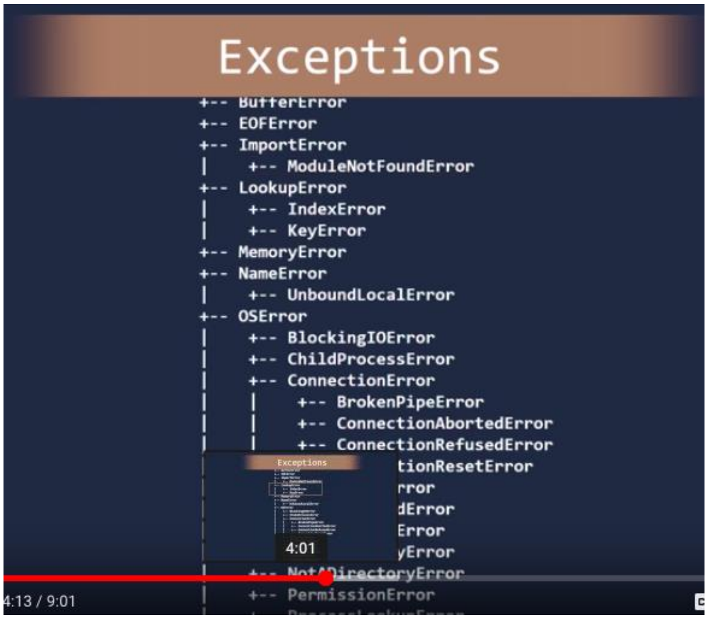
```
€ }q(XfltData1qcnumpy.core.multiarray
_reconstruct
q cnumpy
ndarray
q K  q Cbq   q Rq (KK  qcnumpy
dtype
q    X f8q
KK   q Rq (K X<q
NNNJÿÿÿÿJÿÿÿÿKtq b   C<us2  ¾å?q tq bXfltData2q h h K   q h    q Rq (KK   q h    CtÌàWøéä?q tq
```

- Now, I need to add in some Error Handling. Luckily, I have been dealing with errors for many days and many hours performing this assignment, so it gives me an idea to try using error handling for attempting to import packages I don't have.
- I found a great YouTube video that makes Error Handling make a lot of sense because I am starting to understand the taxonomic nature of classes in object-oriented programming … this is critical for me, because I need to understand the overarching structure of how Python works for me to understand many of the concepts we have learned thus far in isolation.



Exceptions in Python || Python Tutorial || Learn Python Programming

61,780 views

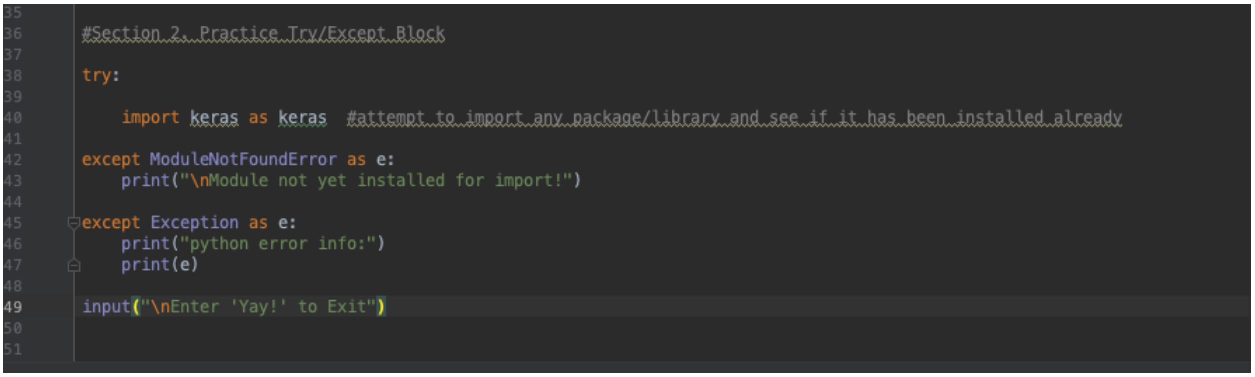


Exceptions

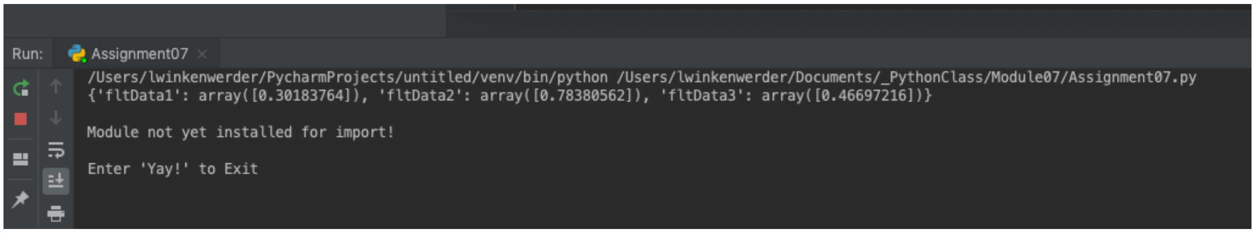
```
+-- BufferError
+-- EOFError
+-- ImportError
|    +-- ModuleNotFoundError
+-- LookupError
|    +-- IndexError
|    +-- KeyError
+-- MemoryError
+-- NameError
|    +-- UnboundLocalError
+-- OSError
|    +-- BlockingIOError
|    +-- ChildProcessError
|    +-- ConnectionError
|    |    +-- BrokenPipeError
|    |    +-- ConnectionAbortedError
|    |    +-- ConnectionRefusedError
                    tionResetError
                    rror
                    dError
                    Error
                    yError
|    +-- NotADirectoryError
|    +-- PermissionError
```

(Source: https://www.youtube.com/watch?v=nlCKrKGHSSk)

- In my Assignment07 script, I then added a section to run a Try/Except block to test for errors that will occur if Python libraries/packages are not yet installed, but the user is trying to import them into their script.
- I tested out importing keras, since my team added keras awhile back to our repo at work and that one particular name was fresh in my mind (I don’t know what it does yet, but will research it).
  - I know I don’t have it installed locally, so I expect it will hit my first ‘except’ and generate an error object, which it does!
- Unfortunately, I am traveling to four states this week and this is all I have time for, so I will upload my homework, post my blog and Git, and bid adieu to this week’s homework!



```

#Section 2, Practice Try/Except Block

try:

    import keras as keras  #attempt to import any package/library and see if it has been installed already

except ModuleNotFoundError as e:
    print("\nModule not yet installed for import!")

except Exception as e:
    print("python error info:")
    print(e)

input("\nEnter 'Yay!' to Exit")


```



```
/Users/lwinkenwerder/PycharmProjects/untitled/venv/bin/python /Users/lwinkenwerder/Documents/_PythonClass/Module07/Assignment07.py
{'fltData1': array([0.30183764]), 'fltData2': array([0.78380562]), 'fltData3': array([0.46697216])}

Module not yet installed for import!

Enter 'Yay!' to Exit
```