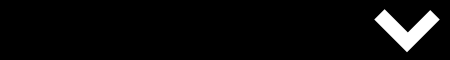
ITP 449

PYTHON Packages

Lecture 3



Topics

- Packages
  - Numpy
  - Pandas
- Dataframes

What are packages?

Packages are code collections (written and shared by many in the community)

- Hundreds of free packages are available
- Packages are written to provide functions (and methods) that allow for easy coding
- Each packages has many python scripts e.g. *1.py, 2.py, 3.py* etc.
- Each script is called a *module* (e.g. *random*)

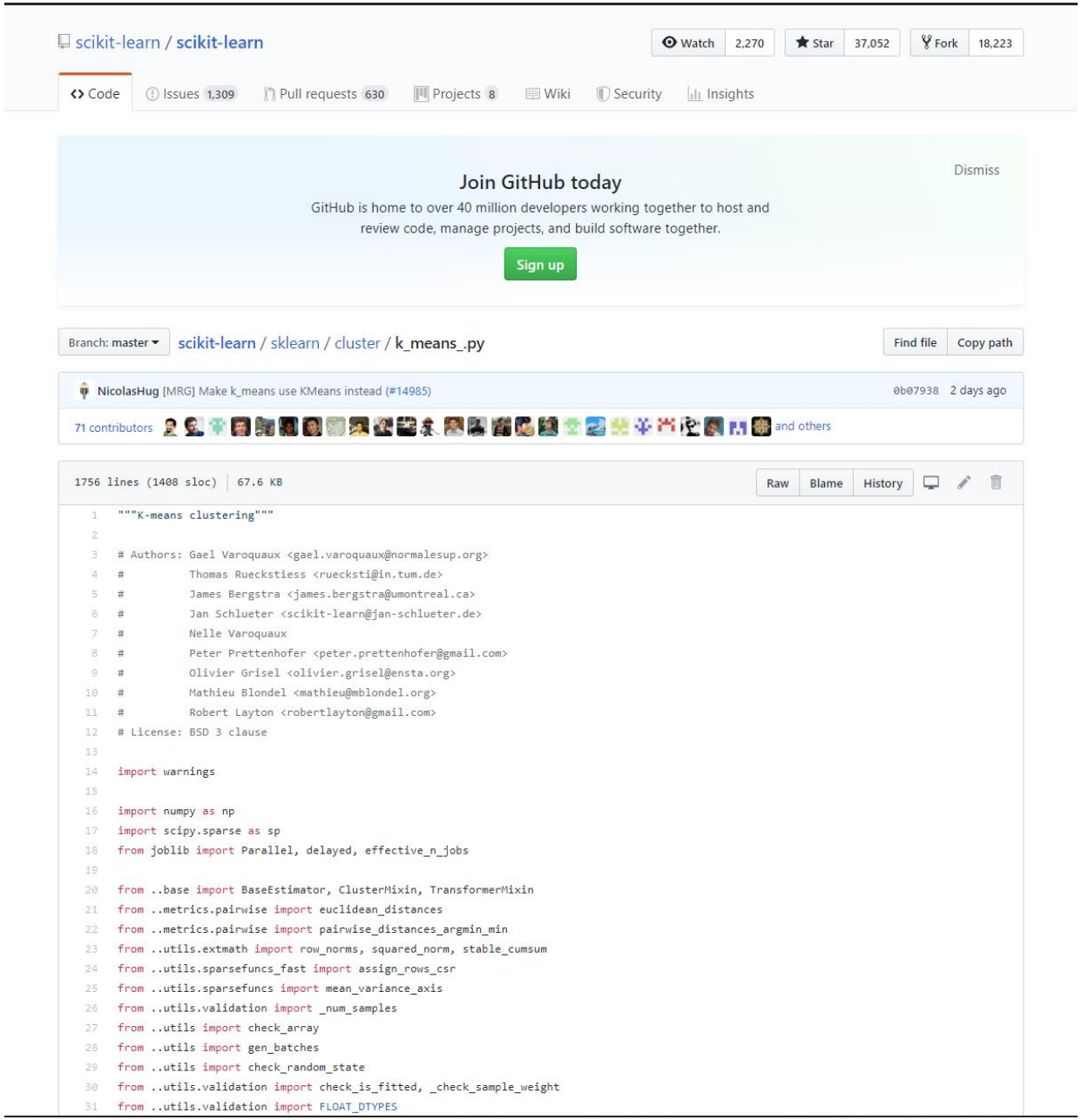
Examples of packages

- **Numpy**: for scientific use, arrays https://numpy.org/
- **Pandas**: time series analysis, plotting https://pandas.pydata.org/pandas-docs/stable/
- **Scipy**: math, science, engineering https://www.scipy.org/
- **Matplotlib**: plotting https://matplotlib.org/
- **Sci-kit**: Machine learning https://scikit-learn.org/stable/

- You can find packages here - https://pypi.org/
- Packages may be dependent on other packages. Prerequisites need to installed first

You can view the source code of packages on github repository

e.g. https://github.com/scikit-learn/scikit-learn



How to install packages?

First *install* the package on your computer

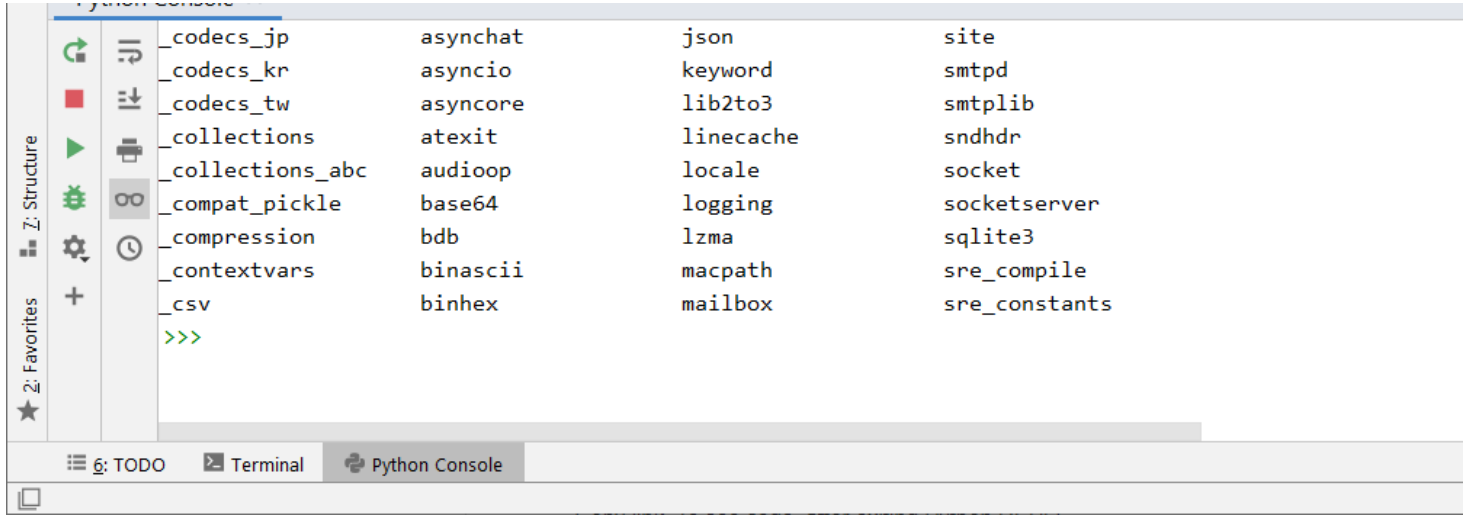
Then *import* it into your python script

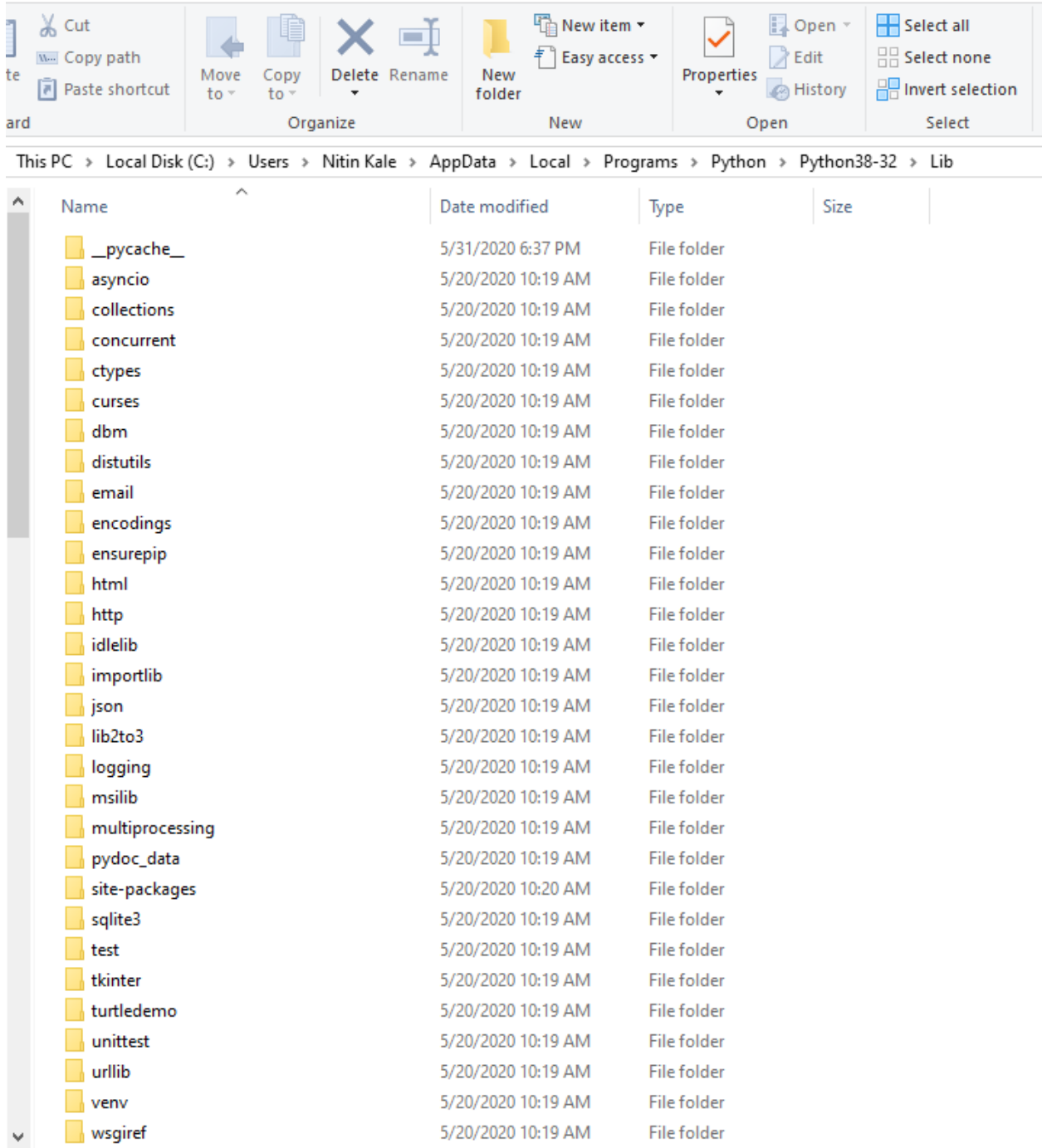
PIP (package installer for python) is a package used to install other packages
https://pip.pypa.io/en/stable/

PIP goes to https://pypi.org/ and downloads packages
Then installs them on your computer

You *don't need to install PIP* on Python3 since it comes with it.

To get a list of packages available on your computer, go to python console
And type
help('modules')




Cut
Copy path
Paste shortcut
Move to
Copy to
Delete
Rename
Organize
New item
Easy access
New folder
New
Properties
Open
Edit
History
Open
Select all
Select none
Invert selection
Select
This PC › Local Disk (C:) › Users › Nitin Kale › AppData › Local › Programs › Python › Python38-32 › Lib
Name
Date modified
Type
Size
__pycache__ 5/31/2020 6:37 PM File folder
asyncio 5/20/2020 10:19 AM File folder
collections 5/20/2020 10:19 AM File folder
concurrent 5/20/2020 10:19 AM File folder
ctypes 5/20/2020 10:19 AM File folder
curses 5/20/2020 10:19 AM File folder
dbm 5/20/2020 10:19 AM File folder
distutils 5/20/2020 10:19 AM File folder
email 5/20/2020 10:19 AM File folder
encodings 5/20/2020 10:19 AM File folder
ensurepip 5/20/2020 10:19 AM File folder
html 5/20/2020 10:19 AM File folder
http 5/20/2020 10:19 AM File folder
idlelib 5/20/2020 10:19 AM File folder
importlib 5/20/2020 10:19 AM File folder
json 5/20/2020 10:19 AM File folder
lib2to3 5/20/2020 10:19 AM File folder
logging 5/20/2020 10:19 AM File folder
msilib 5/20/2020 10:19 AM File folder
multiprocessing 5/20/2020 10:19 AM File folder
pydoc_data 5/20/2020 10:19 AM File folder
site-packages 5/20/2020 10:20 AM File folder
sqlite3 5/20/2020 10:19 AM File folder
test 5/20/2020 10:19 AM File folder
tkinter 5/20/2020 10:19 AM File folder
turtledemo 5/20/2020 10:19 AM File folder
unittest 5/20/2020 10:19 AM File folder
urllib 5/20/2020 10:19 AM File folder
venv 5/20/2020 10:19 AM File folder
wsgiref 5/20/2020 10:19 AM File folder

How to install *numpy*?
Go to your *terminal* in *pycharm*
Then type
>pip install numpy

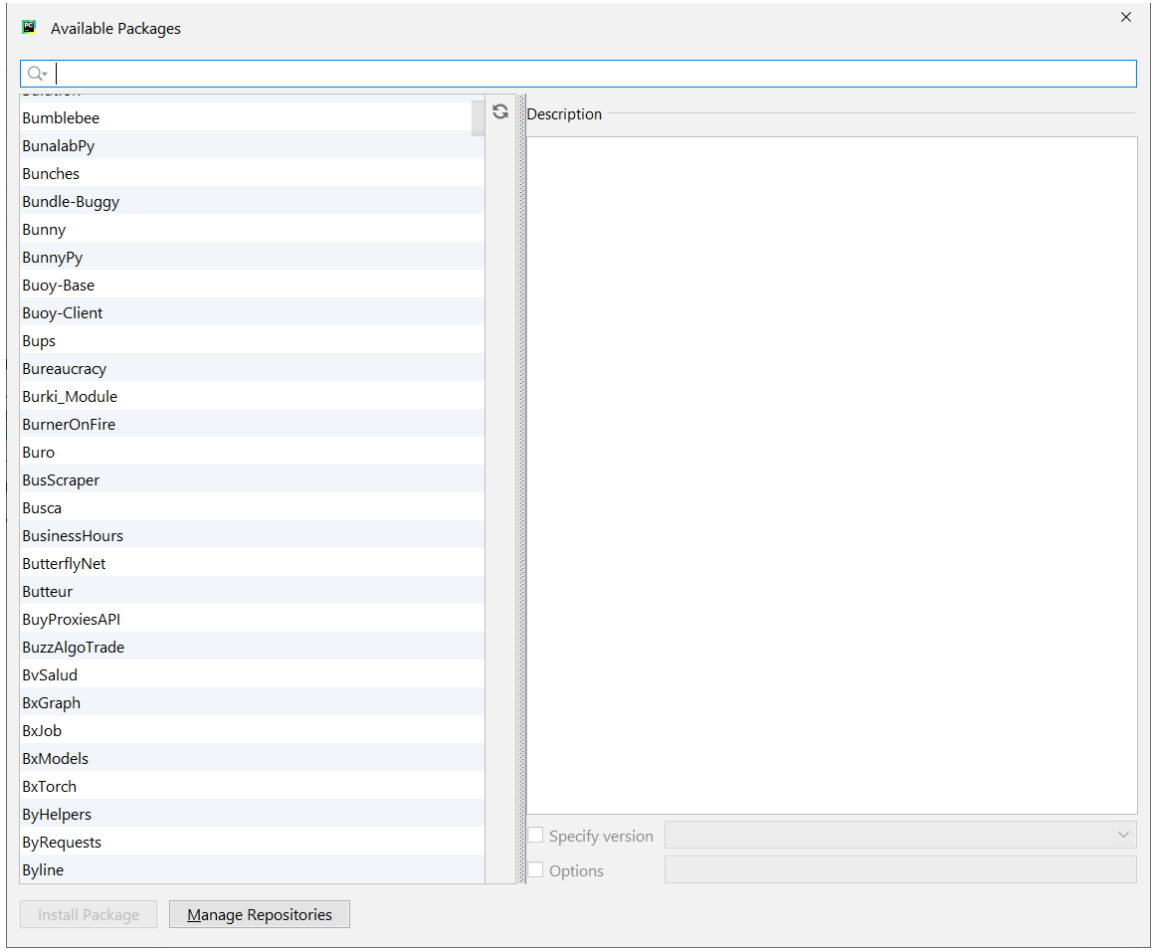
```
ITP449) C:\Users\Nitin Kale\PycharmProjects\ITP449>pip install numpy
ollecting numpy
 Downloading https://files.pythonhosted.org/packages/a8/ce/36f9b4fbc7e675a7c8a3809dd5902e24cecfcdbc006e8a7b2417c2b830a2/numpy-1.17.2-cp37-cp37m-win32.whl (10.8MB)
   100% |████████████████████████████████| 10.8MB 1.8MB/s
nstalling collected packages: numpy
uccessfully installed numpy-1.17.2

ITP449) C:\Users\Nitin Kale\PycharmProjects\ITP449>
```

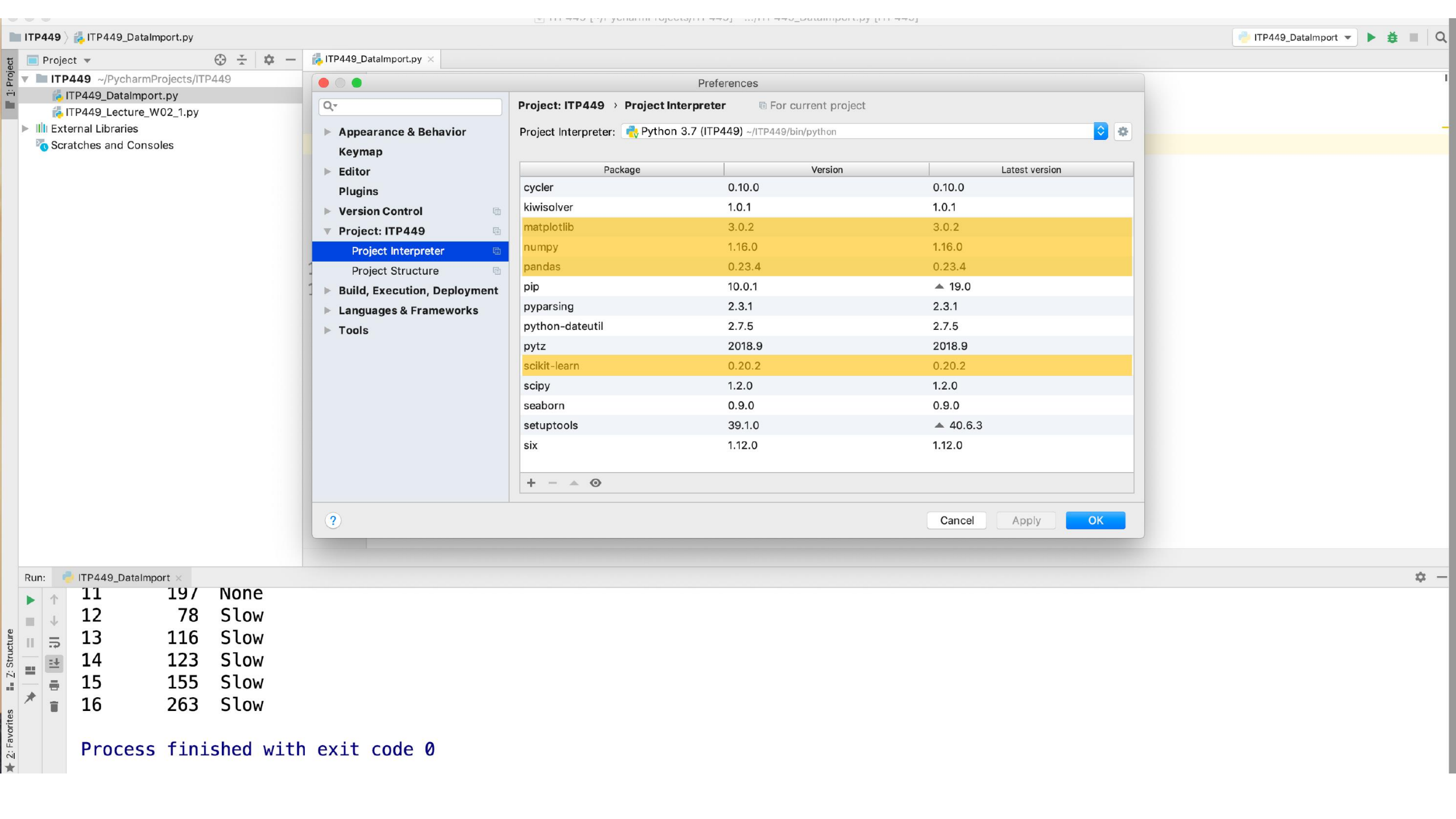
Now, we will install

- Pandas
- Numpy
- Matplotlib
- Scikit-learn
- *…and more as the course progresses*

In *Pycharm*, you can also install packages by

- *File → Settings → Project → Project Interpreter*
- Then *Packages* and *+*
- Then select the package, *Install Package*




ITP449 〉 ITP449_DataImport.py
ITP449_DataImport
Project
ITP449 ~/PycharmProjects/ITP449
ITP449_DataImport.py
ITP449_Lecture_W02_1.py
External Libraries
Scratches and Consoles
ITP449_DataImport.py
Preferences
Appearance & Behavior
Keymap
Editor
Plugins
Version Control
Project: ITP449
Project Interpreter
Project Structure
Build, Execution, Deployment
Languages & Frameworks
Tools
Project: ITP449 › Project Interpreter
For current project
Project Interpreter: Python 3.7 (ITP449) ~/ITP449/bin/python
Package
Version
Latest version
cycler 0.10.0 0.10.0
kiwisolver 1.0.1 1.0.1
matplotlib 3.0.2 3.0.2
numpy 1.16.0 1.16.0
pandas 0.23.4 0.23.4
pip 10.0.1 19.0
pyparsing 2.3.1 2.3.1
python-dateutil 2.7.5 2.7.5
pytz 2018.9 2018.9
scikit-learn 0.20.2 0.20.2
scipy 1.2.0 1.2.0
seaborn 0.9.0 0.9.0
setuptools 39.1.0 40.6.3
six 1.12.0 1.12.0
Cancel
Apply
OK
Run:
ITP449_DataImport
11 197 None
12 78 Slow
13 116 Slow
14 123 Slow
15 155 Slow
16 263 Slow
Process finished with exit code 0

How to use packages?

Import

To use the package

- import it first into you script

  import numpy

- Also *import os.* OS is used to interact with your operating system e.g. for opening files

- Then, to use a function from numpy, you can call it with the syntax *package.function()*

  e.g.*print(numpy.sqrt(25))*

- To shorten numpy, you can alias it

  import numpy as np
  print(np.sqrt(25))

Import Data

File Edit View Navigate Code Refactor Run Tools VCS Window Help ITP449 Master - Slide 19.py - PyCharm

ITP449 Master 〉 Lecture 3 〉 Slide 19.py

Slide 19

Project

ITP449 Master C:\Users\Nitin Kale\P

Lecture 3

grade5ScienceFairProject.csv

Slide 19.py

External Libraries

Scratches and Consoles

Slide 19.py grade5ScienceFairProject.csv

```
1  import os
2  import pandas
3
4  # change the directory to where all ITP449 files are stored.
5  # os.chdir("C:\Users\Nitin Kale\Google Drive\ITP 449\ITP 449 Fall 2020\Lectures")
6  # or place the data file in the Lecture folder
7
8  # read and store a csv file
9  puzzleCompletionTime = pandas.read_csv("grade5ScienceFairProject.csv")
10 print(puzzleCompletionTime)
```

Download the *grade5ScienceFairProject.csv* file into your ITP 449 Lecture folder for pycharm

Run: Slide 19

```
"C:\Users\Nitin Kale\venv\Scripts\python.exe" "C:/Users/Nitin Kale/PycharmProjects/ITP449 Master/Lecture 3/Slide 19
    Seconds Music
0        39  Fast
1        53  Fast
2        56  Fast
3       166  Fast
4       182  Fast
```

6: TODO 4: Run Terminal Python Console Event Log

6:18 CRLF UTF-8 4 spaces Python 3.8 (venv)

1: Project 7: Structure 2: Favorites

NumPy

File Edit View Navigate Code Refactor Run Tools VCS Window Help ITP449 Master - Slide 21.py - PyCharm

ITP449 Master 〉 Lecture 3 〉 Slide 21.py

Slide 21

Project

ITP449 Master C:\Users\Nitin Kale\P
- Lecture 3
  - grade5ScienceFairProject.csv
  - Slide 19.py
  - Slide 21.py
- External Libraries
- Scratches and Consoles

Slide 19.py | Slide 21.py | grade5ScienceFairProject.csv

```
1  odds = [1, 3, 5]
2  evens = [2, 4, 6]
3
4  print(odds + evens)
5  print(odds * evens)
6
```

Odds and evens are lists. Addition and multiplication don't work algebraically

Run: Slide 21

```
"C:\Users\Nitin Kale\venv\Scripts\python.exe" "C:/Users/Nitin Kale/PycharmProjects/ITP449 Master/Lecture 3/Slid
[1, 3, 5, 2, 4, 6]
Traceback (most recent call last):
  File "C:/Users/Nitin Kale/PycharmProjects/ITP449 Master/Lecture 3/Slide 21.py", line 5, in <module>
    print(odds * evens)
TypeError: can't multiply sequence by non-int of type 'list'
```

6: TODO | 4: Run | Terminal | Python Console | Event Log

4:20 UTF-8 4 spaces Python 3.8 (venv)

Do the following:

Multiply the values at the same indices of the *odds* and *evens* lists. Display the product.

File Edit View Navigate Code Refactor Run Tools VCS Window Help ITP449 Master - Slide 23.py - PyCharm

ITP449 Master 〉 Lecture 3 〉 Slide 23.py

Slide 23

Project

ITP449 Master C:\Users\Nitin Kale\P
- Lecture 3
  - grade5ScienceFairProject.csv
  - Slide 19.py
  - Slide 21.py
  - Slide 23.py
- External Libraries
- Scratches and Consoles

Slide 23.py

```
1  odds = [1, 3, 5]
2  evens = [2, 4, 6]
3
4  product = []
5
6  for n in range(3):
7      product.append(odds[n] * evens[n])
8
9  print(product)
10
```

Loop through the lists to multiply.
But this is cumbersome. Let's use numpy.

Run: Slide 23

```
"C:\Users\Nitin Kale\venv\Scripts\python.exe" "C:/Users/Nitin Kale/PycharmProjects/ITP449 Master/Lecture 3/Slid
[2, 12, 30]

Process finished with exit code 0
```

6: TODO 4: Run Terminal Python Console Event Log

10:1 CRLF UTF-8 4 spaces Python 3.8 (venv)

File Edit View Navigate Code Refactor Run Tools VCS Window Help ITP449 Master - Slide 24.py - PyCharm

ITP449 Master 〉 Lecture 3 〉 Slide 24.py

Slide 24

Project

ITP449 Master C:\Users\Nitin Kale\P
- Lecture 3
  - grade5ScienceFairProject.csv
  - Slide 19.py
  - Slide 21.py
  - Slide 23.py
  - Slide 24.py
- External Libraries
- Scratches and Consoles

Slide 24.py

```
1  import numpy as np
2
3  odds = [1, 3, 5]
4  evens = [2, 4, 6]
5
6  # Convert lists to arrays
7  np_odds = np.array(odds)
8  np_evens = np.array(evens)
9
10 product = np_odds * np_evens
11
12 print(product)
13
```

Numpy supports *arrays*

Run: Slide 24

```
"C:\Users\Nitin Kale\venv\Scripts\python.exe" "C:/Users/Nitin Kale/PycharmProjects/ITP449 Master/Lecture 3/Slid
[ 2 12 30]

Process finished with exit code 0
```

6: TODO | 4: Run | Terminal | Python Console | Event Log

13:1 CRLF UTF-8 4 spaces Python 3.8 (venv)

Arrays

Python lists don't allow calculations over the entire list.

Numpy allows for *arrays*. Calculations can be done over the entire array.

You can convert a *list* to an *array*

You can define array like this
A = numpy.array([1,2,3])
B= numpy.array([4,5,6])

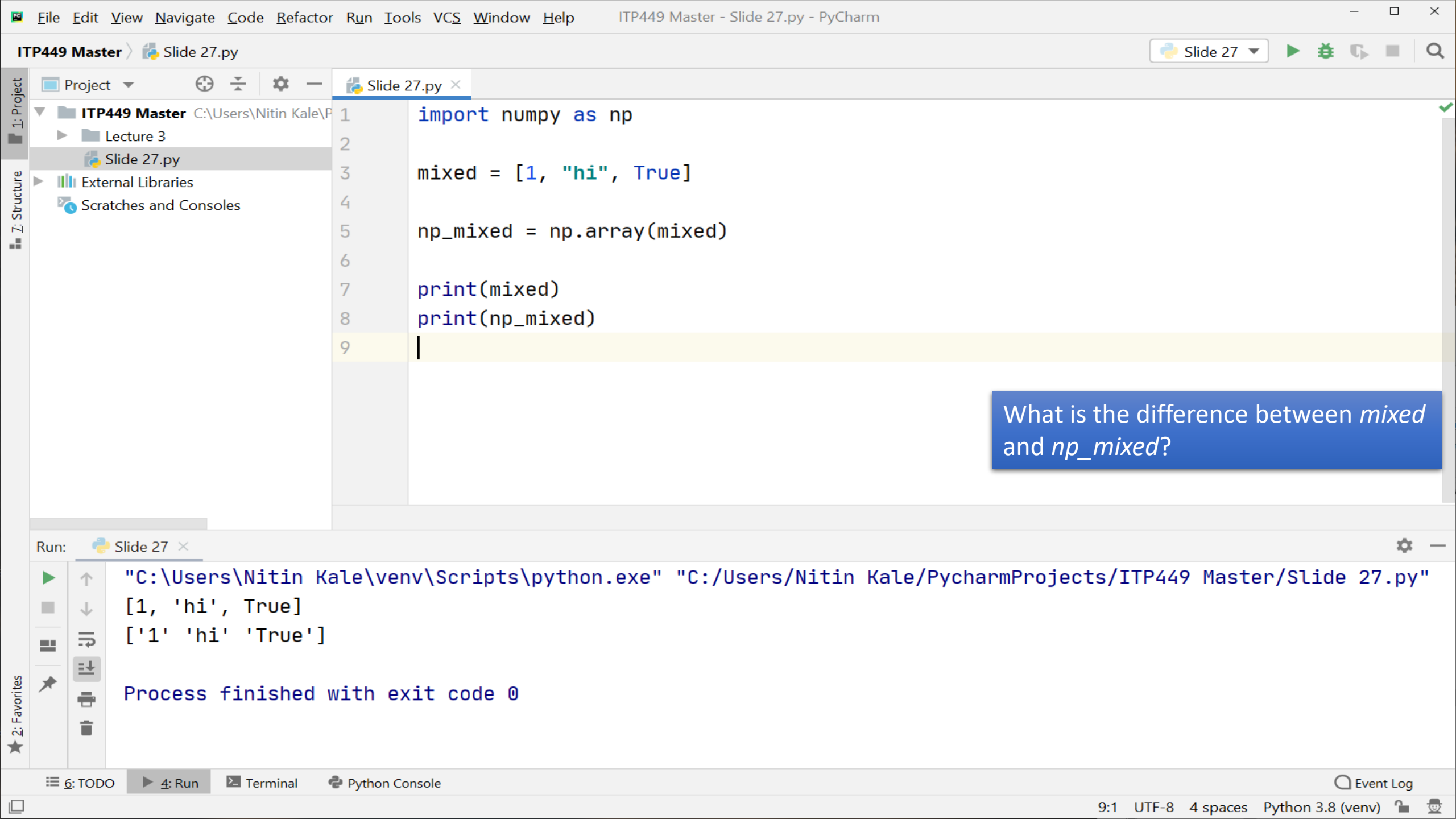
So you can do *A + B,* or *A/B* (element by element, try it)

Numpy arrays contain elements of only one type

So if you create a list of *different* variable types, and *numpy* will convert them to one type
e.g. np.array([2,2.0,False,”Hello”])

Will be interpreted as ['2' ‘2.0' 'False' 'hello’]
All will get converted to strings (or the appropriate variable, *type coercion*)

All elements must be the same *variable type* in an array (unlike in lists)


File Edit View Navigate Code Refactor Run Tools VCS Window Help
ITP449 Master - Slide 27.py - PyCharm
ITP449 Master
Slide 27.py
Slide 27
Project
ITP449 Master C:\Users\Nitin Kale\P
Lecture 3
Slide 27.py
External Libraries
Scratches and Consoles
Slide 27.py
1 import numpy as np
2
3 mixed = [1, "hi", True]
4
5 np_mixed = np.array(mixed)
6
7 print(mixed)
8 print(np_mixed)
9
What is the difference between mixed and np_mixed?
Run:
Slide 27
"C:\Users\Nitin Kale\venv\Scripts\python.exe" "C:/Users/Nitin Kale/PycharmProjects/ITP449 Master/Slide 27.py"
[1, 'hi', True]
['1' 'hi' 'True']
Process finished with exit code 0
6: TODO
4: Run
Terminal
Python Console
Event Log
9:1
UTF-8
4 spaces
Python 3.8 (venv)
1: Project
7: Structure
2: Favorites

File Edit View Navigate Code Refactor Run Tools VCS Window Help ITP449 Master - Slide 28.py - PyCharm

ITP449 Master 〉 Lecture 3 〉 Slide 28.py

Slide 28

Project

ITP449 Master C:\Users\Nitin Kale\P
- Lecture 3
  - grade5ScienceFairProject.csv
  - Slide 19.py
  - Slide 21.py
  - Slide 23.py
  - Slide 24.py
  - Slide 27.py
  - Slide 28.py
- External Libraries
- Scratches and Consoles

Slide 27.py | Slide 28.py

```
1  import numpy as np
2
3  odds = [1, 3, 5, 7]
4  evens = [2, 4, 6, 8]
5
6  np_odds = np.array(odds)
7  np_evens = np.array(evens)
8
9  print(type(odds))
10 print(type(np_odds))
11
```

What is the difference between *odds* and *np_odds*?
Ndarray means N-dimensional array

Run: Slide 28

```
"C:\Users\Nitin Kale\venv\Scripts\python.exe" "C:/Users/Nitin Kale/PycharmProjects/ITP449 Master/Lecture 3/Slid
<class 'list'>
<class 'numpy.ndarray'>

Process finished with exit code 0
```

6: TODO | 4: Run | Terminal | Python Console | Event Log

11:1 CRLF UTF-8 4 spaces Python 3.8 (venv)

1: Project | 7: Structure | 2: Favorites

```python
import numpy as np

np_2d = np.array([[1, 3, 4], [2, 4, 6]])

print(np_2d)
print(np_2d.shape)
```


Run: in_class_coding

```
C:\Users\Reza\Desktop\ITP449_Fall2020\Class\venv\Scripts\python.exe "C:/Users/Reza/Desktop/ITP449_Fall2020/Cl
[[1 3 4]
 [2 4 6]]
(2, 3)

Process finished with exit code 0
```

File Edit View Navigate Code Refactor Run Tools VCS Window Help ITP449 Master - Slide 30.py - PyCharm

ITP449 Master 〉 Lecture 3 〉 Slide 30.py

Slide 30

Project

ITP449 Master C:\Users\Nitin Kale\P

Lecture 3

grade5ScienceFairProject.csv

Slide 19.py

Slide 21.py

Slide 23.py

Slide 24.py

Slide 27.py

Slide 28.py

Slide 29.py

Slide 30.py

Slide 31.py

External Libraries

Scratches and Consoles

Slide 31.py | Slide 30.py

```
1  import numpy as np
2
3  np_2d = np.array([[1, 3, 5], [2, 4, 6]])
4
5  print(np_2d[0])
6  print(np_2d[0][2])
7  print(np_2d[0, 2])
8  print(np_2d[:, 1:3])
9  print(np_2d[1, :])
10
11
12
```

Indices

Run: Slide 30

```
"C:\Users\Nitin Kale\venv\Scripts\python.exe" "C:/Users/Nitin Kale/PycharmProjects/ITP449 Master/Lecture 3/Slid
[1 3 5]
5
5
[[3 5]
 [4 6]]
[2 4 6]
```

6: TODO | 4: Run | Terminal | Python Console | Event Log

10:1 CRLF UTF-8 4 spaces Python 3.8 (venv)

1: Project | 7: Structure | 2: Favorites

File Edit View Navigate Code Refactor Run Tools VCS Window Help ITP449 Master - Slide 31.py - PyCharm

ITP449 Master 〉 Lecture 3 〉 Slide 31.py

Slide 31

Project

ITP449 Master C:\Users\Nitin Kale\P
- Lecture 3
  - grade5ScienceFairProject.csv
  - Slide 19.py
  - Slide 21.py
  - Slide 23.py
  - Slide 24.py
  - Slide 27.py
  - Slide 28.py
  - Slide 29.py
  - Slide 30.py
  - Slide 31.py
- External Libraries
- Scratches and Consoles

Slide 31.py

```
1  import numpy as np
2
3  np_2d = np.array([[1, 3, 5], [2, 4, 6]])
4
5  print(np.mean(np_2d))
6  print(np.mean(np_2d[0]))
7  print(np.mean(np_2d[:, 1:3]))
8
9
```

Functions

Run: Slide 31

```
"C:\Users\Nitin Kale\venv\Scripts\python.exe" "C:/Users/Nitin Kale/PycharmProjects/ITP449 Master/Lecture 3/Slid
3.5
3.0
4.5

Process finished with exit code 0
```

1: Project 7: Structure 2: Favorites

6: TODO 4: Run Terminal Python Console Event Log

8:1 CRLF UTF-8 4 spaces Python 3.8 (venv)

do the following:

Create the Rock, Paper, Scissors game with the following output:

```
Rock (r), paper (p), scissors (s)? x
Try again!

Process finished with exit code 0
```

```
Rock (r), paper (p), scissors (s)? p
You picked Paper and the computer picked Rock

Process finished with exit code 0
```

File Edit View Navigate Code Refactor Run Tools VCS Window Help ITP449 Master - Slide 33.py - PyCharm

ITP449 Master 〉 Lecture 3 〉 Slide 33.py

Slide 33

Project

ITP449 Master C:\Users\Nitin Kale\P

Lecture 3

grade5ScienceFairProject.csv

Slide 19.py

Slide 21.py

Slide 23.py

Slide 24.py

Slide 27.py

Slide 28.py

Slide 29.py

Slide 30.py

Slide 31.py

Slide 33.py

External Libraries

Scratches and Consoles

Slide 33.py

```
1  import numpy as np
2  things = ["Rock", "Paper", "Scissors"]
3  player = input("Rock (r), paper (p), scissors (s)? ")
4  if player == "r":
5      playerIndex = 0
6  elif player == "p":
7      playerIndex = 1
8  elif player == "s":
9      playerIndex = 2
10 else:
11     print("Try again!")
12     playerIndex = 3
13 computerIndex = np.random.randint(low=0, high=2)
14 if playerIndex != 3:
15     print("You picked", things[playerIndex], "and the computer picked",
16           things[computerIndex])
```

Run: Slide 33

```
"C:\Users\Nitin Kale\venv\Scripts\python.exe" "C:/Users/Nitin Kale/PycharmProjects/ITP449 Master/Lecture 3/Slid
Rock (r), paper (p), scissors (s)? p
You picked Paper and the computer picked Rock
```

6: TODO 4: Run Terminal Python Console Event Log

PEP 8: E251 unexpected spaces around keyword / parameter equals 13:39 CRLF UTF-8 4 spaces Python 3.8 (venv)

1: Project 7: Structure 2: Favorites

Pandas

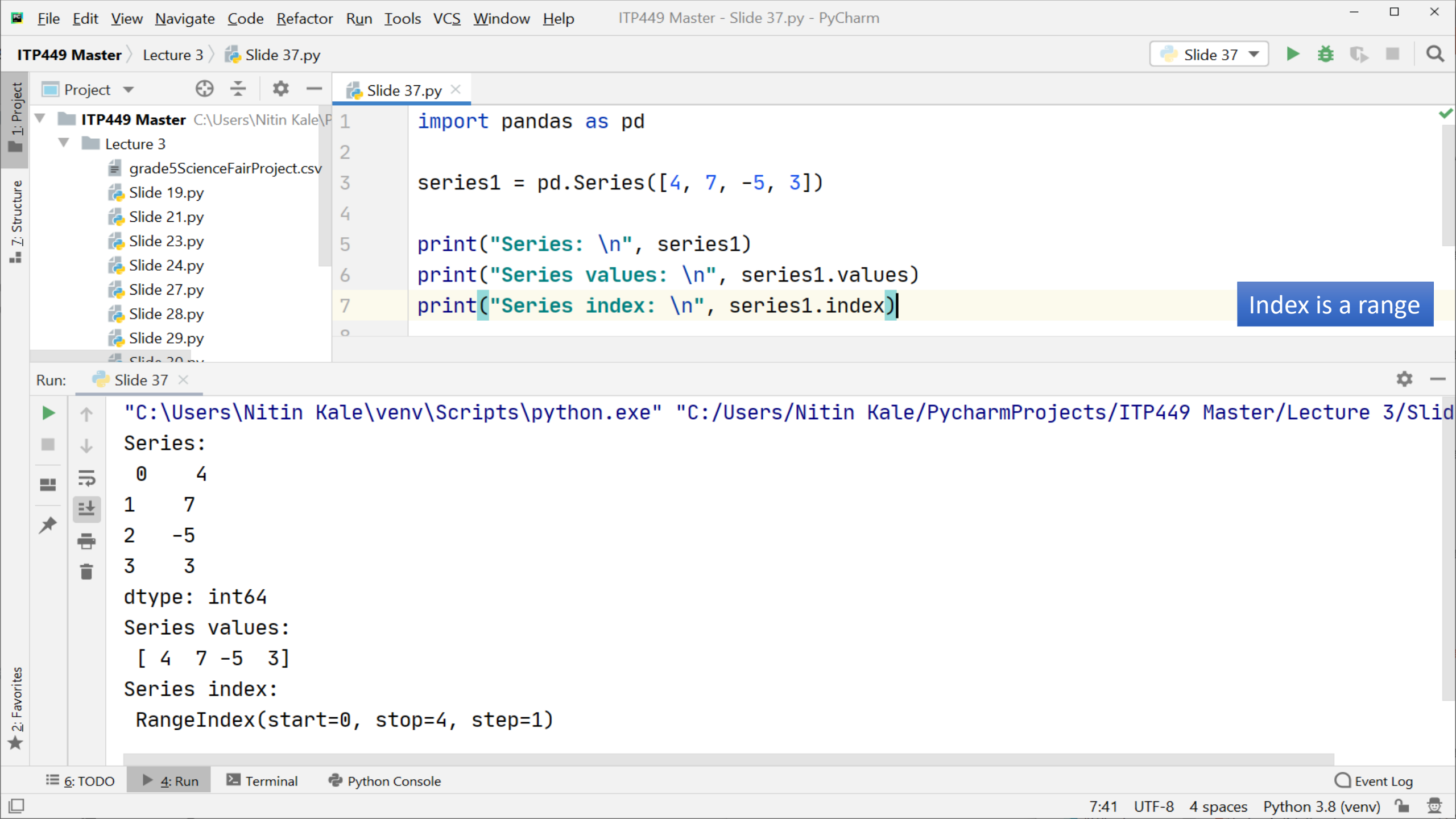
Pandas

Contains data structures and data manipulation tools designed to make data cleansing and analysis fast and easy

Adopts coding style from `Numpy` but main difference is that `Pandas` handles heterogeneous data

Series

A one-dimensional object, similar to an array, list, or column in table. Each item in a Series, however, is assigned to an entry in an index.


File Edit View Navigate Code Refactor Run Tools VCS Window Help
ITP449 Master - Slide 37.py - PyCharm
ITP449 Master 〉 Lecture 3 〉 Slide 37.py
Slide 37
Project
ITP449 Master C:\Users\Nitin Kale\P
Lecture 3
grade5ScienceFairProject.csv
Slide 19.py
Slide 21.py
Slide 23.py
Slide 24.py
Slide 27.py
Slide 28.py
Slide 29.py
Slide 37.py
1 import pandas as pd
2
3 series1 = pd.Series([4, 7, -5, 3])
4
5 print("Series: \n", series1)
6 print("Series values: \n", series1.values)
7 print("Series index: \n", series1.index)
Index is a range
Run: Slide 37
"C:\Users\Nitin Kale\venv\Scripts\python.exe" "C:/Users/Nitin Kale/PycharmProjects/ITP449 Master/Lecture 3/Slid
Series:
0 4
1 7
2 -5
3 3
dtype: int64
Series values:
[4 7 -5 3]
Series index:
RangeIndex(start=0, stop=4, step=1)
6: TODO
4: Run
Terminal
Python Console
Event Log
7:41 UTF-8 4 spaces Python 3.8 (venv)
1: Project
7: Structure
2: Favorites

File Edit View Navigate Code Refactor Run Tools VCS Window Help ITP449 Master - Slide 38.py - PyCharm

ITP449 Master 〉 Lecture 3 〉 Slide 38.py

Slide 38

Project

Slide 19.py
Slide 21.py
Slide 23.py
Slide 24.py
Slide 27.py
Slide 28.py
Slide 29.py
Slide 30.py
Slide 31.py
Slide 33.py
Slide 37.py

Slide 38.py

```
1  import pandas as pd
2
3  series2 = pd.Series([4, 7, -5, 3], index=["d", "b", "a", "c"])
4
5  print("Series: \n", series2)
6  print("Series values: \n", series2.values)
7  print("Series index: \n", series2.index)
```

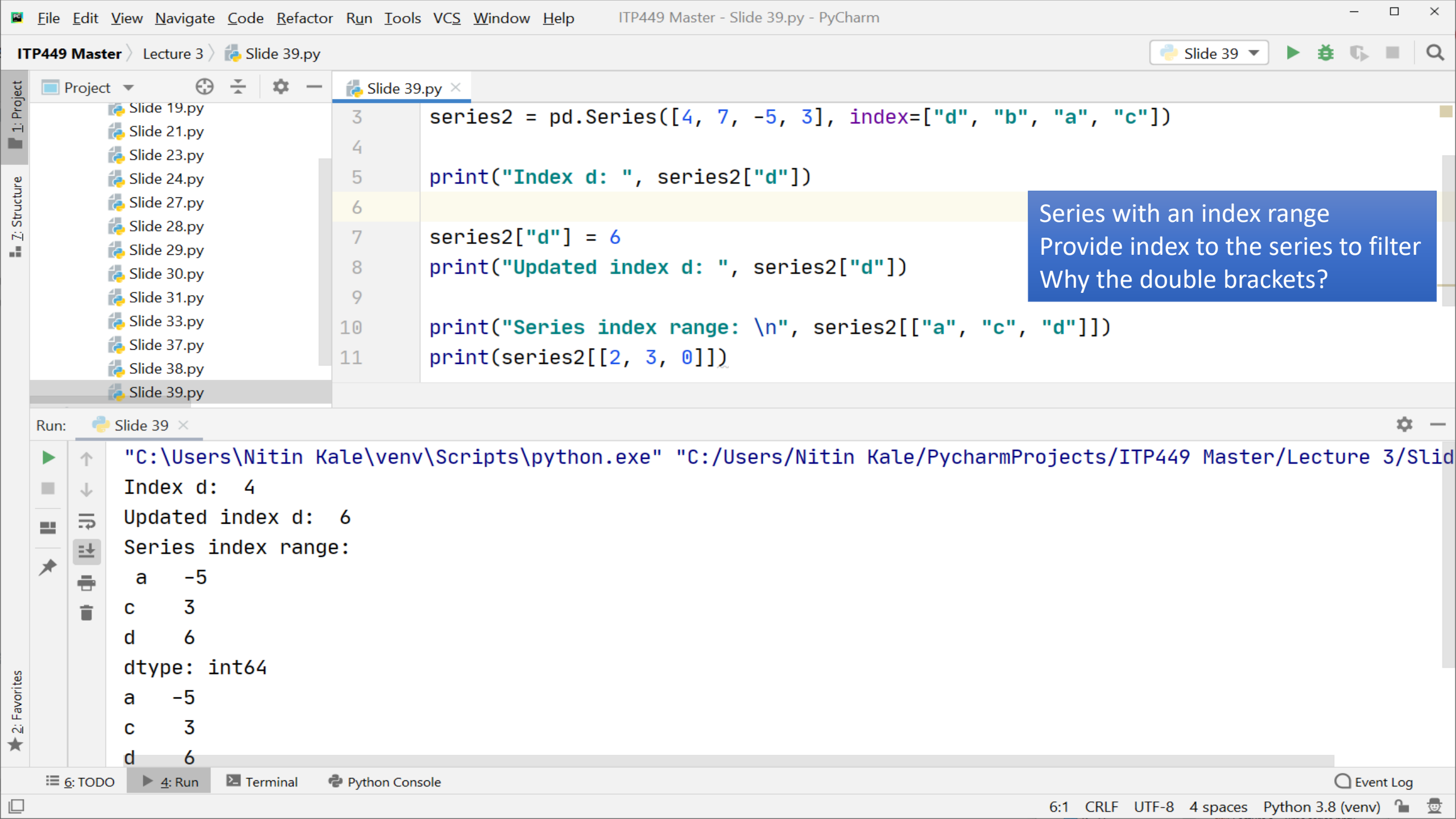
Index is an object

Run: Slide 38

```
"C:\Users\Nitin Kale\venv\Scripts\python.exe" "C:/Users/Nitin Kale/PycharmProjects/ITP449 Master/Lecture 3/Slid
Series:
 d    4
b    7
a   -5
c    3
dtype: int64
Series values:
 [ 4  7 -5  3]
Series index:
 Index(['d', 'b', 'a', 'c'], dtype='object')
```

6: TODO 4: Run Terminal Python Console Event Log

14:1 UTF-8 4 spaces Python 3.8 (venv)


File Edit View Navigate Code Refactor Run Tools VCS Window Help
ITP449 Master - Slide 39.py - PyCharm
ITP449 Master 〉 Lecture 3 〉 Slide 39.py
Slide 39
Project
Slide 19.py
Slide 21.py
Slide 23.py
Slide 24.py
Slide 27.py
Slide 28.py
Slide 29.py
Slide 30.py
Slide 31.py
Slide 33.py
Slide 37.py
Slide 38.py
Slide 39.py
Slide 39.py
3 series2 = pd.Series([4, 7, -5, 3], index=["d", "b", "a", "c"])
4
5 print("Index d: ", series2["d"])
6
7 series2["d"] = 6
8 print("Updated index d: ", series2["d"])
9
10 print("Series index range: \n", series2[["a", "c", "d"]])
11 print(series2[[2, 3, 0]])
Series with an index range
Provide index to the series to filter
Why the double brackets?
Run: Slide 39
"C:\Users\Nitin Kale\venv\Scripts\python.exe" "C:/Users/Nitin Kale/PycharmProjects/ITP449 Master/Lecture 3/Slid
Index d: 4
Updated index d: 6
Series index range:
a -5
c 3
d 6
dtype: int64
a -5
c 3
d 6
1: Project
7: Structure
2: Favorites
6: TODO
4: Run
Terminal
Python Console
Event Log
6:1 CRLF UTF-8 4 spaces Python 3.8 (venv)

File Edit View Navigate Code Refactor Run Tools VCS Window Help ITP449 Master - Slide 40.py - PyCharm

ITP449 Master 〉 Lecture 3 〉 Slide 40.py

Slide 40

Project

Slide 27.py
Slide 28.py
Slide 29.py
Slide 30.py
Slide 31.py
Slide 33.py
Slide 37.py
Slide 38.py
Slide 39.py

Slide 40.py

```
1  import pandas as pd
2  import numpy as np
3
4  series2 = pd.Series([4, 7, -5, 3], index=["d", "b", "a", "c"])
5
6  print("Numpy like filtering: \n", series2[series2 > 0])
7  print("Numpy like arithmetic: \n", series2 * 2)
8  print("Pass series to Numpy function: \n", np.absolute(series2))
```

Run: Slide 40

```
"C:\Users\Nitin Kale\venv\Scripts\python.exe" "C:/Users/Nitin Kale/PycharmProjects/ITP449 Master/Lecture 3/Slide 40.py"
Numpy like filtering:
 d    4
b    7
c    3
dtype: int64
Numpy like arithmetic:
 d     8
b    14
a   -10
c     6
dtype: int64
Pass series to Numpy function:
 d    4
b    7
a    5
c    3
dtype: int64

Process finished with exit code 0
```

1: Project 7: Structure 2: Favorites

6: TODO 4: Run Terminal Python Console Event Log

3:1 CRLF UTF-8 4 spaces Python 3.8 (venv)

Slide 41.py

```
1  import pandas as pd
2  import numpy as np
3
4  series2 = pd.Series([4, 7, -5, 3], index=["d", "b", "a", "c"])
5
6  series2.name = "Numbers"
7  series2.index.name = "Letters"
8
9  print("Value title: ", series2.name)
10 print("Index title: ", series2.index.name)
11 print(series2)
```

Run: Slide 41

```
"C:\Users\Nitin Kale\venv\Scripts\python.exe" "C:/Users/Nitin Kale/PycharmProjects/ITP449 Master/Lecture 3/Slide 41.py"
Value title:  Numbers
Index title:  Letters
Letters
d    4
b    7
a   -5
c    3
Name: Numbers, dtype: int64

Process finished with exit code 0
```

Data Frames

DataFrame

Represents a rectangular table of data (2D) and contains an ordered collection of columns, each of which can be a different value type (numeric, string, boolean, etc.). Has both a row and column index.

Why a dataFrame? For machine learning we usually have a collection of numeric and/or categorical *predictor* variables and a numeric or categorical *target* (response) variable

Creating Data Frames

Columns

rows

	State	Year	Pop
0	Ohio	2001	1.7
1	Ohio	2002	3.6
2	Nevada	2003	2.4
3	Nevada	2004	2.9
4	Nevada	2005	3.2

We will load data mostly from csv files into dataframes

Dataframes can be created from lists, dictionaries, dataframes, series, 2D arrays

You can optionally specify row labels and column labels
If no labels are set, then the rows and columns are labeled 0,1,…

File Edit View Navigate Code Refactor Run Tools VCS Window Help ITP449 Master - Slide 46.py - PyCharm

ITP449 Master 〉 Lecture 3 〉 Slide 46.py

Slide 46

Project

Slide 27.py
Slide 28.py
Slide 29.py
Slide 30.py
Slide 31.py
Slide 33.py
Slide 37.py
Slide 38.py
Slide 39.py
Slide 40.py
Slide 41.py

Slide 46.py

```
1  import pandas as pd
2  data = {"state": ["Ohio", "Ohio", "Nevada", "Nevada", "Nevada"],
3          "year": [2001, 2002, 2001, 2002, 2003],
4          "pop": [1.7, 3.6, 2.4, 2.9, 3.2]}
5
6  frame = pd.DataFrame(data)
7  print(frame)
8  print("First 5 rows: \n", frame.head())
9  print("First 2 rows: \n", frame.head(2))
```

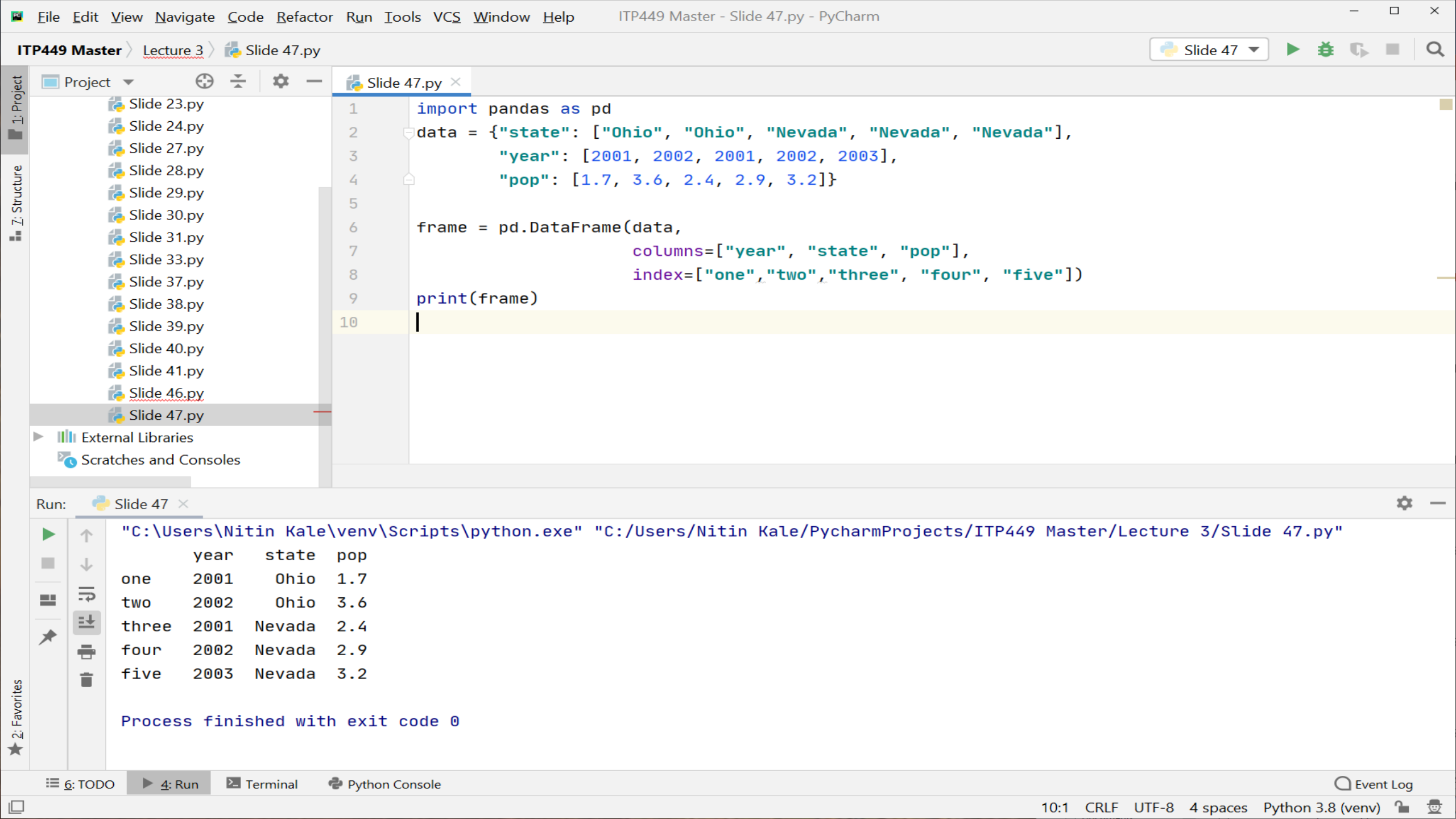
DataFrame from dictionary
print(data.dtypes)

Run: Slide 46

```
    state  year  pop
0    Ohio  2001  1.7
1    Ohio  2002  3.6
2  Nevada  2001  2.4
3  Nevada  2002  2.9
4  Nevada  2003  3.2
First 5 rows:
     state  year  pop
0    Ohio  2001  1.7
1    Ohio  2002  3.6
2  Nevada  2001  2.4
3  Nevada  2002  2.9
4  Nevada  2003  3.2
First 2 rows:
   state  year  pop
0  Ohio  2001  1.7
1  Ohio  2002  3.6
```

6: TODO | 4: Run | Terminal | Python Console | Event Log

2:1 CRLF UTF-8 4 spaces Python 3.8 (venv)


File Edit View Navigate Code Refactor Run Tools VCS Window Help
ITP449 Master - Slide 47.py - PyCharm
ITP449 Master 〉 Lecture 3 〉 Slide 47.py
Slide 47
Project
Slide 47.py
Slide 23.py
Slide 24.py
Slide 27.py
Slide 28.py
Slide 29.py
Slide 30.py
Slide 31.py
Slide 33.py
Slide 37.py
Slide 38.py
Slide 39.py
Slide 40.py
Slide 41.py
Slide 46.py
Slide 47.py
External Libraries
Scratches and Consoles
1 import pandas as pd
2 data = {"state": ["Ohio", "Ohio", "Nevada", "Nevada", "Nevada"],
3 "year": [2001, 2002, 2001, 2002, 2003],
4 "pop": [1.7, 3.6, 2.4, 2.9, 3.2]}
5
6 frame = pd.DataFrame(data,
7 columns=["year", "state", "pop"],
8 index=["one","two","three", "four", "five"])
9 print(frame)
10
Run: Slide 47
"C:\Users\Nitin Kale\venv\Scripts\python.exe" "C:/Users/Nitin Kale/PycharmProjects/ITP449 Master/Lecture 3/Slide 47.py"
year state pop
one 2001 Ohio 1.7
two 2002 Ohio 3.6
three 2001 Nevada 2.4
four 2002 Nevada 2.9
five 2003 Nevada 3.2
Process finished with exit code 0
1: Project
7: Structure
2: Favorites
6: TODO
4: Run
Terminal
Python Console
Event Log
10:1 CRLF UTF-8 4 spaces Python 3.8 (venv)

File Edit View Navigate Code Refactor Run Tools VCS Window Help ITP449 Master - Slide 47.py - PyCharm

ITP449 Master 〉 Lecture 3 〉 Slide 47.py

Slide 47

Project

Slide 23.py
Slide 24.py
Slide 27.py
Slide 28.py
Slide 29.py
Slide 30.py
Slide 31.py
Slide 33.py
Slide 37.py
Slide 38.py
Slide 39.py
Slide 40.py
Slide 41.py
Slide 46.py

Slide 47.py

```
1  import pandas as pd
2  data = {"state": ["Ohio", "Ohio", "Nevada", "Nevada", "Nevada"],
3          "year": [2001, 2002, 2001, 2002, 2003],
4          "pop": [1.7, 3.6, 2.4, 2.9, 3.2]}
5
6  frame = pd.DataFrame(data,
7                       columns=["year", "state", "pop"],
8                       index=["one","two","three", "four", "five"])
9
10 print("Column retrieval: \n", frame["state"])
11 print("Column retrieval: \n", frame[["state", "pop"]])
12 print("Column retrieval: \n", frame.year)
```

Run: Slide 47

```
Column retrieval:
         state  pop
one       Ohio  1.7
two       Ohio  3.6
three   Nevada  2.4
four    Nevada  2.9
five    Nevada  3.2
Column retrieval:
 one      2001
two       2002
three     2001
four      2002
five      2003
Name: year, dtype: int64
```

6: TODO 4: Run Terminal Python Console Event Log

12:41 CRLF UTF-8 4 spaces Python 3.8 (venv)

File Edit View Navigate Code Refactor Run Tools VCS Window Help ITP449 Master - Slide 49.py - PyCharm

ITP449 Master 〉 Lecture 3 〉 Slide 49.py

Slide 49

Project

Slide 29.py
Slide 30.py
Slide 31.py
Slide 33.py
Slide 37.py
Slide 38.py
Slide 39.py
Slide 40.py
Slide 41.py
Slide 46.py
Slide 47.py
Slide 49.py
External Libraries
Scratches and Consoles

Slide 49.py

```
1  import pandas as pd
2  data = {"state": ["Ohio", "Ohio", "Nevada", "Nevada", "Nevada"],
3          "year": [2001, 2002, 2001, 2002, 2003],
4          "pop": [1.7, 3.6, 2.4, 2.9, 3.2]}
5
6  frame = pd.DataFrame(data,
7                       columns=["year", "state", "pop"],
8                       index=["one","two","three", "four", "five"])
9
10 print("Add and populate column:")
11 print(frame["state"] == "Ohio")
12
```

Run: Slide 49

```
"C:\Users\Nitin Kale\venv\Scripts\python.exe" "C:/Users/Nitin Kale/PycharmProjects/ITP449 Master/Lecture 3/Slide 49.py"
Add and populate column:
one       True
two       True
three    False
four     False
five     False
Name: state, dtype: bool

Process finished with exit code 0
```

6: TODO 4: Run Terminal Python Console Event Log

11:32 CRLF UTF-8 4 spaces Python 3.8 (venv)

1: Project 7: Structure 2: Favorites

File Edit View Navigate Code Refactor Run Tools VCS Window Help ITP449 Master - Slide 49.py - PyCharm

ITP449 Master 〉 Lecture 3 〉 Slide 49.py

Slide 49

Project

Slide 30.py
Slide 31.py
Slide 33.py
Slide 37.py
Slide 38.py
Slide 39.py
Slide 40.py
Slide 41.py
Slide 46.py
Slide 47.py
Slide 49.py
External Libraries
Scratches and Consoles

Slide 49.py

```
        "pop": [1.7, 3.6, 2.4, 2.9, 3.2]}

frame = pd.DataFrame(data,
                     columns=["year", "state", "pop"],
                     index=["one","two","three", "four", "five"])

print("Add and populate column:")
print(frame["state"] == "Ohio")
frame["eastern"] = frame["state"] == "Ohio"
print(frame)
print("Delete column:")
del frame["eastern"]
print(frame)
```

Run: Slide 49

```
Name: state, dtype: bool
       year   state  pop  eastern
one    2001    Ohio  1.7     True
two    2002    Ohio  3.6     True
three  2001  Nevada  2.4    False
four   2002  Nevada  2.9    False
five   2003  Nevada  3.2    False
Delete column:
       year   state  pop
one    2001    Ohio  1.7
two    2002    Ohio  3.6
three  2001  Nevada  2.4
four   2002  Nevada  2.9
five   2003  Nevada  3.2

Process finished with exit code 0
```

6: TODO 4: Run Terminal Python Console Event Log

12:44 CRLF UTF-8 4 spaces Python 3.8 (venv)

Series Indexing

File Edit View Navigate Code Refactor Run Tools VCS Window Help ITP449 Master - Slide 52.py - PyCharm

ITP449 Master 〉 Lecture 3 〉 Slide 52.py

Slide 52

Project

Slide 30.py
Slide 31.py
Slide 33.py
Slide 37.py
Slide 38.py
Slide 39.py
Slide 40.py
Slide 41.py
Slide 46.py
Slide 47.py
Slide 49.py

Slide 52.py

```
1  import pandas as pd
2  import numpy as np
3
4  obj = pd.Series(np.arange(4.), index=["a", "b", "c", "d"])
5  print(obj)
6  print("Index name: ", obj["b"])
7  print("Index location: ", obj[1])
8  print("Index multiple locations: \n", obj[[2, 3]])
9  print("Index range: \n", obj[2:4])
10 print("Index multiple names: \n", obj[["b", "c"]])
11 print("Index condition: \n", obj[obj < 2])
```

Run: Slide 52

```
a    0.0
b    1.0
c    2.0
d    3.0
dtype: float64
Index name:  1.0
Index location:  1.0
Index multiple locations:
 c    2.0
d    3.0
dtype: float64
Index range:
 c    2.0
d    3.0
dtype: float64
Index multiple names:
 b    1.0
c    2.0
dtype: float64
Index condition:
 a    0.0
b    1.0
dtype: float64
```

6: TODO 4: Run Terminal Python Console Event Log

10:32 CRLF UTF-8 4 spaces Python 3.8 (venv)

File Edit View Navigate Code Refactor Run Tools VCS Window Help ITP449 Master - Slide 53.py - PyCharm

ITP449 Master 〉 Lecture 3 〉 Slide 53.py

Slide 53

Project

ITP449 Master C:\Users\Nitin Kale\P

Lecture 3

grade5ScienceFairProject.csv

Slide 19.py

Slide 21.py

Slide 23.py

Slide 24.py

Slide 27.py

Slide 28.py

Slide 29.py

Slide 30.py

Slide 53.py

```
1  import pandas as pd
2  import numpy as np
3
4  obj = pd.Series(np.arange(4.), index=["b", "c", "a", "d"])
5
6  print(obj)
7  print(obj["c":"d"])
8  print(obj["d":"b":-1])
9
```

Run: Slide 53

```
"C:\Users\Nitin Kale\venv\Scripts\python.exe" "C:/Users/Nitin Kale/PycharmProjects/ITP449 Master/Lecture 3/Slide 53.py"
b    0.0
c    1.0
a    2.0
d    3.0
dtype: float64
c    1.0
a    2.0
d    3.0
dtype: float64
d    3.0
a    2.0
c    1.0
b    0.0
dtype: float64

Process finished with exit code 0
```

6: TODO 4: Run 5: Debug Terminal Python Console Event Log

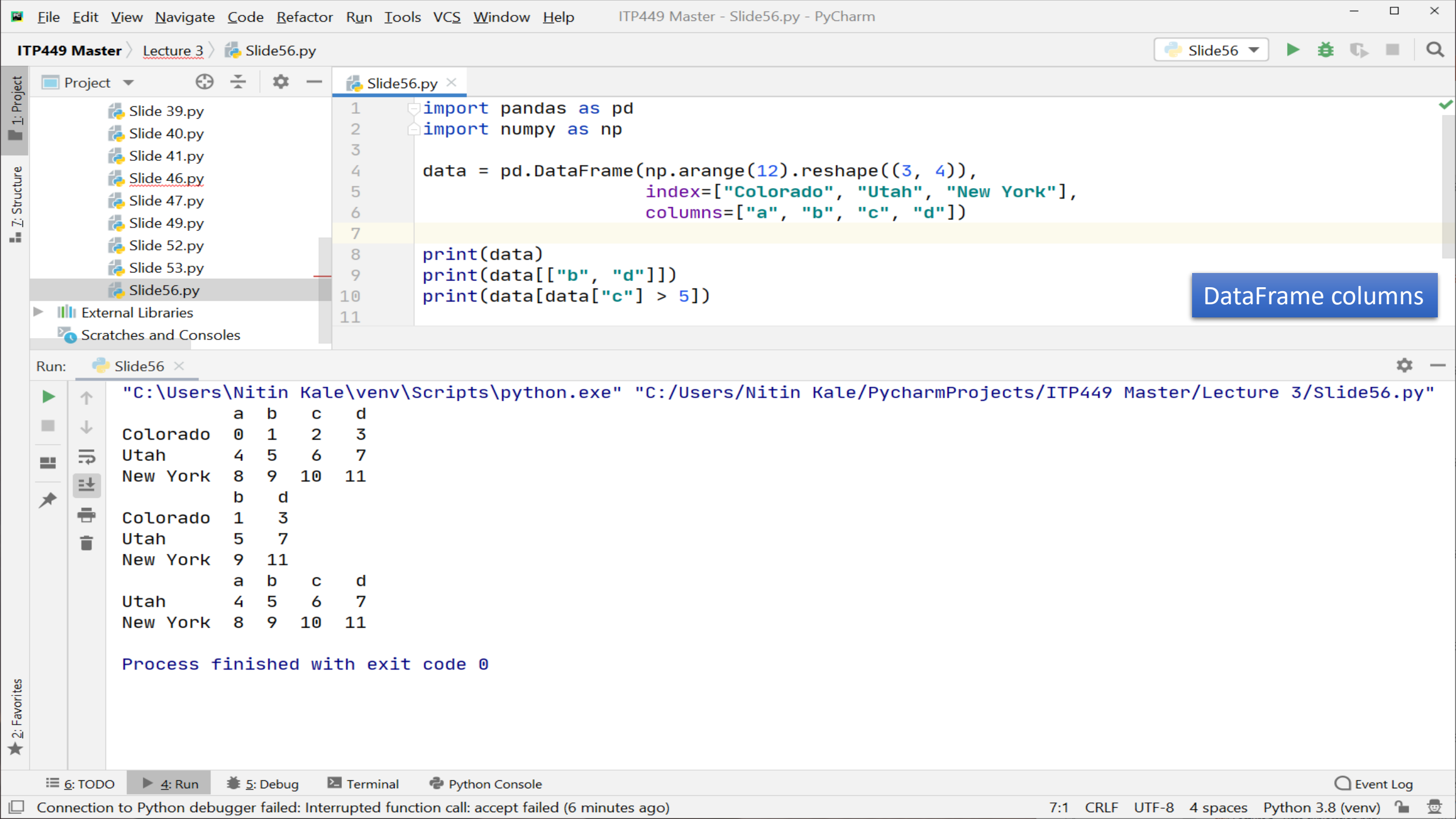
Connection to Python debugger failed: Interrupted function call: accept failed (a minute ago) 9:1 CRLF UTF-8 4 spaces Python 3.8 (venv)

DataFrame Indexing

INDEXING Dataframes

Operation	Syntax	Results in a
Select column	frame[col]	Series
Select row by label	frame.loc[label]	Series
Select row by integer location	frame.iloc[loc]	Series
Slice rows	frame[5:10]	DataFrame
Select rows by boolean vector	frame[bool_vec]	DataFrame

```
print(data.index)
print(data.columns)
```


File Edit View Navigate Code Refactor Run Tools VCS Window Help
ITP449 Master - Slide56.py - PyCharm
ITP449 Master 〉 Lecture 3 〉 Slide56.py
Slide56
Project
Slide 39.py
Slide 40.py
Slide 41.py
Slide 46.py
Slide 47.py
Slide 49.py
Slide 52.py
Slide 53.py
Slide56.py
External Libraries
Scratches and Consoles
Slide56.py
1 import pandas as pd
2 import numpy as np
3
4 data = pd.DataFrame(np.arange(12).reshape((3, 4)),
5 index=["Colorado", "Utah", "New York"],
6 columns=["a", "b", "c", "d"])
7
8 print(data)
9 print(data[["b", "d"]])
10 print(data[data["c"] > 5])
11
DataFrame columns
Run: Slide56
"C:\Users\Nitin Kale\venv\Scripts\python.exe" "C:/Users/Nitin Kale/PycharmProjects/ITP449 Master/Lecture 3/Slide56.py"
a b c d
Colorado 0 1 2 3
Utah 4 5 6 7
New York 8 9 10 11
b d
Colorado 1 3
Utah 5 7
New York 9 11
a b c d
Utah 4 5 6 7
New York 8 9 10 11
Process finished with exit code 0
6: TODO
4: Run
5: Debug
Terminal
Python Console
Event Log
Connection to Python debugger failed: Interrupted function call: accept failed (6 minutes ago)
7:1 CRLF UTF-8 4 spaces Python 3.8 (venv)
1: Project
7: Structure
2: Favorites

File Edit View Navigate Code Refactor Run Tools VCS Window Help ITP449 Master - Slide 57.py - PyCharm

ITP449 Master 〉 Lecture 3 〉 Slide 57.py

Slide 57

Project

Slide 38.py
Slide 39.py
Slide 40.py
Slide 41.py
Slide 46.py
Slide 47.py
Slide 49.py
Slide 52.py
Slide 53.py
Slide56.py
Slide 57.py
External Libraries
Scratches and Consoles

Slide56.py | Slide 57.py

```
1   import pandas as pd
2   import numpy as np
3
4   data = pd.DataFrame(np.arange(12).reshape((3, 4)),
5                       index=["Colorado", "Utah", "New York"],
6                       columns=["a", "b", "c", "d"])
7
8   print(data)
9   print(data > 5)
10
11  data[data > 5] = 0
12  print(data)
13
```

Run: Slide 57

```
"C:\Users\Nitin Kale\venv\Scripts\python.exe" "C:/Users/Nitin Kale/PycharmProjects/ITP449 Master/Lecture 3/Slide 57.py"
          a  b   c   d
Colorado  0  1   2   3
Utah      4  5   6   7
New York  8  9  10  11
              a      b      c      d
Colorado  False  False  False  False
Utah      False  False   True   True
New York   True   True   True   True
          a  b  c  d
Colorado  0  1  2  3
Utah      4  5  0  0
New York  0  0  0  0

Process finished with exit code 0
```

6: TODO | 4: Run | 5: Debug | Terminal | Python Console | Event Log

Connection to Python debugger failed: Interrupted function call: accept failed (38 minutes ago) 13:1 UTF-8 4 spaces Python 3.8 (venv)

File Edit View Navigate Code Refactor Run Tools VCS Window Help ITP449 Master - Slide 58.py - PyCharm

ITP449 Master 〉 Lecture 3 〉 Slide 58.py

Slide 58

Project

Slide 39.py
Slide 40.py
Slide 41.py
Slide 46.py
Slide 47.py
Slide 49.py
Slide 52.py
Slide 53.py
Slide56.py
Slide 57.py
Slide 58.py
External Libraries

Slide 58.py

```
2   import numpy as np
3
4   data = pd.DataFrame(np.arange(12).reshape((3, 4)),
5                       index=["Colorado", "Utah", "New York"],
6                       columns=["a", "b", "c", "d"])
7
8   print("Index rows and columns by index:")
9   print(data)
10  print(data.iloc[2])
11  print(data.iloc[2, [3, 0, 1]])
12  print(data.iloc[[1, 2], [3, 0, 1]])
```

iloc is integer location index
Used for integer based indexing
Use : to display all rows/columns
Data.iloc[:,:]

Run: Slide 58

```
            a   b   c   d
Colorado    0   1   2   3
Utah        4   5   6   7
New York    8   9  10  11
a     8
b     9
c    10
d    11
Name: New York, dtype: int32
d    11
a     8
b     9
Name: New York, dtype: int32
            d   a   b
Utah        7   4   5
New York   11   8   9

Process finished with exit code 0
```

6: TODO 4: Run 5: Debug Terminal Python Console Event Log

Connection to Python debugger failed: Interrupted function call: accept failed (42 minutes ago) 12:36 CRLF UTF-8 4 spaces Python 3.8 (venv)

Project

Slide 40.py
Slide 41.py
Slide 46.py
Slide 47.py
Slide 49.py
Slide 52.py
Slide 53.py
Slide56.py
Slide 57.py
Slide 58.py
Slide 59.py
External Libraries

Slide 59.py

```
1  import pandas as pd
2  import numpy as np
3
4  data = pd.DataFrame(np.arange(12).reshape((3, 4)),
5                      index=["Colorado", "Utah", "New York"],
6                      columns=["a", "b", "c", "d"])
7  print("Index rows and columns by labels:")
8  print(data)
9  print(data.loc["Colorado"])
10 print(data.loc["Colorado", ["b", "c"]])
11 print(data.loc[["Colorado", "Utah"], ["b", "c"]])
12 print(data.loc["Colorado":"Utah", "b":"d"])
```

loc is label based location index
Used for label based indexing

Run: Slide 59

```
Index rows and columns by labels:
          a  b   c   d
Colorado  0  1   2   3
Utah      4  5   6   7
New York  8  9  10  11
a    0
b    1
c    2
d    3
Name: Colorado, dtype: int32
b    1
c    2
Name: Colorado, dtype: int32
          b  c
Colorado  1  2
Utah      5  6
          b  c  d
Colorado  1  2  3
Utah      5  6  7
```

Pandas Arithmetic

FUNCTIONS FOR ARITHMETIC

Function	Description
add	Addition (+)
sub	Subtraction (-)
div	Division (/)
mul	Multiplication (*)
pow	Exponentiation (**)

File Edit View Navigate Code Refactor Run Tools VCS Window Help ITP449 Master - Slide 62.py - PyCharm

ITP449 Master 〉 Lecture 3 〉 Slide 62.py

Slide 62

Project

Slide 46.py
Slide 47.py
Slide 49.py
Slide 52.py
Slide 53.py
Slide56.py
Slide 57.py
Slide 58.py
Slide 59.py
Slide 62.py
External Libraries

Slide 62.py

```
1  import pandas as pd
2  import numpy as np
3
4  data = pd.DataFrame(np.arange(12).reshape((3, 4)),
5                      index=["Colorado", "Utah", "New York"],
6                      columns=["a", "b", "c", "d"])
7
8  print(data)
9  print(np.add(data["a"], data["b"]))
10
```

Run: Slide 62

```
"C:\Users\Nitin Kale\venv\Scripts\python.exe" "C:/Users/Nitin Kale/PycharmProjects/ITP449 Master/Lecture 3/Slide 62.py"
          a  b   c   d
Colorado  0  1   2   3
Utah      4  5   6   7
New York  8  9  10  11
Colorado     1
Utah         9
New York    17
dtype: int32

Process finished with exit code 0
```

6: TODO | 4: Run | 5: Debug | Terminal | Python Console | Event Log

Connection to Python debugger failed: Interrupted function call: accept failed (57 minutes ago) 9:36 UTF-8 4 spaces Python 3.8 (venv)

1: Project | 7: Structure | 2: Favorites

Sorting

File Edit View Navigate Code Refactor Run Tools VCS Window Help ITP449 Master - Slide 64.py - PyCharm

ITP449 Master 〉 Lecture 3 〉 Slide 64.py

Slide 64

Project

Slide 52.py
Slide 53.py
Slide56.py
Slide 57.py
Slide 58.py
Slide 59.py
Slide 62.py
Slide 64.py
External Libraries
Scratches and Consoles

Slide 62.py | Slide 64.py

```
1  import pandas as pd
2  import numpy as np
3
4  series = pd.Series(range(4), index=["d", "a", "b", "c"])
5  print(series)
6  print("Sort series ascending by index: \n", series.sort_index())
7  print("Sort series descending by index: \n", series.sort_index(ascending=False))
8
```

Run: Slide 64

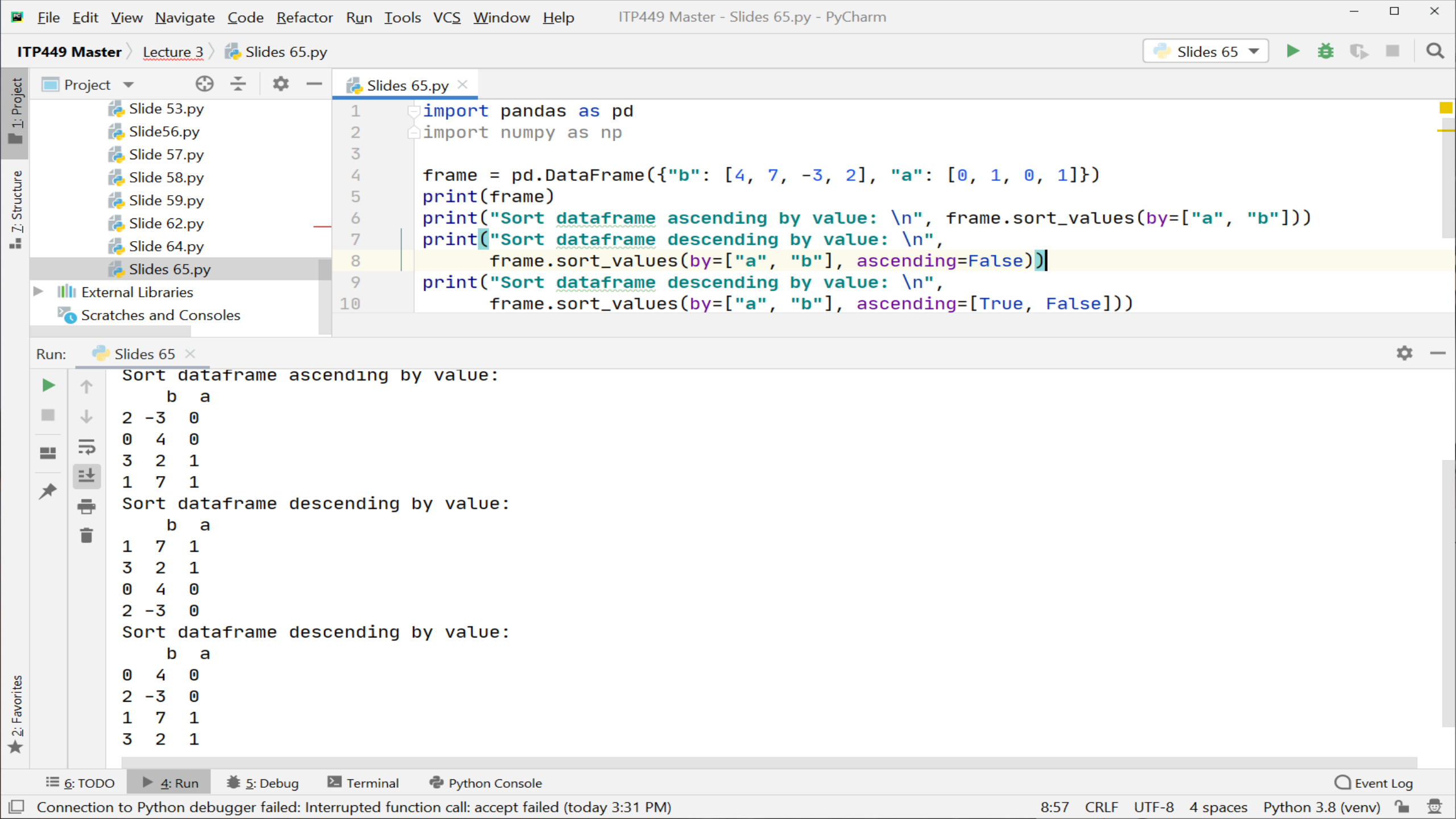
```
"C:\Users\Nitin Kale\venv\Scripts\python.exe" "C:/Users/Nitin Kale/PycharmProjects/ITP449 Master/Lecture 3/Slide 64.py"
d    0
a    1
b    2
c    3
dtype: int64
Sort series ascending by index:
 a    1
b    2
c    3
d    0
dtype: int64
Sort series descending by index:
 d    0
c    3
b    2
a    1
dtype: int64
```

6: TODO | 4: Run | 5: Debug | Terminal | Python Console | Event Log

Connection to Python debugger failed: Interrupted function call: accept failed (1 hour ago)

8:1 CRLF UTF-8 4 spaces Python 3.8 (venv)

1: Project | 7: Structure | 2: Favorites


File Edit View Navigate Code Refactor Run Tools VCS Window Help
ITP449 Master - Slides 65.py - PyCharm
ITP449 Master 〉 Lecture 3 〉 Slides 65.py
Slides 65
Project
Slide 53.py
Slide56.py
Slide 57.py
Slide 58.py
Slide 59.py
Slide 62.py
Slide 64.py
Slides 65.py
External Libraries
Scratches and Consoles
Slides 65.py
1 import pandas as pd
2 import numpy as np
3
4 frame = pd.DataFrame({"b": [4, 7, -3, 2], "a": [0, 1, 0, 1]})
5 print(frame)
6 print("Sort dataframe ascending by value: \n", frame.sort_values(by=["a", "b"]))
7 print("Sort dataframe descending by value: \n",
8 frame.sort_values(by=["a", "b"], ascending=False))
9 print("Sort dataframe descending by value: \n",
10 frame.sort_values(by=["a", "b"], ascending=[True, False]))
Run: Slides 65
Sort dataframe ascending by value:
b a
2 -3 0
0 4 0
3 2 1
1 7 1
Sort dataframe descending by value:
b a
1 7 1
3 2 1
0 4 0
2 -3 0
Sort dataframe descending by value:
b a
0 4 0
2 -3 0
1 7 1
3 2 1
6: TODO
4: Run
5: Debug
Terminal
Python Console
Event Log
Connection to Python debugger failed: Interrupted function call: accept failed (today 3:31 PM)
8:57 CRLF UTF-8 4 spaces Python 3.8 (venv)
1: Project
7: Structure
2: Favorites

do the following:

Combine two Series to form a DataFrame

```
import pandas as pd
import numpy as np

ser1 = pd.Series(list('abcedfghijklmnopqrstuvwxyz'))
ser2 = pd.Series(np.arange(26))
```

```
  col1  col2
0    a     0
1    b     1
2    c     2
3    e     3
4    d     4

Process finished with exit code 0
```

File Edit View Navigate Code Refactor Run Tools VCS Window Help ITP449 Master - Slide 67.py - PyCharm

ITP449 Master 〉 Lecture 3 〉 Slide 67.py

Slide 67

Project

Slide56.py
Slide 57.py
Slide 58.py
Slide 59.py
Slide 62.py
Slide 64.py
Slide 67.py
Slides 65.py
External Libraries
Scratches and Consoles

Slide 67.py

```
1  import pandas as pd
2  import numpy as np
3
4  letters = pd.Series(list("abcdefghijklmnopqrstuvwxyz"))
5  numbers = pd.Series(np.arange(26))
6  data = {"Letters": letters, "Numbers": numbers}
7
8  frame = pd.DataFrame(data)
9  print(frame)
```

Run: Slide 67

```
"C:\Users\Nitin Kale\venv\Scripts\python.exe" "C:/Users/Nitin Kale/PycharmProjects/ITP449 Master/Lecture 3/Slide 67.py"
   Letters  Numbers
0        a        0
1        b        1
2        c        2
3        d        3
4        e        4
5        f        5
6        g        6
7        h        7
8        i        8
9        j        9
10       k       10
11       l       11
12       m       12
13       n       13
14       o       14
15       p       15
16       q       16
```

1: Project 7: Structure 2: Favorites

6: TODO 4: Run 5: Debug Terminal Python Console Event Log

Connection to Python debugger failed: Interrupted function call: accept failed (today 3:31 PM) 7:1 CRLF UTF-8 4 spaces Python 3.8 (venv)

do the following:

Display the 25^{th}, 50^{th} and 75^{th} percentiles for a Series containing 100 random numbers from the standard normal distribution.

Project

Slide 57.py
Slide 58.py
Slide 59.py
Slide 62.py
Slide 64.py
Slide 67.py
Slide 69.py
Slides 65.py
External Libraries
Scratches and Consoles

Slide 69.py

```
1  import pandas as pd
2  import numpy as np
3
4  ser = pd.Series(np.random.randn(100))
5
6  print(ser)
7
8  print(np.percentile(ser, q=[25, 50, 75]))
```

Run: Slide 69

```
"C:\Users\Nitin Kale\venv\Scripts\python.exe" "C:/Users/Nitin Kale/PycharmProjects/ITP449 Master/Lecture 3/Slide 69.py"
0    -0.536177
1    -1.344814
2    -0.874982
3    -1.519546
4     1.824164
        ...
95   -1.040081
96   -0.417140
97   -0.253105
98    0.455927
99    0.372286
Length: 100, dtype: float64
[-0.69197565 -0.07720445  0.72205481]

Process finished with exit code 0
```

Summary

The dataframe data structure will be used extensively this semester

In the next lecture we will learn about data visualization