Microsoft

MLADS

MACHINE LEARNING, AI,
AND DATA SCIENCE CONFERENCE

November 14- 18, 2022

1

Microsoft

Introduction to Python

Shep Sheppard
shep.sheppard@microsoft.com
https://www.linkedin.com/in/sqlshep/
@sqlshep

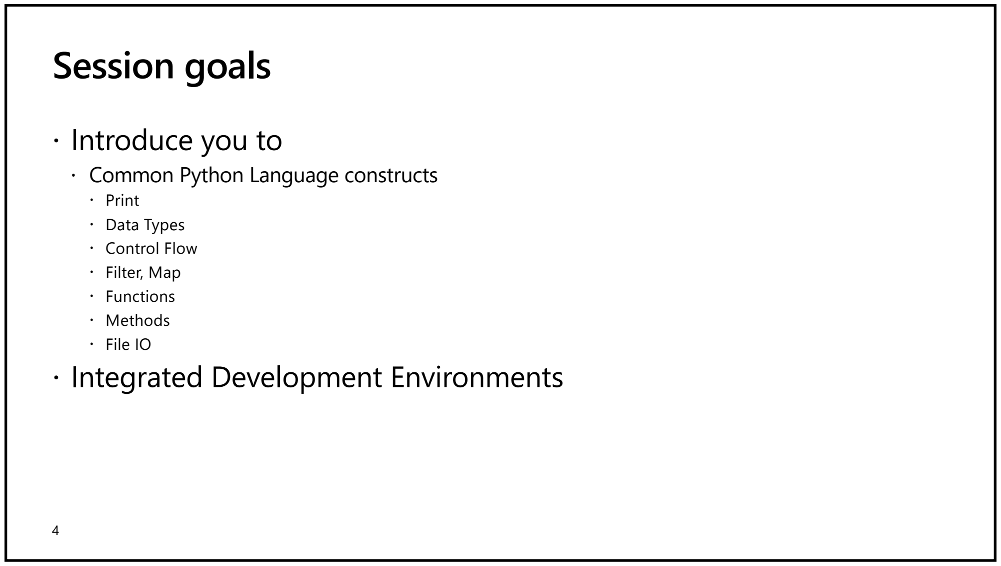
2

Microsoft

Introduction to Python

Shep Sheppard
Customer Engineer
Fast Track ISV

3



Session goals

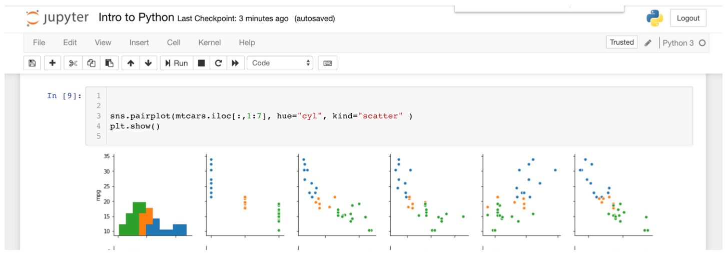
- Introduce you to
  - Common Python Language constructs
    - Print
    - Data Types
    - Control Flow
    - Filter, Map
    - Functions
    - Methods
    - File IO
- Integrated Development Environments

4

4

IDE's and Notebooks

- Jupyter notebook is an open-source web based application that allows code, equations, LaTex, Markup, data visualizations.
- Typically used for exploration, data cleaning, transformation, stats, machine learning



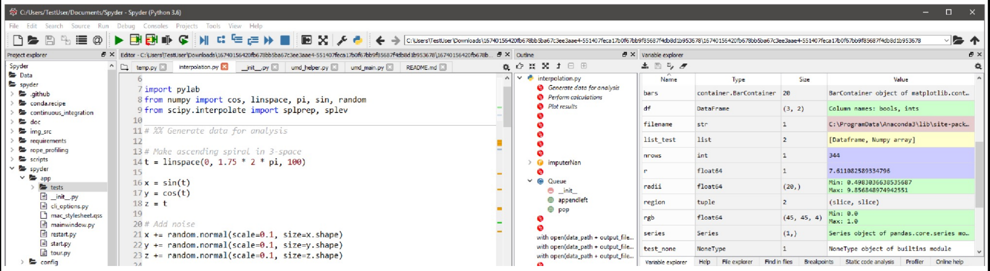
```
sns.pairplot(mtcars.iloc[:,1:7], hue="cyl", kind="scatter" )
plt.show()
```

5

5

Interactive IDE - Spyder

- Spyder is a more typical IDE interface "designed by and for scientists, engineers and data analysts"



```
import pylab
from numpy import cos, linspace, pi, sin, random
from scipy.interpolate import splprep, splev

# %% Generate data for analysis

# Make ascending spiral in 3-space
t = linspace(0, 1.75 * 2 * pi, 100)

x = sin(t)
y = cos(t)
z = t

# Add noise
x += random.normal(scale=0.1, size=x.shape)
y += random.normal(scale=0.1, size=y.shape)
z += random.normal(scale=0.1, size=z.shape)
```

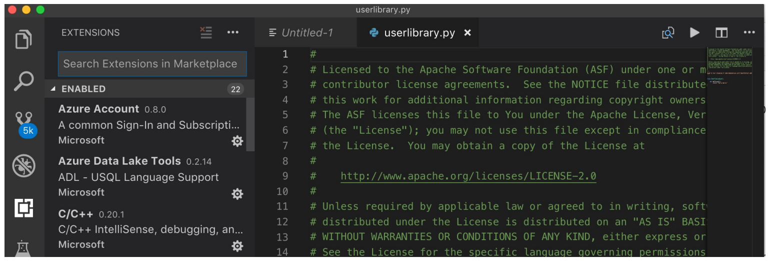
Name	Type	Size	Value
bars	container.BarContainer	20	BarContainer object of matplotlib.cont…
df	DataFrame	(3, 2)	Column names: bools, ints
filename	str	1	C:\ProgramData\Anaconda3\lib\site-pack…
list_test	list	2	[Dataframe, Numpy array]
nrows	int	1	344
r	float64	1	7.611082589334796
radii	float64	(20,)	Min: 0.4983036638535687 Max: 9.856848974942551
region	tuple	2	(slice, slice)
rgb	float64	(45, 45, 4)	Min: 0.0 Max: 1.0
series	Series	(1,)	Series object of pandas.core.series mo…
test_none	NoneType	1	NoneType object of builtins module

6

6

Interactive IDE – Visual Studio Code

- Visual Studio Code is open source and mulit-platform multi language code editor
- IntelliSense, Git integration, thousands of extensions, integrated debugging



Azure Account 0.8.0
A common Sign-In and Subscripti...
Microsoft

Azure Data Lake Tools 0.2.14
ADL - USQL Language Support
Microsoft

C/C++ 0.20.1
C/C++ IntelliSense, debugging, an...
Microsoft

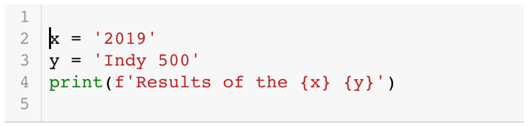
```
#
# Licensed to the Apache Software Foundation (ASF) under one or m
# contributor license agreements.  See the NOTICE file distribute
# this work for additional information regarding copyright owners
# The ASF licenses this file to You under the Apache License, Ver
# (the "License"); you may not use this file except in compliance
# the License.  You may obtain a copy of the License at
#
#    http://www.apache.org/licenses/LICENSE-2.0
#
# Unless required by applicable law or agreed to in writing, soft
# distributed under the License is distributed on an "AS IS" BASI
# WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
# See the License for the specific language governing permissions
```

7

7

Printing

- print() statement will output a string, formatted string or number to console or notebook
- Used for simple out put or debugging code on the fly
- "f" known as the formatted string literal can format and output, not supported in Databricks Python distribution
- In a notebook, unless all lines use print only last print will display



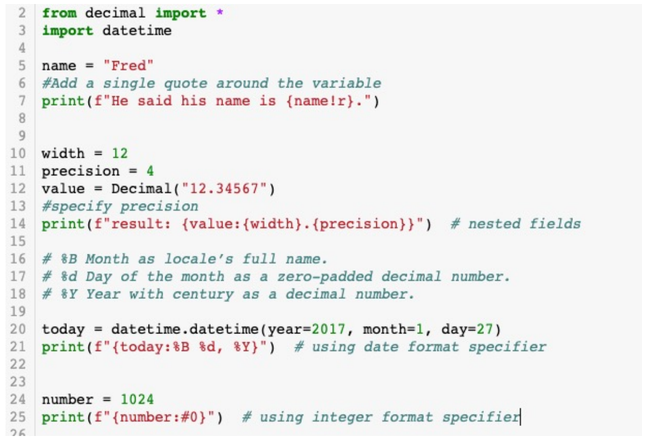
```

x = '2019'
y = 'Indy 500'
print(f'Results of the {x} {y}')

```

8

8

Format String Literals

- Slightly more advanced literals in the print statement



```python
from decimal import *
import datetime

name = "Fred"
#Add a single quote around the variable
print(f"He said his name is {name!r}.")


width = 12
precision = 4
value = Decimal("12.34567")
#specify precision
print(f"result: {value:{width}.{precision}}")  # nested fields

# %B Month as locale’s full name.
# %d Day of the month as a zero-padded decimal number.
# %Y Year with century as a decimal number.

today = datetime.datetime(year=2017, month=1, day=27)
print(f"{today:%B %d, %Y}")  # using date format specifier


number = 1024
print(f"{number:#0}")  # using integer format specifier
```

Types

- type(object) – find out what is it?
- Boolean – Ture or False
- Numeric types
  - Int, float
  - complex - Real and imaginary
- Text – String
- Sequence
  - List, tuple, range
- Binary
  - Byte, Byte array

- Comparisons

<	strict less than
<=	less than or equal
>	strict greater than
>=	greater than or equal
==	equal
!=	not equal
is	object identity
“is not”	negate object identity

Operations on Numeric Types

+	sum	int()	convert to int
-	difference	float()	convert to float
*	product	divmod()	(x // y, x % y)
/	divide or quotient	pow(x,y)	x to the power of y
//	floored quotient	x**y	x to the power of y
%	remainder (modulo)	complex(re, im)	
-x	negate	c.conjugate()	conjugate a complex
+x	unchanged		
abs()	absolute		

11

11

Sequence Types

- Basic Sequences include list, tuple and range
- list([iterable]) – construct of items of the same type
- tuples([iterable]) – immutable sequences
- range(start, stop, [step]) - immutable sequence of numbers

12

12

List []

- Mutable Array of compound data types
- Versatile in that can store a mix of values
- Can store more than one dimension of data
- For more than 2 dimensions of data consider using numpy
- Methods for append, iterating and removing elements via pop and remove

```
myList = ["One", "Two", "Three", "Four","Five", "Six", "Seven"]

myList = [["One",1], ["Two",2],["Three",3], ["Four",4,], ["Five",5]]
```

13

13

Tuple ()

- Immutable Array of compound data types
- Versatile in that can store a mix of values
- Can store more than one dimension of data
- Few methods since it is read only

```

my2dTuple = (["One",1], ["Two",2],["Three",3], ["Four",4,], ["Five",5])

my2dTuple[3][0]


```

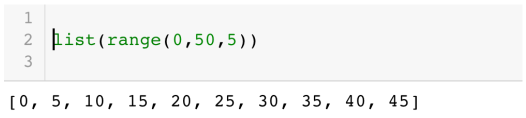
```
'Four'
```

14

14

Range

- Immutable sequence of numbers
- Typically used for looping
- range(*start*, *stop*[, *step*])



```
list(range(0,50,5))
```

```
[0, 5, 10, 15, 20, 25, 30, 35, 40, 45]
```

15

15

Indexing

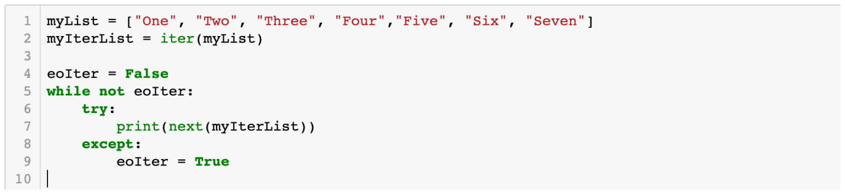
- Used for accessing specific dimensions of lists, tuples and arrays
- myList[:]   - returns everything
- myList[:100] - return the first 100 rows
- myList[100:] – return everything after the first 100 rows
- myList[100:200] return everything between index 100 and 200

16

16

Iterator

- An object representing a stream of data
- Repeated calls to the iterator’s next() method or passing it to the built-in function next() return successive items in the stream.
- Can be created from list, tuple, or range



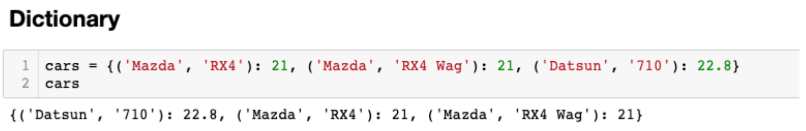
```
myList = ["One", "Two", "Three", "Four","Five", "Six", "Seven"]
myIterList = iter(myList)

eoIter = False
while not eoIter:
    try:
        print(next(myIterList))
    except:
        eoIter = True

```

17

17

Dictionary

- Store Key Value pairs
- Mutable
- Unordered collection
- Iterable
- Cannot mutate a dictionary while iterating it



Dictionary

```
cars = {('Mazda', 'RX4'): 21, ('Mazda', 'RX4 Wag'): 21, ('Datsun', '710'): 22.8}
cars
```

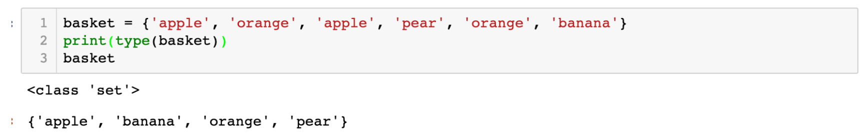
```
{('Datsun', '710'): 22.8, ('Mazda', 'RX4'): 21, ('Mazda', 'RX4 Wag'): 21}
```

18

18

Sets {}

- Use the keyword set or {}
- An unordered collection of unique elements
- Support union and intersect operations



```
basket = {'apple', 'orange', 'apple', 'pear', 'orange', 'banana'}
print(type(basket))
basket
```

```
<class 'set'>
```

```
{'apple', 'banana', 'orange', 'pear'}
```

19

19

Common Control Flow statements

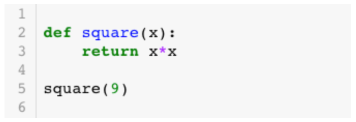
- If - test condition using operator
  - Elif – (optional) test condition inside if
  - Else – (optional) if and elif do not match run else
- For – while some condition true, loop
- While – while some condition is true
- Break – break out of loop

20

20

Functions

- Reusable code
- Functions are the definition of an executable statement
- The function only executed when called explicitly
- Functions can accept a series of inputs and return an output or simply execute on the input
- Variables created in the function are only scoped to the function



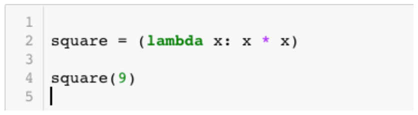
```python

def square(x):
    return x*x

square(9)

```

21

21

Anonymous Functions

- Lambda is small anonymous function
- The are syntactically restricted to a single expression
- Lambda functions can reference variables form the containing scope



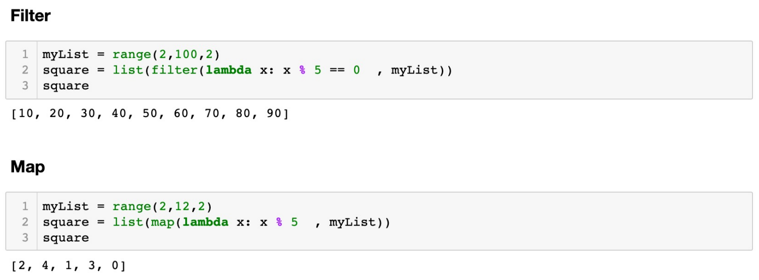
```python

square = (lambda x: x * x)

square(9)

```

22

22

Filter, Map

- Map will return an iterator that applies function to every item of iterable
- Filter returns an iterable list of elements for which function returns true



Filter

```
myList = range(2,100,2)
square = list(filter(lambda x: x % 5 == 0  , myList))
square
```

```
[10, 20, 30, 40, 50, 60, 70, 80, 90]
```

Map

```
myList = range(2,12,2)
square = list(map(lambda x: x % 5  , myList))
square
```

```
[2, 4, 1, 3, 0]
```

23

23

Methods

- Canonically, a method is a procedure associated with a class object
- Different object types have different methods

- String.capitalize()
- String.count()
- String.lower()
- String.replace()
- String.split()

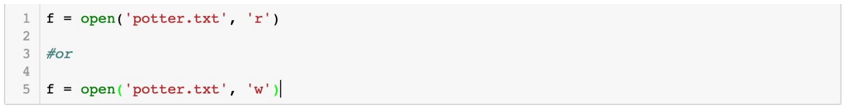
- List.append()
- List.remove()
- List.pop()
- List.clear()
- List.sort()

24

24

Python File IO

- Built in python functions for reading and writing
- Files are typically opened in text mode, indicating you read and write strings.
- Binary files are supported by opening with the "b" option
- Read is supported with the "r" option
- Write is supported with the "w" option
- Combine with "rb+"



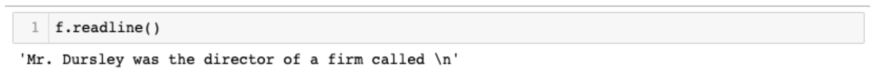
```
f = open('potter.txt', 'r')

#or

f = open('potter.txt', 'w')
```

25

25

File IO

- Reading and writing are done via python methods
- File.read() – read the entire file
- File.readline() – read the next line
- File.seek(5) – goto the 6th byte in the file
- File.write("string") – will write string to the open file(overwrite)
- File.close() when you are done



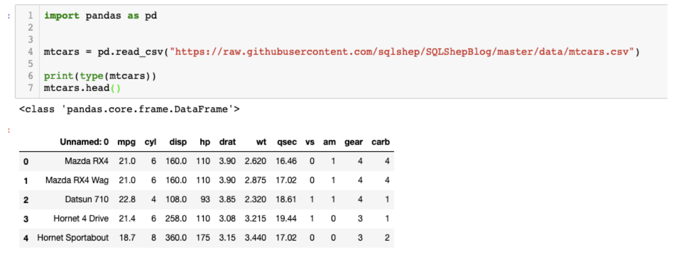
```
f.readline()
'Mr. Dursley was the director of a firm called \n'
```

26

26

IO with CSV and URL

- It is likely you will not be interested in file io using python read or write
- Third party packages make life easier with read from csv, url, etc…
- Pandas package will read csv from URL or disk and import as a dataframe



```python
import pandas as pd


mtcars = pd.read_csv("https://raw.githubusercontent.com/sqlshep/SQLShepBlog/master/data/mtcars.csv")

print(type(mtcars))
mtcars.head()
```

```
<class 'pandas.core.frame.DataFrame'>
```

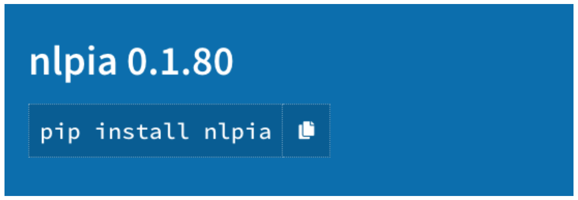
	Unnamed: 0	mpg	cyl	disp	hp	drat	wt	qsec	vs	am	gear	carb
0	Mazda RX4	21.0	6	160.0	110	3.90	2.620	16.46	0	1	4	4
1	Mazda RX4 Wag	21.0	6	160.0	110	3.90	2.875	17.02	0	1	4	4
2	Datsun 710	22.8	4	108.0	93	3.85	2.320	18.61	1	1	4	1
3	Hornet 4 Drive	21.4	6	258.0	110	3.08	3.215	19.44	1	0	3	1
4	Hornet Sportabout	18.7	8	360.0	175	3.15	3.440	17.02	0	0	3	2

27

27

Packages

- Packages are an external collection of Python functions, methods and types.
- Packages are typically specific to the problem they are trying to solve
- There are thousands packages available to be downloaded
- Easily installed via pip in the OS of your choice



nlpia 0.1.80

```
pip install nlpia
```

28

28

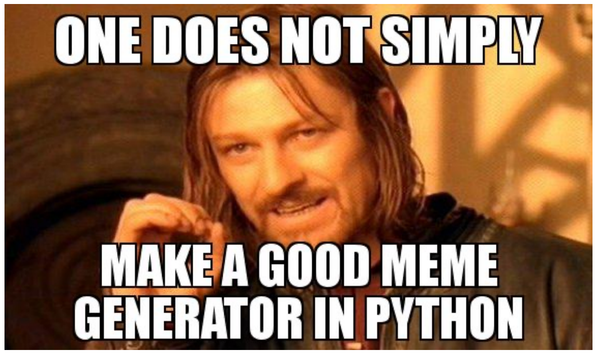
Top Packages for Data Science

- **TensorFlow** - is an open source software library for numerical computation using data flow graphs.
- **Pandas** - a fast, flexible, and expressive data structures designed to make working with "relational" or "labeled" data both easy and intuitive.
- **scikit-learn** – Built on Numpy and SciPy provides data mining and analytictools
- **PyTorch** – Provides tensor computation and DNN
- **Matplotlib** – Python plotting library for high quality graphics
- **Keras** – Neural Network API running on TensorFlow, CNTK, and Theano
- **Numpy** – Fundamental package needed for scientific computing
- **SciPy** – Open source software for mathematics, science and engineering

29

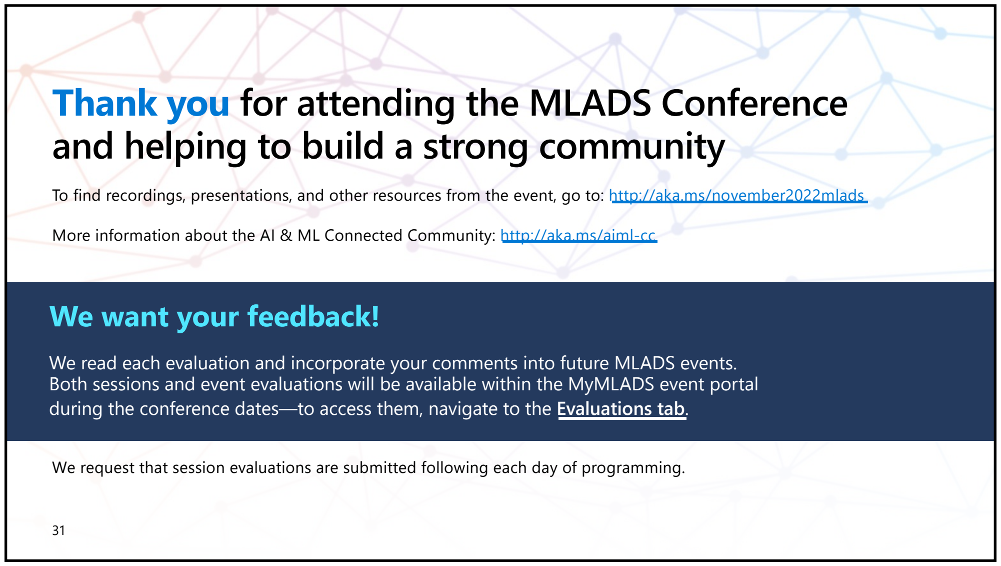
Summary

- Getting started with an IDE
- Print
- Python Data Types
- Control Flow
- Filter, Map
- Functions and Anon Functions
- Methods
- File IO



30

30



Thank you for attending the MLADS Conference and helping to build a strong community

To find recordings, presentations, and other resources from the event, go to: http://aka.ms/november2022mlads

More information about the AI & ML Connected Community: http://aka.ms/aiml-cc

We want your feedback!

We read each evaluation and incorporate your comments into future MLADS events.
Both sessions and event evaluations will be available within the MyMLADS event portal during the conference dates—to access them, navigate to the **Evaluations tab**.

We request that session evaluations are submitted following each day of programming.

31

31

Q&A

This slide is required—do not delete—please read the notes for this slide, then delete this text box

32

32

Microsoft

© Copyright Microsoft Corporation. All rights reserved.

33

© Microsoft Corporation. All rights reserved. MICROSOFT MAKES NO WARRANTIES, EXPRESS, IMPLIED OR STATUTORY, AS TO THE INFORMATION IN THIS PRESENTATION.