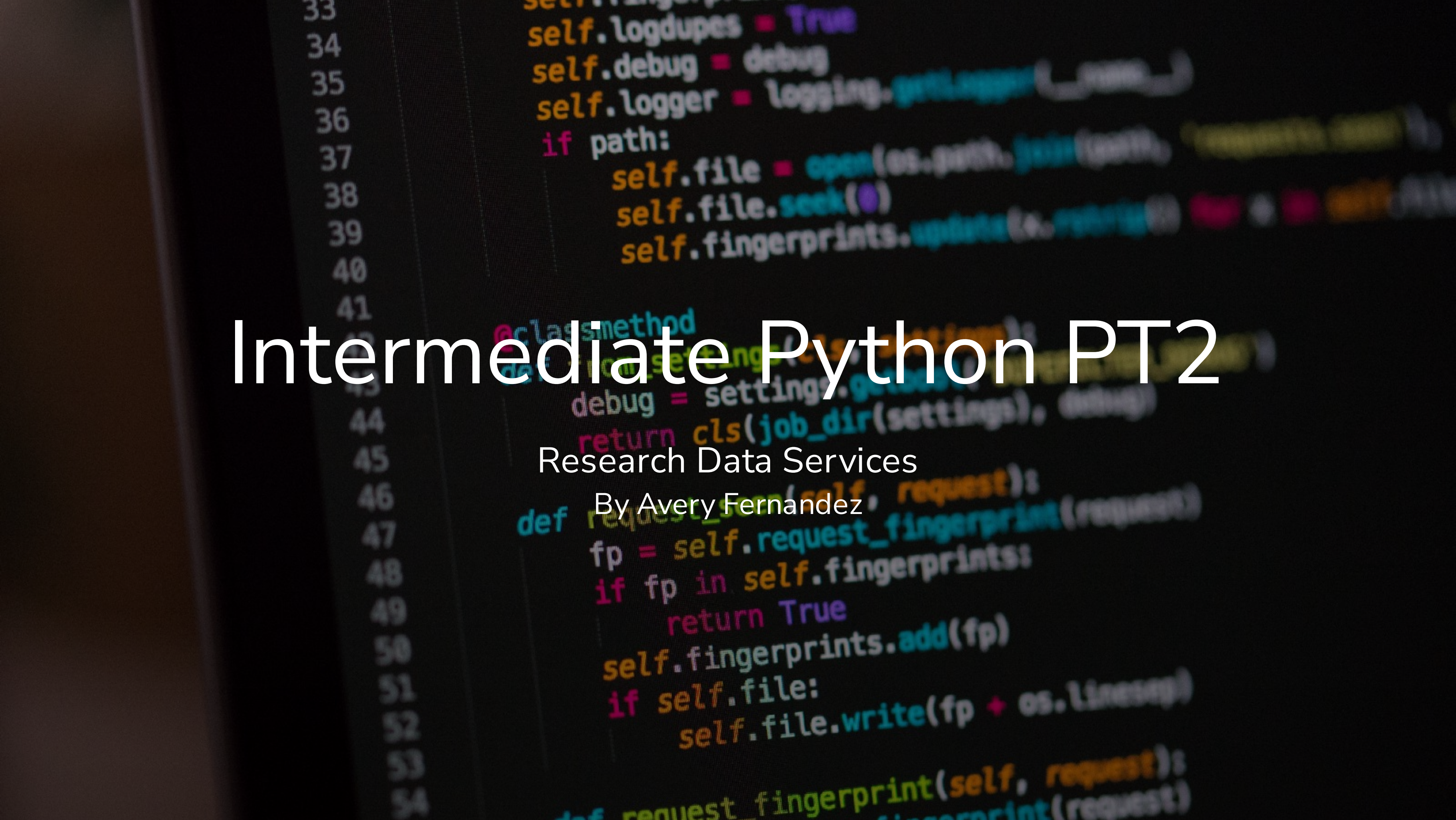


Intermediate Python PT2

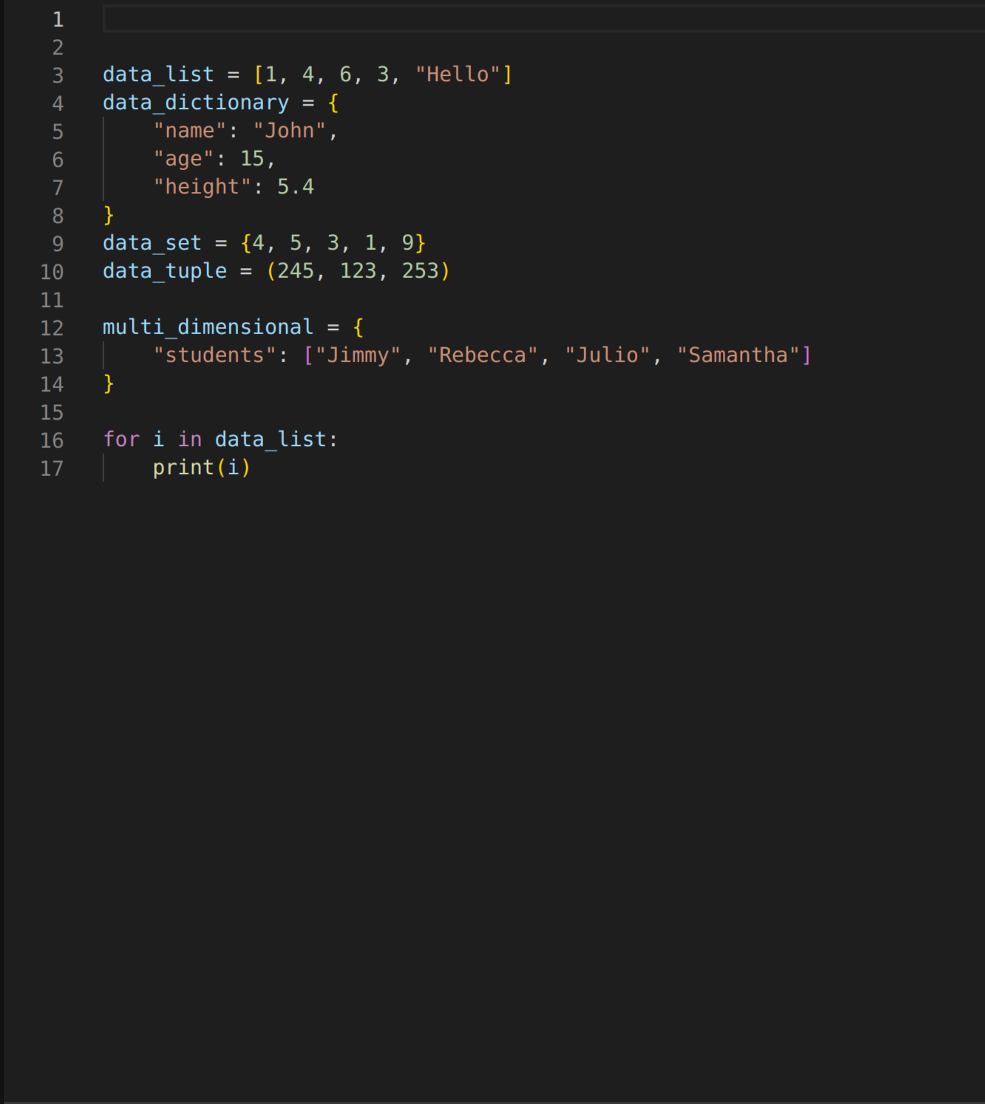
Research Data Services

By Avery Fernandez

Plans For Today

Learning the basics to python

- For Loops with data structures
- Parsing data structures
- Writing/Reading files
- Pip installting python libraries



```python
data_list = [1, 4, 6, 3, "Hello"]
data_dictionary = {
    "name": "John",
    "age": 15,
    "height": 5.4
}
data_set = {4, 5, 3, 1, 9}
data_tuple = (245, 123, 253)

multi_dimensional = {
    "students": ["Jimmy", "Rebecca", "Julio", "Samantha"]
}

for i in data_list:
    print(i)
```

For Loops

With python, looping through these data sets will be a lot easier.

for in range() is just looping through a list

```
for i in range(5):
  print(i)
```

Output

```
0
1
2
3
4
```

Range in list format

```
print(list(range(5)))
```

Output

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

For Loops

Looping is practically all the same, just slightly different

Types of For Loops:

- Lists
- Dictionaries
- Multi-Dimensional

For Loops: Lists

For loops with list can be achieved 2 ways:

- values
- indexes

Syntax:

```python
for i in myList:
    print(i)
```

Output

```
1
3
5
2
5
4
```

Value For Loop

Enumerate adds a counter to For Loops

```python
for idx, i in enumerate(myList):
    print(idx, i)
```

```
0 1
1 3
2 5
3 2
4 5
5 4
```

Index For Loop

Syntax:

```
for {variableName} in range(len({listName})):
    print(listName[variableName])
```

```
for i in range(len(myList)):
    print(i, myList[i])
```

```
0 1
1 3
2 5
3 2
4 5
5 4
```

Exercise 1

Looping through the List and print out each value

```
myList = ["Lists", "Dictionaries", "Sets", "Tuples"]
```

Exercise 1 Code

```
for storageType in myList:
    print(storageType)
```

```
for idx in range(len(myList)):
    print(myList[idx])
```

Output

```
Lists
Dictionaries
Sets
Tuples
```

Extended List Loops

We can use the same system to loop Lists inside Lists

```python
students = [
  ["Daniel", "Sophomore",[90, 20, 100]],
  ['Jacob', 'Sophmore', [90, 20, 30]],
  ['Tristan', 'Junior', [100, 65, 38]],
  ["Julie","Junior", [100, 65, 87]],
]
```

```python
for student in students:
    print(student)
```

Output

```
['Daniel', 'Sophomore', [90, 20, 100]]
['Jacob', 'Sophmore', [90, 20, 30]]
['Tristan', 'Junior', [100, 65, 38]]
['Julie', 'Junior', [100, 65, 87]]
```

Exercise 2

loop through a lists of list and print only the scores, individually

```
students = [
  ["Daniel", "Sophomore",[90, 20, 100]],
  ['Jacob', 'Sophmore', [90, 20, 30]],
  ['Tristan', 'Junior', [100, 65, 38]],
  ["Julie","Junior", [100, 65, 87]],
]
```

Exercise 2 Code

```
students = [
    ["Daniel", "Sophomore",[90, 20, 100]],
    ['Jacob', 'Sophmore', [90, 20, 30]],
    ['Tristan', 'Junior', [100, 65, 38]],
    ["Julie","Junior", [100, 65, 87]],
]

for student in students:
    print(student[0])
    for grade in student[2]:
        print(grade)
```

Output

```
Daniel
90
20
100
Jacob
90
20
30
Tristan
```

For Loops: Dictionaries

You can loop through multiple things based off what you want to do

- keys
- pairs

Dictionaries: Keys For Loop

Syntax:

```
for {variableName} in {dictionary}:
```

Keys Loop

```
states = {
"AL": "Alabama",
"AK": "Alaska",
"AZ": "Arizona",
"AR": "Arkansas"
}
for abbreviations in states:
  print(abbreviations, states[abbreviations])
```

Output

```
AL Alabama
AK Alaska
AZ Arizona
AR Arkansas
```

Dictionaries: Pair For Loop

Syntax:

```
for key,value in dictionary.items():
```

Pairs Loop

```
states = {
"AL": "Alabama",
"AK": "Alaska",
"AZ": "Arizona",
"AR": "Arkansas"
}
for abbreviations,state in states.items():
  print(abbreviations, state)
```

Output

```
AL Alabama
AK Alaska
AZ Arizona
AR Arkansas
```

Exercise 3

Loop through the dictionary in a list

Print out the key and value on the same line:

```
students = [
  {
    "name": "Daniel",
    "year": "Sophmore"
  },
  {
    "name": "Tristan",
    "year": "Junior"
  }
]
```

Exercise 3 Code

```
students = [
  {
    "name": "Daniel",
    "year": "Sophmore"
  },
  {
    "name": "Tristan",
    "year": "Junior"
  }
]

for student in students:
    for key, value in student.items():
        print(key, value)
```

Output

```
name Daniel
year Sophmore
name Tristan
year Junior
```

Using Files in Python

It is super useful to use an external file to save or grab data from

Syntax

```
open({fileName}, "{operation}")
```

Operations

- Read **r**
- Read Binary **rb**
- Write **w**
- Write Binary **wb**
- Append **a**

Reading From a File

```
with open("data.txt", "r") as inFile:
    inFile.read()
```

```
with open("data.txt", "r") as inFile:
    inFile.readline()
```

Write to a File

```python
with open("data.txt", "w") as outFile:
    outFile.write("Hello")
```

```python
with open("data.txt", "a") as outFile:
    outFile.write("more please")
```

External Libraries

We can use pip to install and manage our packages

The commands will be entered inside the terminal

Installing a library

```
pip install {library}
```

List Libraries

```
pip list
```

Unistalling Libraries

```
pip unistall {library}
```

Using Numpy (External Library)

Terminal

```
pip install numpy
```

```
import numpy
print(numpy.random.randint(100))
```

```
from numpy import random
print(random.randint(100))
```

```
import numpy as np
print(np.random.randint(100))
```