Deloitte.

Tech Treks

Python Collections

Welcome!

- ROI leads the industry in designing and delivering customized technology and management training solutions
- Meet your instructor
  - Name
  - Background
  - Contact info
- Let’s get started!

Session Objectives

In this session, you will:

- Explore various Python collections
- Discover the NumPy, pandas, and matplotlib packages
- Use Python to process data from files

What Are Collections?

- A collection is a way of storing multiple values in a single variable
- Collections come in a variety of types:
  - `list` can store any type of data, can be changed, and keeps elements in the order they are entered
  - `tuple` is almost the same as a list but its elements cannot be modified; sometimes that's more efficient
  - `set` is like a list but it only allows a single entry for each unique value; it does not keep things in the order they are entered
  - `dict` is short for dictionary; it is modifiable but it stores data in key/value pairs

Why Collections

- Collections are like arrays or tables
  - They store multiple values which can be accessed by their indexes
  - They can be modified in some cases
  - They are a good way to store complex data and are a building block of data processing in Python

`list`

- `list` is the most fundamental type of collection
  - Each element is referred to by its index number starting with 0 not 1
  - It is flexible and can contain 0, 1, or more items of any data type
  - We can fetch individual elements or ranges of elements
  - Elements can be added and removed and changed
  - The collection can be sorted, filtered, aggregated, and more
  - To make a `list`, we just surround a comma-separated list of items inside of square brackets `[ ]`

```
numbers = [1, 2, 3, 4]
```

`tuple`

- `tuple` is used in many of the same places as a list
  - Each element is referred to by its index number starting with 0 not 1
  - It is flexible and can contain 0, 1, or more items of any data type
  - We can fetch individual elements or ranges of elements
  - A `tuple` cannot be modified after it is assigned to a variable
  - The collection can be sorted, filtered, aggregated, and more
  - To make a `tuple`, we just surround a comma-separated list of items inside of normal parentheses ( )

```
numbers = (1, 2, 3, 4)
```

set

- `set` is used in many of the same places as a list
  - You do not refer to individual elements with an index
  - It is flexible and can contain 0, 1, or more items of any data type
  - A `set` can have elements added or removed
  - The collection only stores unique values once and removes duplicates
  - To make a `set`, we just surround a comma-separated list of items inside of normal curly braces { }

```
numbers = {1, 2, 3, 4, 3}
```

dict

- `dict` is the most complex of the built-in Python collections
  - Instead of using a numerical index, each element has a named key
  - An element in a `dict` can be added, removed, or modified
  - We often refer to `dict` as having key/value pairs or KV
  - To make a `dict`, we just surround a comma-separated list of keys and values separated with a colon (`:`) inside of normal curly braces `{ }`

```
numbers = {'Alpha': 1, 'Beta': 2, 'Gamma': 3}
```

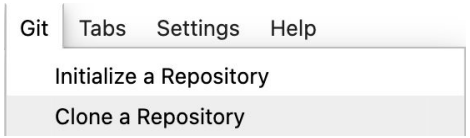
NumPy and pandas

- Collections can get complicated quickly
  - NumPy and pandas are external packages we have to download with `pip`
  - They are so common that almost everyone uses them
  - They make it easier to do manipulate complex structured data
  - The basic object we use in pandas is known as a `DataFrame`

```
from pandas import DataFrame
data = [{'team': 'Leicester', 'player': 'Vardy', 'goals': 24}
        ,{'team': 'Manchester City', 'player': 'Aguero', 'goals': 22}
        ,{'team': 'Arsenal', 'player': 'Sanchez', 'goals': 19}]
```

Jupyter Notebook

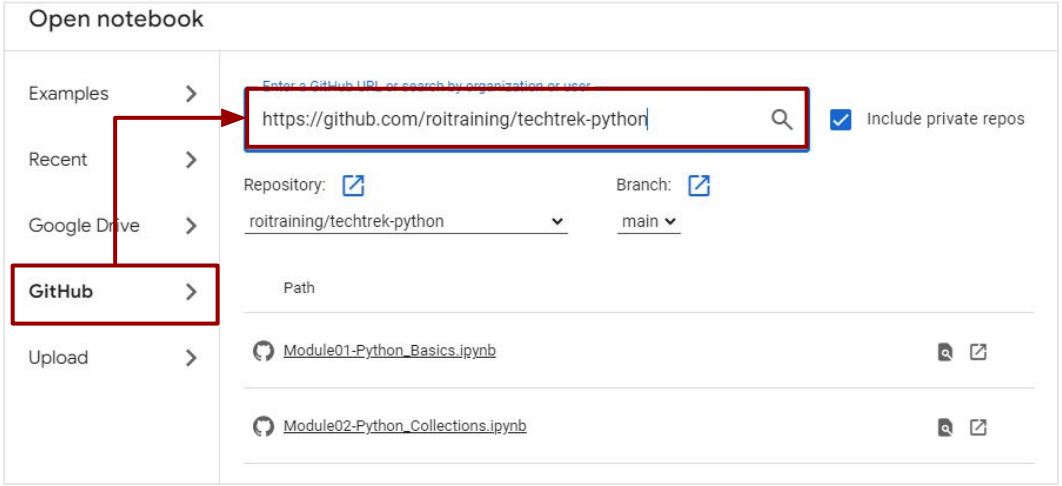
- Jupyter Notebook is perhaps the easiest way to get started with Python
- Let's start a Jupyter session by navigating to: https://notebooks.roitraining.com/
- Select the link nearest to your location
- From the top menu, choose **Git | Clone a Repository** and enter: **`https://github.com/roitraining/techtrek-python`**
  - Sometimes you may need to add `.git` to the end
- Click the **`Module02-Python_Collections.ipynb`**



Note: The Notebook server will only be available during class and one hour afterwards.

After Class

- To access the Notebook after class, go to: https://colab.research.google.com/
- Select **GitHub** and enter https://github.com/roitraining/techtrek-python
- You must have a Google-compatible email account to log in



Session Summary

In this session, you have:

- Explored various Python collections
- Discovered the NumPy, pandas, and matplotlib packages
- Used Python to process data from files

Discussion: Recap