Hello, Python

Introduction to Python

Data Sciences Institute, University of Toronto
Instructor: A Mahfouz | TA: Kaylie Lau
July 2022

Contents:

Welcome!

The course introduces the Python programming language starting from its basics, working up to loading, manipulating, and visualizing real datasets with popular data science libraries. It was designed by the University of Toronto's Data Science Institute for those who have a degree in something other than Computer Science/Statistics who are looking to enhance their data science skills for their career.
All course materials can be found on GitHub: **https://github.com/amfz/dsi-python-workshop**

Land Acknowledgement:

We wish to acknowledge this land on which the University of Toronto operates. For thousands of years it has been the traditional land of the Huron-Wendat, the Seneca, and the Mississaugas of the Credit River. Today, this meeting place is still the home to many Indigenous people from across Turtle Island and we are grateful to have the opportunity to work on this land.

Course Logistics

Course contacts

Instructor: A. Mahfouz (they/them)
Email: amhfz@proton.me
TA: Kaylie Lau (she/her)
Email: kaylie.lau@mail.utoronto.ca
Please include "DSI-Python" or similar in the subject line!

Format

The course runs synchronously over Zoom. It consists of three classes a week for two weeks, or six classes total. Classes are 6 PM - 8 PM EDT on Mondays and Thursdays, and 9 AM - 12 PM EDT on Saturdays. Being mindful of online fatigue, there will be one break during each class where students are encouraged to stretch, grab a drink and snacks, or ask any additional questions.

Prerequisites

Learners are expected to know how to operate a computer and to be able to install programs on their machines. Learners are also expected to be familiar with the parts of a data table or spreadsheet, summary statistics, and basic data visualizations. No prior programming knowledge is required.

Class expectations

The course is a live-coding class. Learners are expected to follow along with the coding in their own Python notebooks. Learners should be active participants while coding and are encouraged to ask questions throughout. Although slides will be available, they should be referenced before or after class, as class will be dedicated to coding with the instructor.

Technology requirements

Learners must have an internet connection and a computer to participate in online activities. Learners are highly encouraged to install the following programs in advance:

- Anaconda from **https://www.anaconda.com/products/individual#Downloads**
- Jupyter Notebook (install within Anaconda Navigator)

If learners prefer, they may use **Google Colab** instead.

Key Texts

- Gries, P., Campbell, J., Montojo, J., & Coron, T. (2017). *Practical programming: An introduction to computer science using python 3.6*.
- Adhikari, DeNero, and Wagner, (2021). *Computational and Inferential Thinking: The Foundations of Data Science*.
- Thomas, R. (2021). *Avoiding Data Disasters*. **https://www.fast.ai/2021/11/04/data-disasters/**
- Boykis, V. (2019). *Neural nets are just people all the way down.* **https://vicki.substack.com/p/neural-nets-are-just-people-all-the**

Setup

Options

Google Colab	Anaconda
• Easy to get started • Less flexible • Runs online through a Google account	• Can be tricky to install • Lots of flexibility • Installed on your own computer
Go to **https://colab.research.google.com/** In the upper left corner, click **File**, then **New notebook**.	Download and install from **anaconda.com/products/distribution#Downloads** Open **Anaconda Navigator**. Then, install and launch **Jupyter Notebook.**

Anaconda and Notebooks

Anaconda is a software suite that consists of Python and several complementary data science applications. It includes a Python *interpreter*, or a program that translates Python code into machine instructions. It also includes *Jupyter Notebook*, a web application that lets us combine pieces of executable code, text, images, and visualizations in a single document, or notebook. Even though Jupyter Notebook is a web application, notebooks are stored locally on your computer, not hosted online by default.
Sections of a notebook are called *cells*. Cells can be run independently of each other, and in any order.

Google Colab

Google Colab is a Jupyter Notebook environment hosted on the cloud -- that is, on Google's servers. Google Colab provides a standardized notebook environment, with common data science libraries already installed.

Check your Python version

Try adding and running your first code cells.

1. Add a cell by clicking the **+** in the toolbar (Jupyter Notebooks) or **+ Code** (Google Colab).
2. Enter the following: `!python --version`.
3. Press `ctrl + enter` to run the cell.

You should see Python 3.6 or higher.

```
In [1]: !python --version
```

```
Python 3.10.4
```

Adding text

Notebooks use markdown, a markup language, for formatting text. IBM publishes a **Jupyter-specific markdown cheat sheet**.
To add a text cell:

1. (Jupyter Notebooks) Click **+** in the toolbar. Then in the dropdown menu to the right, switch the cell from Code to **Markdown**.
2. (Google Colab) Click **+ Text**
3. Add some text. When you are done editing, press `ctrl + enter` to run the cell and format your text.

Course Tour

Getting Started

What are the core features of Python?

Topics

- Data types and variables
- Expressions and statements
- Functions
- Dealing with errors
- Comments and readability

What is Python?

Python is a general-purpose programming language first released in 1991. It has since become a popular language for data science, thanks to an enthusiastic community and a large ecosystem of code libraries and tools that make it easier to perform common tasks throughout the data science life cycle.

```python
# create a variable
degrees_celsius = 25

# convert to Fahrenheit
(9 / 5) * degrees_celsius + 32

# make it a function
def c_to_f(degrees_c):
    return (9 / 5) * degrees_c + 32
```

Dealing with Reality

What tools does Python give us for working with collections of values and for writing programs that change their behaviour based on data?

Topics

- Logic
- Conditionals
- Iteration
- Sequences and collections of values

```python
numbers = [2, 52, 18, 27, 5, 1937]

for number in numbers:
    if number % 2 == 0:
        print('Even')
    else:
        print('Odd')
```

In/Out

How can we bring in outside code and data to accomplish tasks? How can we get data out of our programs?

Topics

- Modules
- Working with files
- Object-oriented programming

```python
import os

for f in os.listdir('folder'):
    if f.endswith('.csv'):
        # process file
```

Doing More with Data

How can we work with real data for analysis?

Topics

- `numpy` and `pandas`
- Loading tabular data
- Exploring data
- Wrangling data
- Reshaping and combining datasets

In [2]:
```
import pandas as pd

neighbourhoods = pd.read_csv('../data/neighbourhood-profiles-2016-140-model.csv')
neighbourhoods[['Characteristic', 'City of Toronto']].head()
```

Out[2]:

	Characteristic	City of Toronto
0	Neighbourhood Number	NaN
1	TSNS2020 Designation	NaN
2	Population, 2016	2,731,571
3	Population, 2011	2,615,060
4	Population Change 2011-2016	4.50%

Visualizing Data

How do we create visualizations in Python?

Topics

- `matplotlib`, `seaborn` and `plotly`
- Bar charts, histograms, scatterplots, line charts
- Faceting and layering

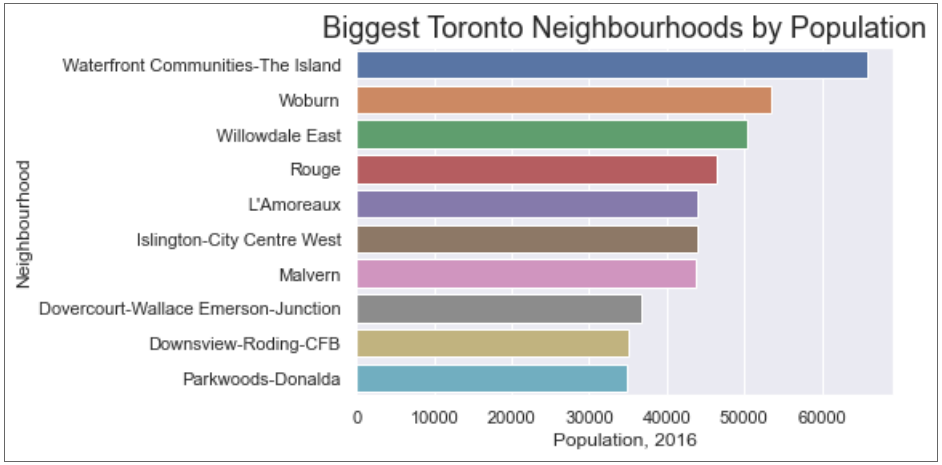
In [4]:
```
import seaborn as sns

sns.set_theme()

(sns.barplot(data=neighbourhoods.head(10),
             y='Neighbourhood',
             x='Population, 2016')
    .set_title('Biggest Toronto Neighbourhoods by Population',
               fontdict={'fontsize': 18}))
```

Out[4]:

```
Text(0.5, 1.0, 'Biggest Toronto Neighbourhoods by Population')
```



Do It Again

How can we make our work easier to reproduce and/or maintain?

Topics

- Environments
- Testing and debugging
- Moving beyond notebooks

In [5]:
```
import doctest

def c_to_f(degrees_c):
    '''
    Convert degrees Celsius to Fahrenheit
    >>> c_to_f(0)
    32.0
    >>> c_to_f(10)
    50.0
    '''
    return (9 / 5) * degrees_c + 32

doctest.testmod()
```

Out[5]:

```
TestResults(failed=0, attempted=2)
```

Ethics

What do we even mean when we talk about ethics in tech?

Topics

- What does "tech ethics" usually entail?
- Where do our data and models come from?
- What responsibility do we have for the data we collect and use?

Professional Skills

How can we prepare for a data science job interview?

Topics

- Technical resources
- Common questions

Let’s get started!