Data Dive Week 0: Intro To Programming

The Data Dive will run as a competition between teams to produce and present a data science project that answers a particular question. Programming, using languages like Python, is the common way to perform data cleaning, analysis, visualisation, and machine learning, and will likely be used by all groups.

Don't worry if you have little experience with programming, this guide aims to cover the basics, and more programming for data science will be covered in the remaining sessions of the Data Dive. If you have some level of programming skill, but want to recap some of the basics you will need, this might also be a good place to start!

Python

Python is the top programming language used for data science and machine learning, popular for being versatile and having very readable syntax.

Working With Python

There are multiple ways to use Python to program. You could:

- use it inside Google Colab
- install it on your machine, and use an IDE such as VSCode

You can access Google Colab from within Google Drive (by going New -> More -> Google Colaboratory), and this will create a 'notebook' where you can write and run snippets of Python code from the browser.

If you want to use Python locally, you can install it from this site, and also find and install your desired IDE and any Python extensions it may require.

More details on setting up your project will be given in Week 1!

Python Tutorial

Python code is written line-by-line, and executed in that order. You may see comments (lines of code that do not run, written just for the programmer's reference) written after a '#'.

```python
print("This is a line of code") # this is a comment
```

In Python, you can create variables (which store values) of a variety of data types, and assign them using '='. Below are the most common data types:

```python
name = "Rick"    # String (text)
age = 48         # Integer (a whole number)
height = 167.5   # Float (a real number
student = False  # Boolean (true/false)

names = ["Rick", "Alice", "Bob"]              # List (a set of multiple values of any type, that can be changed once created)
ages = {"Rick": 48, "Alice": 30, "Bob": 44}   # Dictionary (a mapping of key-value pairs
heights = (167.5, 174.2, 181.0)               # Tuple (a set of multiple values, that cannot be changed)
```

These data types have certain ways of allowing you to modify their values, add to them, compare them, etc. See these pages for more details on operators and specific data types.

There are also built-in functions in Python you can use to perform certain tasks, a common one being 'print', to print (show) the given value to the console:

```python
print(name)
# you will see "Rick" be displayed
```

Though code runs line-by-line, you can control when, if, or how often certain sections of code run by using loops and conditional statements:

```python
for i in range(10): # the indented lines will run 10 times
      print("Hello World")
      print(i)

for n in names: # the indented code will run for each value in the list 'names'
      print(n)

while (height < 200): # the indented lines will run, so long as the condition (that the value of 'height' is less than 200) is true
      height = height * 1.2

if (age == 48): # the indented lines below the if statement will run if the condition (that the value of 'age' is 48) is true
```

```
        print("You are 48!")
else: # otherwise, the lines below 'else' will run
        print("You are not 48.")
```

For data science, you will likely be working with different libraries (imported code that lets you perform extra tasks). Examples of common libraries being imported are shown below:

Python
```
import numpy as np # NumPy, for working with arrays and performing mathematical calculations
import pandas as pd # pandas, for working with 'dataframes', very common for data processing and analysis
import matplotlib.pyplot as plt # matplotlib, for creating visualisations

# using the 'as' keyword to give an alias to these libraries is generally the convention
```

If you want to know how to install (if it is not already included) and import a certain library you want, it can be useful to check on sites like PyPi and the library's documentation for how to do this.

You will also likely want to import data from files, such as csvs. This can be done in multiple ways, including using libraries like pandas to create a 'dataframe' (table-like object with rows and columns).

Python
```
# using pandas
import pandas as pd
df = pd.read_csv("csv_filepath.csv")

# using the csv library
import csv
with open("csv_filepath.csv", "r") as csv_file:
        file_reader = csv.reader(csv_file)
        for row in file_reader:
                print(row)
```

More details of working with libraries like pandas, numpy, and matplotlib will be given in Weeks 2 and 3!

Check out these resources for more details:

- DataSoc's Python Guide
- w3schools Python Tutorial
- datacamp Intro To Python for DS Course

SQL

SQL is used for working with relational databases, and can be used inside relation database management systems (RDBMS), or within languages like Python by using libraries like sqlite3 or sqlalchemy.

Check out DataSoc's SQL Guide for common SQL syntax.

R

The R language is commonly used for statistical work and data visualisation.

Check out DataSoc's R Guide for details of how to get started using R.