

Day 1: *Variables and Numbers in Python*

The chilly wetness of the west coast's winter had descended upon Vancouver, and Dot wasn't too happy about it. The rain pattered down on the city's streets all day long, the dark rainclouds obscuring the beautiful view of the Pacific coastline that Dot loved so much. Staring out the window from the home they remotely worked from, Dot began to think that it was time to take a trip. Dot had been saving up their vacation time at work — everything lined up perfectly. They even had a friend who would be eager to watch over their pet snake, Python. The more they thought about it, the more appealing the idea seemed. It was time for Dot to take a courageous leap and go international...to Europe and beyond!

Dot opened a browser on their laptop and started looking into flights. Their savings account wasn't bottomless, and the trip would be expensive, so they had to make sure to do things efficiently. Plus, the intercontinental flight would be very long, and Dot got restless on planes — they wanted to book a flight that would give them the best value per hour of flight duration. Dot wanted to fly to Europe from the Canadian city that would give them the best combination of value/duration. Dot could manually browse through all the available flights to compare them, but they knew there was a better way to do things — we can use data skills to map out the flights for them and decide from which city they should fly out.

Tutorial

Jupyter Notebooks

This document that you're currently reading is called a "Jupyter Notebook". It's similar to a text document, but you can do many things with it:

- run code
- show graphs
- pull data from various databases

We will learn all of those things during the 21 days.

The most important thing in Jupyter Notebooks is: **everything is a cell**

Notebooks are organized as a set of "*cells*". Each cell can contain different types of content: like Python code, images or even human readable text (markdown), like the one you're currently reading.

You can edit these cells by double clicking on them. Try editing the following cell:

 **Edit this cell** 

When you double click the cell it will open an `edit mode`.

Once you've made the changes, you have to **"run"** the cell to display the changes. You can see how cell can be run in the following cell.

There are many keyboard shortcuts that will let you interact with your documents, run code and make other changes. For example, there are two shortcuts to execute a cell:

1. `shift + return` : Run cell and advance to the next one.
2. `ctrl + return` : Run the cell but don't change focus.

There are a couple of differences between `edit` and `command` mode:

- Edit mode is indicated by a blue (can be green in some versions) cell border and a prompt showing in the editor area. Here you can type text or code inside a cell
- Command mode is indicated by a grey cell border with a blue left margin. Here you can edit the whole of the notebook. e.g Keyboard short cuts `A` to add a cell above or `B` to add a cell below.
- If you are focused on a cell and in edit mode, you can enter command mode with `esc` key, or hit `enter` to toggle back to edit mode

Here are a few examples of what content in cells can look like. Try running the following cells:

```
In [3]: # empty cell
```

```
In [4]: # Python Code
print(1+1)
```

2

```
In [ ]: 2 + 2
```

While in command mode, you can create new empty cells by using following shortcuts:

1. `A` : to create cell **above** the one you are currently at
2. `B` : to create cell **below** the one you are currently at

Now, let's go to learn some basics of Python programming!!

Variables in Python

In Python, variables are structures that store values within them. A variable can be in the form of a string, an integer, or even a list.

```
var_a = 'This is a string.'
```

```
var_b = 2022
```

```
var_c = 2022.3
```

```
var_d = ['This', 'is', 'a', 'list', 'on', 'the', 21, 'day data challenge.']
```

Note: In the example above, `var_a` is a **string** and `var_d` is a **list**. We can ignore these variables for now because we will learn more about them in the next challenges.

Numbers

There are two main numeric data types in Python, **integers** and **float** (`var_b` and `var_c` above respectively).

The integer data types refer to **whole numbers**, while float data types refer to any numeric value that isn't a whole, like a **fraction or a decimal**.

In Python as well as any other programming language, there are a variety of math operations you can use to do basic arithmetic.

To learn more about numbers in Python, take a look at these articles [here](#) and [here](#).

Challenge

Dot wants to find a connection to Europe that will give them the best value per hour of travel time. The following tickets are available for the connection:

- Vancouver - Toronto: 250 CAD, travel time 3.5 hours
- Vancouver - Ottawa: 280 CAD, travel time 4 hours
- Vancouver - Montreal: 240 CAD, travel time 4 hours
- Vancouver - Edmonton: 150 CAD, travel time 1.5 hours
- Vancouver - Calgary: 180 CAD, travel time 1 hour

These tickets are available for the second leg of the trip:

- Ottawa - Berlin: 1350 CAD, layover: 3.5 hours, travel time 9 hours
- Montreal - London: 1300 CAD, layover: 2 hours, travel time 8 hours
- Edmonton - London: 1200 CAD, layover: 5 hours, travel time 10 hours
- Calgary - London: 1400 CAD, layover: 2.5 hours, travel time 10 hours
- Toronto - Munich: 990 CAD, layover: 1.5 hours, travel time 9.5 hours

Using math operators in Python, find out which flight will give Dot the best value per hour of travel time.

```
value = price_paid / travel_time
```

In [7]:

```
van_tor_price = 250
van_ott_price = 280
van_mon_price = 240
van_edm_price = 150
van_cal_price = 180

van_tor_travel_time = 3.5
van_ott_travel_time = 4
van_mon_travel_time = 4
van_edm_travel_time = 1.5
van_cal_travel_time = 1

ott_ber_price = 1350
mon_lon_price = 1300
edm_lon_price = 1290
cal_lon_price = 1400
```

```
tor_mun_price = 990

ott_layover = 3.5
mon_layover = 2
edm_layover = 5
cal_layover = 2.5
tor_layover = 1.5

ott_ber_travel_time = 9
mon_lon_travel_time = 8
edm_lon_travel_time = 10
cal_lon_travel_time = 10
tor_mun_travel_time = 9.5
```

In [9]:

SOLUTION

```
print((van_tor_price + tor_mun_price) / (van_tor_travel_time + tor_layover + tor_mun_tr
print((van_ott_price + ott_ber_price) / (van_ott_travel_time + ott_layover + ott_ber_tr
print((van_mon_price + mon_lon_price) / (van_mon_travel_time + mon_layover + mon_lon_tr
print((van_edm_price + edm_lon_price) / (van_edm_travel_time + edm_layover + edm_lon_tr
print((van_cal_price + cal_lon_price) / (van_cal_travel_time + cal_layover + cal_lon_tr
```

85.51724137931035

98.78787878787878

110.0

87.27272727272727

117.03703703703704

The flight with the best value is Vancouver -> Toronto -> Munich so we will board flight to Toronto