Types of Variable

In Python, you can handle and manipulate different types of variables. Each has its own specificities and benefits. We will not go through every single one of them but rather focus on the main ones that you will have to use in this course. For each of the following code examples, you can run the code in Google Colab to view the given output.

Numeric Variables

The most basic variable type is numeric. This can contain integer or decimal (or float) numbers, and some mathematical operations can be performed on top of them.

Let's use an integer variable called var1 that will take the value 8 and another one called var2 with the value 160.88, and add them together with the + operator, as shown here:

```
var1 = 8
var2 = 160.88
var1 + var2
```

```
168.88
```

In Python, you can perform other mathematical operations on numerical variables, such as multiplication (with the * operator) and division (with /).

Text Variables

Another interesting type of variable is string, which contains textual information. You can create a variable with some specific text using the single or double quote, as shown in the following example:

```
var3 = 'Hello, '
var4 = 'World'
print(var3)
print(var4)
```

```
Hello,
World
```

Python also provides an interface called f-strings for printing text with the value of defined variables. It is very handy when you want to print results with additional text to make it more readable and interpret results. It is also quite common to use f-strings to print logs. You will need to add f before the quotes (or double quotes) to specify that the text will be an f-string. Then you can add an existing variable inside the quotes and display the text with the value of this variable. You need to wrap the variable with curly brackets, {}. For instance, if we want to print Text: before the values of var3 and var4, we will write the following code:

```
print(f"Text: {var3} {var4}!")
```

```
Text: Hello,  World!
```

You can also perform some text-related transformations with string variables, such as capitalizing or replacing characters. For instance, you can concatenate the two variables together with the + operator:

```
var3 + var4
```

```
'Hello, World'
```

Python List

Another very useful type of variable is the list. It is a collection of items that can be changed (you can add, update, or remove items). To declare a list, you will need to use square brackets, [ ], like this:

```
var5 = ['I', 'love', 'data', 'science']
print(var5)
```

```
['I', 'love', 'data', 'science']
```

A list can have different item types, so you can mix numerical and text variables in it:

```
var6 = ['Packt', 15019, 2020, 'Data Science']
print(var6)
```

```
['Packt', 15019, 2020, 'Data Science']
```

An item in a list can be accessed by its index (its position in the list). To access the first (index 0) and third elements (index 2) of a list, you do the following:
Note : In Python, all indexes start at 0.

```
print(var6[0])
print(var6[2])
```

```
Packt
2020
```

Python provides an API to access a range of items using the : operator. You just need to specify the starting index on the left side of the operator and the ending index on the right side. The ending index is always excluded from the range. So, if you want to get the first three items (index 0 to 2), you should do as follows:

```
print(var6[0:3])
```

```
['Packt', 15019, 2020]
```

You can also iterate through every item of a list using a for loop. If you want to print every item of the var6 list, you should do this:

```
for item in var6:
    print(item)
```

```
Packt
15019
2020
Data Science
```

You can add an item at the end of the list using the .append() method:

```
var6.append('Python')
print(var6)
```

```
['Packt', 15019, 2020, 'Data Science', 'Python']
```

To delete an item from the list, you use the .remove() method:

```
var6.remove(15019)
print(var6)
```

```
['Packt', 2020, 'Data Science', 'Python']
```

Python Dictionary

Another very popular Python variable used by data scientists is the dictionary type. For example, it can be used to load JSON data into Python so that it can then be converted into a DataFrame . A dictionary contains multiple elements, like a list, but each element is organized as a key-value pair. ***A dictionary contains multiple elements, like a list, but each element is organized as a key-value pair. A dictionary is not indexed by numbers but by keys. So, to access a specific value, you will have to call the item by its corresponding key***. To define a dictionary in Python, you will use curly brackets, {}, and specify the keys and values separated by :, as shown here:
Note: Each key-value pair in a dictionary needs to be unique.

```
var7 = {'Topic': 'Data Science', 'Language': 'Python'}
print(var7)
```

```
{'Topic': 'Data Science', 'Language': 'Python'}
```

To access a specific value, you need to provide the corresponding key name. For instance, if you want to get the value Python, you do this:

```
var7['Language']
```

```
'Python'
```

Python provides a method to access all the key names from a dictionary, .keys(), which is used as shown in the following code snippet.There is also a method called .values(), which is used to access all the values of a dictionary.

```
print(var7.keys())
print(var7.values())
```

```
dict_keys(['Topic', 'Language'])
dict_values(['Data Science', 'Python'])
```

You can iterate through all items from a dictionary using a for loop and the .items() method

```
for key, value in var7.items():
    print(key)
    print(value)
```

```
Topic
Data Science
Language
Python
```

You can add a new element in a dictionary by providing the key name like this:

```
var7['Publisher'] = 'Packt'
print(var7)
```

```
{'Topic': 'Data Science', 'Language': 'Python', 'Publisher': 'Packt'}
```

You can delete an item from a dictionary with the del command:

```
del var7['Publisher']
print(var7)
```

```
{'Topic': 'Data Science', 'Language': 'Python'}
```

Exercise 1.01:

Creating a Dictionary That Will Contain Machine Learning Algorithms

1. Create a list called algorithm that will contain the following elements: Linear Regression, Logistic Regression, RandomForest, and a3c

```
algorithm = ['Linear Regression' , 'Logistic Regression' , 'Random Forest' , \
             'a3c']
algorithm
```

```
['Linear Regression', 'Logistic Regression', 'Random Forest', 'a3c']
```

2. create a list called learning that will contain the following elements: Supervised, Supervised, Supervised, and Reinforcement

```
learning = ['Supervised' , 'Supervised' , 'Supervised' , \
            'Reinforcement']
learning
```

```
['Supervised', 'Supervised', 'Supervised', 'Reinforcement']
```

3. Create a list called algorithm_type that will contain the following elements: Regression, Classification, Regression or Classification, and Game AI

```
algorithm_type = ['Regression' , 'Classification' , \
                  'Regression or Classification' , 'Game AI']

```

```
4 algorithm_type
```

```
['Regression', 'Classification', 'Regression or Classification', 'Game AI']
```

4. Add an item called k-means into the algorithm list using the .append() method

```
1 algorithm.append('k-means')
2
3 print(algorithm)
```

```
['Linear Regression', 'Logistic Regression', 'Random Forest', 'a3c', 'k-means', 'k-means']
```

5. add the Unsupervised item into the learning list using the .append() method

```
1 learning.append('Unsupervised')
2
3 print(learning)
```

```
['Supervised', 'Supervised', 'Supervised', 'Reinforcement', 'Unsupervised']
```

6. Add the Clustering item into the algorithm_type list using the .append() method

```
1 algorithm_type.append('Clustering')
2
3 print(algorithm_type)
```

```
['Regression', 'Classification', 'Regression or Classification', 'Game AI', 'Clustering']
```

7. Create an empty dictionary called machine_learning using curly brackets, {}

```
1 machine_learning = {}
```

8. Create a new item in machine_learning with the key as algorithm and the value as all the items from the algorithm list

```
1 machine_learning['algorithm'] = algorithm
2
3 print(machine_learning)
```

```
{'algorithm': ['Linear Regression', 'Logistic Regression', 'Random Forest', 'a3c', 'k-means', 'k-means']}
```

9. Create a new item in machine_learning with the key as learning and the value as all the items from the learning list

```
1 machine_learning['learning'] = learning
2
3 print(machine_learning)
```

```
{'algorithm': ['Linear Regression', 'Logistic Regression', 'Random Forest', 'a3c', 'k-means', 'k-means'], 'learning': ['Supervise
```

10. create a new item in machine_learning with the key as algorithm_type and the value as all the items from the algorithm_type list

```
1 machine_learning['algorithm_type'] = algorithm_type
2
3 print(machine_learning)
```

```
{'algorithm': ['Linear Regression', 'Logistic Regression', 'Random Forest', 'a3c', 'k-means', 'k-means'], 'learning': ['Supervise
```