Data Collections : Set

Set is the third collection that we are learning about (After going over list & dictionary).

A simple set looks like this :

```
fruit_set_collection = {“banana”, “apple”, “berry”}
```

What is the main difference between a ‘List’ and a ‘Set’ ?

- A list can contain the same value in several cells, while a set collection cannot have the same value in more than 1 cell.

- List is an ordered collection. Meaning, it has an index and cell values can be accessed by the index. Set in the other hand is an unordered collections (same as dictionary), and does not have an index value for each cell

- Set is written with curly brackets, in a list we’ll use round brackets

- In ‘set’, once an item is created you cannot change it. But, you can add new items. While in a ‘list’ you can change any items that you want.

How to access items in a ‘set’ ?

We cannot access a ‘set’ cell by index. But what we can do is, to check if it exists by using a boolean statement.

For example

If we want to know whether a ‘banana’ item is inside the ‘set’, we should use

```
print("banana" in fruit_set_collection )
output -> ‘True’
```

How to delete items of a ‘set’ ?

We can remove items from a cell by using the ‘.remove()’ action or the ‘.discard()’ actions.

For example:

```
fruit_set_collection.remove(“berry”)
```

How to add items to a ‘set’ ?

We can remove items from a cell by using the ‘.add()’ action or the ‘.update()’ actions.

For example:

```
fruit_set_collection.add(“cherry”)
```

Duplicate items, what will happen if we have it ?

If we will have duplicate items inside a ‘set’, once printing it, python will ignore the dublications and will print only 1 of the duplicated items.

For example:

```
fruit_set_collection = {“banana”, “apple”, “berry”, “banana”}
print(fruit_set_collection )

output -> {'banana', 'apple', 'berry'}
```