by. Null Science

Learning Python #2 "Data Type in Python"

Null Science Team

Leader : Fahmi Hamzah

Member :

- **Aditya Hermawan**
- **Fara Rizkhi Karunia**
- **Muh. Aksa Hadim**
- **Nafiatul Risa**
- **Salsabilla Atasyaputri Setyawan**

Data Types in Python

As we all known before, there are many data types in python. In the previous presentation explains about numeric, string, and boolean data types which are the basic on python.

This presentation explains about :

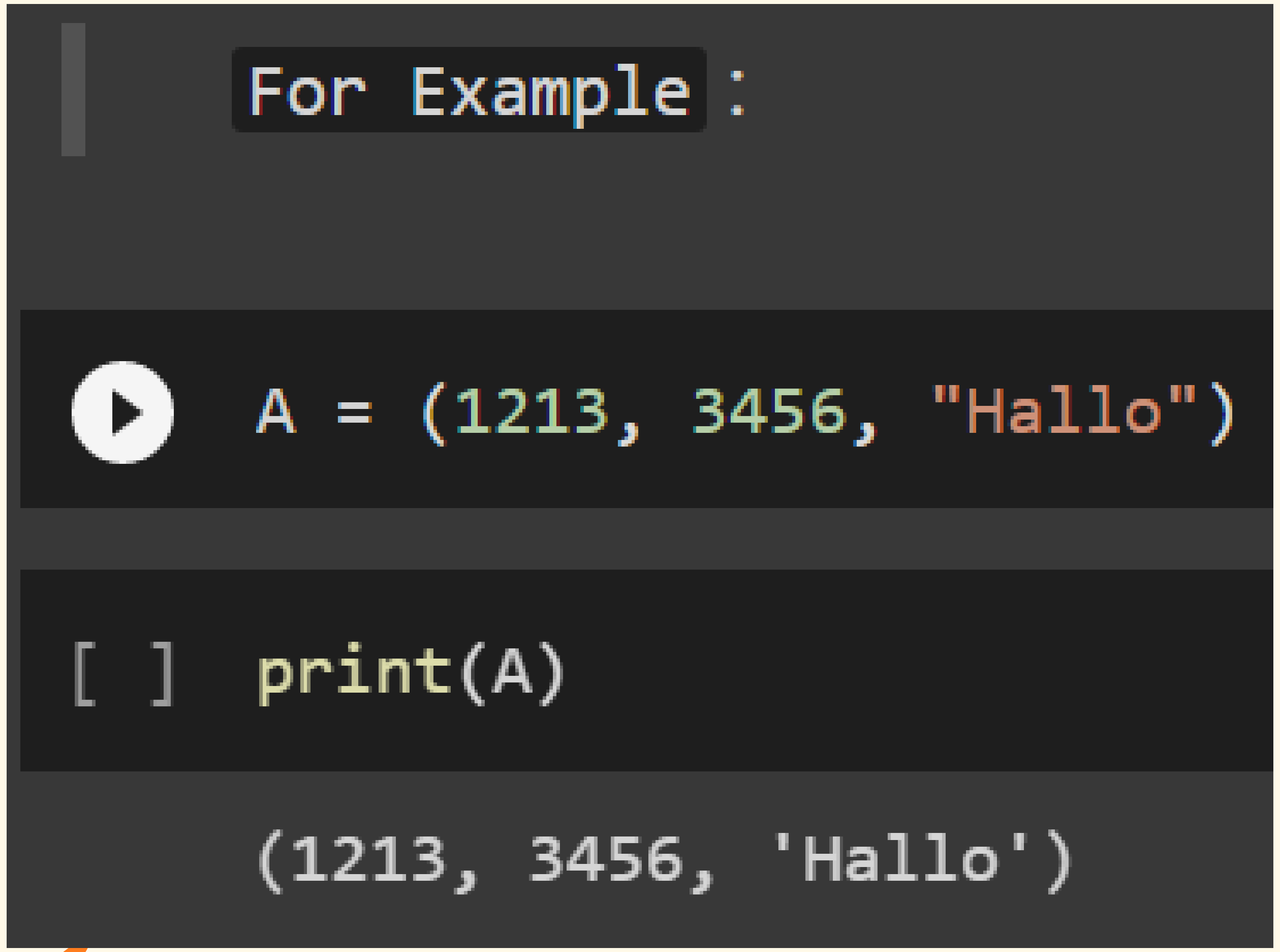
- Tuples
- List
- Set
- Dictionary

TUPLE

A tuple in Python is a data structure used to store a set of data. Tuples are **immutable**, meaning that the contents of the tuple cannot be changed or deleted. However, we can fill it with various values and objects.

HOW TO CREATE TUPLE?

Tuples are defined by **parentheses** () and elements separated by **commas.**

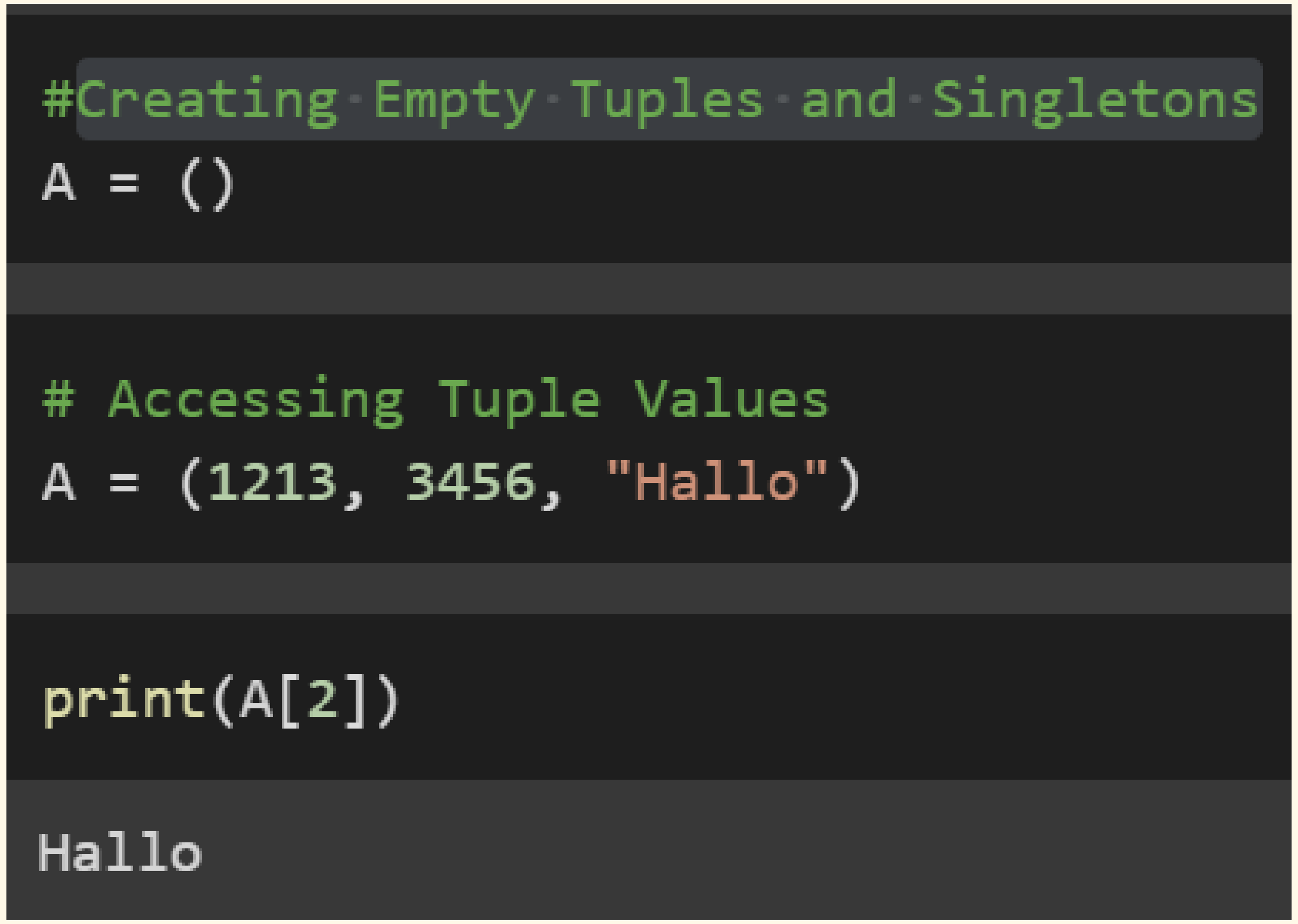


For Example :

```
A = (1213, 3456, "Hallo")
```

```
print(A)
```

```
(1213, 3456, 'Hallo')
```



```
#Creating Empty Tuples and Singletons
A = ()
```

```
# Accessing Tuple Values
A = (1213, 3456, "Hallo")
```

```
print(A[2])
```

```
Hallo
```

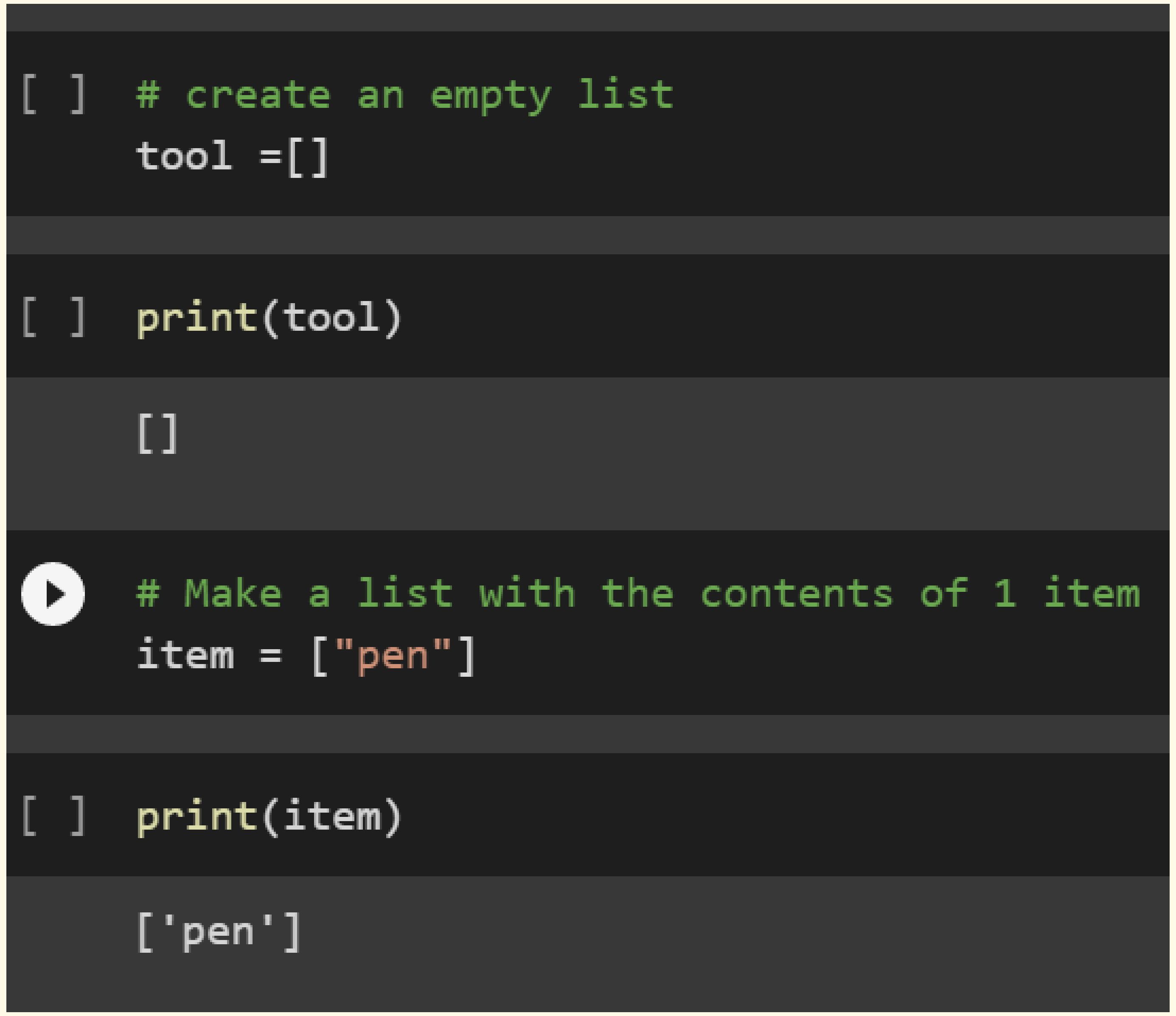
LIST

List is a data structure **(ordered sequence)** in python that can store more than one data, similar but different with array. The difference is, list elements in python can use **different data types.**

Every data in list can be **accessed** with an **index** starting from 0

HOW TO CREATE LIST?

List are defined by **square bracket [] and elements separated by commas.**



```
# create an empty list
tool =[]
```

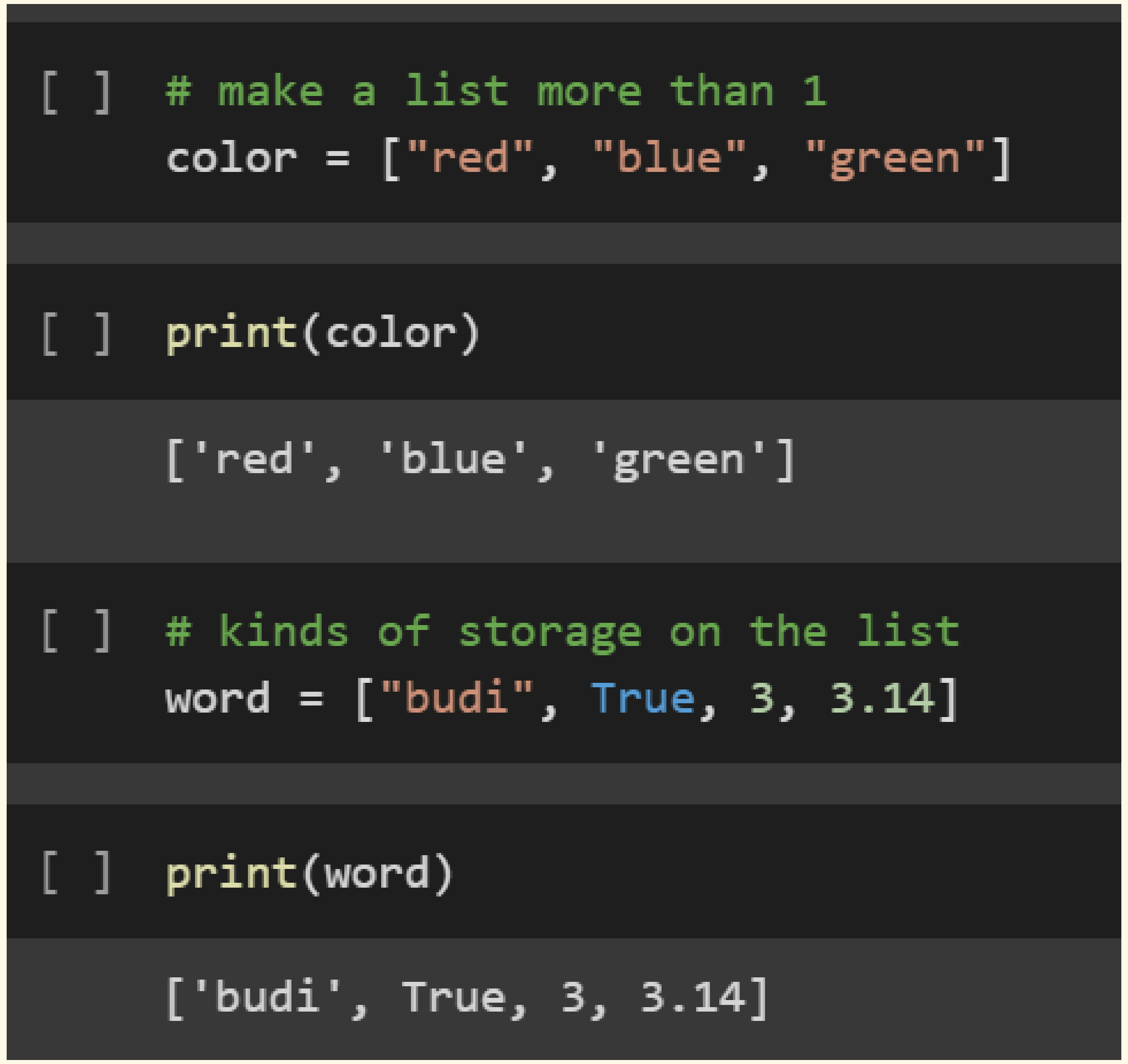
```
print(tool)
```

```
[]
```

```
# Make a list with the contents of 1 item
item = ["pen"]
```

```
print(item)
```

```
['pen']
```



```
# make a list more than 1
color = ["red", "blue", "green"]
```

```
print(color)
```

```
['red', 'blue', 'green']
```

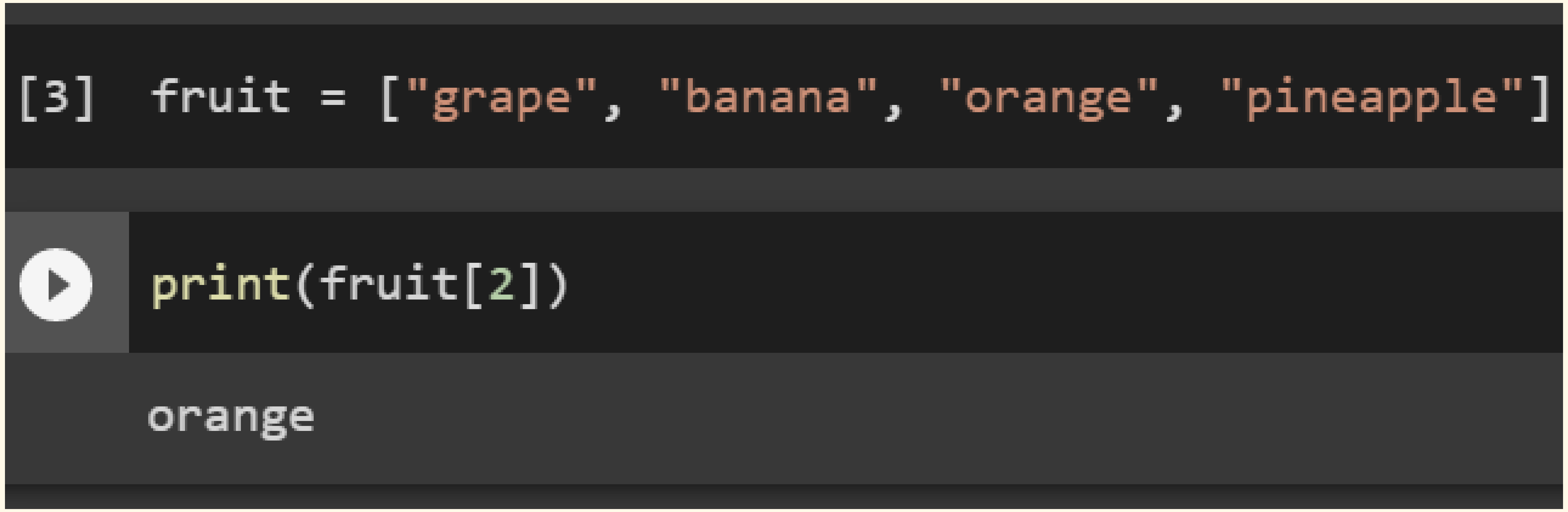
```
# kinds of storage on the list
word = ["budi", True, 3, 3.14]
```

```
print(word)
```

```
['budi', True, 3, 3.14]
```

HOW TO RETRIEVE VALUE FROM LIST?

How to retrieve a value in a list can be done with calling **list_name[Index]**



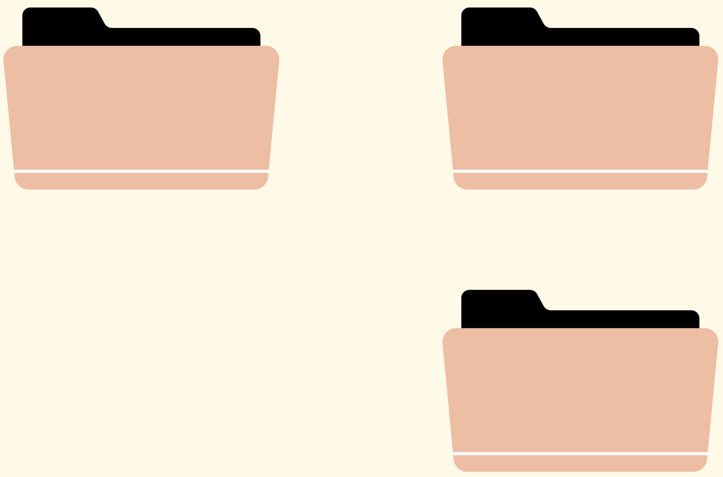
```
[3]  fruit = ["grape", "banana", "orange", "pineapple"]
```

```
print(fruit[2])
```

```
orange
```

SET

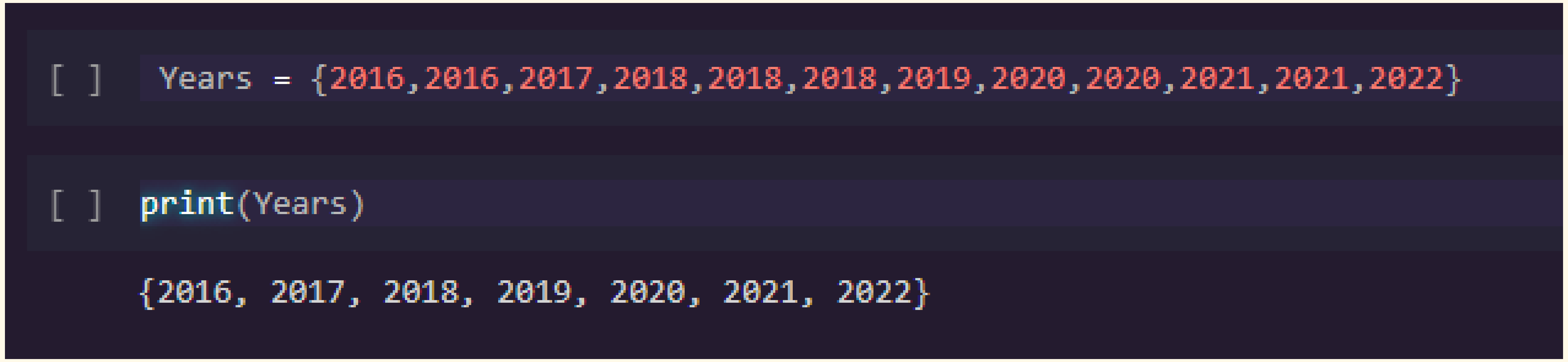
Data Set is a collection of items that are **unique and unordered (underordered collection).** It has the characteristic of curly braces and each element is separated using **curly brackets** with **commas.**



DIFFERENT TYPES OF SETS IN PYTHON

- Set of integers Numeral = {1, 2, 3}
- Set of mixed data types Mixed = {176.5, "Hello", (100, 200, 300)}

← EXAMPLE



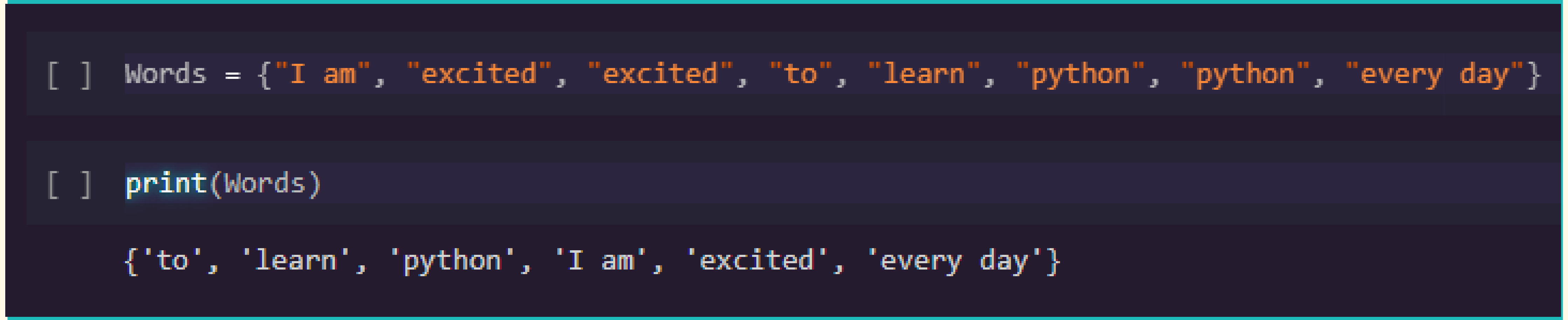
```
[ ]  Years = {2016,2016,2017,2018,2018,2018,2019,2020,2020,2021,2021,2022}

[ ]  print(Years)

     {2016, 2017, 2018, 2019, 2020, 2021, 2022}
```

TAKEN ONLY UNIQUE AND DUPLICATES DO NOT APPEAR FROM THE YEARS DATA

EXAMPLE



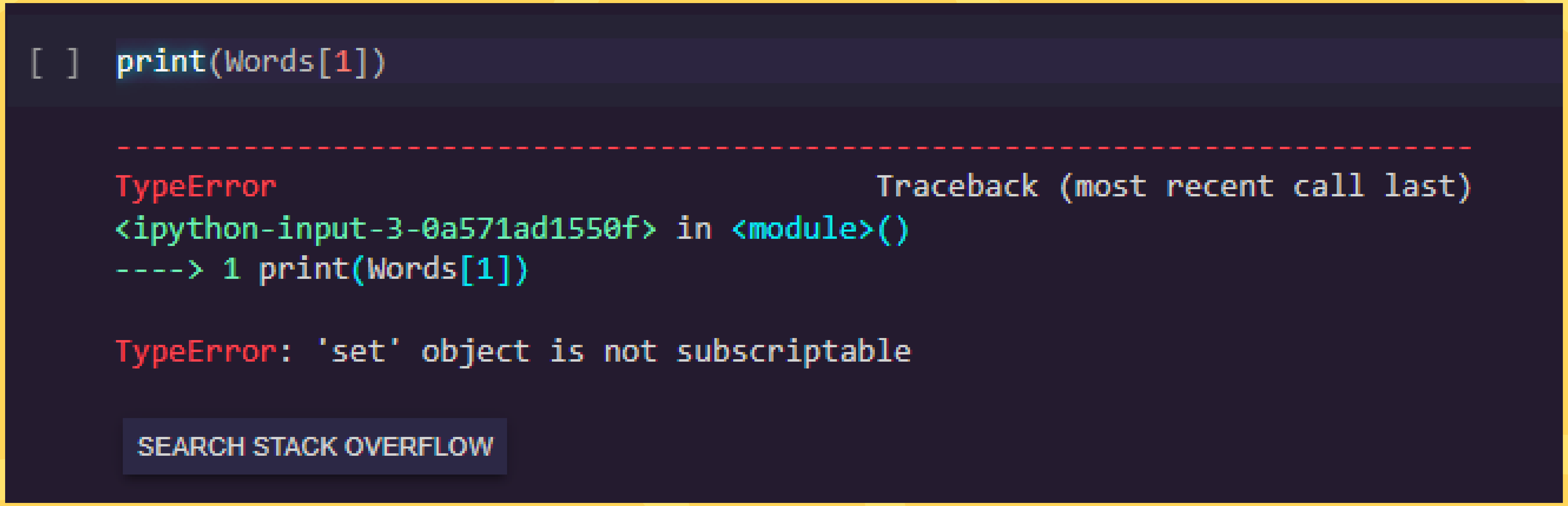
```
[ ] Words = {"I am", "excited", "excited", "to", "learn", "python", "python", "every day"}

[ ] print(Words)

    {'to', 'learn', 'python', 'I am', 'excited', 'every day'}
```

Taken only unique and duplicates do not appear from the Words data.

Don't Forget That !



```
[ ]  print(Words[1])

---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
<ipython-input-3-0a571ad1550f> in <module>()
----> 1 print(Words[1])

TypeError: 'set' object is not subscriptable
```

The result is an error because the data set type does not have an index.

✓

Let's see →

What if we keep accessing set members using the index ?

DICTIONARY

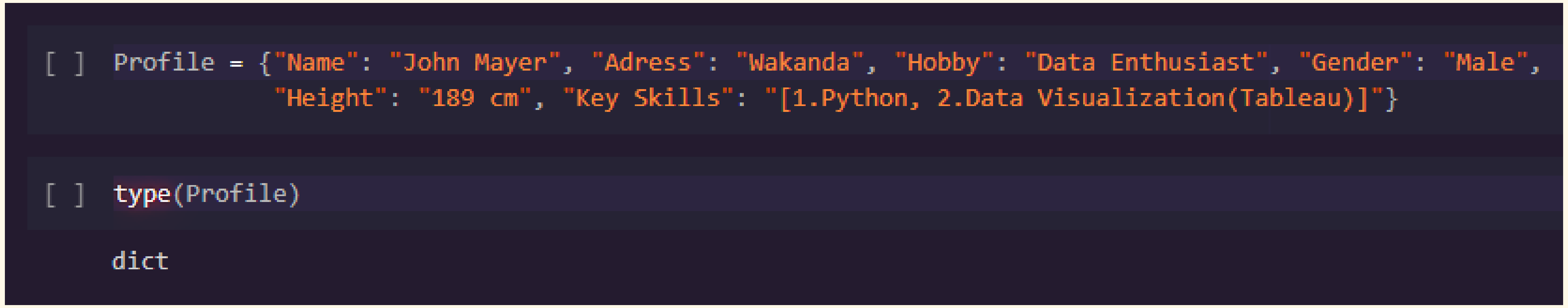
Dictionary is an **unordered collection** of items. Each item of a dictionary has a **key/value pair**. Dictionaries are optimized to retrieve values when the key is known.

In python, the dictionary is defined by curly and the following is an additional definition.

- Each key-value pair element is separated by a comma(,)
- Key and value are separated by a two-point (:)
- Key and value can be any type of variable/object

Different types of dictionaries in Python :

- Empty Dictionary dict = {}
- Dictionary with integer keys dict = {1: 'apple', 2: 'ball'}
- Dictionary with mixed keys dict = {'name': 'John', 1: [2, 4, 3]}

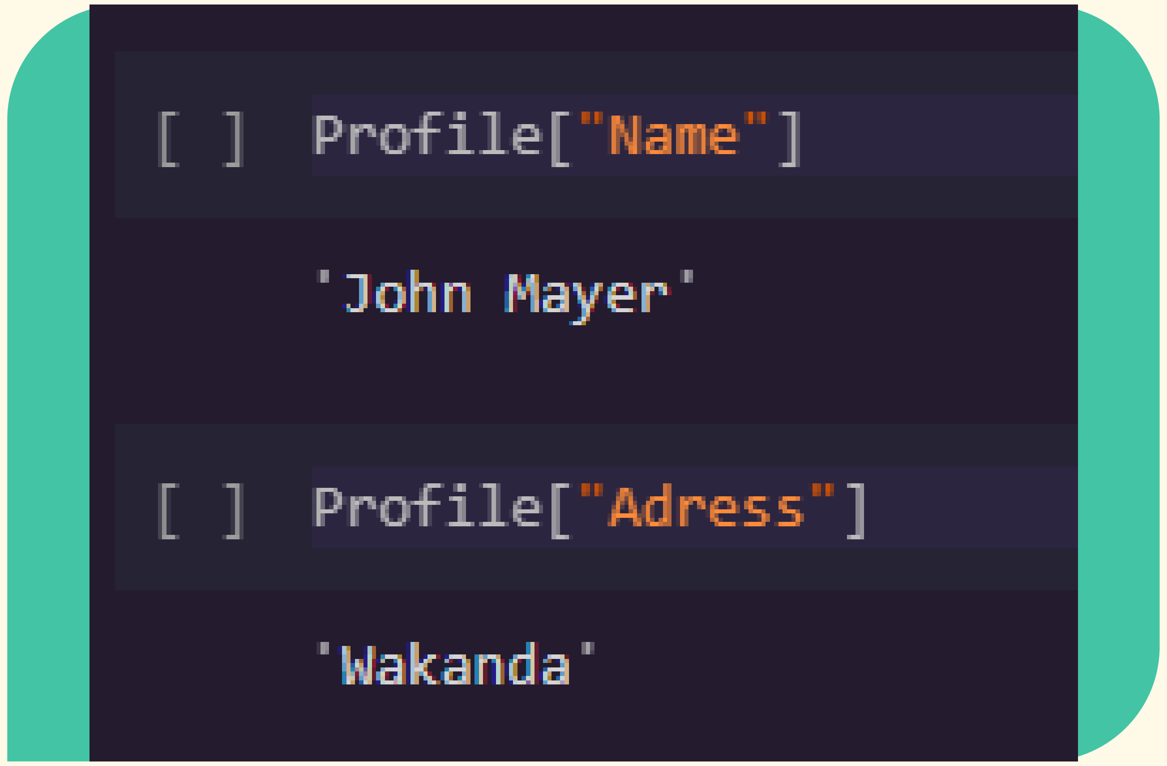


```
[ ]  Profile = {"Name": "John Mayer", "Adress": "Wakanda", "Hobby": "Data Enthusiast", "Gender": "Male",
                "Height": "189 cm", "Key Skills": "[1.Python, 2.Data Visualization(Tableau)]"}

[ ]  type(Profile)

     dict
```

← EXAMPLE

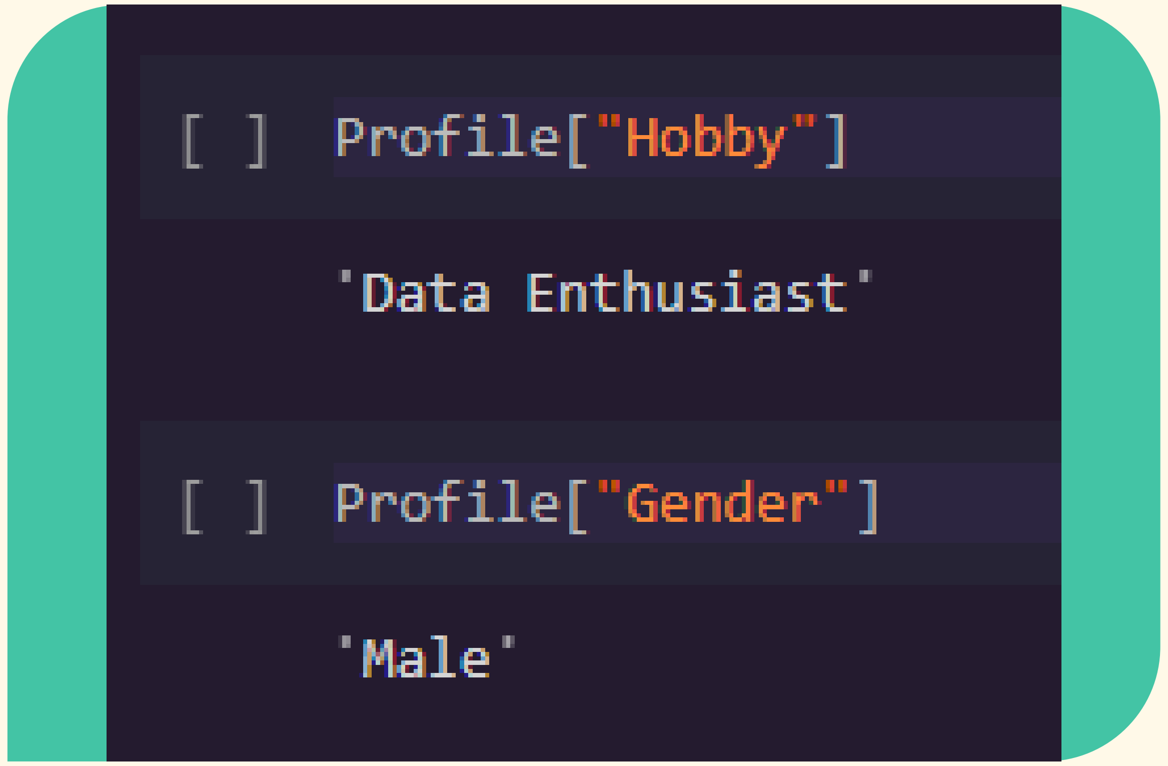


```
[ ]  Profile["Name"]

     'John Mayer'

[ ]  Profile["Adress"]

     'Wakanda'
```

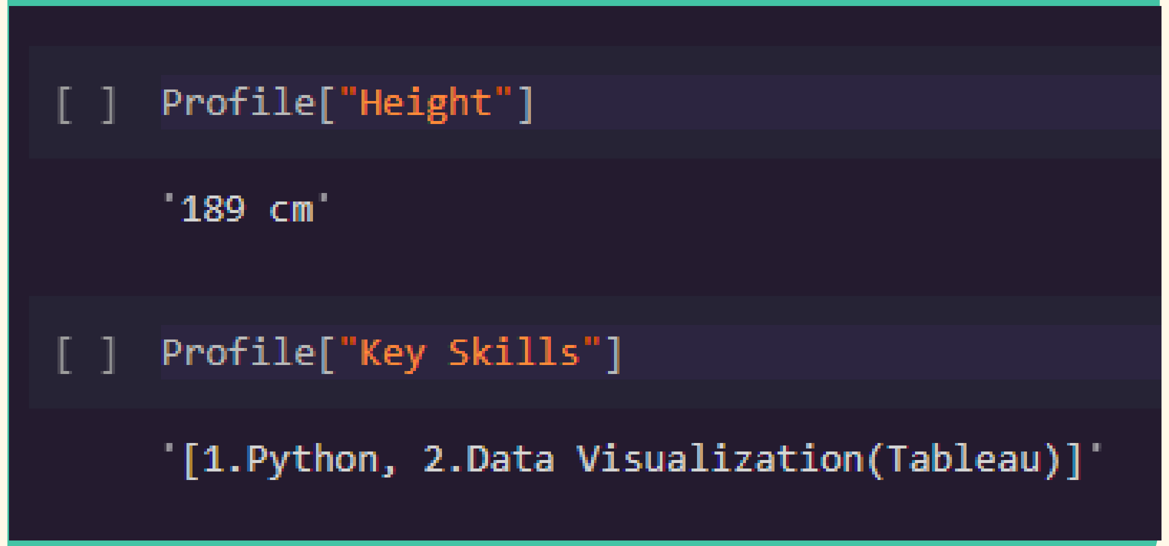


```
[ ]  Profile["Hobby"]

     'Data Enthusiast'

[ ]  Profile["Gender"]

     'Male'
```



```
[ ]  Profile["Height"]

     '189 cm'

[ ]  Profile["Key Skills"]

     '[1.Python, 2.Data Visualization(Tableau)]'
```

Let's see →

Another Example

THANK YOU

&

Wait For The Next Portofolio!