



# ADVANCED PYTHON DATA STRUCTURE

List, Tuple, Set, Dictionary

Matplotlib Group - Data Science Track B

# Python Data Structure

## MATPLOTLIB GROUP

1. Tobias Mikha Sulistiyo
2. Daud Ibadurahman
3. Putri Reghina Hilmi Prasati
4. Sari Yuliastuti
5. Fitri Alfaqrina

## OUTLINES

1. List
2. Tuple
3. Set
4. Dictionary

## TYPE 1

### List

Data types List is an ordered sequence of items. It is one of the most used datatype in Python. List is very flexible, all the items in a list do not need to be of the same type. Declaring a list is pretty straightforward. Items separated by commas are enclosed within brackets [ ].

# DATA TYPES LIST

## Creating a List

```
[1] a = ["Saturday", 19, 9.25, 2022]
    print (a)

['Saturday', 19, 9.25, 2022]

[2] type (a)

list
```

We can use the slicing operator [ ] to access an element of the list by index. The index starts from 0 in python.

```
[3] print (a[0])
    print (a[1])
    print (a[2])
    print (a[3])

Saturday
19
9.25
2022
```

# DATA TYPES LIST

## List Slicing

We can access a range of items in a List by using the slicing operator (:). When using List slicing, a [ m:n ], means returning list elements from element with index m to element with index n-1.

Slicing in the String are similar to list, the slicing operator can also be used on strings to return the string elements. String can be changed only if they are intact (cannot be changed element by element). Slicing rules on string are the same as for list.

```
[5] print (a[0:2])  
    #return list 'a' with elements at index 0 and 1  
    ['Saturday', 19]  
  
[6] print (a[1:4])  
    #return list 'a' with elements at index 1, 2, and 3  
    [19, 9.25, 2022]  
  
[7] print (a[:3])  
    #return list 'a' from beginning to index 2  
    ['Saturday', 19, 9.25]  
  
[8] print (a[1:])  
    #return list 'a' from index 1 to end  
    [19, 9.25, 2022]
```

# DATA TYPES LIST

## Change List Elements

List are mutable, their elements can be changed. We can use the assignment operator = to change an item or a range of items.

```
✓ [5] print (a[0:2])  
    #return list 'a' with elements at index 0 and 1  
    ['Saturday', 19]
```

```
✓ [6] print (a[1:4])  
    #return list 'a' with elements at index 1, 2, and 3  
    [19, 9.25, 2022]
```

```
✓ [7] print (a[:3])  
    #return list 'a' from beginning to index 2  
    ['Saturday', 19, 9.25]
```

```
✓ [8] print (a[1:])  
    #return list 'a' from index 1 to end  
    [19, 9.25, 2022]
```

Elements in a list can be added by using append(), extend(), or insert() methods.

```
✓ [14] b = [1,2,3,"n","o"]  
       print (b)  
       [1, 2, 3, 'n', 'o']  
✓ [15] b.append("w")  
       #adding an element to the end of the list 'b'  
✓ [16] print (b)  
       [1, 2, 3, 'n', 'o', 'w']
```

```
✓ [18] b.extend(["d","o"," ", "i","t"])  
✓ [19] print (b)  
       [1, 2, 3, 'n', 'o', 'w', 'd', 'o', ' ', 'i', 't']
```

```
✓ [20] b.insert(3,4)  
       #adding item 4 at index 3  
✓ [21] print (b)  
       [1, 2, 3, 4, 'n', 'o', 'w', 'd', 'o', ' ', 'i', 't']
```

# DATA TYPES LIST

## Delete List Element

We can delete one or more items from a list using the Python `del` statement. It can even delete the list entirely. We also can use `remove()` to remove the given item or `pop()` to remove an item at the given index.

```
[22] c = [19, 3, 2022, 10.43, "a", "m"]
      print (c)
[19, 3, 2022, 10.43, 'a', 'm']

[23] del c[3]
      #deleting item of the list 'c' at index 3

[24] print (c)
[19, 3, 2022, 'a', 'm']
```

```
[26] c.remove('a')
      #remove item "a" in the list c

[27] print (c)
[19, 3, 2022, 'm']
```

```
[29] c.pop(3)
      #remove list 'c' element at index 3
'm'

print(c)
[19, 3, 2022]
```

# DATA TYPES LIST

## Len and Count in List

`len()` used to find out the length (number of items or elements) in a List.. `count(x)` returns the count of the number of elements `x` in the List.

```
✓ [31] d = ["s",1,"a",2,"r",3,"a",4,4,"r","i"]  
    print (d)  
  
      ['s', 1, 'a', 2, 'r', 3, 'a', 4, 4, 'r', 'i']  
  
✓ [32] len(d)  
      #find out the length of the list 'd'  
  
      11
```

```
✓ [34] d.count("r")  
      2  
  
✓ [35] d.count(4)  
      2
```



# DATA TYPES LIST

## Sort and Reverse in the List

Sort items in a list in ascending order and Reverse the order of items in the list.

```
✓ [36] e = [56,77,84,33,24,58,1,2022]
    f = ["s","a","t","u","r","d","a","y"]
    print (e)
    print (f)

[56, 77, 84, 33, 24, 58, 1, 2022]
['s', 'a', 't', 'u', 'r', 'd', 'a', 'y']

✓ [37] e.sort()
    f.sort()

✓ [38] print (e)
    print (f)

[1, 24, 33, 56, 58, 77, 84, 2022]
['a', 'a', 'd', 'r', 's', 't', 'u', 'y']
```

```
✓ [41] e.reverse()
    f.reverse()

✓ [42] print (e)
    print (f)

[2022, 84, 77, 58, 56, 33, 24, 1]
['y', 'u', 't', 's', 'r', 'd', 'a', 'a']
```

# DATA TYPES LIST

## Index in the List

Index method in the List used to returns the index of the first matched item

```
[43] g = [1,2,4,5,1,7,9,10,4,7]
      print (g)

      [1, 2, 4, 5, 1, 7, 9, 10, 4, 7]

      g.index(4)

      2
```

## TYPE 2

### Tuple

Data types Tuple is a data structure used to store a set of data. The tuple is immutable, meaning that the contents of the tuple cannot be changed or deleted. However, we can fill it with various values and objects. A tuple is created by placing all the items (elements) inside parentheses ( ), separated by commas. A tuple can have any number of items and they may be of different types (integer, float, list, string, etc.).

# DATA TYPES TUPLE

## Creating a Tuple

You can use parentheses '()' to create a Tuple

```
✓ 0d ▶ pathway = ("DST_B", 50, 19.0, "PM")  
    print (pathway)  
  
    ('DST_B', 50, 19.0, 'PM')  
  
✓ 0d [2] type (pathway)  
      tuple
```

# DATA TYPES TUPLE

## Difference Between List and Tuple

The difference between the Tuple and List is we cannot change the elements of a tuple once it is assigned whereas we can change the elements of a list.

```
✓ [13] c = [21, "Fitri", 2.5, 'i']  
Od      print (c)
```

```
[21, 'Fitri', 2.5, 'i']
```

```
✓ [14] type(c)
```

```
list
```

```
✓ [23] b = (21, 'Fitri', 2, 5, 'i')  
Od      print (b)
```

```
(21, 'Fitri', 2, 5, 'i')
```

```
✓ [24] type(b)
```

```
tuple
```

## TYPE 3

### Set

Data types Set is a collective data type used to store multiple values in a single variable with the condition: stored member values must be unique (not duplicates), member values that have been entered cannot be changed anymore and are set to be irregular or not sequential – which means they cannot be accessed by index.

# Data Types Set

TYPE 3

## Creating a Set

Sets are used to store multiple items in a single variable.

```
✓ [1] group={1, 'Me', 0.37}
✓ [2] type(group)
set
```

When using data set if there are duplicate items, it will only show 1 item.

```
✓ [3] example1={8,9,8,8,7,9,9,7,7, 'a', 'a', 'd', 3,3,5,2,2,1,1, 'c', 'c', 'b', 'b'}
✓ [4] len(example1)
11
```

# Data Types Set

TYPE 3

## Creating a Set

Set data type are unordered, it means that the items in a set do not have a defined order

```
✓ 0 d [6] print(example1)
{1, 2, 3, 5, 7, 8, 9, 'd', 'a', 'c', 'b'}
```

When we use Set, we can't replace or change value in set. But We can add and remove value in set

```
✓ 0 d [6] example1.add(321)
✓ 0 d [7] print(example1)
{1, 2, 3, 321, 5, 7, 8, 9, 'd', 'a', 'c', 'b'}
✓ 0 d [8] example1.remove('b')
✓ 0 d [9] print(example1)
{1, 2, 3, 321, 5, 7, 8, 9, 'd', 'a', 'c'}
```



## TYPE 4

### Dictionary

Data types Dictionary can be defined as a collection of data stored in key/value pairs. Keys must be an immutable data type (such as string, integer or tuple), while values in a dictionary can be any Python data type.

# Data Types Dictionary

TYPE 4

## Creating a Dictionary

```
[1] matplotlibgroup = {"Member1": "Tobias", "Member2": "Daud", "Member3": "Rere", "Member4": "Sari", "Member5": "Fitri"}  
[2] type(matplotlibgroup)  
dict  
[6] print(matplotlibgroup)  
{'Member1': 'Tobias', 'Member2': 'Daud', 'Member3': 'Rere', 'Member4': 'Sari', 'Member5': 'Fitri'}
```

When we use dictionary, to call the data must to call the key so the value will appear

```
[8] print(matplotlibgroup['Member5'])  
Fitri
```

## Data Types Dictionary

TYPE 4

### Creating a Dictionary

We can Add Dictionary value by make new key and value.

```
[9] matplotlibgroup['Mentor']='Suwarti'

[10] print(matplotlibgroup)

{'Member1': 'Tobias', 'Member2': 'Daud', 'Member3': 'Rere', 'Member4': 'Sari', 'Member5': 'Fitri', 'Mentor': 'Suwarti'}
```

To make Dictionary Keys must be an immutable data type (such as string, integer or tuple), while values in a dictionary can be any Python data type

# Thanks!

That's our powerpoint for Advanced Python Data Structure. For the github, you can access in the link on our LinkedIn comment section!

See you next time!

**Matplotlib Group - Data Science Track B**