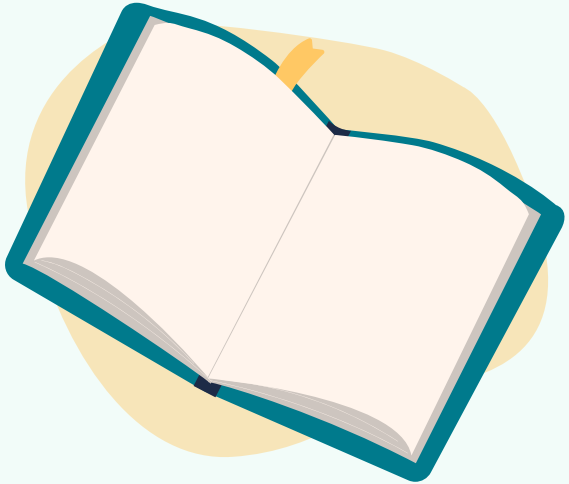
Kampus Merdeka
INDONESIA JAYA
x
MyEduSolve

Numpy Group Summary #2

Python Data Type : List,Tuple,Dictionary,Set



The Teams

Atika Rahma

Muhammad Atthariq Ramadhan

Muhammad Iqbal Rustan

Daril Hana Salsabila

Donny P.Tambunan

Table of contents

01 List

02 Tuple

03 Set

04 Dictionary

01

List

Data strings that store various data types
and their contents can be changed

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

```
[ ]  #example
     list = [1,"two",3.0]
```

```
[ ]  # extract element
     list[0]
```

```
'1'
```

```
[ ]  type(list[0])
```

```
str
```

```
[ ]  list[-2]
```

```
'two'
```

```
[ ]  type(list[2])
```

```
str
```

```
[ ]  # extract range
     list[0:3]
```

```
['one', 'two', 'three']
```

```
list[ :3]
```

```
['1', 'two', '3.0']
```

- display the specified contents of a list by using an index
- use negative index to display data from behind

```
[ ] list[0:3:2]
```

```
['1', '3.0']
```

```
[ ] #len -> Return the number of items in a list
    len(list)
```

```
4
```

```
[ ] #del -> delete an element in list
    del list[0]
    print(list)
```

```
['two', 3.0]
```

```
[ ] #append -> To add an item to the end of the list
    list.append("four")
    print(list)
```

```
['two', 3.0, 'four']
```

```
[ ] #extend ->  merges another list L to the end
    list.extend([5,"six",7.0])
    print(list)
```

```
['two', 3.0, 'four', 5, 'six', 7.0]
```

- Delete an item from the list using the del statement. With this statement, we can remove any index from the list item.
- append() menambahkan elemen baru pada list
- The len () function is used to find the length (number of items or members) of an object such as a sequence
- extend is used to add multiple elements to an existing list

- remove () removes the item from the list according to the defined value

- insert () Adds a new item to the list at a specified position

- pop () Removes the last item in the list, or it can also delete the item at a defined position

- index () Returns the first index of the defined item

- count () Returns the number of items in the specified list

```
[ ] list2 = ["eight",9.0,10]
    list.extend(list2)
    print(list)
```

```
['two', 3.0, 'four', 5, 'six', 7.0, 'eight', 9.0, 10]
```

```
[ ] #insert -> inserts the specified value at the specified position.
    list.insert(3,3.5)
    print(list)
```

```
['two', 3.0, 'four', 3.5, 5, 'six', 7.0, 'eight', 9.0, 10]
```

```
[ ] #remove -> removes the specified element in list
    list.remove(10)
    print(list)
```

```
['two', 3.0, 'four', 3.5, 5, 'six', 7.0, 'eight', 9.0]
```

```
[ ] #pop -> removes the specified index in list
    list.pop(0)
    print(list)
```

```
[3.0, 'four', 3.5, 5, 'six', 7.0, 'eight', 9.0]
```

```
[ ] #index -> return the first index of the value in the list
    list.index(3.0)
```

```
0
```

```
[ ] #count -> returns the number of elements with the specified value in list
    number = [2,2,2,2,3,3,3,1,1,1,1,1,4,4,5]
    number.count(4)
```

```
2
```

```
[ ] #sort -> sorts the list ascending by default in list
    number.sort()
    print(number)
```

```
    [1, 1, 1, 1, 1, 2, 2, 2, 2, 3, 3, 3, 4, 4, 5]
```

```
[ ] #reverse -> reverses the sorting order of the elements in list
    number.reverse()
    print(number)
```

```
    [5, 4, 4, 3, 3, 3, 2, 2, 2, 2, 1, 1, 1, 1, 1]
```

```
[15] list = [1,"two",3.0]
```

```
#slice -> returns a slice object in list
print(list[0])
print(list[1])
print(list[2])
```

```
1
two
3.0
```

- sort () Sorts a list
- reverse () Reverses the position of each item in the list
- Slicing list is a technique for values in a list

02

Tuple

Data strings that store various data types and their contents can be changed

- Tuples are the same as lists. It is equally used to store set data. Both can accommodate various types of data in one set. It's just that once assigned a value, the tuple cannot be changed anymore.

```
[14] Data = (2022,"Universitas Indonesia","march")

     #Create a Tuple
```

```
print(Data[1])

#Accessing Tuple values
```

```
Universitas Indonesia
```

03

Set

A data type used to store multiple values in a single variable and which are irregular and have unique values (no duplication)

- Set puts its members in curly braces {}.
- Sets do not have an index sequence number unlike lists and tuples.
- Sets have unique data / no duplication.
- Set members cannot be changed, but we can add, update and delete sets.
- Sets are separated by commas.
- Sets can be created with the set () function.

Practice

```
[6] Month = {"March","June","May","August","September","October","December","February","January","March","April","May","January","July","June","July","August","September"}
    #Create a Set

[7] print(Month)
    #Accesing Set Values
    {'July', 'June', 'October', 'September', 'May', 'January', 'April', 'March', 'February', 'December', 'August'}

[9] type(Month)
    #Check Data Type
    set
```

04

Dictionary

Data strings that store various data types in the form of a pointer and value pairs

```
[17] Data_Mahasiswa = {"NIM": "46136434","Nama Mahasiswa": "Donny Tambunan","Angkatan": 2019, "IPK":3.51}

     #Create a Dictionary

[18] print(Data_Mahasiswa)

     #Accessing Dictionary Values

     {'NIM': '46136434', 'Nama Mahasiswa': 'Donny Tambunan', 'Angkatan': 2019, 'IPK': 3.51}
```

```
[19] Data_Mahasiswa["NIM"]

     '46136434'

[20] type(Data_Mahasiswa)

     dict

     Data_Mahasiswa["Nama Mahasiswa"]

     'Donny Tambunan'

[22] Data_Mahasiswa["Angkatan"]

     2019

[23] Data_Mahasiswa["IPK"]

     3.51
```

Practice

- Python dictionaries are different from lists or tuples. Because each sequence contains a key and a value. Each key is separated from its value by a colon (:), Items are separated by commas, and are all enclosed in curly braces. An empty dictionary without items is written with just two curly braces, like this: {}.
- The dictionary value can be of any type, but the key must be an immutable data type such as a string, number, or tuple.

ThankYou

Kampus Merdeka
INDONESIA JAYA
x
MyEduSolve