

Get Know about Data Type in Python

Pandas Group – Data Science Track B
My Edu Solve

Ajie Prasetya | Dewa Ayu Rai I.M. | M. Ramadhoni | Firdausi Nurus Sa'idah | Vivi Indah Fitriani | I K. Juni Saputra

Collection Data Type in Python

Python has 4 built-in data structures that can be used to hold a collection of objects.

The built-in data structures are : list, tuple, set and dictionary.

They can be distinguished into mutable, immutable, set type, and mappings respectively.

Learning Objective



To understand what list, tuple, set, and dictionary are in Python.

To find out the difference between each collective data type in python

To learn how to use list, tuple, set, and dictionary in Python programs.

To learn how to use indexing and slicing to access data in Python programs.

1. List

A list is a data structure in python that is a mutable, or changable, ordered sequence of elements. Each element or value that is inside of a list called an item. Just as strings are defined as characters between quotes, lists are defined by having values between square brackets.

```
[ ] University = ["itb","uns","ugm","undiksha","binus","unram"]
```

```
[ ] print(University)
```

```
['itb', 'uns', 'ugm', 'undiksha', 'binus', 'unram']
```

```
[ ] print(University[2])
```

```
ugm
```

list.append()

Append is the method to adds an element at the end of the list.

```
[ ] University.append("unair")  
  
[ ] print(University)  
['itb', 'uns', 'ugm', 'undiksha', 'binus', 'unram', 'unair']
```

list.extend()

This method appends each element of the iterable (tuple, string, or list) to the end of the list and increases the length of the list by the number of elements of the iterable passed as an argument.

```
[ ] University.extend(["telkom","unpad"])  
  
[ ] print(University)  
['itb', 'uns', 'ugm', 'undiksha', 'binus', 'unram', 'unair', 'telkom', 'unpad']
```

list.insert()

The element passed as the argument can be inserted at any desired position by passing the index along with it as a parameter.

```
[ ] University.insert(3,"unud")
```

```
[ ] print(University)
```

```
['itb', 'uns', 'ugm', 'unud', 'undiksha', 'binus', 'unram', 'unair', 'telkom', 'unpad']
```

list.remove()

It removes the first matching element (which is passed as an argument) from the list.

```
[ ] University.remove("ugm")
```

```
[ ] print(University)
```

```
['itb', 'uns', 'unud', 'undiksha', 'binus', 'unram', 'unair', 'telkom', 'unpad']
```

len(list)

The len() function returns the length of the list.

```
[ ] len(University)
```

```
9
```

list.sort()

The sort() method sorts the elements of a given list in a specific ascending or descending order.

```
[ ] University.sort()
```

```
[ ] print(University)
```

```
['binus', 'itb', 'telkom', 'unair', 'undiksha', 'unpad', 'unram', 'uns', 'unud']
```

list.reverse()

The reverse method use for reverse the list

```
[ ] University.reverse()  
  
[ ] print(University)  
['unud', 'uns', 'unram', 'unpad', 'undiksha', 'unair', 'telkom', 'itb', 'binus']
```

list.index()

Returns the first index of a value in the list.

```
[ ] University.index("unpad")  
3  
  
[ ] print(University)  
['unud', 'uns', 'unram', 'unpad', 'undiksha', 'unair', 'telkom', 'itb', 'binus']
```


2. Slicing

- Slicing is a technique for selecting a data or some data from a data set.
- Slicing can be used in the use of number, string, list, or tuple data types.

Slicing in number

```
[ ] Height = 156, 170.4, 160, 167, 163.7, 172
```

```
[ ] print(Height[2])    #This command will display the Height of the body in the index 2  
    print(Height[:])    #This command indicates all the data in the list, because the index [:] shows all the data  
    print(Height[1:3])   #In the use of slicing, the last order is not read. Thus the writing [1:3] executed is the index of the 1st to the 2nd  
    print(Height[:4])    #command [:4] indicates that the data selected is from index 0 (as default) to 3rd index (because the 4th index is not readable)  
    print(Height[0:6:2]) #This command indicates that the selected data is from index 0 to 6 where it will jump 2 index
```

```
160  
(156, 170.4, 160, 167, 163.7, 172)  
(170.4, 160)  
(156, 170.4, 160, 167)  
(156, 160, 163.7)
```

Slicing in string

Strings have similarities to list, so operators `[]` can also be used on strings to retrieve their contents or even substring. Whole strings are mutable, while elements are immutable.

```
[ ] Job = "Data Science and Data Analyst"
```

```
print(Job[5])      #Take the fifth character from the Job string
print(Job[:13])    #Take the index of characters 0 to 12th from the Job string
print(Job[17:])    #Take the index of characters 17th to last from the Job string
print(Job[-5])     #Take Take 5th character from the back of the Job string
print(Job[-4:])    #Take the last 4 characters until they run out of the Job string
print(Job[2:14:3]) #Take characters with index 2nd to 13th from Job string and each take jump 3 index
```

```
S
Data Science
Data Analyst
a
lyst
tSee
```

Slicing in string

```
✓ [1] Biodata = ["Neymar", 20, 175.2, "Bandung", "Microsoft Office Specialist"]
0s

✓ [2] print(Biodata[2])    #Take the second index from the Biodata list
0s    print(Biodata[:1])    #Take the zero index from the Biodata list
    print(Biodata[-3])    #Take the index to three in reverse from Biodata list
    print(Biodata[0:7])   #Take the index 0 to 6 from the Biodata list. Although the Biodata only reaches the 3rd index, it will only display all existing lists.

175.2
['Neymar']
175.2
['Neymar', 20, 175.2, 'Bandung', 'Microsoft Office Specialist']
```

Slicing in tuple

```
[ ] Pathway = ("Data Science Track", "Career Ready Track", "Ms. Office Specialist Expert",
              "Adobe Certified Professional in Visual Design", "Adobe Certified Professional in Video Design", "21st Century Educator")

[ ] print(Pathway[1])    #Take the index-1 from tuple
    print(Pathway[-2])   #Take the second last item from tuple
    print(Pathway[3:5])  #Take the index 3 to 4 from tuple
    print(Pathway[-4:-1]) #Take the index from -4 to -2 beside tuple

Career Ready Track
Adobe Certified Professional in Video Design
('Adobe Certified Professional in Visual Design', 'Adobe Certified Professional in Video Design')
('Ms. Office Specialist Expert', 'Adobe Certified Professional in Visual Design', 'Adobe Certified Professional in Video Design')
```

3. Tuple

A tuple is a collection of objects which ordered and immutable. Tuples are sequences, just like lists. The differences between tuples and lists are the tuples cannot be changed unlike lists and tuples use parentheses(), whereas lists use square brackets[].

A tuple can have any number of items and they may be of different types (integer, float, list, string, etc).


```
[ ] data1 = ("Chris",21,170.8,"Jakarta","Data Science Track")  
    data2 = ("Rayna",19,165.5,"Denpasar","Career Ready Track")
```


```
[ ] print("the type of data is",type(data1))  
    print("the type of data is",type(data2))
```

```
the type of data is <class 'tuple'>  
the type of data is <class 'tuple'>
```

Access **tuple** item

To access an item in tuples, we can use square brackets to slice along with the index or to get the values available at that index. For example:

```
 print(data1[0]) #The index of 0 refers to the first item
```

```
 Chris
```

```
[ ] print(data1[-2]) #The index of -2 refers to the second last item in tuple
```

```
Jakarta
```

```
[ ] print(data1[1:3]) #The search will start at index 1 (included) and end at index 3 (excluded
```

```
(21, 170.8)
```

```
[ ] print(data1[:5]) #The index will start from the beginning to index 4
```

```
('Chris', 21, 170.8, 'Jakarta', 'Data Science Track')
```

Tuple length

To determine how many items a tuple has, you can use the `len()` function:

```
[ ] class_program = ("Microsft Excel Associate","Microsoft Excel Expert","Phyton","Azure DP-900", "Azure AZ-900")
```

```
[ ] print(len(class_program))
```

```
5
```

4. Set

Data Set is a collection of items that are unique and unordered collection. Sets are formed with curly brackets {} and separated by commas (.). In Set we can also do duplicate item removal.

```
[ ] Name = {"E","L","E","N","A"}  
    Phone_number = {0,5,6,3,7,6,2,7}  
  
[ ] print(Name) # This command will display the name, and we will know that the duplicate data was be deleted  
    print(Phone_number) # Twin numbers are only displayed one in sequence  
  
{'N', 'E', 'A', 'L'}  
{0, 2, 3, 5, 6, 7}
```

Because it's unordered, then the items in the data cannot be taken partially/slicing.

```
[ ] print(Name[2]) # This command will be an Error, according to the character of the data Set  
  
-----  
TypeError                                Traceback (most recent call last)  
<ipython-input-85-d48c5c2cfecb> in <module>()  
----> 1 print(Name[2]) # This command will be an Error, according to the character of the data Set  
  
TypeError: 'set' object is not subscriptable
```

5. Dictionary

- Dictionary is a data type whose members consist of key:value pairs (key:value). Dictionary is unordered (unordered) so that its members do not have an index.
- A dictionary is created by placing its members inside curly braces { }, separated by commas.
- The dictionary members consist of key:value pairs. The key must be unique, there cannot be two identical keys in the dictionary.

```
[ ] identity = {"name": "andi", "class": "data science track B", "class hours": 2}
```

```
[ ] identity["name"] # This command will display the "value" of the key named "name"
```

```
'andi'
```

```
[ ] identity["class"] # This command will display the "value" of the key named "class"
```

```
'data science track B'
```

```
[ ] identity["class hours"] # This command will display the "value" of the key named "class hours"
```

```
2
```


When “key” is integer

```
[ ] pathway = {1: "Data Science Track", 2: "Career Ready Track", 3: "Adobe Certified Professional in Visual Design"}  
  
[ ] pathway[2] # This command will return the "value" of the integer "key" named "2"  
      'Career Ready Track'  
  
[ ] pathway[1] # This command will return the "value" of the integer "key" named "1"  
      'Data Science Track'
```

When “value” is list

```
"Value" = List  
  
[ ] MyEduSolve = {"Class MyEduSolve": ["Pathway Class", "Career Mentoring Class", "1 on 1 Class", "Masterclass", "Sandbox Class"]}  
  
[ ] MyEduSolve["Class MyEduSolve"] # This command will display the "value" in the form of a "list" of "key" named "class MyEduSolve"  
      ['Pathway Class',  
       'Career Mentoring Class',  
       '1 on 1 Class',  
       'Masterclass',  
       'Sandbox Class']
```

Thank You

