Linear and Associative Data Structure

Linear Data Structure are ordered list and follows zero-based indexing. Elements are accessed by index value. Eg. List, Tuple and String.

Associative Data Structure do not follows zero-based indexing. Elements are accessed by an associated key value. Eg. Dictionary.

Dictionary

It is an ordered, and mutable assciative data structure.

1. Creating Dictionary: It is created by mentioning the keys and values within the curly bracket.

Dict_list = {key1:value1, key2:value2, ...}

```
marks = {'Physics':45, 'Maths': 50 , 'Chemistry': 48}
print(marks)
```

```
{'Physics': 45, 'Maths': 50, 'Chemistry': 48}
```

```
# Empty String is acceptable
marks = {'':45, 'Maths': 50 , 'Chemistry': 48}
print(marks)
```

```
{'': 45, 'Maths': 50, 'Chemistry': 48}
```

```
type(marks)
```

```
dict
```

It does not allow duplication that is Key remains unique and the current value will be overwrited by the last updated value.

```
marks = {'Physics':45, 'Maths': 50 , 'Chemistry': 48, 'Physics':35}
print(marks)
```

```
{'Physics': 35, 'Maths': 50, 'Chemistry': 48}
```

Values of any data type are allowed.

```
stud = {'Name':'Ankit Verma', 'Enrollment_Number': 210023, 'marks':[25, 45, 35]}
print(stud)
```

```
{'Name': 'Ankit Verma', 'Enrollment_Number': 210023, 'marks': [25, 45, 35]}
```

Any immutable type can also be used as key in the Dictionary.

```
Datee = {('Mon',23,'May'):34, ('Tues',24,'May'):32, ('Wed',25,'May'):36}
print(Datee)
```

```
{('Mon', 23, 'May'): 34, ('Tues', 24, 'May'): 32, ('Wed', 25, 'May'): 36}
```

```
Datee = {['Mon',23,'May']:34, ('Tues',24,'May'):32, ('Wed',25,'May'):36}
print(Datee)
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
<ipython-input-4-866f159ad082> in <module>
```

New key:value pair can be added to dictionary at any time in the program.

```
stud = {'Name':'Ankit Verma', 'Enrollment_Number': 210023, 'marks':[25, 45, 35]}
print('Before:',stud)
stud["Garde"]='C'
print('After:',stud)
```

```
Before: {'Name': 'Ankit Verma', 'Enrollment_Number': 210023, 'marks': [25, 45, 35]}
After: {'Name': 'Ankit Verma', 'Enrollment_Number': 210023, 'marks': [25, 45, 35], 'Garde': 'C'}
```

Value in the dictionary can be updated as follows:

```
stud = {'Name':'Ankit Verma', 'Enrollment_Number': 210023, 'marks':[25, 45, 35],'Grade':'C'}
print('Before:',stud)
stud["Grade"]='B'
print('After:',stud)
```

```
Before: {'Name': 'Ankit Verma', 'Enrollment_Number': 210023, 'marks': [25, 45, 35], 'Grade': 'C'}
After: {'Name': 'Ankit Verma', 'Enrollment_Number': 210023, 'marks': [25, 45, 35], 'Grade': 'B'}
```

Value can also be updated using a **method update()**

```
stud = {'Name':'Ankit Verma', 'Enrollment_Number': 210023, 'marks':[25, 45, 35],'Grade':'C'}
print('Before:',stud)
stud.update({"Grade":'B'})
print('After:',stud)
```

```
Before: {'Name': 'Ankit Verma', 'Enrollment_Number': 210023, 'marks': [25, 45, 35], 'Grade': 'C'}
After: {'Name': 'Ankit Verma', 'Enrollment_Number': 210023, 'marks': [25, 45, 35], 'Grade': 'C', 'Grade1': 'B'}
```

Using the method: **update()** , if the key is not present in the dictionary, then a new key:value pair will be added to the dictionary.

```
stud = {'Name':'Ankit Verma', 'Enrollment_Number': 210023, 'marks':[25, 45, 35],'Grade':'C'}
print('Before:',stud)
stud.update({"Status":'Pass'})
print('After:',stud)
```

```
Before: {'Name': 'Ankit Verma', 'Enrollment_Number': 210023, 'marks': [25, 45, 35], 'Grade': 'C'}
After: {'Name': 'Ankit Verma', 'Enrollment_Number': 210023, 'marks': [25, 45, 35], 'Grade': 'C', 'Status': 'Pass'}
```

To Delete the item with specific key, use **method pop()** or **del keyword**

```
stud = {'Name':'Ankit Verma', 'Enrollment_Number': 210023, 'marks':[25, 45, 35],'Grade':'C'}
print('Before:',stud)
stud.pop("Grade")
print('After:',stud)
```

```
Before: {'Name': 'Ankit Verma', 'Enrollment_Number': 210023, 'marks': [25, 45, 35], 'Grade': 'C'}
After: {'Name': 'Ankit Verma', 'Enrollment_Number': 210023, 'marks': [25, 45, 35]}
```

```
stud = {'Name':'Ankit Verma', 'Enrollment_Number': 210023, 'marks':[25, 45, 35],'Grade':'C'}
print('Before:',stud)
del stud["Grade"]
print('After:',stud)
```

```
Before: {'Name': 'Ankit Verma', 'Enrollment_Number': 210023, 'marks': [25, 45, 35], 'Grade': 'C'}
After: {'Name': 'Ankit Verma', 'Enrollment_Number': 210023, 'marks': [25, 45, 35]}
```

Error will be popped if a key which is not present in the dictionary is mentioned in the pop method.

```
stud = {'Name':'Ankit Verma', 'Enrollment_Number': 210023, 'marks':[25, 45, 35],'Grade':'C'}
print('Before:',stud)
stud.pop("Status")
print('After:',stud)
```

```
Before: {'Name': 'Ankit Verma', 'Enrollment_Number': 210023, 'marks': [25, 45, 35], 'Grade': 'C'}
---------------------------------------------------------------------------
KeyError                                  Traceback (most recent call last)
<ipython-input-21-88c743e71813> in <module>
      1 stud = {'Name':'Ankit Verma', 'Enrollment_Number': 210023, 'marks':[25, 45, 35],'Grade':'C'}
      2 print('Before:',stud)
----> 3 stud.pop("Status")
      4 print('After:',stud)

KeyError: 'Status'
```

SEARCH STACK OVERFLOW

To remove last inserted item in the dictionary, use **method popitem()**

```
stud = {'Name':'Ankit Verma', 'Enrollment_Number': 210023, 'marks':[25, 45, 35],'Grade':'C'}
print('Before:',stud)
stud.popitem()
print('After:',stud)
stud.popitem()
print('Again:',stud)
```

```
Before: {'Name': 'Ankit Verma', 'Enrollment_Number': 210023, 'marks': [25, 45, 35], 'Grade': 'C'}
After: {'Name': 'Ankit Verma', 'Enrollment_Number': 210023, 'marks': [25, 45, 35]}
Again: {'Name': 'Ankit Verma', 'Enrollment_Number': 210023}
```

To remove all the items from the dictionary, use **method clear()**

```
stud = {'Name':'Ankit Verma', 'Enrollment_Number': 210023, 'marks':[25, 45, 35],'Grade':'C'}
print('Before:',stud)
stud.clear()
print('After:',stud)
```

```
Before: {'Name': 'Ankit Verma', 'Enrollment_Number': 210023, 'marks': [25, 45, 35], 'Grade': 'C'}
After: {}
```

Dictionaries are defined as objects with the data type 'dict'

```
print(type(stud))
```

```
<class 'dict'>
```

2. Access values from Dictionary

Value from the dictionary can be accessed by mentioning the key inside the square bracket.

```
stud = {'Name':'Ankit Verma', 'Enrollment_Number': 210023, 'marks':[25, 45, 35]}
print(stud['marks'])
```

```
[25, 45, 35]
```

Value from the dictionary can also be accessed by using **method get()**.

```
stud = {'Name':'Ankit Verma', 'Enrollment_Number': 210023, 'marks':[25, 45, 35]}
print(stud.get('Name'))
```

```
Ankit Verma
```

To get the list of all the Keys in the Dictionary use **method keys()**.

```
stud = {'Name':'Ankit Verma', 'Enrollment_Number': 210023, 'marks':[25, 45, 35]}
x=stud.keys()
print(x)
```

```
dict_keys(['Name', 'Enrollment_Number', 'marks'])
```

To get the list of all values in the Dictionary use **method values()**.

```
stud = {'Name':'Ankit Verma', 'Enrollment_Number': 210023, 'marks':[25, 45, 35]}
y=stud.values()
print(y)
```

```
dict_values(['Ankit Verma', 210023, [25, 45, 35]])
```

```
stud = {'Name':'Ankit Verma', 'Enrollment_Number': 210023, 'marks':[25, 45, 35]}
y=stud.items()
print(y)
```

```
dict_items([('Name', 'Ankit Verma'), ('Enrollment_Number', 210023), ('marks', [25, 45, 35])])
```

in Operator for Dictionary

```
stud = {'Name':'Ankit Verma', 'Enrollment_Number': 210023, 'marks':[25, 45, 35]}
'Name' in stud
```

```
True
```

```
'Grade' in stud
```

```
False
```

```
stud = {'Name':'Ankit Verma', 'Enrollment_Number': 210023, 'marks':[25, 45, 35]}
for x in stud:
  print(x,':',stud[x])
```

```
Name : Ankit Verma
Enrollment_Number : 210023
marks : [25, 45, 35]
```

To make a copy of dictionary, use **built-in function dict()** or **method copy()**

```
num = 10
k=10
list=[1,2,3]
list1= list
list1[1]=13
print(list)
```

```
[1, 13, 3]
```

```
stud = {'Name':'Ankit Verma', 'Enrollment_Number': 210023, 'marks':[25, 45, 35]}
# it will just create a reference to dictionary "stud"
stud1 = stud
stud1['Name']='sfdgjhsfgd'
print(stud)
```

```
{'Name': 'sfdgjhsfgd', 'Enrollment_Number': 210023, 'marks': [25, 45, 35]}
```

```
stud = {'Name':'Ankit Verma', 'Enrollment_Number': 210023, 'marks':[25, 45, 35]}
stud2 = stud.copy()
stud2['Name']='sfdgjhsfgd'
print('Öriginal Dictionary:',stud)
print('Modified Dictionary:',stud2)
```

```
Öriginal Dictionary: {'Name': 'Ankit Verma', 'Enrollment_Number': 210023, 'marks': [25, 45, 35]}
Modified Dictionary: {'Name': 'sfdgjhsfgd', 'Enrollment_Number': 210023, 'marks': [25, 45, 35]}
```

```
stud = {'Name':'Ankit Verma', 'Enrollment_Number': 210023, 'marks':[25, 45, 35]}
stud2 = dict(stud)
stud2['Name']='sfdgjhsfgd'
print('Öriginal Dictionary:',stud)
print('Modified Dictionary:',stud2)
```

```
Öriginal Dictionary: {'Name': 'Ankit Verma', 'Enrollment_Number': 210023, 'marks': [25, 45, 35]}
Modified Dictionary: {'Name': 'sfdgjhsfgd', 'Enrollment_Number': 210023, 'marks': [25, 45, 35]}
```

Nested Dictionary

```
stud1 = {'Name':'Ankit Verma', 'Enrollment_Number': 210023, 'marks':[25, 45, 35]}
stud2 = {'Name':'Sahil', 'Enrollment_Number': 210024, 'marks':[45, 35, 35]}
stud3 = {'Name':'Vineet', 'Enrollment_Number': 210025, 'marks':[35, 45, 35]}

Student_data = { "student1" : stud1,
  "student2" : stud2,
  "student3" : stud3
}

print(Student_data)
```

```
{'student1': {'Name': 'Ankit Verma', 'Enrollment_Number': 210023, 'marks': [25, 45, 35]}, 'student2': {'Name': 'Sahil', 'Enroll
```