

```
In [1]: myTuple = ("apple", "mango", "banana")
        print(len(myTuple))
```

3

```
In [3]: t = ("berry",)
        print(type(t))
```

<class 'tuple'>

```
In [5]: myTuple = ("apple", "mango", "banana")
        myTuple.append("cherry")
        print(myTuple)
```

```
-----
--
AttributeError                                Traceback (most recent call las
t)
Cell In[5], line 2
      1 myTuple = ("apple", "mango", "banana")
----> 2 myTuple.append("cherry")
      3 print(myTuple)

AttributeError: 'tuple' object has no attribute 'append'
```

```
In [6]: myTuple = ("apple", "mango", "banana")
        y = list(myTuple)
        y.append("cherry")
        myTuple = tuple(y)
        print(myTuple)
```

('apple', 'mango', 'banana', 'cherry')

```
In [7]: mytuple = ("apple", "mango", "banana")
        (green, yellow, red) = mytuple
        print(green)
        print(yellow)
        print(red)
```

apple
mango
banana

```
In [9]: mytuple = ("apple", "mango", "banana", "kiwi", "orange", "cherry")
        (green, yellow, *red) = mytuple
        print(green)
        print(yellow)
        print(red)
```

apple
mango
['banana', 'kiwi', 'orange', 'cherry']

```
In [10]: mytuple = ("apple", "mango", "banana", "kiwi", "orange", "cherry")
         (green, *yellow, red) = mytuple
         print(green)
         print(yellow)
         print(red)
```

```
apple
['mango', 'banana', 'kiwi', 'orange']
cherry
```

```
In [13]: #Doesnt Print multiple values
         myset = {"apple", "mango", "banana", "mango", "kiwi", "orange", "cherry"}
         print(myset)
```

```
{'cherry', 'apple', 'mango', 'orange', 'kiwi', 'banana'}
```

```
In [15]: thisset = {"apple", "banana", "cherry"}
         thisset.add("orange")
         print(thisset)
```

```
{'banana', 'cherry', 'orange', 'apple'}
```

```
In [17]: thisset = {"apple", "mango", "cherry"}
         tropical = {"pineapple", "papaya", "banana"}
         thisset.update(tropical)
         print(thisset)
```

```
{'apple', 'cherry', 'mango', 'pineapple', 'papaya', 'banana'}
```

```
In [18]: set1 = {"a", "b", "c"}
         set2 = {1, 2, 3}
         set1.update(set2)
         print(set1)
```

```
{1, 2, 3, 'b', 'c', 'a'}
```

```
In [19]: x = {"apple", "banana", "cherry"}
         y = {"google", "microsoft", "apple"}
         x.symmetric_difference_update(y)
         print(x)
```

```
{'banana', 'cherry', 'microsoft', 'google'}
```

Dictionary

Dictionary is another type of collection which uses two values, like key and value. It is used to store data values in key value pairs. Dictionary is a collection which is unordered, changeable and doesn't allow duplicate values. They are Written with curly brackets and have key and value.

```
In [20]: mydict = {
          "brand" : "Maruti",
          "model" : "Swift",
          "year" : 2020
        }
print(mydict)

{'brand': 'Maruti', 'model': 'Swift', 'year': 2020}
```

```
In [22]: mydict = {
          "brand" : "Maruti",
          "model" : "Swift",
          "year" : 2020
        }
print(mydict["brand"])

Maruti
```

```
In [23]: mydict = {
          "brand" : "Maruti",
          "model" : "Swift",
          "year" : 2020,
          "year" : 2021,
        }
print(mydict)

{'brand': 'Maruti', 'model': 'Swift', 'year': 2021}
```

```
In [24]: print(len(mydict))

3
```

```
In [25]: mydict = {
          "brand" : "Maruti",
          "model" : "Swift",
          "year" : 2020,
          "electric" : False,
          "color" : ["red", "white", "blue"]
        }
print(mydict)

{'brand': 'Maruti', 'model': 'Swift', 'year': 2020, 'electric': False, 'color': ['red', 'white', 'blue']}
```

```
In [26]: print(type(mydict))

<class 'dict'>
```

```
In [27]: x = mydict["model"]
print(x)

Swift
```

```
In [28]: x = mydict.get("model")
print(x)
```

Swift

```
In [29]: x = mydict.keys()
print(x)
```

dict_keys(['brand', 'model', 'year', 'electric', 'color'])

```
In [30]: x = mydict.values()
print(x)
```

dict_values(['Maruti', 'Swift', 2020, False, ['red', 'white', 'blue']])

```
In [32]: mydict = {
    "brand" : "Maruti",
    "model" : "Swift",
    "year" : 2020
}

if "name" in mydict:
    print("Yes: This is one of the key")
else:
    print("No")
```

No

```
In [33]: mydict = {
    "brand" : "Maruti",
    "model" : "Swift",
    "year" : 2020
}

if "model" in mydict:
    print("Yes: This is one of the key")
else:
    print("No")
```

Yes: This is one of the key

```
In [34]: mydict = {
    "brand" : "Maruti",
    "model" : "Swift",
    "year" : 2020
}

x = mydict.keys()
print(x)
mydict["color"] = "White"
print(x)
```

dict_keys(['brand', 'model', 'year'])
dict_keys(['brand', 'model', 'year', 'color'])

```
In [35]: mydict["year"] = 2018  
print(mydict)
```

```
{'brand': 'Maruti', 'model': 'Swift', 'year': 2018, 'color': 'White'}
```

```
In [37]: mydict.update({"year" : 2015})  
print(mydict)
```

```
{'brand': 'Maruti', 'model': 'Swift', 'year': 2015, 'color': 'White'}
```

```
In [38]: mydict = {  
    "brand" : "Maruti",  
    "model" : "Swift",  
    "year" : 2020  
}  
  
mydict.popitem()  
print(mydict)
```

```
{'brand': 'Maruti', 'model': 'Swift'}
```

```
In [39]: mydict = {  
    "brand" : "Maruti",  
    "model" : "Swift",  
    "year" : 2020  
}  
  
del mydict["model"]  
print(mydict)
```

```
{'brand': 'Maruti', 'year': 2020}
```

```
In [43]: ▶ mydict = {
            "brand" : "Maruti",
            "model" : "Swift",
            "year" : 2020
        }

del mydict
print(mydict)
```

```
-----
--
NameError                                Traceback (most recent call las
t)
Cell In[43], line 8
      1 mydict = {
      2     "brand" : "Maruti",
      3     "model" : "Swift",
      4     "year" : 2020
      5 }
      7 del mydict
----> 8 print(mydict)

NameError: name 'mydict' is not defined
```

```
In [41]: ▶ mydict = {
            "brand" : "Maruti",
            "model" : "Swift",
            "year" : 2020
        }

mydict.clear()
print(mydict)

{}
```

```
In [44]: ▶ mydict = {
            "brand" : "Maruti",
            "model" : "Swift",
            "year" : 2020
        }

for x in mydict:
    print(x)

brand
model
year
```

```
In [48]: > for x in mydict:  
          print(mydict[x])
```

```
Maruti  
Swift  
2020
```

```
In [49]: > for x in mydict.keys():  
          print(x)
```

```
brand  
model  
year
```

```
In [50]: > for x,y in mydict.items():  
          print(x,y)
```

```
brand Maruti  
model Swift  
year 2020
```

```
In [51]: > mydict2 = mydict.copy()  
          print(mydict2)
```

```
{'brand': 'Maruti', 'model': 'Swift', 'year': 2020}
```

```
In [53]: > mydict2 = dict(mydict)  
          print(mydict2)
```

```
{'brand': 'Maruti', 'model': 'Swift', 'year': 2020}
```

```
In [54]: > myfamily = {  
            "child1" : {  
                "name" : "Utkrist",  
                "year" : 2001  
            },  
            "child2" : {  
                "name" : "Jaiswal",  
                "year" : 2002  
            }  
        }  
  
print(myfamily)
```

```
{'child1': {'name': 'Utkrist', 'year': 2001}, 'child2': {'name': 'Jaiswal',  
'year': 2002}}
```

Question 1

Python program to find the sum of all items in a dictionary

```
In [75]: mydict = {
          "a" : 10,
          "b" : 20,
          "c" : 30,
          "d" : 45
        }
result = sum(mydict.values())

print("Sum is:", result)

Sum is: 105
```

Question 2

Python program to find the size of a Dictionary

```
In [76]: mydict = {
          'a': 10,
          'b': 20,
          'c': 30,
          "d" : 255
        }

print("Size is:", len(mydict))

Size is: 4
```

Question 3

Python program to Sort Dictionaries by Key or Value

```
In [77]: mydict = {
          'b': 20,
          'a': 10,
          'c': 30,
          "d" : 255
        }

x = sorted(mydict.values())
print(x)

y = sorted(mydict.keys())
print(y)

[10, 20, 30, 255]
['a', 'b', 'c', 'd']
```


Question 4

Sort list of dictionaries by values in Python – Using lambda function

```
In [82]: mydicts = [  
    {'key': 'b', 'value': 20},  
    {'key': 'a', 'value': 10},  
    {'key': 'c', 'value': 30}  
]  
  
sorted_list_by_values = sorted(mydicts, key = lambda x: x["value"])  
print("Sorted by values:", sorted_list_by_values)
```

```
Sorted by values: [{'key': 'a', 'value': 10}, {'key': 'b', 'value': 20},  
{ 'key': 'c', 'value': 30}]
```

Question 5

Program to Merge two Dictionaries

```
In [85]: dict1 = {  
    'a': 10,  
    'b': 20  
}  
  
dict2 = {  
    'b': 30,  
    'c': 40,  
    'd': 11  
}  
  
merged_dict = dict1.copy()  
merged_dict.update(dict2)  
  
print("Merged dictionary:", merged_dict)
```

```
Merged dictionary: {'a': 10, 'b': 30, 'c': 40, 'd': 11}
```

Question 6

Program to Find all duplicate characters in string

```
In [88]: ▶ str = "Volley Ball"

charCount = {}
duplicates = []

for char in str:
    charCount[char] = charCount.get(char, 0) + 1

for char, count in charCount.items():
    if count > 1:
        duplicates.append(char)

print(duplicates)

['l']
```

Question 7

Program to Replace String by Kth Dictionary value

```
In [100]: ▶ replacementDict = {
    'name': 'John',
    'city': 'New York'
}

instr = "name city"
words = instr.split()
for i in range(len(words)):
    if words[i] in replacementDict:
        words[i] = replacementDict[words[i]]

outstr = " Welcome to ".join(words)
print(outstr)

John Welcome to New York
```

Question 8

Python | Remove all duplicates words from a given sentence

```
In [102]: ▶ sentence = "I am Utkrist Jaiswal I am Utkrist Jaiswal"
unique_words = set(sentence.split())
unique_sentence = " ".join(unique_words)
print(unique_sentence)

Utkrist I Jaiswal am
```

Question 9

Program to Coun the frequencies in a list using dictionary in Python

```
In [103]: ▶ lst = [1, 2, 3, 1, 2, 1, 4, 5, 4, 3, 5, 6, 7, 7, 7]

frequency_dict = {}

for element in lst:

    if element in frequency_dict:

        frequency_dict[element] += 1
    else:
        frequency_dict[element] = 1

for key, value in frequency_dict.items():
    print(f"{key}: {value}")

1: 3
2: 2
3: 2
4: 2
5: 2
6: 1
7: 3
```

Question 10

Program to create grade calculator in Python using dictionary

```
In [107]: ▶ ranges = {
    (0, 39) : "F",
    (40, 59) : "E",
    (60, 69) : "D",
    (70, 79) : "C",
    (80, 89) : "B",
    (90, 100) : "A"
}

mygrades=[95,89,56,50,90]
avggrade=sum(mygrades)/len(mygrades)

for range_, lettergrade in ranges.items():
    if range_[0] <= avggrade <= range_[1]:
        finalgrade=lettergrade
        break
    else:
        finalgrade="Invalid"

print(finalgrade)
```

C

