

Tuple

- Collection of items or group of items
- tuple is immutable
- tuple()
- symbol () paranthesis

In [1]: t = tuple()
print(type(t))

<class 'tuple'>

In [3]: t1 = (1,2,3,4,5)
print(type(t1))

<class 'tuple'>

In [4]: t2 = (1,"apssdc",7.8)
t2

Out[4]: (1, 'apssdc', 7.8)

In [5]: t2[1] = "srkit"

TypeError Traceback (most recent call last)
<ipython-input-5-1b3416c60977> in <module>
----> 1 t2[1] = "srkit"

TypeError: 'tuple' object does not support item assignment

- tuple indexing

In [6]: t1

Out[6]: (1, 2, 3, 4, 5)

In [7]: t1[1]

Out[7]: 2

In [8]: t1[-1]

Out[8]: 5

- tuple slicing

In [9]: t1[2:5]

Out[9]: (3, 4, 5)

In [10]: t1[1:]

Out[10]: (2, 3, 4, 5)

- Methods in tuple

In [11]: print(dir(tuple))

['__add__', '__class__', '__contains__', '__delattr__', '__dir__', '__doc__', '__eq__', '__format__', '__ge_'
_', '__getattr__', '__getitem__', '__getnewargs__', '__gt__', '__hash__', '__init__', '__init_subclass_'
_', '__iter__', '__le__', '__len__', '__lt__', '__mul__', '__ne__', '__new__', '__reduce__', '__reduce_ex__',
['__repr__', '__rmul__', '__setattr__', '__sizeof__', '__str__', '__subclasshook__', 'count', 'index']

- tuplevariable.count(item)

In [12]: t = (1,2,1,3,1,1)
t

Out[12]: (1, 2, 1, 3, 1, 1)

In [14]: t.count(1)

Out[14]: 4

```
In [15]: t.count(2)

Out[15]: 1

    • tuplevariable.index(item)

In [16]: t.index(3)

Out[16]: 3

In [17]: t.index(2)

Out[17]: 1

In [18]: min(t)

Out[18]: 1

In [19]: max(t)

Out[19]: 3

In [20]: sum(t)

Out[20]: 9

In [22]: sum(t2)

-----
TypeError                                Traceback (most recent call last)
<ipython-input-22-3fd5fddf4b2a> in <module>
----> 1 sum(t2)

TypeError: unsupported operand type(s) for +: 'int' and 'str'

In [21]: t2

Out[21]: (1, 'apssdc', 7.8)

In [ ]:
```

Dictionary

- dictionary is acollection of unordered items stored in the form of key and value i.e pair.
- we can seperate key ,value i.e pair by using symbol : (colon)
- creating a dictionary
- dict()
- symbol {} curly braces

```
In [23]: d = dict()
print(type(d))

<class 'dict'>

In [24]: d = {}
print(type(d))

<class 'dict'>

In [25]: students = {"A1203":"sasidhar","A1217":"Aihika","A1218":"christopher"}
students

Out[25]: {'A1203': 'sasidhar', 'A1217': 'Aihika', 'A1218': 'christopher'}

In [26]: len(students)

Out[26]: 3

In [32]: # for accessing the keys in dictionary
for rollnumber in students:
    print(rollnumber)

A1203
A1217
A1218
```

- Dictionaryvariable[key]

```
In [30]: students["A1218"]
```

Out[30]: 'christopher'

```
In [31]: students["christopher"]
```

```
-----
KeyError                                Traceback (most recent call last)
<ipython-input-31-ceda955d3a75> in <module>
----> 1 students["christopher"]

KeyError: 'christopher'
```

```
In [34]: # For accessing the values in dictionary
for rollnumber in students:
    print(students[rollnumber])
```

sasidhar
Aihika
christopher

- adding a new item into dictionary
- dictvariable[newkey] = value

```
In [35]: students["A1245"] = "Hemanth"
```

```
In [36]: students
```

Out[36]: {'A1203': 'sasidhar',
'A1217': 'Aihika',
'A1218': 'christopher',
'A1245': 'Hemanth'}

```
In [37]: students["A1203"] = "sasidhar"
```

```
In [38]: students
```

Out[38]: {'A1203': 'sasidhar',
'A1217': 'Aihika',
'A1218': 'christopher',
'A1245': 'Hemanth'}

- updating a value for existing key
- dictvariable[key]=newvalue

```
In [39]: students["A1203"] = "Kallam Sasidhar"
```

```
In [40]: students
```

Out[40]: {'A1203': 'Kallam Sasidhar',
'A1217': 'Aihika',
'A1218': 'christopher',
'A1245': 'Hemanth'}

```
In [41]: new = {"names":"apssdc","names":"srkit"}
print(new)
```

{'names': 'srkit'}

- Methods in dictionary

```
In [42]: print(dir(dict))
```

['_class_', '__contains__', '__delattr__', '__delitem__', '__dir__', '__doc__', '__eq__', '__format__', '__ge_e__', '__getattribute__', '__getitem__', '__gt__', '__hash__', '__init__', '__init_subclass__', '__iter__', '_ _le_', '__len__', '__lt__', '__ne__', '__new__', '__reduce__', '__reduce_ex__', '__repr__', '__setattr__', '_ _setitem__', '__sizeof__', '__str__', '__subclasshook__', 'clear', 'copy', 'fromkeys', 'get', 'items', 'keys', 'pop', 'popitem', 'setdefault', 'update', 'values']

- clear()
- it clears all the items in dictionary

In [44]: new

Out[44]: {'names': 'srkit'}

In [45]: new.clear()

In [46]: new

Out[46]: {}

- copy()

In [49]: students

Out[49]: {'A1203': 'Kallam Sasidhar',
'A1217': 'Aihika',
'A1218': 'christopher',
'A1245': 'Hemanth'}

In [50]: names = students.copy()
names

Out[50]: {'A1203': 'Kallam Sasidhar',
'A1217': 'Aihika',
'A1218': 'christopher',
'A1245': 'Hemanth'}

In [51]: students["A1234"] = "srkit"

In [52]: students

Out[52]: {'A1203': 'Kallam Sasidhar',
'A1217': 'Aihika',
'A1218': 'christopher',
'A1245': 'Hemanth',
'A1234': 'srkit'}

In [53]: names

Out[53]: {'A1203': 'Kallam Sasidhar',
'A1217': 'Aihika',
'A1218': 'christopher',
'A1245': 'Hemanth'}

- fromkeys(collection_of_keys,optional_default_value)

In [54]: keys = ["a","b","c","d"]
{}.fromkeys(keys)

Out[54]: {'a': None, 'b': None, 'c': None, 'd': None}

In [55]: {}.fromkeys(keys,100)

Out[55]: {'a': 100, 'b': 100, 'c': 100, 'd': 100}

- get(key,defaultvalue)
- to get a value for a given key

In [57]: students["A1203"]

Out[57]: 'Kallam Sasidhar'

In [63]: students.get("A1203","rollnumber is not aviable")

Out[63]: 'Kallam Sasidhar'

In [62]: print(students.get("A1215","rollnumber is not aviable "))

rollnumber is not aviable

In [60]: students

Out[60]: {'A1203': 'Kallam Sasidhar',
'A1217': 'Aihika',
'A1218': 'christopher',
'A1245': 'Hemanth',
'A1234': 'srkit'}

- keys()--> to get the keys in dictionary

```
In [65]: students.keys()
```

Out[65]: dict_keys(['A1203', 'A1217', 'A1218', 'A1245', 'A1234'])

- values() -> to get the values in dictionary

```
In [66]: students.values()
```

Out[66]: dict_values(['Kallam Sasidhar', 'Aihika', 'christopher', 'Hemanth', 'srkit'])

```
In [67]: students.items()
```

Out[67]: dict_items([('A1203', 'Kallam Sasidhar'), ('A1217', 'Aihika'), ('A1218', 'christopher'), ('A1245', 'Hemanth'), ('A1234', 'srkit')])

- pop(key)
- remove the item related to given data if available, otherwise prints keyerror

```
In [68]: students.pop("A1203")
```

Out[68]: 'Kallam Sasidhar'

```
In [69]: students
```

Out[69]: {'A1217': 'Aihika',
 'A1218': 'christopher',
 'A1245': 'Hemanth',
 'A1234': 'srkit'}

```
In [70]: students.pop("A1290")
```

```
-----  
KeyError                                Traceback (most recent call last)  
<ipython-input-70-121666e58932> in <module>  
----> 1 students.pop("A1290")  
  
KeyError: 'A1290'
```

- popitem()
- to remove the last item in dictionary

```
In [71]: students
```

Out[71]: {'A1217': 'Aihika',
 'A1218': 'christopher',
 'A1245': 'Hemanth',
 'A1234': 'srkit'}

```
In [72]: students.popitem()
```

Out[72]: ('A1234', 'srkit')

```
In [73]: students
```

Out[73]: {'A1217': 'Aihika', 'A1218': 'christopher', 'A1245': 'Hemanth'}

- update()

```
In [74]: students["A1217"] = "Gunda Aihika"
```

```
In [75]: students
```

Out[75]: {'A1217': 'Gunda Aihika', 'A1218': 'christopher', 'A1245': 'Hemanth'}

```
In [82]: a = {"home": "house", "mobile": "contactnumber"}  
         b = {"mobile": "contactnumber", "address": "dat"}
```

```
In [83]: print(a)  
         print(b)
```

```
{'home': 'house', 'mobile': 'contactnumber'}  
{'mobile': 'contactnumber', 'address': 'dat'}
```

```
In [85]: a.update(b)
print(a)
```

{'home': 'house', 'mobile': 'contactnumber', 'address': 'dat'}

```
In [87]: b.update(a)
print(b)
```

{'mobile': 'contactnumber', 'address': 'dat', 'home': 'house'}

- setdefault(key,optional value)
- it can add a new item if given key is not avialble

```
In [76]: names = {"1":"a",2:"b"}
```

```
In [77]: names
```

Out[77]: {'1': 'a', 2: 'b'}

```
In [78]: names.setdefault(5,"e")
```

Out[78]: 'e'

```
In [79]: names
```

Out[79]: {'1': 'a', 2: 'b', 5: 'e'}

```
In [80]: names.setdefault("1","l")
```

Out[80]: 'a'

```
In [81]: names
```

Out[81]: {'1': 'a', 2: 'b', 5: 'e'}

```
In [88]: a = {"names":["aihika","supriya","hemanth"],"rollnumbers":[123,124,125]}
a
```

Out[88]: {'names': ['aihika', 'supriya', 'hemanth'], 'rollnumbers': [123, 124, 125]}

```
In [89]: a["names"]
```

Out[89]: ['aihika', 'supriya', 'hemanth']

```
In [90]: a["rollnumbers"]
```

Out[90]: [123, 124, 125]

```
In [92]: a["rollnumbers"][1]
```

Out[92]: 124

```
In [93]: a1 = {"names":["aihika","supriya","hemanth"],"rollnumbers":[123,124,(123,678)]}
a1
#678
```

Out[93]: {'names': ['aihika', 'supriya', 'hemanth'], 'rollnumbers': [123, 124, (123, 678)]}

```
In [95]: a1["rollnumbers"]
```

Out[95]: [123, 124, (123, 678)]

```
In [97]: a1["rollnumbers"][2][1]
```

Out[97]: 678

```
In [98]: a1 = {"names":["aihika","supriya","hemanth"],"rollnumbers":[123,124,(123,[678,89])]}
a1
#678
```

Out[98]: {'names': ['aihika', 'supriya', 'hemanth'], 'rollnumbers': [123, 124, (123, [678, 89])]}

```
In [105]: a1["rollnumbers"][2][1][1]
```

Out[105]: 89

```
In [106]: a3 ={"names":{"apssdc":"srkit"},"rollnumbers":{"123":1234}}
a3
```

Out[106]: {'names': {'apssdc': 'srkit'}, 'rollnumbers': {'123': 1234}}

```
In [108]: #srkit
a3["names"]["apssdc"]
```

Out[108]: 'srkit'

```
In [ ]:
```

```
In [115]: # find the frequency of characters in the given sentence

a = "apssdc is conducting online workshops"

#{ "a":1, "p":2, "s":4, .....}
freq = {}

for char in a:#a,p
    freq[char] = a.count(char)#freq[p]=a.count("p")
print(freq)

{'a': 1, 'p': 2, 's': 5, 'd': 2, 'c': 3, ' ': 4, 'i': 3, 'o': 4, 'n': 4, 'u': 1, 't': 1, 'g': 1, 'l': 1, 'e': 1, 'w': 1, 'r': 1, 'k': 1, 'h': 1}
```

```
In [113]: a = "apssdc"
a.count('s')
```

Out[113]: 2

- creating a contact

```
In [126]: contact = {}
def contactapp(name,number):
    if name not in contact:
        contact[name] = number
        print(name," contact is added")
    else:
        print(name," is already exists")
    return contact
contactapp("alekhya",9876543210)
contactapp("chandana",8796543217)
contactapp("chandana",89297642324)
contactapp("archana",9765421567)

alekhya  contact is added
chandana  contact is added
chandana  is already exists
archana  contact is added
```

Out[126]: {'alekhya': 9876543210, 'chandana': 8796543217, 'archana': 9765421567}

- search for the name ,if name is exists then print name and number,otherwise print does not exists
- update the number in your contacts

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```