int

float

string or char

list

tuple

set

dictionay ¶

tuple

tuple is immutable

() -->this is the declaration of tuple

In [1]:

```
names = ("siva","sai","ganga")
```

In [2]:

```
names
```

Out[2]:

```
('siva', 'sai', 'ganga')
```

In [3]:

```
names[0]
```

Out[3]:

```
'siva'
```

In [4]:

```
names[-1]
```

Out[4]:

```
'ganga'
```

In [5]:

```
names[0:2]
```

Out[5]:

```
('siva', 'sai')
```

In [6]:

```
dir(tuple)
```

Out[6]:

```
['__add__',
 '__class__',
 '__contains__',
 '__delattr__',
 '__dir__',
 '__doc__',
 '__eq__',
 '__format__',
 '__ge__',
 '__getattribute__',
 '__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']
```

In [7]:

```
dir(list)
```

Out[7]:

```
['__add__',
 '__class__',
 '__contains__',
 '__delattr__',
 '__delitem__',
 '__dir__',
 '__doc__',
 '__eq__',
 '__format__',
 '__ge__',
 '__getattribute__',
 '__getitem__',
 '__gt__',
 '__hash__',
 '__iadd__',
 '__imul__',
 '__init__',
 '__init_subclass__',
 '__iter__',
 '__le__',
 '__len__',
 '__lt__',
 '__mul__',
 '__ne__',
 '__new__',
 '__reduce__',
 '__reduce_ex__',
 '__repr__',
 '__reversed__',
 '__rmul__',
 '__setattr__',
 '__setitem__',
 '__sizeof__',
 '__str__',
 '__subclasshook__',
 'append',
 'clear',
 'copy',
 'count',
 'extend',
 'index',
 'insert',
 'pop',
 'remove',
 'reverse',
 'sort']
```

In [8]:

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

In [9]:

```
li.append(6)
```

In [10]:

```
li
```

Out[10]:

```
[1, 2, 3, 4, 5, 6]
```

In [11]:

```
p=(1,2,3,4)
```

In [12]:

```
p.add(6)
```

```
---------------------------------------------------------------------------
-
AttributeError                            Traceback (most recent call las
t)
<ipython-input-12-ef3bda61d0c7> in <module>
----> 1 p.add(6)

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

In [13]:

```
p[5]=10
```

```
---------------------------------------------------------------------------
-
TypeError                                 Traceback (most recent call las
t)
<ipython-input-13-4cbe88a2b0e9> in <module>
----> 1 p[5]=10

TypeError: 'tuple' object does not support item assignment
```

In [ ]:

```
tup = (100,200,300,200,100)

tup.count(200)
```

In [ ]:

```
## index
tup1=(1,2,3,5,7)

tup1.index(3)
```

In [ ]:

```
tup2=(1,2,3,4,3,5)

tup2.index(3)
```

In [14]:

```
tup2.index(3,3)
```

```
---------------------------------------------------------------------------
-
NameError                                 Traceback (most recent call las
t)
<ipython-input-14-51f4f0af445d> in <module>
----> 1 tup2.index(3,3)

NameError: name 'tup2' is not defined
```

len --> this is used to find the length of the tuple

In [ ]:

```
t = (1,"s",5.6,45)
```

In [ ]:

```
len(t)
```

set

it is used for store the multiple data types of values

it is declare with { }

it won't allow the duplicates , underordered list

In [ ]:

```
set1 = {1,2,2,3,4,5,4}

print(set1)
```

In [ ]:

```
dir(set)
```

In [ ]:

```
set1={1,2,3,5,7,11}

set1.add(13)
```

In [ ]:

```
set1
```

In [ ]:

```
set2={"odd","even",2,4,3,5,7.8}
set2.add(9.2)
print(set2)
```

In [ ]:

```
set3= {"a","b",[1,2,3],(5,6),10}
```

In [ ]:

```
set4 = {1,2,(3,4),6}
print(set4)
```

In [ ]:

```
## clear
print(set4,"before using the clear method")
set4.clear()
print(set4,"after using the clear method")
```

In [ ]:

```
num_list1 = {1,3,5,7,9,11}
num_list2 = {1,2,3,5,7,11}

# difference

num_list1.difference(num_list2)
```

In [ ]:

```
num_list2.difference(num_list1)
```

In [ ]:

```
# difference_update

print(num_list1,"num_list1")
print(num_list2,"num_list2")

print(num_list1.difference(num_list2),"output")

print(num_list1,"after num_list1")
print(num_list2,"after num_list2")
```

In [ ]:

```
print(num_list1,"num_list1")
print(num_list2,"num_list2")

print(num_list1.difference_update(num_list2),"output")

print(num_list1,"after num_list1")
print(num_list2,"after num_list2")
```

In [ ]:

```
num_list1.discard()
```

In [ ]:

```
num_list1
```

In [ ]:

```
num_list1.discard(9)
```

In [ ]:

```
num_list1
```

In [ ]:

```
## intersection

test1 = {"sankhar","kalyan","anil","ayyappa","srinu"}
test2 = {"siva","gopi","raj kumar","anil","sankhar"}

test1.intersection(test2)
```

In [ ]:

```
# intersection_update

test1.intersection_update(test2)
```

In [ ]:

```
test1
```

In [ ]:

```
t1 = {1,5,10,15,20}
t2 = {6,12,18,24,28}

t1.isdisjoint(t2)
```

In [ ]:

```
t3 = {1,2,3,5,7}
t4 = {2,4,6,8,10}

t3.isdisjoint(t4)
```

In [ ]:

```
## issubset

s1 = {"sairam","lakshmi"}
s2 = {"sai","ram","sairam","gopi","hari","lakshmi"}

s1.issubset(s2) -->true

s2.issubset(s1) --> false
```

In [ ]:

```
h1 = {1,2,5}
h2 = {1,2,4,6,7}

h1.issubset(h2)
```

In [ ]:

```
# issuperset

d1 = {"siva",1,12.5,41}
d2 = {"sai","sankhar",10,12.5,41}

d1.issuperset(d2)
```

In [ ]:

```
g1 = {2,4,6,8,10}
g2 = {2,8,10}

g1.issuperset(g2)
```

In [ ]:

```
# pop
print(g1) # g1={2,4,6,8,10}
g1.pop()
print(g1)
```

In [ ]:

```
g1.pop()
```

In [ ]:

```
g1
```

In [ ]:

```
g1.pop(8)
```

In [ ]:

```
# remove

weeks = {"mon","tue","wed","th","fri","sat"}

weeks.remove("sat")
```

In [ ]:

```
weeks
```

In [ ]:

```
# pop and remove both for deleting
# but in pop it removes any value
# but in remove method it removes the what we mention in method

weeks.remove('mon')
```

In [ ]:

```
weeks
```

In [ ]:

```
# symmetric_difference

odd={1,3,5,7,9,11,13,15}
prime={2,3,5,7,11,13,17}

odd.symmetric_difference(prime)
```

In [ ]:

```
#symmetric_difference_update
```

In [ ]:

```
team1 = {"hemanth","kumar","hari","giri"}
team2 = {"ganga","manga","gopi","krishna"}

# union

team_new = team1.union(team2)

team_new
```

In [ ]:

```
team_new.update("anil")

print(team_new)
```

In [ ]:

```
team_new.add("anil")
```

In [ ]:

```
team_new
```

In [ ]:

```
team_new.update(100)
team_new
```

In [ ]:

```
t1={1,5,6,"s"}
```

In [15]:

```
t1.update({1,2,3})
```

```
---------------------------------------------------------------------------
-
NameError                                 Traceback (most recent call las
t)
<ipython-input-15-5dd1a270aad4> in <module>
----> 1 t1.update({1,2,3})

NameError: name 't1' is not defined
```

In [16]:

```
t1
```

```
---------------------------------------------------------------------------
-
NameError                                 Traceback (most recent call las
t)
<ipython-input-16-5db19043943a> in <module>
----> 1 t1

NameError: name 't1' is not defined
```

In [ ]:

```
t1.update(5)
```

In [ ]:

```
t1.update({10})
```

In [17]:

```
t1
```

```
---------------------------------------------------------------------------
-
NameError                                 Traceback (most recent call las
t)
<ipython-input-17-5db19043943a> in <module>
----> 1 t1

NameError: name 't1' is not defined
```

dictionary

In [19]:

```
# 1. by using indexing
# Example
# p=[1,4,5,6]
# p[0]=10
```

In [20]:

```
# it is also used for strong the different data
# in this we can declare our own keys or index(list,tuple)
# in values stored in the form " key: value "
# {} --> declaration of dictionary
```

In [21]:

```
dict1 = {"name":"ranganayakulu","company":"Apssdc","job":"Multi skill"}

print(dict1)
```

```
{'name': 'ranganayakulu', 'company': 'Apssdc', 'job': 'Multi skill'}
```

In [22]:

```
# keys --> immutable
# values --> mutable

dict2={1:"one","second":2,3:"three"}

dict2
```

Out[22]:

```
{1: 'one', 'second': 2, 3: 'three'}
```

In [23]:

```
dict3 = {"names":["vijay","raja","gopal"],"company":["dell","google","hp"]}

dict3
```

Out[23]:

```
{'names': ['vijay', 'raja', 'gopal'], 'company': ['dell', 'google', 'hp']}
```

In [24]:

```
dir(dict)
```

Out[24]:

```
['__class__',
 '__contains__',
 '__delattr__',
 '__delitem__',
 '__dir__',
 '__doc__',
 '__eq__',
 '__format__',
 '__ge__',
 '__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']
```

In [25]:

```
dict2
```

Out[25]:

```
{1: 'one', 'second': 2, 3: 'three'}
```

In [26]:

```
dict3
```

Out[26]:

```
{'names': ['vijay', 'raja', 'gopal'], 'company': ['dell', 'google', 'hp']}
```

In [27]:

```
# how to get values in dict

dict3['names']
```

Out[27]:

```
['vijay', 'raja', 'gopal']
```

In [28]:

```
dict3['company']
```

Out[28]:

```
['dell', 'google', 'hp']
```

In [29]:

```
# keys()

dict3.keys()
```

Out[29]:

```
dict_keys(['names', 'company'])
```

In [30]:

```
dict3['names']
```

Out[30]:

```
['vijay', 'raja', 'gopal']
```

In [31]:

```
dict3['names'][1]
```

Out[31]:

```
'raja'
```

In [32]:

```
dict3['names'][0:2]
```

Out[32]:

```
['vijay', 'raja']
```

In [33]:

```
# values

dict3.values()
```

Out[33]:

```
dict_values([['vijay', 'raja', 'gopal'], ['dell', 'google', 'hp']])
```

In [34]:

```
# items()

dict3.items()
```

Out[34]:

```
dict_items([('names', ['vijay', 'raja', 'gopal']), ('company', ['dell', 'g
oogle', 'hp'])])
```

In [35]:

```
dict4 = dict3.copy()
```

In [36]:

```
dict4
```

Out[36]:

```
{'names': ['vijay', 'raja', 'gopal'], 'company': ['dell', 'google', 'hp']}
```

In [37]:

```
dict3
```

Out[37]:

```
{'names': ['vijay', 'raja', 'gopal'], 'company': ['dell', 'google', 'hp']}
```

In [38]:

```
dict4
```

Out[38]:

```
{'names': ['vijay', 'raja', 'gopal'], 'company': ['dell', 'google', 'hp']}
```

In [39]:

```
dict4["salary"]
```

```
---------------------------------------------------------------------------
-
KeyError                                  Traceback (most recent call las
t)
<ipython-input-39-8023a4f84563> in <module>
----> 1 dict4["salary"]

KeyError: 'salary'
```

In [40]:

```
dict4.setdefault('salary')
```

In [41]:

```
dict4
```

Out[41]:

```
{'names': ['vijay', 'raja', 'gopal'],
 'company': ['dell', 'google', 'hp'],
 'salary': None}
```

In [42]:

```
print(dict4['names'])
dict4.get('names')
```

```
['vijay', 'raja', 'gopal']
```

Out[42]:

```
['vijay', 'raja', 'gopal']
```

In [43]:

```
dict4['names']
```

Out[43]:

```
['vijay', 'raja', 'gopal']
```

In [44]:

```
for val in dict4.values():
    print(val)
```

```
['vijay', 'raja', 'gopal']
['dell', 'google', 'hp']
None
```

In [45]:

```
for key in dict4.keys():
    if key=="salary":
        #jhjhjk
    else:
        print(key)
```

```
  File "<ipython-input-45-1f45532e02b2>", line 4
    else:
       ^
IndentationError: expected an indented block
```

In [46]:

```
dict3
```

Out[46]:

```
{'names': ['vijay', 'raja', 'gopal'], 'company': ['dell', 'google', 'hp']}
```

In [47]:

```
dict4
```

Out[47]:

```
{'names': ['vijay', 'raja', 'gopal'],
 'company': ['dell', 'google', 'hp'],
 'salary': None}
```

In [48]:

```
dict4['salary']=120
```

In [49]:

```
dict4
```

Out[49]:

```
{'names': ['vijay', 'raja', 'gopal'],
 'company': ['dell', 'google', 'hp'],
 'salary': 120}
```

In [50]:

```
dict4['salary']=150
```

In [51]:

```
dict4
```

Out[51]:

```
{'names': ['vijay', 'raja', 'gopal'],
 'company': ['dell', 'google', 'hp'],
 'salary': 150}
```