

Name: Suryakant Upadhyay

PRN:- 20220802043

Div :- A1

* TUPLES *

In [3]:

Write a Python code to create a tuple with all integer values.

```
my_tuple=(5,12,13,14)
print(my_tuple)
```

(5, 12, 13, 14)

In [4]:

Write a Python code to create a tuple with all float values.

```
my_tuple=(1.2,4.5,6.7,8.9)
print(my_tuple)
```

(1.2, 4.5, 6.7, 8.9)

In [5]:

Write a Python code to create a tuple with all string values.

```
my_tuple=("Adnan","Anuj","Shivom")
print(my_tuple)
```

('Suryakant', 'Anuj', 'Shivom')

In [6]:

Write a Python code to create a tuple with Boolean values.

```
my_tuple=(True,False,False)
print(my_tuple)
```

(True, False, False)

In [8]:

Write a Python code to create a tuple with an integer, float, string and Boolean values.

```
my_tuple=(1,4.5,"Adnan",True)
print(my_tuple)
```

(1, 4.5, 'Suryakant', True)

In [9]:

Write a Python code to create an empty tuple.

```
my_tuple=()
print(my_tuple)
```

()

In [10]:

Write a Python code to create a tuple with numbers, notation without parenthesis.

```
my_tuple=1,2,5,8,9
print(my_tuple)
```

(1, 2, 5, 8, 9)

In [12]:

Write a Python code to create a tuple from an iterable object.

```
my_tuple=tuple([True,False])
print(my_tuple)
```

(True, False)

In [13]:

Write a Python code to demonstrate duplicates.

```
my_tuple=(5,12,45,12,4)
print(my_tuple)
```

(5, 12, 45, 12, 4)

In [18]:

```
# Write a Python code to cut first two items from a tuple.
```

```
my_tuple=("Red","Blue","Green","Yellow")
my_tuple[0:2]
```

Out[18]:

```
('Red', 'Blue')
```

In [22]:

```
# Write a Python code to cut second item from a tuple.
```

```
my_tuple=("Red","Blue","Green","Yellow")
my_tuple[1]
```

Out[22]:

```
'Blue'
```

In [23]:

```
# Write a Python code to cut second and third element from a tuple.
```

```
my_tuple=("Red","Blue","Green","Yellow")
my_tuple[1:3]
```

Out[23]:

```
('Blue', 'Green')
```

In [24]:

```
# Write a Python code to cut first three items from a tuple.
```

```
my_tuple=("Red","Blue","Green","Yellow")
my_tuple[0:3]
```

Out[24]:

```
('Red', 'Blue', 'Green')
```

In [28]:

```
# Write a Python code to create copy of original tuple.
```

```
my_tuple=("Red","Blue","Green","Yellow")
my_tuple1=my_tuple
print(my_tuple1)
```

```
('Red', 'Blue', 'Green', 'Yellow')
```

In [33]:

```
# Write a Python code to add an element in a tuple.
```

```
my_tuple=(1,2,3,4,5)
my_list=list(my_tuple)
my_list.append(30)
my_tuple=tuple(my_list)
print(my_tuple)
```

```
(1, 2, 3, 4, 5, 30)
```

In [37]:

```
# Write a Python code to add an element and add a tuple to the existing tuple.
```

```
my_tuple=(4,6,8,2,3)
my_tuple1=my_tuple + (9,)
print(my_tuple1)
```

```
(4, 6, 8, 2, 3, 9)
```

In [39]:

```
# Write a Python code to remove an item from a tuple.
```

```
my_tuple=("Apple","Banana","Cherry")
my_list=list(my_tuple)
my_list.remove("Banana")
my_tuple=tuple(my_list)
print(my_tuple)
```

```
('Apple', 'Cherry')
```

In [45]:

```
# Write a Python code to delete the tuple completely.
```

```
my_tuple=("Apple","Banana","Cherry")
del(my_tuple)
print(my_tuple)
```

NameError Traceback (most recent call last)

```
Input In [45], in <cell line: 5>()
      3 my_tuple=("Apple","Banana","Cherry")
      4 del(my_tuple)
----> 5 print(my_tuple)
```

NameError: name 'my_tuple' is not defined

In [46]:

```
# Write a Python code to join two or more tuples.
```

```
my_tuple=("Adnan","Anuj","Shivom")
my_tuple1=(1,2,3,4)
my_tuple2=my_tuple1+my_tuple
print(my_tuple2)
```

(1, 2, 3, 4, 'Suryakant', 'Anuj', 'Shivom')

In [47]:

```
# Write a Python code to multiply a tuple.
```

```
my_tuple=("Adnan","Anuj","Shivom")
(my_tuple*2)
```

Out[47]:

('Suryakant', 'Anuj', 'Shivom', 'Suryakant', 'Anuj', 'Shivom')

PYTHON TUPLES - BUILT IN METHODS

In [48]:

```
# Write a Python code to demonstrate index() method.
```

```
my_tuple=(1,4,5,6,8)
my_tuple.index(5)
```

Out[48]:

2

In [49]:

```
# Write a Python code to demonstrate count() method.
```

```
my_tuple=(4,55,62,55,48)
my_tuple.count(55)
```

Out[49]:

2

TUPLE IN BUILT METHODS/FUNCTIONS

In [51]:

```
# Write a Python code to returns the index of the first element with the specified value.
```

```
my_tuple=("Adnan","Anuj","Shivom")
my_tuple.index("Anuj")
```

Out[51]:

1

In [52]:

```
# Write a Python code to return the number of elements with the specified value.
```

```
my_tuple=("Adnan","Anuj","Shivom","Adnan")
my_tuple.count("Adnan")
```

Out[52]:

2

In [53]:

```
# Various Python function also works with Tuple Data Structure
```

SET-(SET BASICS)

In [54]:

```
# Write a Python code to create set with integer values.
```

```
my_set={0,1,5,6,8}  
print(my_set)
```

```
{0, 1, 5, 6, 8}
```

In [55]:

```
# Write a Python code to create set with duplicate integer values.
```

```
my_set={1,2,3,2,4,5,2}  
print(my_set)
```

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

In [64]:

```
# Write a Python code to create an empty set using built in set() function.
```

```
my_set={}  
set()
```

Out[64]:

```
set()
```

In [67]:

```
# Write a Python code to create a non-empty set of integers using built in set() function.
```

```
my_set={0,5,4,55,5,6}  
set(my_set)
```

Out[67]:

```
{0, 4, 5, 6, 55}
```

In [68]:

```
# Write a Python code to create a non-empty set of all float values using built in set() function.
```

```
my_set={5.5,4.6,12.8,76.6}  
set(my_set)
```

Out[68]:

```
{4.6, 5.5, 12.8, 76.6}
```

In [69]:

```
# Write a Python code to create a non-empty set of all string values using built in set() function.
```

```
my_set={"Adnan","Anuj","Shivom"}  
set(my_set)
```

Out[69]:

```
{'Suryakant', 'Anuj', 'Shivom'}
```

In [70]:

```
# Write a Python code to create a non-empty set of all Boolean values using built in set() function.
```

```
my_set={True,False,True}  
set(my_set)
```

Out[70]:

```
{False, True}
```

In [54]:

```
# Write a Python code to demonstrate that Python Set cannot have mutable items.
```

```
my_set={4,5,6,[3,5]}
print(my_set)
```

```
-----
TypeError                                Traceback (most recent call last)
Input In [74], in <cell line: 3>()
      1 # Write a Python code to demonstrate that Python Set cannot have mutable items.
----> 3 my_set={4,5,6,[3,5]}
      4 print(my_set)
```

TypeError: unhashable type: 'list'

SET BUILT IN METHODS/FUNCTIONS

In [77]:

```
# Write a Python code to add an element to the set.
```

```
my_set={"Adnan","Anuj","Shivom"}
my_set.add("Aniruddha")
print(my_set)
```

```
{'Shivom', 'Aniruddha', 'Anuj', 'Suryakant'}
```

In [79]:

```
# Write a Python code to remove all the elements from the set.
```

```
my_set={"Adnan","Anuj","Shivom"}
my_set.clear()
print(my_set)
```

```
set()
```

In [81]:

```
# Write a Python code to returns a copy of the set.
```

```
my_set={"Adnan","Anuj","Shivom"}
my_set.copy()
print(my_set)
```

```
{'Shivom', 'Anuj', 'Suryakant'}
```

In [83]:

```
# Write a Python code to perform difference operation on set.
```

```
my_set={"Adnan","Anuj","Shivom"}
my_set1={"Google","Microsoft","Anuj"}
my_set2=my_set.difference(my_set1)
print(my_set2)
```

```
{'Shivom', 'Suryakant'}
```

In [85]:

```
# Write a Python code to perform difference_update operation on set.
```

```
my_set={"Adnan","Anuj","Shivom"}
my_set1={"Google","Microsoft","Anuj"}
my_set.difference_update(my_set1)
print(my_set)
```

```
{'Shivom', 'Suryakant'}
```

In [87]:

```
# Write a Python code to remove specified item.
```

```
my_set={"Adnan","Anuj","Shivom"}
my_set.discard("Anuj")
print(my_set)
```

```
{'Shivom', 'Suryakant'}
```

In [89]:

```
# Write a Python code to perform intersection operation on set.
```

```
my_set={"Adnan","Anuj","Shivom"}
my_set1={"Google","Microsoft","Anuj"}
my_set2=my_set.intersection(my_set1)
print(my_set2)
```

```
{'Anuj'}
```

In [91]:

```
# Write a Python code to removes the items in this set that are not present in other, specified set(s).
```

```
my_set={"Adnan","Anuj","Shivom"}
my_set1={"Google","Microsoft","Anuj"}
my_set.isdisjoint(my_set1)
```

Out[91]:

```
False
```

In [92]:

```
# Write a Python code to demonstrate issubset() method.
```

```
my_set={"Adnan","Anuj","Shivom"}
my_set1={"Pratik","Aniruddha","Anuj","Adnan","Shivom"}
my_set.issubset(my_set1)
```

Out[92]:

```
True
```

In [94]:

```
# Write a Python code to demonstrate issuperset() method.
```

```
my_set={"Pratik","Aniruddha","Anuj","Adnan","Shivom"}
my_set1={"Adnan","Anuj","Shivom"}
my_set.issuperset(my_set1)
```

Out[94]:

```
True
```

In [98]:

```
# Write a Python code to demonstrate pop() method.
```

```
my_set={"Adnan","Anuj","Shivom"}
my_set.pop()
print(my_set)
```

```
{'Anuj', 'Suryakant'}
```

In [107]:

```
# Write a Python code to demonstrate symmetric_difference() method.
```

```
my_set={"Adnan","Anuj","Shivom"}
my_set1={"Google","Microsoft","Anuj"}
my_set.symmetric_difference(my_set1)
```

Out[107]:

```
{'Suryakant', 'Google', 'Microsoft', 'Shivom'}
```

In [103]:

```
# Write a Python code to demonstrate symmetric_difference_update() method.
```

```
my_set={"Adnan","Anuj","Shivom"}
my_set1={"Google","Microsoft","Anuj"}
my_set.symmetric_difference_update(my_set1)
print(my_set)
```

```
{'Microsoft', 'Google', 'Suryakant', 'Shivom'}
```

In [104]:

```
# Write a Python code to demonstrate union() method.
```

```
my_set={"Adnan","Anuj","Shivom"}
my_set1={"Google","Microsoft","Anuj"}
my_set.union(my_set1)
```

Out[104]:

```
{'Suryakant', 'Anuj', 'Google', 'Microsoft', 'Shivom'}
```

In [106]:

```
# Write a Python code to demonstrate update() method.
```

```
my_set={"Adnan","Anuj","Shivom"}
my_set1={"Google","Microsoft","Anuj"}
my_set.update(my_set1)
print(my_set)

{'Shivom', 'Anuj', 'Microsoft', 'Google', 'Suryakant'}
```

DICTIONARY BASICS

In [108]:

```
# Write a Python code to create an empty dictionary.
```

```
dict()

Out[108]:
{}

In [111]:
```

```
# Write a Python code to create non-empty dictionary.
```

```
my_dict={"Suzuki":"Baleno","Renault":"Triber","Tata":"Safari"}
print(my_dict)

{'Suzuki': 'Baleno', 'Renault': 'Triber', 'Tata': 'Safari'}
```

In [116]:

```
# Write a Python code to access elements of dictionary using key.
```

```
my_dict={"Suzuki":"Baleno","Renault":"Triber","Tata":"Safari"}
my_dict["Renault"]

Out[116]:
'Triber'

In [119]:
```

```
# Write a Python code to access elements of dictionary using get method.
```

```
my_dict={"Suzuki":"Baleno","Renault":"Triber","Tata":"Safari"}
my_dict.get("Suzuki")

Out[119]:
'Baleno'

In [121]:
```

```
# Write a Python code to add an element to the dictionary.
```

```
my_dict={"Suzuki":"Baleno","Renault":"Triber","Tata":"Safari"}
my_dict["Volkswagen"]="Polo"
print(my_dict)

{'Suzuki': 'Baleno', 'Renault': 'Triber', 'Tata': 'Safari', 'Volkswagen': 'Polo'}
```

In [133]:

```
# Write a Python code to add an element to the dictionary using update method.
```

```
my_dict={"Suzuki":"Baleno","Renault":"Triber","Tata":"Safari"}
my_dict.update({"Volkswagen":"Polo"})
print(my_dict)

{'Suzuki': 'Baleno', 'Renault': 'Triber', 'Tata': 'Safari', 'Volkswagen': 'Polo'}
```

In [134]:

```
# Write a Python code to iterate through using dict.items() method.
```

```
my_dict={"Suzuki":"Baleno","Renault":"Triber","Tata":"Safari"}
my_dict.items()

Out[134]:
dict_items([('Suzuki', 'Baleno'), ('Renault', 'Triber'), ('Tata', 'Safari')])
```

In [140]:

```
# Write a Python code to iterate through using keys() method.
```

```
my_dict={"Suzuki":"Baleno","Renault":"Triber","Tata":"Safari"}  
my_dict.keys()
```

Out[140]:

```
dict_keys(['Suzuki', 'Renault', 'Tata'])
```

In [141]:

```
# Write a Python code to iterate through using values() method.
```

```
my_dict={"Suzuki":"Baleno","Renault":"Triber","Tata":"Safari"}  
my_dict.values()
```

Out[141]:

```
dict_values(['Baleno', 'Triber', 'Safari'])
```

In [143]:

```
# Write a Python code to remove all items from the dictionary.
```

```
my_dict={"Suzuki":"Baleno","Renault":"Triber","Tata":"Safari"}  
my_dict.clear()  
print(my_dict)
```

```
{}
```

In [144]:

```
# Write a Python code to copy all the elements from the dictionary.
```

```
my_dict={"Suzuki":"Baleno","Renault":"Triber","Tata":"Safari"}  
my_dict.copy()
```

Out[144]:

```
{'Suzuki': 'Baleno', 'Renault': 'Triber', 'Tata': 'Safari'}
```

In []: