In [1]:
```
my_tuple2 = (1, 2, 3, 'SYBTECH')
for x in my_tuple2:
 print(x)
print(my_tuple2)
print(my_tuple2[0])
print(my_tuple2[:])
print(my_tuple2[3][4])
```

```
1
2
3
SYBTECH
(1, 2, 3, 'SYBTECH')
1
(1, 2, 3, 'SYBTECH')
E
```

In [3]:
```
my_tuple = (1,2,3)
my_tuple = my_tuple + (4,5,6)
print(my_tuple)
my_tuple = (1,2,3, ['hindi','python'
])
my_tuple[3][0] = 'english'
print(my_tuple)
print(my_tuple.count(2))
print(my_tuple.index(['english','python']))
```

```
(1, 2, 3, 4, 5, 6)
(1, 2, 3, ['english', 'python'])
1
3
```

In [4]:
```
my_dict =()
print(my_dict)
my_dict = {1:'Python', 2:'Java'}
print(my_dict)
```

```
()
{1: 'Python', 2: 'Java'}
```

In [6]:
```
my_dict = {'First': 'Python', 'Second': 'Java'}
print(my_dict)
my_dict['Second'] = 'C++'
print(my_dict)
my_dict['Third'] = 'Ruby'
print(my_dict)
```

```
{'First': 'Python', 'Second': 'Java'}
{'First': 'Python', 'Second': 'C++'}
{'First': 'Python', 'Second': 'C++', 'Third': 'Ruby'}
```

In [8]:
```
Dict = {'Name':['Riya','Pranav','Vaishali'],1:[1,2,3,4]}
Dict.values()
```

Out[8]:
```
dict_values([['Riya', 'Pranav', 'Vaishali'], [1, 2, 3, 4]])
```

In [9]:
```
my_dict = {'First':'Python','Second':
'Java', 'Third':'Ruby'}
a=my_dict.pop('Third')
print('Value:',a)
print('Dictionary:', my_dict)
b= my_dict.popitem()
print('Key, value pair:', b)
print('Dictionary', my_dict)
my_dict.clear()
print('n', my_dict)
```

```
Value: Ruby
Dictionary: {'First': 'Python', 'Second': 'Java'}
Key, value pair: ('Second', 'Java')
Dictionary {'First': 'Python'}
n {}
```

In [ ]:

Devanshu Surana
1032210755
PC-12
Batch-C1

MIT-WPU | MIT WORLD PEACE UNIVERSITY | PUNE

Python Programming
Assignment - 5

Problem statement: Different operations on dicitionary and tuple data structure.

Aim: write a python program to create, append and remove etc. operation on Dictionary and Tuple

Objectives: To learn and implement Dicitionary and Tuple data structure.

Theory:
Different operations performed on:
- Dictionary
- clear() - Remove all items from dictionary
- copy() - Returns a shallow copy of dictionary
- get() - Returns the value for given key
- items() - Returns the list with all dictionary keys with values.
- pop() - Returns and removes the element with given key.
- pop item() = Returns and removes the key-value pair from dictionary.
- updates() - updates the dictionary with the elements from another dictionary.

- Tuples:
  - index() - find in the tuple and returns the index of the given value where its available
  - count() - Returns the frequency of occurence of a specified value.
  - len() - Returns length of the tuple or size of tuple.

Platform: Windows - Python Editor (Jupyter)

Conclusion: Studied python dicitionary and Tuple data structure.

FAQ's

Ans 1) output: [1,3,5,7,9]

Ans 2) output: ()

Ans 3)

Lists	Dictionary
• List is a collection of index value pairs.	• Dictionary is a hashed structure of key and value pairs.
• List is created by placing elements in [] separated by commas ",".	• Dictionary is created by placing elements in {} as "key":"value", each key value pair is separated by commas ",".
• The indices of lists are integers starting from 0.	• The keys of dictionary can be of any data type.
• The elements are accessed via indices.	• The elements are accessed via Key-values.
• order of elements are maintained	• No-guarantee for maintaining order.

Ans 4) Python has tuple assignment feature which enables one to assign more than one variable at a time. In packing, we place value into a new tuple while in unpacking we extract those values back into variables.