In [1]:

```
Agenda = '''

1. Case Study:
Dictionary Case Study

2. Built in Functions
a) lambda
b) filter
c) map
d) sum
e) max
f) min
g) set
h) reversed
i) zip
j) sorted
k) enumerate

3. List Comprehension

4. Dictionary Comprehension

# DAY 3

5. Module
a) How to import ?
b) Built in math
c) Create User Define Module
d) os
e) sys
f) pickle / unpickle ( Object Serialisation )
g) glob

Group Discussion and Quiz

'''


```

In [3]:

```
print(Agenda)
```

```
1. Case Study:
Dictionary Case Study

2. Built in Functions
a) lambda
b) filter
c) map
d) sum
e) max
f) min
g) set
h) reversed
i) zip
j) sorted
k) enumerate

3. List Comprehension
4. Dictionary Comprehension
5. Module
a) How to import ?
b) Built in math
c) Create User Define Module
d) os
e) sys
f) pickle / unpickle ( Object Serialisation )
g) glob

Group Discussion and Quiz
```

In [4]:

```
print('''Dictionay is an unorder collection of unique,
immutable key and value pairs. ''')
```

```
Dictionay is an unorder collection of unique,
immutable key and value pairs.
```

In [5]:

```
# Create Dictionary

Dictionary1 = {}

print(type(Dictionary1))

Dictionary2 = dict()

print(type(Dictionary2))

```

```
<class 'dict'>
<class 'dict'>
```

In [6]:

```
# Update / PUT / POST Dictionary

Dictionary1['first_name'] = 'Surendra'

Dictionary1['last_name'] = 'Panpaliya'

Dictionary1['city'] = 'Pune'

Dictionary1['Age'] = 42

Dictionary1['Email'] = 'surendra@gktcs.com'

print(Dictionary1)

```

```
{'first_name': 'Surendra', 'last_name': 'Panpaliya', 'city': 'Pune',
'Age': 42, 'Email': 'surendra@gktcs.com'}
```

In [17]:

```
# Read / Access / GET Dictionary1

print(Dictionary1.get('first_name'))

print("\n")

print(Dictionary1['last_name'])

print("\n")

# List Dictionary keys

print(Dictionary1.keys())

print("\n")

# List Dictionary Values

print(Dictionary1.values())

print("\n")

# List Dictionary items

print(Dictionary1.items())

print("\n")
```

```
Surendra


Panpaliya


dict_keys(['first_name', 'last_name', 'city', 'Age', 'Email'])


dict_values(['Surendra', 'Panpaliya', 'Pune', 42, 'surendra@gktcs.co
m'])


dict_items([('first_name', 'Surendra'), ('last_name', 'Panpaliya'),
('city', 'Pune'), ('Age', 42), ('Email', 'surendra@gktcs.com')])
```

In [18]:

```
1 dir(Dictionary1)
```

Out[18]:

```
['__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 [19]:

```
# Delete Dictionary items

Dictionary1.keys()
```

Out[19]:

```
dict_keys(['first_name', 'last_name', 'city', 'Age', 'Email'])
```

In [20]:

```
help(Dictionary1.popitem)
```

```
Help on built-in function popitem:

popitem(...) method of builtins.dict instance
    D.popitem() -> (k, v), remove and return some (key, value) pair
as a
    2-tuple; but raise KeyError if D is empty.
```

In [21]:

```
help(Dictionary1.pop)
```

```
Help on built-in function pop:

pop(...) method of builtins.dict instance
    D.pop(k[,d]) -> v, remove specified key and return the correspon
ding value.
    If key is not found, d is returned if given, otherwise KeyError
is raised
```

In [22]:

```
Dictionary1.pop('Age')
```

Out[22]:

```
42
```

In [23]:

```
Dictionary1
```

Out[23]:

```
{'first_name': 'Surendra',
 'last_name': 'Panpaliya',
 'city': 'Pune',
 'Email': 'surendra@gktcs.com'}
```

In [24]:

```
Dictionary1.popitem()
```

Out[24]:

```
('Email', 'surendra@gktcs.com')
```

In [25]:

```
Dictionary1
```

Out[25]:

```
{'first_name': 'Surendra', 'last_name': 'Panpaliya', 'city': 'Pune'}
```

In [33]:

```
print('''Books Case Study

Scenarios:

1. Create Publisher Dictionary, which include

3 publishers data.


2. Write Publisher fields as follows:

a) publisher_name
b) publisher_location
c) publisher_email
d) publisher_contact

3. Create Author Dictionary, which include

3 Authors data.

```

```
4. Write Author fields as follows:

a) author_first_name
b) author_last_name
b) author_location
c) author_email
d) author_contact

5. Write Book fields as follows:

a) book_title
b) book_publisher ( One to One Relation)
c) book_authors  ( One to Many Relations)
d) book_pub_date
e) book_web

6. Fetch Second Author first name from Books
Dictionary.

7.Update Last Name of Second Author using Books

Dictionary Only.

8. Books Authors field should refer to 2 authors
from author dictionary. Note You can use list to
provide authors list to book_authors field.

9. Fetch Publisher Name from books and
Update it.

10. Create Books Menu

Welcome to Books Library Creation.
Select following option:
1. Publisher Creation
2. Author Creation
3. Add New Publisher
4. Add New Author
5. Book Creation
6. Add New Book
7. Display Publishers
8. Display Authors
9. Display Books
10. Display Books Title
11. Exit Library
'''



''')
```

Books Case Study

Scenarios:

1. Create Publisher Dictionary, which include

3 publishers data.

2. Write Publisher fields as follows:

a) publisher_name
b) publisher_location
c) publisher_email
d) publisher_contact

3. Create Author Dictionary, which include

3 Authors data.

4. Write Author fields as follows:

a) author_first_name
b) author_last_name
b) author_location
c) author_email
d) author_contact

5. Write Book fields as follows:

a) book_title
b) book_publisher (One to One Relation)
c) book_authors (One to Many Relations)
d) book_pub_date
e) book_web

6. Fetch Second Author first name from Books
Dictionary.

7.Update Last Name of Second Author using Books

Dictionary Only.

8. Books Authors field should refer to 2 authors
from author dictionary. Note You can use list to
provide authors list to book_authors field.

9. Fetch Publisher Name from books and
Update it.

10.

In [32]:

```
1
2 Pub1={'pid':1,'pname':'Orielly','pcity':'Bangalore',
3       'pweb':'www.orielly.org'}
```

```
Pub2={'pid':2,'pname':'Tata','pcity':'Chennai',
      'pweb':'www.tata.org'}
Pub3={'pid':3,'pname':'willey','pcity':'Delhi',
      'pweb':'www.willey.org'}
Pub4={'pid':4,'pname':'bpb','pcity':'Pune',
      'pweb':'www.bpb.org'}

Publishers={'P1':Pub1,'P2':Pub2,'P3':Pub3}

Publishers['P4']=Pub4

print("\n"*2)

print(Publishers)

print("\n"*2)

Auth1={'id':1,'fname':'Surendra','lname':'Panpaliya',
       'city':'Pune'}
Auth2={'id':2,'fname':'Satish','lname':'Panpaliya',
       'city':'Bangalore'}

Authors={'A1':Auth1,'A2':Auth2}

Authors['A3']={'id':3,'fname':'Suresh','lname':'Patil','city':'Bangalore'}

print(Authors)

Book1={'title':'Python Cookbook',
       'publisher':Publishers['P2'],
      'Authors':[Authors['A1'],Authors['A2']],
      'pdate':'2019-03-20'}

Book2={'title':'Ruby Cookbook',
       'publisher':Publishers['P1'],
      'Authors':[Authors['A2'],Authors['A3']],
       'pdate':'2019-02-18'}

Book3={'title':'Perl Cookbook',
       'publisher':Publishers['P3'],
      'Authors':[Authors['A1'],Authors['A3']],
       'pdate':'2019-03-17'}

Lib={'B1':Book1,'B2':Book2,'B3':Book3}

def create_publisher():
    '''Create Publisher'''
    pid = int(input("Enter Publisher id:"))
    pname = input("Enter Publisher Name: ")
    pcity = input("Enter Publisher Name: ")
    pweb = input("Enter Publisher Name: ")
    Pub={'pid':pid,'pname':pname,'pcity': pcity,
         'pweb': pweb }

```

{'P1': {'pid': 1, 'pname': 'Orielly', 'pcity': 'Bangalore', 'pweb':
'www.orielly.org'}, 'P2': {'pid': 2, 'pname': 'Tata', 'pcity': 'Chen
nai', 'pweb': 'www.tata.org'}, 'P3': {'pid': 3, 'pname': 'willey', '
pcity': 'Delhi', 'pweb': 'www.willey.org'}, 'P4': {'pid': 4, 'pname'
: 'bpb', 'pcity': 'Pune', 'pweb': 'www.bpb.org'}}

{'A1': {'id': 1, 'fname': 'Surendra', 'lname': 'Panpaliya', 'city':
'Pune'}, 'A2': {'id': 2, 'fname': 'Satish', 'lname': 'Panpaliya', 'c
ity': 'Bangalore'}, 'A3': {'id': 3, 'fname': 'Suresh', 'lname': 'Pat
il', 'city': 'Bangalore'}}

Out[32]:

{'B1': {'title': 'Python Cookbook',
  'publisher': {'pid': 2,
   'pname': 'Tata',
   'pcity': 'Chennai',
   'pweb': 'www.tata.org'},
  'Authors': [{'id': 1,
    'fname': 'Surendra',
    'lname': 'Panpaliya',
    'city': 'Pune'},
   {'id': 2, 'fname': 'Satish', 'lname': 'Panpaliya', 'city': 'Banga
lore'}],
  'pdate': '2019-03-20'},
 'B2': {'title': 'Ruby Cookbook',
  'publisher': {'pid': 1,
   'pname': 'Orielly',
   'pcity': 'Bangalore',
   'pweb': 'www.orielly.org'},
  'Authors': [{'id': 2,
    'fname': 'Satish',
    'lname': 'Panpaliya',
    'city': 'Bangalore'},
   {'id': 3, 'fname': 'Suresh', 'lname': 'Patil', 'city': 'Bangalore
'}],
  'pdate': '2019-02-18'},
 'B3': {'title': 'Perl Cookbook',
  'publisher': {'pid': 3,
   'pname': 'willey',
   'pcity': 'Delhi',
   'pweb': 'www.willey.org'},
  'Authors': [{'id': 1,
    'fname': 'Surendra',
    'lname': 'Panpaliya',
    'city': 'Pune'},
   {'id': 3, 'fname': 'Suresh', 'lname': 'Patil', 'city': 'Bangalore
'}],
  'pdate': '2019-03-17'}}

In [34]:

```
1 Menu='''
```

```
Welcome to Books Library Creation.
Select following option:
1. Publisher Creation
2. Author Creation
3. Add New Publisher
4. Add New Author
5. Book Creation
6. Add New Book
7. Display Publishers
8. Display Authors
9. Display Books
10. Display Books Title
11. Exit Library
'''


print(Menu)

while True:
    op = int(input("Enter option: "))
    if op == 1:
        print('Publisher Creation:')

    elif op == 2:
        print("Author Creation:")

    elif op == 3:
        print("Add New Publisher:")

    elif op == 4:
        print("Add New Author:")

    elif op == 5:
        print("Book Creation:")

    elif op == 6:
        print("Add New Book:")

    elif op == 7:
        print("Display Publishers:")

    elif op == 8:
        print("Display Authors:")

    elif op == 9:
        print("Display Books:")

    elif op == 10:
        print("Display Books Title:")

    elif op == 11:
        print("Thanks for using Book Library!!")
        break

    else:
        print("Enter option from [1..11] Only")

```

```
Welcome to Books Library Creation.
Select following option:
1. Publisher Creation
2. Author Creation
3. Add New Publisher
4. Add New Author
5. Book Creation
6. Add New Book
7. Display Publishers
8. Display Authors
9. Display Books
10. Display Books Title
11. Exit Library

Enter option: 11
Thanks for using Book Library!!
```

In [43]:

```python
def create_publisher():
    '''Create Publisher'''
    global Pub
    Pub = {}
    Pub['pid'] = int(input("Enter Publisher id:"))
    Pub['pname'] = input("Enter Publisher Name: ")
    Pub['pcity'] = input("Enter Publisher city: ")
    Pub['pweb'] = input("Enter Publisher Web: ")
    return Pub
```

In [44]:

```python
print(create_publisher())
```

```
Enter Publisher id:2
Enter Publisher Name: sure
Enter Publisher city: per
Enter Publisher Web: dfdf
{'pid': 2, 'pname': 'sure', 'pcity': 'per', 'pweb': 'dfdf'}
```

In [45]:

```
def display_dict():
    '''Display Dictionary'''
    return Pub

display_dict()
```

Out[45]:

```
{'pid': 2, 'pname': 'sure', 'pcity': 'per', 'pweb': 'dfdf'}
```

In [50]:

```
def create_publisher():
    '''Create Publisher'''
    global Pub
    Pub = {}
    Pub['pid'] = int(input("Enter Publisher id:"))
    Pub['pname'] = input("Enter Publisher Name: ")
    Pub['pcity'] = input("Enter Publisher city: ")
    Pub['pweb'] = input("Enter Publisher Web: ")
    return Pub

Publishers = {}
print("Enter Publisher1 Details:")
Publishers['P1'] = create_publisher()
print("Enter Publisher2 Details:")
Publishers['P2'] = create_publisher()
print("Enter Publisher3 Details:")
Publishers['P3'] = create_publisher()
print("Publishers Dictionary is")
print(Publishers)
```

```
Enter Publisher id:1
Enter Publisher Name: surendra
Enter Publisher city: pune
Enter Publisher Web: www.gktcs.com
Enter Publisher id:2
Enter Publisher Name: narendra
Enter Publisher city: blr
Enter Publisher Web: www.dsd
Enter Publisher id:3
Enter Publisher Name: sdsd
Enter Publisher city: sd
Enter Publisher Web: sdsds
{'P1': {'pid': 1, 'pname': 'surendra', 'pcity': 'pune', 'pweb': 'www
.gktcs.com'}, 'P2': {'pid': 2, 'pname': 'narendra', 'pcity': 'blr',
'pweb': 'www.dsd'}, 'P3': {'pid': 3, 'pname': 'sdsd', 'pcity': 'sd',
'pweb': 'sdsds'}}
```

In [49]:

```
update_dictionary()
```

```
Enter Publisher id:4
Enter Publisher Name: proe
Enter Publisher city: sdssd
Enter Publisher Web: sdsd
```

Out[49]:

```
{'P2': {'pid': 4, 'pname': 'proe', 'pcity': 'sdssd', 'pweb': 'sdsd'}
}
```

In [51]:

```
Built_in_Functions = '''
a) lambda
b) filter
c) map
d) sum
e) max
f) min
g) set
h) reversed
i) zip
j) sorted
k) enumerate '''
```

In [52]:

```
'''lambda is an annonymus single line function'''
```

Out[52]:

```
'lambda is an annonymus single line function'
```

In [53]:

```
add = lambda num1, num2 : num1 + num2
```

In [54]:

```
1 add
```

Out[54]:

```
<function __main__.<lambda>(num1, num2)>
```

In [55]:

```
1 add(34,12)
```

Out[55]:

```
46
```

In [59]:

```
1 hello = lambda : 'Hello Python'
```

In [60]:

```
1 hello()
```

Out[60]:

```
'Hello Python'
```

In [61]:

```
1 # filter('function','Sequence')
```

In [73]:

```
1 seq = list(range(2,25))
2 print(seq)
```

```
[2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20,
21, 22, 23, 24]
```

In [63]:

```
func1 = lambda num : num % 2 != 0 and num % 3 != 0
```

In [65]:

```
func1(5)
```

Out[65]:

```
True
```

In [67]:

```
list(filter(func1, seq))
```

Out[67]:

```
[5, 7, 11, 13, 17, 19, 23]
```

In [82]:

```
list(filter(lambda num : num % 2 != 0 and num % 3 != 0,
            seq))
```

Out[82]:

```
[5, 7, 11, 13, 17, 19, 23]
```

In [70]:

```
fdata = float(5)
```

In [71]:

```
fdata
```

Out[71]:

```
5.0
```

In [75]:

```
books = [ 'cbook', 'cppbook','javabook']

'cbook' in books
```

Out[75]:

```
True
```

In [76]:

```
'perl' in books
```

Out[76]:

```
False
```

In [81]:

```
Dictionary1.__contains__('first_name2')
```

Out[81]:

```
False
```

In [79]:

```
Dictionary1
```

Out[79]:

```
{'first_name': 'Surendra', 'last_name': 'Panpaliya', 'city': 'Pune'}
```

In [84]:

```
fun2 = lambda book : book in books
```

In [85]:

```
fun2('python')
```

Out[85]:

```
False
```

In [86]:

```
books
```

Out[86]:

```
['cbook', 'cppbook', 'javabook']
```

In [87]:

```
fun2('cbook')
```

Out[87]:

```
True
```

In [89]:

```
Lib = ['java','cpp','perl','cbook',
       'cppbook', 'javabook']
```

In [90]:

```
fun2 = lambda book : book in Lib
fun2('perl')
```

Out[90]:

```
True
```

In [92]:

```
list(filter(fun2, Lib))
```

Out[92]:

```
['java', 'cpp', 'perl', 'cbook', 'cppbook', 'javabook']
```

In [93]:

```
list(map(len, ['surendra','narendra',
               'satish','panpaliya']))
```

Out[93]:

```
[8, 8, 6, 9]
```

In [94]:

```
seq1 = list(range(1,10))
print(seq1)
```

```
[1, 2, 3, 4, 5, 6, 7, 8, 9]
```

In [95]:

```
seq2 = list(range(10,20))
print(seq2)
```

```
[10, 11, 12, 13, 14, 15, 16, 17, 18, 19]
```

In [96]:

```
print(seq1 + seq2)
```

```
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19]
```

In [98]:

```
1 list(map(lambda num1, num2 : num1 + num2, seq1,seq2))
```

Out[98]:

```
[11, 13, 15, 17, 19, 21, 23, 25, 27]
```

In [99]:

```
1 print(Agenda)
```

```
1. Case Study:
Dictionary Case Study

2. Built in Functions
a) lambda
b) filter
c) map
d) sum
e) max
f) min
g) set
h) reversed
i) zip
j) sorted
k) enumerate

3. List Comprehension
4. Dictionary Comprehension
5. Module
a) How to import ?
b) Built in math
c) Create User Define Module
d) os
e) sys
f) pickle / unpickle ( Object Serialisation )
g) glob

Group Discussion and Quiz
```

In [100]:

```
sum([3,4,6,8,9])
```

Out[100]:

30

In [101]:

```
sum(seq1)
```

Out[101]:

45

In [102]:

```
max([3,4,6,8,9])
```

Out[102]:

9

In [103]:

```
min([3,4,6,8,9])
```

Out[103]:

3

In [104]:

```
print('''set is unorder collection of unique
immutable objects''')
```

```
set is unorder collection of unique
immutable objects
```

In [105]:

```
# Set Create Operation

book_list1 = ['python','perl','php','python','perl']

book_list2 = ['python','perl','java','jython',
              'perl','java']



book_set1 = set(book_list1)
print(book_set1)


book_set2 = set(book_list2)
print(book_set2)

```

```
{'perl', 'python', 'php'}
{'perl', 'jython', 'java', 'python'}
```

In [107]:

```
# Union of set

book_union_set = book_set1.union(book_set2)

print(book_union_set)

book_union_set1 = book_set1 | book_set2

print(book_union_set1)
```

```
{'jython', 'perl', 'java', 'python', 'php'}
{'jython', 'perl', 'java', 'python', 'php'}
```

In [116]:

```
# Intersection of set

book_intersection_set = book_set1.intersection(book_set2)


print(book_intersection_set)

book_intersection_set1 = book_set1 & book_set2

print(book_intersection_set1)
```

```
{'perl', 'python'}
{'perl', 'python'}
```

In [109]:

```
help(book_set1.intersection_update)
```

```
Help on built-in function intersection_update:

intersection_update(...) method of builtins.set instance
    Update a set with the intersection of itself and another.
```

In [110]:

```
help(book_set1.intersection)
```

```
Help on built-in function intersection:

intersection(...) method of builtins.set instance
    Return the intersection of two sets as a new set.

    (i.e. all elements that are in both sets.)
```

In [118]:

```
print(book_set1)
print(book_set2)

book_set1 ^ book_set2
```

```
{'perl', 'python'}
{'perl', 'jython', 'java', 'python'}
```

Out[118]:

```
{'java', 'jython'}
```

In [119]:

```
'java' in book_union_set
```

Out[119]:

```
True
```

In [120]:

```
sorted_books = sorted(book_union_set)

print(sorted_books)
```

```
['java', 'jython', 'perl', 'php', 'python']
```

In [122]:

```
reverse_sorted_books = sorted(book_union_set, \
                              reverse=True)

print(reverse_sorted_books)
```

```
['python', 'php', 'perl', 'jython', 'java']
```

In [124]:

```
print(list(reversed(sorted_books)))
```

```
['python', 'php', 'perl', 'jython', 'java']
```

In [127]:

```
list(enumerate(sorted_books))
```

Out[127]:

```
[(0, 'java'), (1, 'jython'), (2, 'perl'), (3, 'php'), (4, 'python')]
```

In [129]:

```
for index, book in list(enumerate(sorted_books,1)):
    print(index, book)
```

```
1 java
2 jython
3 perl
4 php
5 python
```

In [130]:

```
Que = ['Name','City','Age','Contact']

Ans = ['Surendra','Pune','42','9975072320']

list(zip(Que,Ans))
```

Out[130]:

```
[('Name', 'Surendra'),
 ('City', 'Pune'),
 ('Age', '42'),
 ('Contact', '9975072320')]
```

In [132]:

```
for Q, A in zip(Que,Ans):
    print("What's your %s and it is %s"%(Q,A))
```

```
What's your Name and it is Surendra
What's your City and it is Pune
What's your Age and it is 42
What's your Contact and it is 9975072320
```

In [133]:

```
print('''List comprehension is a comprehensive way of
writing python script inside list ''')
```

```
List comprehension is a comprehensive way of
writing python script inside list
```

In [138]:

```
price_data = [ 50, 70, 300, 560 ]

double_price = [ num*2 for num in price_data ]

print(double_price)
```

```
[100, 140, 600, 1120]
```

In [136]:

```
[ (num,num*2) for num in price_data ]
```

Out[136]:

```
[(50, 100), (70, 140), (300, 600), (560, 1120)]
```

In [137]:

```
[ num**3 for num in price_data ]
```

Out[137]:

```
[125000, 343000, 27000000, 175616000]
```

In [141]:

```
price_data = [ 50, 70, 300, 560 ]

[ num for num in price_data if num % 7 !=0  ]

```

Out[141]:

```
[50, 300]
```

In [142]:

```
words = ['abc', 'def', 'ghi']
```

In [146]:

```
for word in words:
    for char in word:
        print(char)
```

```
a
b
c
d
e
f
g
h
i
```

In [150]:

```
[ ch for w in words for ch in w ]
```

Out[150]:

```
['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i']
```

In [154]:

```
char_ascii = [ (char,ord(char)) for word in words
   for char in word ]
print(char_ascii)
```

```
[('a', 97), ('b', 98), ('c', 99), ('d', 100), ('e', 101), ('f', 102)
, ('g', 103), ('h', 104), ('i', 105)]
```

In [152]:

```
for ch in 'abc':
    print(ch)
```

```
a
b
c
```

In [153]:

```
1 ord('s')
```

Out[153]:

```
115
```

In [155]:

```
1 char_dict =  dict(char_ascii)
2
3 char_dict
```

Out[155]:

```
{'a': 97,
 'b': 98,
 'c': 99,
 'd': 100,
 'e': 101,
 'f': 102,
 'g': 103,
 'h': 104,
 'i': 105}
```

In [156]:

```
1 QA_Dict = dict(list(zip(Que,Ans)))
2 print(QA_Dict)
```

```
{'Name': 'Surendra', 'City': 'Pune', 'Age': '42', 'Contact': '997507
2320'}
```

In [157]:

```
1 { Que : Ans for Que, Ans in QA_Dict.items() }
```

Out[157]:

```
{'Name': 'Surendra', 'City': 'Pune', 'Age': '42', 'Contact': '997507
2320'}
```

In [159]:

```
1 Inv_Dict = { Ans : Que for Que, Ans in QA_Dict.items() }
```

In [160]:

```
1 Inv_Dict
```

Out[160]:

```
{'Surendra': 'Name', 'Pune': 'City', '42': 'Age', '9975072320': 'Con
tact'}
```

In [ ]:

```
1
```