In [63]:

```
Agenda = '''
Object Oriented Programming
Class vs Object
Object Variable vs Class Variable
Use of self
Constructor __init__
Polymorphism : Overriding
use of super function

Module
What is Module in Python?
User Define Module.
How to import Modules?
math
os, sys,
Pickle,
JSON,
Regular Expression,
XML
YML Concept
YAML/JINJA— What/how/why do we use
File Handling ,
Exception Handling
User Define Exception
[ Project Assignment to Participants ]
'''
```

In [ ]:

```
Dict = {}
Dict.update({'a':'apple'})
print(Dict)

List1 = []
List1.append('apple')
print(List1)
```

In [ ]:

```
class BankApp:
    '''Bank Application'''

    def info(self):
        '''This is Bank Information Method'''
        print("Bank Information Method")


Bank1 = BankApp()
Bank1.info()
help(Bank1)
```

In [ ]:

```
class BankApp:
    '''Bank Application'''

    def __init__(self):  # Constructor Method
        '''Initialize BankApp Class'''
        print("Initilisation of Bank Application")


    def info(self):
        '''This is Bank Information Method'''
        print("Bank Information Method")



Bank1 = BankApp()
```

In [ ]:

```
class BankApp:
    '''Bank Application'''

    def __init__(self, bname, bloc ):  # Constructor Method
        '''Initialize BankApp Class'''
        # bname and bloc are local variables
        print("Bank Name is %s" %bname)
        print("Bank Location is %s" %bloc)

    def info(self):
        '''This is Bank Information Method'''
        print("Bank Information Method")



Bank1 = BankApp("HDFC Bank","Pune")
Bank1.bname
```

In [ ]:

```
class BankApp:
    '''Bank Application'''

    def __init__(self, bname, bloc ):  # Constructor Method
        '''Initialize BankApp Class'''
        self.bname = bname  # Object Variable
        self.bloc = bloc

    def info(self, branch):
        '''This is Bank Information Method'''
        self.branch = branch
        print("Bank Information Method")
        print("Bank Name is %s" %self.bname)
        print("Bank Location is %s" %self.bloc)
        print("Bank Branch is %s" %self.branch)


Bank1 = BankApp("HDFC Bank","Pune")
Bank1.info('Kothrud')
```

In [ ]:

```
class BankApp:
    '''Bank Application'''
    ifsc = 0  # Class Variable

    def __init__(self, bname, bloc ):  # Constructor Method
        '''Initialize BankApp Class'''
        BankApp.ifsc += 1  # Incrementing for all object
        self.bname = bname  # Object Variable
        self.bloc = bloc
        print("Hdfc ifsc hdfc000%d"%BankApp.ifsc)

    def info(self, branch):
        '''This is Bank Information Method'''
        self.branch = branch
        print("Bank Name is %s" %self.bname)
        print("Bank Location is %s" %self.bloc)
        print("Bank Branch is %s" %self.branch)


Hdfc1 = BankApp("HDFC Bank","Pune")
Hdfc1.info('Kothrud')
print("\n")
Hdfc2 = BankApp("HDFC Bank","Pune")
Hdfc2.info('Sinhgad Road')
```

In [ ]:

```
class BankApp:
    '''Bank Application'''
    ifsc = 0  # Class Variable

    def __init__(self, bname, bloc ):  # Constructor Method
        '''Initialize BankApp Class'''
        BankApp.ifsc += 1  # Incrementing for all object
        self.bname = bname  # Object Variable
        self.bloc = bloc
        print("Hdfc ifsc hdfc000%d"%BankApp.ifsc)

    def info(self, branch):
        '''This is Bank Information Method'''
        self.branch = branch
        print("Bank Name is %s" %self.bname)
        print("Bank Location is %s" %self.bloc)
        print("Bank Branch is %s" %self.branch)


class Customer(BankApp):
    '''Bank Customer Class '''

    def cinfo(self, cname):
        '''This is Customer Information Method'''
        self.cname = cname
        print("Customer Name is %s" %self.cname)




Hdfc1 = Customer("HDFC Bank","Pune")
Hdfc1.info('Kothrud')
Hdfc1.cinfo('Surendra')
print("\n")
Hdfc2 = Customer("HDFC Bank","Pune")
Hdfc2.info('Sinhgad Road')
Hdfc2.cinfo('Amit')

help(Hdfc1)
```

In [ ]:

```
class BankApp:
    '''Bank Application'''
    ifsc = 0  # Class Variable

    def __init__(self, bname, bloc ):  # Constructor Method
        '''Initialize BankApp Class'''
        BankApp.ifsc += 1  # Incrementing for all object
        self.bname = bname  # Object Variable
        self.bloc = bloc
        print("Hdfc ifsc hdfc000%d"%BankApp.ifsc)

    def info(self, branch):
        '''This is Bank Information Method'''
        self.branch = branch
        print("Bank Name is %s" %self.bname)
        print("Bank Location is %s" %self.bloc)
        print("Bank Branch is %s" %self.branch)


class Customer(BankApp):
    '''Bank Customer Class '''

    def cinfo(self, cname):
        '''This is Customer Information Method'''
        self.cname = cname
        print("Customer Name is %s" %self.cname)

    def info(self, cloc, cstate):
        '''This is Customer Location Info'''
        self.cloc = cloc
        self.cstate = cstate
        print("Customer Location %s and State %s"\
              %(cloc, cstate))


Hdfc1 = Customer("HDFC Bank","Pune")
Hdfc1.info('Anand Nagar','Maharashtra')
Hdfc1.cinfo('Surendra')
print("\n")
Hdfc2 = Customer("HDFC Bank","Pune")
Hdfc2.info('Sinhgad Road','Maharashtra')
Hdfc2.cinfo('Amit')
```

In [ ]:

```
class BankApp:
    '''Bank Application'''
    ifsc = 0  # Class Variable

    def __init__(self, bname, bloc ):  # Constructor Method
        '''Initialize BankApp Class'''
        BankApp.ifsc += 1  # Incrementing for all object
        self.bname = bname  # Object Variable
        self.bloc = bloc
        print("Hdfc ifsc hdfc000%d"%BankApp.ifsc)

    def info(self, branch):
        '''This is Bank Information Method'''
        self.branch = branch
        print("Bank Name is %s" %self.bname)
        print("Bank Location is %s" %self.bloc)
        print("Bank Branch is %s" %self.branch)


class Customer(BankApp):
    '''Bank Customer Class '''

    def cinfo(self, cname):
        '''This is Customer Information Method'''
        self.cname = cname
        print("Customer Name is %s" %self.cname)

    def info(self, branch, cloc, cstate):
        '''This is Customer Location Info'''
        BankApp.info(self,branch)
        self.cloc = cloc
        self.cstate = cstate
        print("Customer Location %s and State %s"\
              %(cloc, cstate))


Hdfc1 = Customer("HDFC Bank","Pune")
Hdfc1.info('Kothrud','Anand Nagar','Maharashtra')
Hdfc1.cinfo('Surendra')
```

In [ ]:

```
%%writefile BankAppex.py
'''Bank Application Module '''

version = 1.0
name = "Bank Application Module"

class BankApp:
    '''Bank Application'''
    ifsc = 0  # Class Variable

    def __init__(self, bname, bloc ):  # Constructor Method
        '''Initialize BankApp Class'''
        BankApp.ifsc += 1  # Incrementing for all object
        self.bname = bname  # Object Variable
        self.bloc = bloc
        print("Hdfc ifsc hdfc000%d"%BankApp.ifsc)

    def info(self, branch):
        '''This is Bank Information Method'''
        self.branch = branch
        print("Bank Name is %s" %self.bname)
        print("Bank Location is %s" %self.bloc)
        print("Bank Branch is %s" %self.branch)


class Customer(BankApp):
    '''Bank Customer Class '''

    def cinfo(self, cname):
        '''This is Customer Information Method'''
        self.cname = cname
        print("Customer Name is %s" %self.cname)

    def info(self, branch, cloc, cstate):
        '''This is Customer Location Info'''
        super().info(branch)
        self.cloc = cloc
        self.cstate = cstate
        print("Customer Location %s and State %s"\
              %(cloc, cstate))

if __name__ == '__main__':
    Hdfc1 = Customer("HDFC Bank","Pune")
    Hdfc1.info('Kothrud','Anand Nagar','Maharashtra')
    Hdfc1.cinfo('Surendra')
```

In [ ]:

```
run BankAppex.py
```

In [ ]:

```
run BankApp.py
```

In [1]:

```
import BankAppex
```

In [2]:

```
help(BankAppex)
```

```
Help on module BankAppex:

NAME
    BankAppex - Bank Application Module

CLASSES
    builtins.object
        BankApp
            Customer

    class BankApp(builtins.object)
     |  Bank Application
     |
     |  Methods defined here:
     |
     |  __init__(self, bname, bloc)
     |      Initialize BankApp Class
     |
     |  info(self, branch)
     |      This is Bank Information Method
     |
     |  ----------------------------------------
----------------------------
     |  Data descriptors defined here:
     |
     |  __dict__
     |      dictionary for instance variables (if
defined)
     |
     |  __weakref__
     |      list of weak references to the object
```

```
(if defined)
     |
     |  ----------------------------------------
----------------------------
     |  Data and other attributes defined here:
     |
     |  ifsc = 0

    class Customer(BankApp)
     |  Bank Customer Class
     |
     |  Method resolution order:
     |      Customer
     |      BankApp
     |      builtins.object
     |
     |  Methods defined here:
     |
     |  cinfo(self, cname)
     |      This is Customer Information Method
     |
     |  info(self, branch, cloc, cstate)
     |      This is Customer Location Info
     |
     |  ----------------------------------------
----------------------------
     |  Methods inherited from BankApp:
     |
     |  __init__(self, bname, bloc)
     |      Initialize BankApp Class
     |
     |  ----------------------------------------
----------------------------
     |  Data descriptors inherited from BankApp:
     |
     |  __dict__
     |      dictionary for instance variables (if
defined)
     |
     |  __weakref__
     |      list of weak references to the object
(if defined)
     |
     |  ----------------------------------------
----------------------------
     |  Data and other attributes inherited from B
ankApp:
     |
```

```
     |  ifsc = 0

DATA
    name = 'Bank Application Module'
    version = 1.0

FILE
    /Users/surendra/BankAppex.py
```

In [3]:

```
# cd __pycache__/
```

In [4]:

```
dir(BankAppex)
```

Out[4]:

```
['BankApp',
 'Customer',
 '__builtins__',
 '__cached__',
 '__doc__',
 '__file__',
 '__loader__',
 '__name__',
 '__package__',
 '__spec__',
 'name',
 'version']
```

In [6]:

```
BankAppex.name
```

Out[6]:

```
'Bank Application Module'
```

In [7]:

```
BankAppex.version
```

Out[7]:

```
1.0
```

In [11]:

```
help(BankAppex.Customer)
```

```
Help on class Customer in module BankAppex:

class Customer(BankApp)
 |  Bank Customer Class
 |
 |  Method resolution order:
 |      Customer
 |      BankApp
 |      builtins.object
 |
 |  Methods defined here:
 |
 |  cinfo(self, cname)
 |      This is Customer Information Method
 |
 |  info(self, branch, cloc, cstate)
 |      This is Customer Location Info
 |
 |  ----------------------------------------------
----------------------
 |  Methods inherited from BankApp:
 |
 |  __init__(self, bname, bloc)
 |      Initialize BankApp Class
 |
 |  ----------------------------------------------
----------------------
 |  Data descriptors inherited from BankApp:
 |
 |  __dict__
 |      dictionary for instance variables (if defi
ned)
 |
 |  __weakref__
 |      list of weak references to the object (if
defined)
 |
 |  ----------------------------------------------
----------------------
 |  Data and other attributes inherited from BankA
pp:
 |
 |  ifsc = 0
```

In [12]:

```
Customer1 = BankAppex.Customer('HDFC Bank','Pune')
```

```
Hdfc ifsc hdfc0001
```

In [13]:

```
dir(Customer1)
```

Out[13]:

```
['__class__',
 '__delattr__',
 '__dict__',
 '__dir__',
 '__doc__',
 '__eq__',
 '__format__',
 '__ge__',
 '__getattribute__',
 '__gt__',
 '__hash__',
 '__init__',
 '__init_subclass__',
 '__le__',
 '__lt__',
 '__module__',
 '__ne__',
 '__new__',
 '__reduce__',
 '__reduce_ex__',
 '__repr__',
 '__setattr__',
 '__sizeof__',
 '__str__',
 '__subclasshook__',
 '__weakref__',
 'bloc',
 'bname',
 'cinfo',
 'ifsc',
 'info']
```

In [14]:

```
Customer1.bname
```

Out[14]:

```
'HDFC Bank'
```

In [16]:

```
help(Customer1.info)
```

```
Help on method info in module BankAppex:

info(branch, cloc, cstate) method of BankAppex.Cus
tomer instance
    This is Customer Location Info
```

In [17]:

```
Customer1.info('Kothrud','Pune','Maharashtra')
```

```
Bank Name is HDFC Bank
Bank Location is Pune
Bank Branch is Kothrud
Customer Location Pune and State Maharashtra
```

In [18]:

```
import BankAppex as BA

BA.name
```

Out[18]:

```
'Bank Application Module'
```

In [19]:

```
BA.version
```

Out[19]:

```
1.0
```

In [22]:

```
B1 = BA.BankApp('HDFC','Pune')
```

```
Hdfc ifsc hdfc0003
```

In [23]:

```
B1.bloc
```

Out[23]:

```
'Pune'
```

In [24]:

```
B1.bname
```

Out[24]:

```
'HDFC'
```

In [26]:

```
B1.info('Kothrud')
```

```
Bank Name is HDFC
Bank Location is Pune
Bank Branch is Kothrud
```

In [29]:

```
from BankAppex import name, version
```

In [31]:

```
print(name)
print(version)
```

```
Bank Application Module
1.0
```

In [32]:

```
from BankAppex import *
```

In [33]:

```
name
```

Out[33]:

```
'Bank Application Module'
```

In [35]:

```
version
```

Out[35]:

```
1.0
```

In [37]:

```
B1 = BankApp('HDFC','Warje')
```

```
Hdfc ifsc hdfc0004
```

In [38]:

```
B1.bloc
```

Out[38]:

```
'Warje'
```

In [39]:

```
import os
```

In [41]:

```
#dir(os)
```

In [42]:

```
os.getcwd()
```

Out[42]:

```
'/Users/surendra'
```

In [43]:

```
os.mkdir('accenture')
```

In [48]:

```
os.listdir('/Users/surendra/accenture/')
```

Out[48]:

```
[]
```

In [53]:

```
pwd
```

Out[53]:

```
'/Users/surendra'
```

In [61]:

```
os.system('ls -l *.txt')
```

Out[61]:

```
0
```

In [62]:

```
import math
```

In [63]:

```
math.factorial(5)
```

Out[63]:

```
120
```

In [64]:

```
math.sin(90)
```

Out[64]:

```
0.8939966636005579
```

In [65]:

```
math.pi
```

Out[65]:

```
3.141592653589793
```

In [66]:

```
dir(math)
```

Out[66]:

```
['__doc__',
 '__file__',
 '__loader__',
 '__name__',
 '__package__',
 '__spec__',
 'acos',
 'acosh',
 'asin',
 'asinh',
 'atan',
 'atan2',
 'atanh',
 'ceil',
 'copysign',
 'cos',
 'cosh',
 'degrees',
 'e',
 'erf',
 'erfc',
 'exp',
 'expm1',
 'fabs',
 'factorial',
 'floor',
 'fmod',
 'frexp',
 'fsum',
 'gamma',
 'gcd',
 'hypot',
 'inf',
 'isclose',
 'isfinite',
```

```
 'isinf',
 'isnan',
 'ldexp',
 'lgamma',
 'log',
 'log10',
 'log1p',
 'log2',
 'modf',
 'nan',
 'pi',
 'pow',
 'radians',
 'sin',
 'sinh',
 'sqrt',
 'tan',
 'tanh',
 'tau',
 'trunc']
```

In [68]:

```
#help(math)
```

In [69]:

```
import sys

sys.platform
```

Out[69]:

```
'darwin'
```

In [70]:

```
sys.version
```

Out[70]:

```
'3.6.9 |Anaconda, Inc.| (default, Jul 30 2019, 13:
42:17) \n[GCC 4.2.1 Compatible Clang 4.0.1 (tags/R
ELEASE_401/final)]'
```

In [71]:

```
sys.version_info
```

Out[71]:

```
sys.version_info(major=3, minor=6, micro=9, releas
elevel='final', serial=0)
```

In [72]:

```
sys.path
```

Out[72]:

```
['/Users/surendra/anaconda3/lib/python36.zip',
 '/Users/surendra/anaconda3/lib/python3.6',
 '/Users/surendra/anaconda3/lib/python3.6/lib-dynl
oad',
 '',
 '/Users/surendra/anaconda3/lib/python3.6/site-pac
kages',
 '/Users/surendra/anaconda3/lib/python3.6/site-pac
kages/aeosa',
 '/Users/surendra/anaconda3/lib/python3.6/site-pac
kages/IPython/extensions',
 '/Users/surendra/.ipython']
```

In [73]:

```
sys.path.append(r'/Users/surendra/accenture/')
```

In [74]:

```
sys.path
```

Out[74]:

```
['/Users/surendra/anaconda3/lib/python36.zip',
 '/Users/surendra/anaconda3/lib/python3.6',
 '/Users/surendra/anaconda3/lib/python3.6/lib-dynl
oad',
 '',
 '/Users/surendra/anaconda3/lib/python3.6/site-pac
kages',
 '/Users/surendra/anaconda3/lib/python3.6/site-pac
kages/aeosa',
 '/Users/surendra/anaconda3/lib/python3.6/site-pac
kages/IPython/extensions',
 '/Users/surendra/.ipython',
 '/Users/surendra/accenture/']
```

In [75]:

```
dir(sys)
```

Out[75]:

```
['__displayhook__',
 '__doc__',
 '__excepthook__',
 '__interactivehook__',
 '__loader__',
 '__name__',
 '__package__',
 '__spec__',
 '__stderr__',
 '__stdin__',
 '__stdout__',
 '_clear_type_cache',
 '_current_frames',
 '_debugmallocstats',
 '_getframe',
 '_git',
 '_home',
 '_xoptions',
 'abiflags',
 'api_version',
 'argv',
 'base_exec_prefix',
```

```
'base_prefix',

'builtin_module_names',
'byteorder',
'call_tracing',
'callstats',
'copyright',
'displayhook',
'dont_write_bytecode',
'exc_info',
'excepthook',
'exec_prefix',
'executable',
'exit',
'flags',
'float_info',
'float_repr_style',
'get_asyncgen_hooks',
'get_coroutine_wrapper',
'getallocatedblocks',
'getcheckinterval',
'getdefaultencoding',
'getdlopenflags',
'getfilesystemencodeerrors',
'getfilesystemencoding',
'getprofile',
'getrecursionlimit',
'getrefcount',
'getsizeof',
'getswitchinterval',
'gettrace',
'hash_info',
'hexversion',
'implementation',
'int_info',
'intern',
'is_finalizing',
'last_traceback',
'last_type',
'last_value',
'maxsize',
'maxunicode',
'meta_path',
'modules',
'path',
'path_hooks',
'path_importer_cache',
'platform',
'prefix',
```

```
 'ps1',
 'ps2',
 'ps3',
 'set_asyncgen_hooks',
 'set_coroutine_wrapper',
 'setcheckinterval',
 'setdlopenflags',
 'setprofile',
 'setrecursionlimit',
 'setswitchinterval',
 'settrace',
 'stderr',
 'stdin',
 'stdout',
 'thread_info',
 'version',
 'version_info',
 'warnoptions']
```

In [81]:

```
sys.argv[1:]
```

Out[81]:

```
['-f',
 '/Users/surendra/Library/Jupyter/runtime/kernel-a
42d8fbc-3dfd-40ab-8e46-94b55333f309.json']
```

In [84]:

```
#sys.modules.keys()
```

In [85]:

```
sys.modules.get('os')
```

Out[85]:

```
<module 'os' from '/Users/surendra/anaconda3/lib/p
ython3.6/os.py'>
```

In [88]:

```
sys.modules.get('BankAppex')
```

Out[88]:

```
<module 'BankAppex' from '/Users/surendra/BankAppe
x.py'>
```

In [108]:

```
import pickle

Fruits = {'a' : ['apple','apricot'],
          'b': ('banana','blueberry'), 'o':'orange' }

print(Fruits)

pickle_fruits = pickle.dumps(Fruits)

print("\n Pickle Fruits::\n",pickle_fruits)

print("\n Pickle Data Type is :")
print(type(pickle_fruits))

# Unpickle Bytes Data / Serialise data

unpickle_fruits = pickle.loads(pickle_fruits)

print("\n Unpickle Data is \n", unpickle_fruits)

print("Is data equal?", Fruits == unpickle_fruits )
print("Is data same?", Fruits is unpickle_fruits )
print(id(Fruits))
print(id(unpickle_fruits))
```

```
{'a': ['apple', 'apricot'], 'b': ('banana', 'blueb
erry'), 'o': 'orange'}

 Pickle Fruits::
 b'\x80\x03}q\x00(X\x01\x00\x00\x00aq\x01]q\x02(X\
x05\x00\x00\x00appleq\x03X\x07\x00\x00\x00apricotq
\x04eX\x01\x00\x00\x00bq\x05X\x06\x00\x00\x00banan
aq\x06X\t\x00\x00\x00blueberryq\x07\x86q\x08X\x01\
x00\x00\x00oq\tX\x06\x00\x00\x00orangeq\nu.'

 Pickle Data Type is :
<class 'bytes'>

 Unpickle Data is
 {'a': ['apple', 'apricot'], 'b': ('banana', 'blue
berry'), 'o': 'orange'}
Is data equal? True
Is data same? False
4452880960
4481964576
```

In [104]:

```
import json

Fruits = {'a': ['apple','apricot'],
          'b': ('banana','blueberry'), 'o':'orange' }

print(Fruits)

pickle_fruits = json.dumps(Fruits)

print("\n Pickle Fruits::\n",pickle_fruits)

print("\n Pickle Data Type is :")
print(type(pickle_fruits))

# Unpickle Bytes Data / Serialise data

unpickle_fruits = json.loads(pickle_fruits)

print("\n Unpickle Data is \n", unpickle_fruits)

print("Is data equal?", Fruits == unpickle_fruits )
print("Is data same?", Fruits is unpickle_fruits )
print(id(Fruits))
print(id(unpickle_fruits))
```

```
{'a': ['apple', 'apricot'], 'b': ('banana', 'blueb
erry'), 'o': 'orange'}

 Pickle Fruits::
 {"a": ["apple", "apricot"], "b": ["banana", "blue
berry"], "o": "orange"}

 Pickle Data Type is :
<class 'str'>

 Unpickle Data is
 {'a': ['apple', 'apricot'], 'b': ['banana', 'blue
berry'], 'o': 'orange'}
Is data equal? False
Is data same? False
4481964576
4481965944
```

In [1]:

```
#help(pickle)
```

In [2]:

```
import re

patt = re.compile('[a-z]+', re.I )
print(patt)
```

```
re.compile('[a-z]+', re.IGNORECASE)
```

In [3]:

```
match_data = patt.match('Surendra Panpaliya')
print(match_data)
```

```
<_sre.SRE_Match object; span=(0, 8), match='Surend
ra'>
```

In [4]:

```
match_data = patt.match('9975072320 Surendra Panpaliya')
print(match_data)
```

```
None
```

In [5]:

```
search_data = patt.search('9975072320 Surendra Panpaliya')
print(search_data)
```

```
<_sre.SRE_Match object; span=(11, 19), match='Sure
ndra'>
```

In [6]:

```
data = '''
Name : Surendra Panpaliya
Mobile Number : 7620379390
Email : surendra@gktcs.com
Account Number : 624001053983

Name : Narendra Panpaliya
Mobile Number : 8620379390
Email : surendra.panpaliya@gmail.com
Account Number : 624001053984

Name : Satish Panpaliya
Mobile Number : 9620379390
Email : Surendra@gktcs.co.in
Account Number : 424001053983

Name : Sanket Panpaliya
Mobile Number : 6620379390
Email : Surendra@gktcs.com
Account Number : 624001053986
'''
```

In [9]:

```
Mobiles = re.findall(r'\b\d{10}\b',data)
print(Mobiles)
```

```
['7620379390', '8620379390', '9620379390', '662037
9390']
```

In [10]:

```
Mobiles = re.findall('\\b\d{10}\\b',data)
print(Mobiles)
```

```
['7620379390', '8620379390', '9620379390', '662037
9390']
```

In [11]:

```
Accounts = re.findall('\\b\d{12}\\b',data)
print(Accounts)
```

```
['624001053983', '624001053984', '424001053983', '
624001053986']
```

In [12]:

```
Emails = re.findall(r'\w+@\w+.\w{2,5}',data)
print(Emails)
```

```
['surendra@gktcs.com', 'panpaliya@gmail.com', 'Sur
endra@gktcs.co', 'Surendra@gktcs.com']
```

In [18]:

```
Emails = re.findall(r'[.\w]+@\w+[.\w]{2,6}',data)
print(Emails)
```

```
['surendra@gktcs.com', 'surendra.panpaliya@gmail.c
om', 'Surendra@gktcs.co.in', 'Surendra@gktcs.com']
```

In [19]:

```
re.sub('\\b\d{10}\\b','xxxMobilexxx',data)
```

Out[19]:

```
'\nName : Surendra Panpaliya \nMobile Number : xxx
Mobilexxx\nEmail : surendra@gktcs.com\nAccount Num
ber : 624001053983\n\nName : Narendra Panpaliya \n
Mobile Number : xxxMobilexxx\nEmail : surendra.pan
paliya@gmail.com\nAccount Number : 624001053984\n\
nName : Satish Panpaliya \nMobile Number : xxxMobi
lexxx\nEmail : Surendra@gktcs.co.in\nAccount Numbe
r : 424001053983\n\nName : Sanket Panpaliya \nMobi
le Number : xxxMobilexxx\nEmail : Surendra@gktcs.c
om\nAccount Number : 624001053986\n'
```

In [20]:

```
re.sub('\\b\d{12}\\b','xxxAccountsxxx',data)
```

Out[20]:

```
'\nName : Surendra Panpaliya \nMobile Number : 762
0379390\nEmail : surendra@gktcs.com\nAccount Numbe
r : xxxAccountsxxx\n\nName : Narendra Panpaliya \n
Mobile Number : 8620379390\nEmail : surendra.panpa
liya@gmail.com\nAccount Number : xxxAccountsxxx\n\
nName : Satish Panpaliya \nMobile Number : 9620379
390\nEmail : Surendra@gktcs.co.in\nAccount Number
: xxxAccountsxxx\n\nName : Sanket Panpaliya \nMobi
le Number : 6620379390\nEmail : Surendra@gktcs.com
\nAccount Number : xxxAccountsxxx\n'
```

In [22]:

```
re.sub(r'[.\w]+@\w+[.\w]{2,6}','xxEmailsxx',data)
```

Out[22]:

```
'\nName : Surendra Panpaliya \nMobile Number : 762
0379390\nEmail : xxEmailsxx\nAccount Number : 6240
01053983\n\nName : Narendra Panpaliya \nMobile Num
ber : 8620379390\nEmail : xxEmailsxx\nAccount Numb
er : 624001053984\n\nName : Satish Panpaliya \nMob
ile Number : 9620379390\nEmail : xxEmailsxx\nAccou
nt Number : 424001053983\n\nName : Sanket Panpaliy
a \nMobile Number : 6620379390\nEmail : xxEmailsxx
\nAccount Number : 624001053986\n'
```

In [23]:

```
Mobiles = re.split(r'\b\d{10}\b',data)
print(Mobiles)
```

```
['\nName : Surendra Panpaliya \nMobile Number : ',
'\nEmail : surendra@gktcs.com\nAccount Number : 62
4001053983\n\nName : Narendra Panpaliya \nMobile N
umber : ', '\nEmail : surendra.panpaliya@gmail.com
\nAccount Number : 624001053984\n\nName : Satish P
anpaliya \nMobile Number : ', '\nEmail : Surendra@
gktcs.co.in\nAccount Number : 424001053983\n\nName
: Sanket Panpaliya \nMobile Number : ', '\nEmail :
Surendra@gktcs.com\nAccount Number : 624001053986\
n']
```

In [28]:

```
Accounts = re.split('\\b\d{12}\\b',data,5)
print(Accounts)
```

```
['\nName : Surendra Panpaliya \nMobile Number : 76
20379390\nEmail : surendra@gktcs.com\nAccount Numb
er : ', '\n\nName : Narendra Panpaliya \nMobile Nu
mber : 8620379390\nEmail : surendra.panpaliya@gmai
l.com\nAccount Number : ', '\n\nName : Satish Panp
aliya \nMobile Number : 9620379390\nEmail : Surend
ra@gktcs.co.in\nAccount Number : ', '\n\nName : Sa
nket Panpaliya \nMobile Number : 6620379390\nEmail
: Surendra@gktcs.com\nAccount Number : ', '\n']
```

In [51]:

```
%%writefile country_data.xml
<?xml version="1.0"?>
<data>
    <country name="Dubai">
        <rank>1</rank>
        <year>2019</year>
        <gdppc>141100</gdppc>
        <neighbor name="Saudi" direction="E"/>
        <neighbor name="Abudabi" direction="W"/>
    </country>
    <country name="Singapore">
        <rank>4</rank>
        <year>2011</year>
        <gdppc>59900</gdppc>
        <neighbor name="Malaysia" direction="N"/>
    </country>
    <country name="Panama">
        <rank>68</rank>
        <year>2011</year>
        <gdppc>13600</gdppc>
        <neighbor name="Costa Rica" direction="W"/>
        <neighbor name="Colombia" direction="E"/>
    </country>
    <country name="India">
        <rank>5</rank>
        <year>2018</year>
        <gdppc>136</gdppc>
        <neighbor name="Singapoor" direction="W"/>
        <neighbor name="Malasiya" direction="E"/>
    </country>
</data>
```

```
Overwriting country_data.xml
```

In [30]:

```
%%writefile xmlex.py

import xml.etree.ElementTree as ET
tree = ET.parse('country_data.xml')
root = tree.getroot()

print(root)

#Reading the data from a string:
#country_data="country_data in string format"
#root = ET.fromstring(country_data)

print(root.tag)
#'data'
print(root.attrib)
#{}

print(root[0][1].text)

for neighbor in root.iter('neighbor'):
    print(neighbor.attrib)

for country in root.findall('country'):
    rank = country.find('rank').text
    name = country.get('name')
    print(name, rank)

for rank in root.iter('rank'):
    new_rank = int(rank.text) + 1
    rank.text = str(new_rank)
    rank.set('updated', 'yes')

tree.write('output.xml')
```

```
Writing xmlex.py
```

In [58]:

```
import xml.etree.ElementTree as ET

tree = ET.parse('country_data.xml')
dir(tree)

# Get root node

root = tree.getroot()

print(root)
print(root.tag)
print(root.attrib)
print(root[0])
print(root[0][0].text)
print(root[0][1].text)
print(root[1])
print(root[1][0].text)
print(root[1][1].text)

for neighbor in root.iter('neighbor'):
    print(neighbor.attrib)


for country in root.findall('country'):
    rank = country.find('rank').text
    name = country.get('name')
    print(name, rank)

for rank in root.iter('rank'):
    new_rank = int(rank.text) + 1
    rank.text = str(new_rank)
    print(rank.text)
    rank.set('updated', 'yes')
    print(rank.attrib)

tree.write('new_country.xml')
```

```
<Element 'data' at 0x10894a228>
data
{}
<Element 'country' at 0x1088afea8>
1
2019
<Element 'country' at 0x109934c78>
4
2011
{'name': 'Saudi', 'direction': 'E'}
{'name': 'Abudabi', 'direction': 'W'}
{'name': 'Malaysia', 'direction': 'N'}
{'name': 'Costa Rica', 'direction': 'W'}
{'name': 'Colombia', 'direction': 'E'}
{'name': 'Singapoor', 'direction': 'W'}
{'name': 'Malasiya', 'direction': 'E'}
Dubai 1
Singapore 4
Panama 68
India 5
2
{'updated': 'yes'}
5
{'updated': 'yes'}
69
{'updated': 'yes'}
6
{'updated': 'yes'}
```

In [35]:

```
dir(root)
```

Out[35]:

```
['__class__',
 '__copy__',
 '__deepcopy__',
 '__delattr__',
 '__delitem__',
 '__dir__',
 '__doc__',
 '__eq__',
 '__format__',
 '__ge__',
 '__getattribute__',
 '__getitem__',
 '__getstate__',
```

'__gt__',
'__hash__',
'__init__',
'__init_subclass__',
'__le__',
'__len__',
'__lt__',
'__ne__',
'__new__',
'__reduce__',
'__reduce_ex__',
'__repr__',
'__setattr__',
'__setitem__',
'__setstate__',
'__sizeof__',
'__str__',
'__subclasshook__',
'append',
'attrib',
'clear',
'extend',
'find',
'findall',
'findtext',
'get',
'getchildren',
'getiterator',
'insert',
'items',
'iter',
'iterfind',
'itertext',
'keys',
'makeelement',
'remove',
'set',
'tag',
'tail',
'text']

In [ ]:

```
%load new_country.xml
<data>
    <country name="Dubai">
        <rank updated="yes">2</rank>
        <year>2019</year>
        <gdppc>141100</gdppc>
        <neighbor direction="E" name="Saudi" />
        <neighbor direction="W" name="Abudabi" />
    </country>
    <country name="Singapore">
        <rank updated="yes">5</rank>
        <year>2011</year>
        <gdppc>59900</gdppc>
        <neighbor direction="N" name="Malaysia" />
    </country>
    <country name="Panama">
        <rank updated="yes">69</rank>
        <year>2011</year>
        <gdppc>13600</gdppc>
        <neighbor direction="W" name="Costa Rica" />
        <neighbor direction="E" name="Colombia" />
    </country>
    <country name="India">
        <rank updated="yes">6</rank>
        <year>2018</year>
        <gdppc>136</gdppc>
        <neighbor direction="W" name="Singapoor" />
        <neighbor direction="E" name="Malasiya" />
    </country>
</data>
```

In [59]:

```
import yaml

ymldata= '''
   - eggs
   - ham
   - spam
   - French basil salmon terrine
   '''

with open('foo.yaml','w') as f:
    f.write(ymldata)
```

In [61]:

```
#dir(yaml)
```

In [63]:

```
import yaml

stream = open("foo.yaml", 'r')
data = yaml.load(stream, Loader=yaml.FullLoader)
print(data)
for item in data:
    print(item)
```

```
['eggs', 'ham', 'spam', 'French basil salmon terri
ne']
eggs
ham
spam
French basil salmon terrine
```

In [64]:

```
%%writefile items.yaml

raincoat: 1
coins: 5
books: 23
spectacles: 2
chairs: 12
pens: 6
```

```
Overwriting items.yaml
```

In [65]:

```
%%writefile data.yaml
cities:
  - Bratislava
  - Kosice
  - Trnava
  - Moldava
  - Trencin
---
companies:
  - Eset
  - Slovnaft
  - Duslo Sala
  - Matador Puchov
```

```
Overwriting data.yaml
```

In [66]:

```
import yaml

with open('items.yaml') as f:

    data = yaml.load(f, Loader=yaml.FullLoader)
    print(data)
```

```
{'raincoat': 1, 'coins': 5, 'books': 23, 'spectacl
es': 2, 'chairs': 12, 'pens': 6}
```

In [67]:

```
import yaml

users = [{'name': 'John Doe', 'occupation': 'gardener'},
         {'name': 'Lucy Black', 'occupation': 'teacher'}]

print(yaml.dump(users))
```

```
- name: John Doe
  occupation: gardener
- name: Lucy Black
  occupation: teacher
```

In [69]:

```
pip install netmiko
```

```
Collecting netmiko
  Downloading https://files.pythonhosted.org/packa
ges/26/05/dbe9c97c39f126e7b8dc70cf897dcad557dbd579
703f2e3acfd3606d0cee/netmiko-2.4.2-py2.py3-none-an
y.whl (144kB)
     |████████████████████████████████| 153kB 228k
B/s eta 0:00:01
Collecting scp>=0.13.2 (from netmiko)
  Downloading https://files.pythonhosted.org/packa
ges/4d/7a/3d76dc5ad8deea79642f50a572e1c057cb27e8b4
27f83781a2c05ce4e5b6/scp-0.13.2-py2.py3-none-any.w
hl
Collecting pyserial (from netmiko)
  Downloading https://files.pythonhosted.org/packa
ges/0d/e4/2a744dd9e3be04a0c0907414e2a01a7c88bb3915
cbe3c8cc06e209f59c30/pyserial-3.4-py2.py3-none-any
.whl (193kB)
     |████████████████████████████████| 194kB 345k
B/s eta 0:00:01
Collecting textfsm (from netmiko)
  Downloading https://files.pythonhosted.org/packa
ges/bd/27/0b149b6da3e47cc8daebace6920093114392171a
8f5c24f1f2ad9a9e9c4d/textfsm-1.1.0-py2.py3-none-an
y.whl
Requirement already satisfied: setuptools>=38.4.0
in ./anaconda3/lib/python3.6/site-packages (from n
etmiko) (41.4.0)
Collecting paramiko>=2.4.3 (from netmiko)
  Downloading https://files.pythonhosted.org/packa
ges/06/1e/1e08baaaf6c3d3df1459fd85f0e7d2d6aa916f33
958f151ee1ecc9800971/paramiko-2.7.1-py2.py3-none-a
ny.whl (206kB)
     |████████████████████████████████| 215kB 325k
B/s eta 0:00:01
Collecting future (from textfsm->netmiko)
  Downloading https://files.pythonhosted.org/packa
ges/45/0b/38b06fd9b92dc2b68d58b75f900e97884c45bedd
2ff83203d933cf5851c9/future-0.18.2.tar.gz (829kB)
     |████████████████████████████████| 829kB 695k
B/s eta 0:00:01     |████████████
| 399kB 333kB/s eta 0:00:02
Requirement already satisfied: six in ./anaconda3/
lib/python3.6/site-packages (from textfsm->netmiko
) (1.12.0)
```

```
Collecting bcrypt>=3.1.3 (from paramiko>=2.4.3->ne
tmiko)
  Downloading https://files.pythonhosted.org/packa
ges/62/20/4c94f3f8dfc6b8720c8bc903ce2951ec6397ad86
4e3a64b4abdced014514/bcrypt-3.1.7-cp34-abi3-macosx
_10_6_intel.whl (53kB)
     |████████████████████████████████| 61kB 5.9MB
/s eta 0:00:011
Collecting pynacl>=1.0.1 (from paramiko>=2.4.3->ne
tmiko)
  Using cached https://files.pythonhosted.org/pack
ages/cd/c9/4d6deea9d900345b1243cc3e6265110697fae5e
594689cfa1ba34e0ef4a8/PyNaCl-1.3.0-cp34-abi3-macos
x_10_6_intel.whl
Requirement already satisfied: cryptography>=2.5 i
n ./anaconda3/lib/python3.6/site-packages (from pa
ramiko>=2.4.3->netmiko) (2.7)
Requirement already satisfied: cffi>=1.1 in ./anac
onda3/lib/python3.6/site-packages (from bcrypt>=3.
1.3->paramiko>=2.4.3->netmiko) (1.12.3)
Requirement already satisfied: asn1crypto>=0.21.0
in ./anaconda3/lib/python3.6/site-packages (from c
ryptography>=2.5->paramiko>=2.4.3->netmiko) (1.0.1
)
Requirement already satisfied: pycparser in ./anac
onda3/lib/python3.6/site-packages (from cffi>=1.1-
>bcrypt>=3.1.3->paramiko>=2.4.3->netmiko) (2.19)
Building wheels for collected packages: future
  Building wheel for future (setup.py) ... done
  Created wheel for future: filename=future-0.18.2
-cp36-none-any.whl size=491057 sha256=e4455deee371
0ba015d605f675fa3ffef29885cbe96a49ada9bc364f05915e
f3
  Stored in directory: /Users/surendra/Library/Cac
hes/pip/wheels/8b/99/a0/81daf51dcd359a9377b110a8a8
86b3895921802d2fc1b2397e
Successfully built future
Installing collected packages: bcrypt, pynacl, par
amiko, scp, pyserial, future, textfsm, netmiko
Successfully installed bcrypt-3.1.7 future-0.18.2
netmiko-2.4.2 paramiko-2.7.1 pynacl-1.3.0 pyserial
-3.4 scp-0.13.2 textfsm-1.1.0
Note: you may need to restart the kernel to use up
dated packages.
```

In [70]:

```
import netmiko
```

In [71]:

```
dir(netmiko)
```

Out[71]:

```
['BaseConnection',
 'CNTL_SHIFT_6',
 'ConnectHandler',
 'FileTransfer',
 'InLineTransfer',
 'NetMikoAuthenticationException',
 'NetMikoTimeoutException',
 'Netmiko',
 'NetmikoAuthError',
 'NetmikoTimeoutError',
 'SCPConn',
 'SSHDetect',
 '__all__',
 '__builtins__',
 '__cached__',
 '__doc__',
 '__file__',
 '__loader__',
 '__name__',
 '__package__',
 '__path__',
 '__spec__',
 '__version__',
 '_textfsm',
 'a10',
 'accedian',
 'alcatel',
 'apresia',
 'arista',
 'aruba',
 'base_connection',
 'calix',
 'checkpoint',
 'ciena',
 'cisco',
 'cisco_base_connection',
 'citrix',
 'cloudgenix',
```

```
'coriant',
'dell',
'eltex',
'endace',
'enterasys',
'extreme',
'f5',
'file_transfer',
'flexvnf',
'fortinet',
'hp',
'huawei',
'ipinfusion',
'juniper',
'keymile',
'linux',
'log',
'logging',
'mellanox',
'mikrotik',
'mrv',
'netapp',
'netmiko_globals',
'oneaccess',
'ovs',
'paloalto',
'platforms',
'pluribus',
'py23_compat',
'quanta',
'rad',
'redispatch',
'ruckus',
'scp_functions',
'scp_handler',
'ssh_autodetect',
'ssh_dispatcher',
'ssh_exception',
'terminal_server',
'ubiquiti',
'unicode_literals',
'utilities',
'vyos']
```

In [1]:

```
from netmiko import ConnectHandler

# connection properties described in a dictionary
csr = {
    'device_type': 'cisco_ios',
    'ip': '192.168.1.1',
}
# the ** unpacks a dictionary
#csr_session = ConnectHandler(**csr)
#print(csr_session.send_command('show ip int brief'))
```

In [3]:

```
from jinja2 import Environment

HTML = """
<html>
<head>
<title>{{ title }}</title>
</head>
<body>
Hello.
</body>
</html>
"""

def print_html_doc():
    print(Environment().from_string(HTML).render(title='Hello
Gist from GutHub'))

if __name__ == '__main__':
    print_html_doc()
```

```
<html>
<head>
<title>Hellow Gist from GutHub</title>
</head>
<body>
Hello.
</body>
</html>
```

In [5]:

```
from jinja2 import Template

class Person:

    def __init__(self, name, age):

        self.name = name
        self.age = age

    def getAge(self):
        return self.age

    def getName(self):
        return self.name


person = Person('Surendra', 42)

tm = Template("My name is {{ per.getName() }} and I am {{ per.
getAge() }}")
msg = tm.render(per=person)

print(msg)
```

```
My name is Surendra and I am 42
```

In [6]:

```
from jinja2 import Template

person = { 'name': 'Person', 'age': 34 }

tm = Template("My name is {{ per.name }} \
and I am {{ per.age }}")
# tm = Template("My name is {{ per['name'] }} and I am {{ per[
'age'] }}")
msg = tm.render(per=person)

print(msg)
```

```
My name is Person and I am 34
```

In [7]:

```
from jinja2 import Template

data = '''
{% raw %}
His name is {{ name }}
{% endraw %}
'''

tm = Template(data)
msg = tm.render(name='Peter')

print(msg)
```

```
His name is {{ name }}
```

In [8]:

```
from jinja2 import Template, escape

data = '<a>Today is a sunny day</a>'

tm = Template("{{ data | e}}")
msg = tm.render(data=data)

print(msg)
print(escape(data))
```

```
&lt;a&gt;Today is a sunny day&lt;/a&gt;
&lt;a&gt;Today is a sunny day&lt;/a&gt;
```

In [12]:

```
%%writefile showpersons.txt

persons = [
    {'name': 'Andrej', 'age': 34},
    {'name': 'Mark', 'age': 17},
    {'name': 'Thomas', 'age': 44},
    {'name': 'Lucy', 'age': 14},
    {'name': 'Robert', 'age': 23},
    {'name': 'Dragomir', 'age': 54}
]
```

```
Overwriting showpersons.txt
```

In [13]:

```
from jinja2 import Environment, FileSystemLoader

persons = [
    {'name': 'Andrej', 'age': 34},
    {'name': 'Mark', 'age': 17},
    {'name': 'Thomas', 'age': 44},
    {'name': 'Lucy', 'age': 14},
    {'name': 'Robert', 'age': 23},
    {'name': 'Dragomir', 'age': 54}
]

file_loader = FileSystemLoader('templates')

env = Environment(loader=file_loader)

template = env.get_template('showpersons.txt')

output = template.render(persons=persons)
print(output)
```

```
---------------------------------------------------
-------------------------
TemplateNotFound
Traceback (most recent call last)
<ipython-input-13-04c0e37918c2> in <module>
     13 env = Environment(loader=file_loader)
     14
---> 15 template = env.get_template('showpersons.t
xt')
     16
     17 output = template.render(persons=persons)
```

```
~/anaconda3/lib/python3.6/site-packages/jinja2/env
ironment.py in get_template(self, name, parent, gl
obals)
    828         if parent is not None:
    829             name = self.join_path(name,
parent)
--> 830         return self._load_template(name,
self.make_globals(globals))
    831
    832     @internalcode

~/anaconda3/lib/python3.6/site-packages/jinja2/env
ironment.py in _load_template(self, name, globals)
    802
template.is_up_to_date):
    803                 return template
--> 804         template = self.loader.load(self,
name, globals)
    805         if self.cache is not None:
    806             self.cache[cache_key] =
template

~/anaconda3/lib/python3.6/site-packages/jinja2/loa
ders.py in load(self, environment, name, globals)
    111         # first we try to get the source f
or this template together
    112         # with the filename and the uptoda
te function.
--> 113         source, filename, uptodate = self.
get_source(environment, name)
    114
    115         # try to load the code from the by
tecode cache if there is a

~/anaconda3/lib/python3.6/site-packages/jinja2/loa
ders.py in get_source(self, environment, template)
    185                     return False
    186             return contents, filename,
uptodate
--> 187         raise TemplateNotFound(template)
    188
    189     def list_templates(self):

TemplateNotFound: showpersons.txt
```

In [10]:

```
%%writefile showpersons.txt

persons = [
    {'name': 'Andrej', 'age': 34},
    {'name': 'Mark', 'age': 17},
    {'name': 'Thomas', 'age': 44},
    {'name': 'Lucy', 'age': 14},
    {'name': 'Robert', 'age': 23},
    {'name': 'Dragomir', 'age': 54}
]
```

```
Writing showpersons.txt
```

In [15]:

```
%%writefile showminors.txt

{% for person in persons %}
    {% if person.age < 18 %}
        {{- person.name }}
    {% endif %}
{%- endfor %}
```

```
Writing showminors.txt
```

In [16]:

```
from jinja2 import Environment, FileSystemLoader

persons = [
    {'name': 'Andrej', 'age': 34},
    {'name': 'Mark', 'age': 17},
    {'name': 'Thomas', 'age': 44},
    {'name': 'Lucy', 'age': 14},
    {'name': 'Robert', 'age': 23},
    {'name': 'Dragomir', 'age': 54},
]

file_loader = FileSystemLoader('templates')
env = Environment(loader=file_loader)
env.trim_blocks = True
env.lstrip_blocks = True
env.rstrip_blocks = True

template = env.get_template('showminors.txt')

output = template.render(persons=persons)
print(output)
```

```
---------------------------------------------------
-------------------------
TemplateNotFound
Traceback (most recent call last)
<ipython-input-16-bb71cefd1bf6> in <module>
     16 env.rstrip_blocks = True
     17
---> 18 template = env.get_template('showminors.tx
t')
     19
     20 output = template.render(persons=persons)

~/anaconda3/lib/python3.6/site-packages/jinja2/env
ironment.py in get_template(self, name, parent, gl
obals)
    828         if parent is not None:
    829             name = self.join_path(name,
parent)
--> 830         return self._load_template(name,
self.make_globals(globals))
    831
    832     @internalcode
```

```
~/anaconda3/lib/python3.6/site-packages/jinja2/env
ironment.py in _load_template(self, name, globals)
    802
template.is_up_to_date):
    803                     return template
--> 804             template = self.loader.load(self,
name, globals)
    805             if self.cache is not None:
    806                 self.cache[cache_key] =
template

~/anaconda3/lib/python3.6/site-packages/jinja2/loa
ders.py in load(self, environment, name, globals)
    111             # first we try to get the source f
or this template together
    112             # with the filename and the uptoda
te function.
--> 113             source, filename, uptodate = self.
get_source(environment, name)
    114
    115             # try to load the code from the by
tecode cache if there is a

~/anaconda3/lib/python3.6/site-packages/jinja2/loa
ders.py in get_source(self, environment, template)
    185                         return False
    186                 return contents, filename,
uptodate
--> 187             raise TemplateNotFound(template)
    188
    189     def list_templates(self):

TemplateNotFound: showminors.txt
```

In [21]:

```
from jinja2 import Template

fname = input("Enter your first name: ")
lname = input("Enter your last name: ")


tm = Template("My First Name  is  {{ fname }}")
tm1 = Template("My Last Name  is  {{ lname }}")
msg = tm.render(fname=fname)
msg1 = tm1.render(lname=lname)
print(msg)
print(msg1)
```

```
Enter your first name: Surendra
Enter your last name: Panpaliya
My First Name  is  Surendra
My Last Name  is  Panpaliya
```

In [23]:

```
from jinja2 import Template

fname = input("Enter your first name: ")
lname = input("Enter your last name: ")


tm = Template("My First Name  is  {{ fname }} and \
My Last Name  is  {{ lname }}")
msg = tm.render(fname=fname, lname=lname )
print(msg)
```

```
Enter your first name: Surendra
Enter your last name: Panpaliya
My First Name  is  Surendra and My Last Name  is
Panpaliya
```

File Handling in Python

In [28]:

```
devices = '''
cisco router
cisco switches
Juniper Routers
Juniper Switches
'''

devices1 = '''cisco Firewall
cisco IOS 7200
Juniper 3600
Juniper 3000
'''

# Step1  Open file with write mode

with open("devices.txt","w") as file1:
    file1.write(devices)

# Step2  Open file with read mode

with open("devices.txt") as file1:
    print(file1.read())


# Step3  Open file with append mode

with open("devices.txt","a") as file1:
    file1.write(devices1)

# Step4  Open file with read mode

with open("devices.txt") as file1:
    print(file1.read())
```

```
cisco router
cisco switches
Juniper Routers
Juniper Switches


cisco router
cisco switches
Juniper Routers
Juniper Switches
cisco Firewall
cisco IOS 7200
Juniper 3600
Juniper 3000
```

In [33]:

```
devices = '''cisco router
cisco switches
Juniper Routers
Juniper Switches
'''

devices1 = '''cisco Firewall
cisco IOS 7200
Juniper 3600
Juniper 3000
'''

# Step1  Open file with write mode

with open("devices.txt","w") as file1:
    file1.write(devices)

# Step2  Open file with read mode

with open("devices.txt") as file1:
    print(file1.read())


# Step3  Open file with append mode

with open("devices.txt","a") as file1:
    file1.write(devices1)

# Step4  Open file with read mode
```

```
with open("devices.txt") as file1:

    devices_list = file1.readlines()
    print(devices_list)
    # insert new line in the devices list
    devices_list.insert(3,'Cisco 2200 Series\n')
    print(devices_list)


with open("devices_new.txt","w") as file1:
    # Write devices_list in devices_new file
    file1.writelines(devices_list)


with open("devices_new.txt") as file1:
    # Read devices_new file
    print(file1.read())
```

```
cisco router
cisco switches
Juniper Routers
Juniper Switches

['cisco router\n', 'cisco switches\n', 'Juniper Ro
uters\n', 'Juniper Switches\n', 'cisco Firewall\n'
, 'cisco IOS 7200\n', 'Juniper 3600\n', 'Juniper 3
000\n']
['cisco router\n', 'cisco switches\n', 'Juniper Ro
uters\n', 'Cisco 2200 Series\n', 'Juniper Switches
\n', 'cisco Firewall\n', 'cisco IOS 7200\n', 'Juni
per 3600\n', 'Juniper 3000\n']
cisco router
cisco switches
Juniper Routers
Cisco 2200 Series
Juniper Switches
cisco Firewall
cisco IOS 7200
Juniper 3600
Juniper 3000
```

In [35]:

```
data = int(input("Enter Some data:"))
```

```
Enter Some data:hello

---------------------------------------------------
-------------------------
ValueError
Traceback (most recent call last)
<ipython-input-35-42575b99cd48> in <module>
----> 1 data = int(input("Enter Some data:"))

ValueError: invalid literal for int() with base 10
: 'hello'
```

In [38]:

```
try:
    data = int(input("Enter Some data:"))
except ValueError as emsg:
    print("Integer Data Only::",emsg)
else:
    print("Your data is %d"%data)
```

```
Enter Some data:hello
Integer Data Only:: invalid literal for int() with
base 10: 'hello'
```

In [39]:

```
while True:
    try:
        data = int(input("Enter Some data:"))
    except ValueError as emsg:
        print("Integer Data Only::",emsg)
    else:
        print("Your data is %d"%data)
        break
```

```
Enter Some data:hrer
Integer Data Only:: invalid literal for int() with
base 10: 'hrer'
Enter Some data:344
Your data is 344
```

In [43]:

```
help(Exception)
```

```
Help on class Exception in module builtins:

class Exception(BaseException)
 |  Common base class for all non-exit exceptions.
 |
 |  Method resolution order:
 |      Exception
 |      BaseException
 |      object
 |
 |  Methods defined here:
 |
 |  __init__(self, /, *args, **kwargs)
 |      Initialize self.  See help(type(self)) for
accurate signature.
 |
 |  __new__(*args, **kwargs) from builtins.type
 |      Create and return a new object.  See help(
type) for accurate signature.
 |
 |  ----------------------------------------------
----------------------
 |  Methods inherited from BaseException:
 |
 |  __delattr__(self, name, /)
 |      Implement delattr(self, name).
 |
 |  __getattribute__(self, name, /)
 |      Return getattr(self, name).
 |
 |  __reduce__(...)
 |      helper for pickle
 |
 |  __repr__(self, /)
 |      Return repr(self).
 |
 |  __setattr__(self, name, value, /)
 |      Implement setattr(self, name, value).
 |
 |  __setstate__(...)
 |
 |  __str__(self, /)
 |      Return str(self).
 |
```

```
  |  with_traceback(...)
  |      Exception.with_traceback(tb) --
  |      set self.__traceback__ to tb and return se
lf.
  |
  |  ----------------------------------------------
----------------------
  |  Data descriptors inherited from BaseException:
  |
  |  __cause__
  |      exception cause
  |
  |  __context__
  |      exception context
  |
  |  __dict__
  |
  |  __suppress_context__
  |
  |  __traceback__
  |
  |  args
```

In [56]:

```
%%writefile shortmod.py

'''ShortInput Exception Module'''

class ShortInput(Exception):
    '''Short Input Exception for short Message'''
    def __init__(self, length, atleast ):
        '''Initialise ShortInput Exception Class'''
        self.length = length
        self.atleast = atleast
        print("You Enter %d bytes of data \
Expected %d bytes"%(length, atleast))

if __name__ == '__main__':
    ob = ShortInput(6,8)
```

```
Overwriting shortmod.py
```

In [61]:

```
%%writefile ShortInputEx.py

from shortmod import ShortInput

while True:
    try:
        data = input("Enter some data: ")
        if(len(data) < 6):
            raise ShortInput(len(data),6)
    except ShortInput as e:
        print("Enter minimum %d bytes of data"%e.atleast)
    except Exception as e1:
        print("Other Exception",e1)
    else:
        print("Data is %s"%data)
        break
    finally:
        print("It is Always Executed")
```

```
Overwriting ShortInputEx.py
```

In [62]:

```
run ShortInputEx.py
```

```
Enter some data: hello
You Enter 5 bytes of data Expected 6 bytes
Enter minimum 6 bytes of data
It is Always Executed
Enter some data: surendra
Data is surendra
It is Always Executed
```

In [ ]:

```

```