In [27]:

```
Agenda='''
1. regular expression
2. Fileinput, glob, sys and re
case study.'
3.Itertools and Collections
framework ( generator,
yield, deque )
4.Threading, Logging and
Debugging ( Example
Optional Topic )
5. numpy
6. scipy
7. Unittesting ( if time permit)
8. Decorator
9. Group Activities
'''
```

In [2]:

```
import re
```

In [3]:

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

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

In [10]:

```
m=p.match('7897854 Surendra Panpaliya')
m
```

In [11]:

```
m.group()
```

```
---------------------------------------------------------------------
-------
AttributeError                            Traceback (most recent cal
l last)
<ipython-input-11-b9a6738fa293> in <module>()
----> 1 m.group()

AttributeError: 'NoneType' object has no attribute 'group'
```

In [6]:

```
1 m.span()
```

Out[6]:

```
(0, 8)
```

In [7]:

```
1 m.start()
```

Out[7]:

```
0
```

In [8]:

```
1 m.end()
```

Out[8]:

```
8
```

In [12]:

```
1 m1=p.search('7897854 Surendra Panpaliya')
2 m1
```

Out[12]:

```
<_sre.SRE_Match object; span=(8, 16), match='Surendra'>
```

In [13]:

```
1 m1.group()
```

Out[13]:

```
'Surendra'
```

In [14]:

```
1 m1.span()
```

Out[14]:

```
(8, 16)
```

In [15]:

```
1 Mobiles=re.findall('\d{10}',open('profile.txt').read())
2 print(Mobiles)
```

```
['9975072320', '6240010539', '8308761477', '8240010539']
```

In [16]:

```
1 cat profile.txt
```

```
Total 16 + years of Experience in Corporate Training and Consultancy
.
Director, International Corporate Trainer and Consultant at GKTCS In
novations Pvt. Ltd. Pune.
500 + Corporate Training
This is new line inserted
Python Programming And Shell Scripting
This is new line inserted
Taught 10000 + Academic & Industrial Trainees.
vt. Ltd., Rolta India Ltd., Tieto, Siemens,   Tech Mahindra, HCL, OR
ACLE, Verizon, Ariecent Technology, Helwett Packard, Patni, Shell Gr
oup Lagos, Africa, Tieto,
Synechron Techologies, Johnson Control, Alstom India Ltd, Airbus, Ca
pgemini, Harman International Pvt. Ltd, Amadeus, Vodaphone, Sears IT
services ,RSystem, Delhi, Crest Pune etc
 My Email address is surendra@gktcs.com
  My Mobile Number is 9975072320
  My Account Number 624001053983
  My Another Email address is surendra.panpaliya@gmail.com
  My Alternate Phone no 8308761477
  My Another Account No. 824001053987Thread for Appending new line
Thread for Appending new line
This is New Line Now
This is New Line Now
It's last line in the file
It's last line in the file
It's last line in the file
It's last line in the file
It's last line in the file
It's last line in the file
It's last line in the file
It's last line in the file
It's last line in the file
This is New Line Now
```

In [18]:

```
1 Mobiles=re.findall('\\b\d{10}\\b',open('profile.txt').read())
2 print(Mobiles)
```

```
['9975072320', '8308761477']
```

In [19]:

```
1 Mobiles=re.findall(r'\b\d{10}\b',open('profile.txt').read())
2 print(Mobiles)
```

```
['9975072320', '8308761477']
```

In [24]:

```
data1=r'''
Total 18 + years of Experience in Corporate \n
Training and Consultancy.
Director, International \t Corporate Trainer and Consultant
\n at GKTCS Innovations Pvt. Ltd. Pune.
500 + Corporate Training \n
This is new line inserted
Python Programming \n  And Shell Scripting
'''
```

In [25]:

```
print(data1)
```

```
Total 18 + years of Experience in Corporate \n
Training and Consultancy.
Director, International \t Corporate Trainer and Consultant
\n at GKTCS Innovations Pvt. Ltd. Pune.
500 + Corporate Training \n
This is new line inserted
Python Programming \n  And Shell Scripting
```

In [26]:

```
data1
```

Out[26]:

```
'\nTotal 18 + years of Experience in Corporate \\n\nTraining and Con
sultancy. \nDirector, International \\t Corporate Trainer and Consul
tant \n\\n at GKTCS Innovations Pvt. Ltd. Pune.\n500 + Corporate Tra
ining \\n\nThis is new line inserted\nPython Programming \\n  And Sh
ell Scripting\n'
```

In [31]:

```
Emails=re.findall(r'\b[.\w]+@\w+.[a-z.]{2,7}\b',open('profile.txt').read())
print(Emails)
```

```
['surendra@gktcs.com', 'surendra.panpaliya@gmail.com']
```

In [32]:

```
Mobiles=re.sub(r'\b\d{10}\b','xxMobilexx',open('profile.txt').read())
print(Mobiles)
```

```
Total 16 + years of Experience in Corporate Training and Consultancy
.
Director, International Corporate Trainer and Consultant at GKTCS In
novations Pvt. Ltd. Pune.
500 + Corporate Training
This is new line inserted
Python Programming And Shell Scripting
This is new line inserted
Taught 10000 + Academic & Industrial Trainees.
vt. Ltd., Rolta India Ltd., Tieto, Siemens,   Tech Mahindra, HCL, OR
ACLE, Verizon, Ariecent Technology, Helwett Packard, Patni, Shell Gr
oup Lagos, Africa, Tieto,
Synechron Techologies, Johnson Control, Alstom India Ltd, Airbus, Ca
pgemini, Harman International Pvt. Ltd, Amadeus, Vodaphone, Sears IT
services ,RSystem, Delhi, Crest Pune etc
 My Email address is surendra@gktcs.com
  My Mobile Number is xxMobilexx
  My Account Number 624001053983
  My Another Email address is surendra.panpaliya@gmail.com
  My Alternate Phone no xxMobilexx
  My Another Account No. 824001053987Thread for Appending new line
Thread for Appending new line
This is New Line Now
This is New Line Now
It's last line in the file
It's last line in the file
It's last line in the file
It's last line in the file
It's last line in the file
It's last line in the file
It's last line in the file
It's last line in the file
It's last line in the file
This is New Line Now
```

In [34]:

```
Emails=re.sub(r'\b[.\w]+@\w+.[a-z.]{2,7}\b','Emailxx@xx',open('profile.txt').
print(Emails)
```

```
Total 16 + years of Experience in Corporate Training and Consultancy
.
Director, International Corporate Trainer and Consultant at GKTCS In
novations Pvt. Ltd. Pune.
500 + Corporate Training
This is new line inserted
Python Programming And Shell Scripting
This is new line inserted
Taught 10000 + Academic & Industrial Trainees.
vt. Ltd., Rolta India Ltd., Tieto, Siemens,   Tech Mahindra, HCL, OR
ACLE, Verizon, Ariecent Technology, Helwett Packard, Patni, Shell Gr
oup Lagos, Africa, Tieto,
Synechron Techologies, Johnson Control, Alstom India Ltd, Airbus, Ca
pgemini, Harman International Pvt. Ltd, Amadeus, Vodaphone, Sears IT
services ,RSystem, Delhi, Crest Pune etc
 My Email address is Emailxx@xx
  My Mobile Number is 9975072320
  My Account Number 624001053983
  My Another Email address is Emailxx@xx
  My Alternate Phone no 8308761477
  My Another Account No. 824001053987Thread for Appending new line
Thread for Appending new line
This is New Line Now
This is New Line Now
It's last line in the file
It's last line in the file
It's last line in the file
It's last line in the file
It's last line in the file
It's last line in the file
It's last line in the file
It's last line in the file
It's last line in the file
This is New Line Now
```

In [35]:

```
Emails=re.split(r'\b[.\w]+@\w+.[a-z.]{2,7}\b',open('profile.txt').read())
print(Emails)
```

```
['\nTotal 16 + years of Experience in Corporate Training and Consult
ancy.\nDirector, International Corporate Trainer and Consultant at G
KTCS Innovations Pvt. Ltd. Pune.\n500 + Corporate Training\nThis is 
new line inserted\nPython Programming And Shell Scripting\nThis is n
ew line inserted\nTaught 10000 + Academic & Industrial Trainees.\nvt
. Ltd., Rolta India Ltd., Tieto, Siemens,  Tech Mahindra, HCL, ORAC
LE, Verizon, Ariecent Technology, Helwett Packard, Patni, Shell Grou
p Lagos, Africa, Tieto, \nSynechron Techologies, Johnson Control, Al
stom India Ltd, Airbus, Capgemini, Harman International Pvt. Ltd, Am
adeus, Vodaphone, Sears IT services ,RSystem, Delhi, Crest Pune etc\
n My Email address is ', '\n  My Mobile Number is 9975072320\n  My A
ccount Number 624001053983\n  My Another Email address is ', "\n  My
Alternate Phone no 8308761477\n  My Another Account No. 824001053987
Thread for Appending new line\nThread for Appending new line\nThis i
s New Line Now\nThis is New Line Now\nIt's last line in the file\nIt
's last line in the file\nIt's last line in the file\nIt's last line
in the file\nIt's last line in the file\nIt's last line in the file\
nIt's last line in the file\nIt's last line in the file\nIt's last l
ine in the file\nThis is New Line Now\n"]
```

In [49]:

```
import glob
Files=glob.glob("pr*.txt")
```

In [47]:

```
import fileinput

FR=fileinput.input(Files)
#print(FR)
for line in FR:
    print(FR.filename(),line.strip(),FR.lineno())
```

```
profile4.txt  1
profile4.txt Total 16 + years of Experience in Corporate Training an
d Consultancy. 2
profile4.txt Director, International Corporate Trainer and Consultan
t at GKTCS Innovations Pvt. Ltd. Pune. 3
profile4.txt 500 + Corporate Training 4
profile4.txt Taught 10000 + Academic & Industrial Trainees. 5
profile4.txt Major Skills :-> 6
profile4.txt  7
profile4.txt 1. Active Python 2.7, Ipython, Python 3.5, Python xml p
arsing, Python Database Management, Machine Learning in Python, Ansi
ble, Scikit learning, Numpy, Scipy, Pandas, Matplotlib, OpenCV, Data
Analysis , Pylab, Anaconda, XML, JSON, REST Api, Python Socket Progr
amming, Python Decorator, Python Multithreading, Python Unit Testing
, Django,ORM, Django Forms, Django Models, Pylon, MYSQL Database, SQ
LAlchemy, Jython, Iron Python, Kivy, Robot framework, Selenium, Appi
um, MongoDB etc. 8
profile4.txt 2. Ruby, Rails, Rspec, Pry, Appium, Cucumber,Watir, And
roid, Automation Testing,Calalbash, Selenium etc. 9
```

In [48]:

```
dir(FR)
```

Out[48]:

```
['__class__',
 '__del__',
 '__delattr__',
 '__dict__',
 '__dir__',
 '__doc__',
 '__enter__',
 '__eq__',
 '__exit__',
 '__format__',
 '__ge__',
 '__getattribute__',
 '__getitem__',
 '__gt__',
 '__hash__',
 '__init__',
 '__init_subclass__',
 '__iter__',
 '__le__',
 '__lt__',
 '__module__',
 '__ne__',
```

```
'__new__',
'__next__',
'__reduce__',
'__reduce_ex__',
'__repr__',
'__setattr__',
'__sizeof__',
'__str__',
'__subclasshook__',
'__weakref__',
'_backup',
'_backupfilename',
'_file',
'_filelineno',
'_filename',
'_files',
'_inplace',
'_isstdin',
'_mode',
'_openhook',
'_output',
'_readline',
'_savestdout',
'_startlineno',
'close',
'filelineno',
'filename',
'fileno',
'isfirstline',
'isstdin',
'lineno',
'nextfile',
'readline']
```

In [66]:

```
import fileinput
import glob
import re

Files=glob.glob("pr*.txt")
fileinput.close()
FR=fileinput.input(Files)
for line in FR:
    if re.search(r'\b\d{10}\b',line):
        print(" %s \t %s\t %s\t"%(FR.filename(),line.strip(),FR.lineno()))
```

```
profile3.txt    My Mobile Number is 9975072320  47
profile3.txt    My Alternate Phone no 8308761477        50
profile3.txt    My Mobile Number is 9975072320  61
profile3.txt    My Alternate Phone no 8308761477        64
profile.txt     My Mobile Number is 9975072320  85
profile.txt     My Alternate Phone no 8308761477        88
```

In [67]:

```
import collections
```

```
2 dir(collections)
```

Out[67]:

```
['AsyncGenerator',
 'AsyncIterable',
 'AsyncIterator',
 'Awaitable',
 'ByteString',
 'Callable',
 'ChainMap',
 'Collection',
 'Container',
 'Coroutine',
 'Counter',
 'Generator',
 'Hashable',
 'ItemsView',
 'Iterable',
 'Iterator',
 'KeysView',
 'Mapping',
 'MappingView',
 'MutableMapping',
 'MutableSequence',
 'MutableSet',
 'OrderedDict',
 'Reversible',
 'Sequence',
 'Set',
 'Sized',
 'UserDict',
 'UserList',
 'UserString',
 'ValuesView',
 '_Link',
 '_OrderedDictItemsView',
 '_OrderedDictKeysView',
 '_OrderedDictValuesView',
 '__all__',
 '__builtins__',
 '__cached__',
 '__doc__',
 '__file__',
 '__loader__',
 '__name__',
 '__package__',
 '__path__',
 '__spec__',
 '_chain',
 '_class_template',
 '_collections_abc',
 '_count_elements',
 '_eq',
 '_field_template',
 '_heapq',
 '_iskeyword',
 '_itemgetter',
```

```
 '_proxy',
 '_recursive_repr',
 '_repeat',
 '_repr_template',
 '_starmap',
 '_sys',
 'abc',
 'defaultdict',
 'deque',
 'namedtuple']
```

In [68]:

```
from collections import deque

DQ=deque(['python','jython','perl','java'])
print(DQ)
```

```
deque(['python', 'jython', 'perl', 'java'])
```

In [69]:

```
DQ.appendleft('CPP')
print(DQ)
```

```
deque(['CPP', 'python', 'jython', 'perl', 'java'])
```

In [70]:

```
DQ.popleft()
print(DQ)
```

```
deque(['python', 'jython', 'perl', 'java'])
```

In [71]:

```
DQ.rotate()
print(DQ)
```

```
deque(['java', 'python', 'jython', 'perl'])
```

In [72]:

```
from collections import OrderedDict

OD=OrderedDict()
OD
```

Out[72]:

```
OrderedDict()
```

In [73]:

```
OD['p']='python'
```

In [74]:

```
1 OD
```

Out[74]:

```
OrderedDict([('p', 'python')])
```

In [75]:

```
1 OD.update({'j':'jython'})
```

In [76]:

```
1 OD
```

Out[76]:

```
OrderedDict([('p', 'python'), ('j', 'jython')])
```

In [77]:

```
1 OD.keys()
```

Out[77]:

```
odict_keys(['p', 'j'])
```

In [78]:

```
1 OD.values()
```

Out[78]:

```
odict_values(['python', 'jython'])
```

In [79]:

```
1 D=dict(OD)
```

In [80]:

```
1 D
```

Out[80]:

```
{'j': 'jython', 'p': 'python'}
```

In [82]:

```
1 #dir(OD)
```

In [83]:

```
def gy():
    x,y=40,6
    yield x
    yield y
    yield x+y
    yield x*y
    yield x/y
    yield x%y
    return
```

In [84]:

```
g=gy()
print(g)
```

```
<generator object gy at 0x109cbec50>
```

In [85]:

```
print(g.__next__())
print(g.__next__())
print(g.__next__())
print(g.__next__())
print(g.__next__())
print(g.__next__())
```

```
40
6
46
240
6.666666666666667
4
```

In [86]:

```
print(g.__next__())
print(g.__next__())
```

```
---------------------------------------------------------------------
-------
StopIteration                             Traceback (most recent cal
l last)
<ipython-input-86-5e43d22919c1> in <module>()
----> 1 print(g.__next__())
      2 print(g.__next__())

StopIteration:
```

In [87]:

```
g1=gy()
print(g1)
for i in g1:
    print(i)
```

```
<generator object gy at 0x109bd7360>
40
6
46
240
6.666666666666667
4
```

In [88]:

```
def fib(n):
    '''This is fibbonaci sequence for n number'''
    a,b=0,1   # parallel assignment
    yield a
    while b<n:
        yield b
        a,b=b,a+b  # swapping of a and b value
```

In [89]:

```
f=fib(50)
for i in f:
    print(i)
```

```
0
1
1
2
3
5
8
13
21
34
```

In [90]:

```
#import threading
```

In [91]:

```
import numpy as np

L1=list(range(1,10))
L2=list(range(11,20))
print(L1)
print(L2)
```

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

In [92]:

```
L3=L1+L2
print(L3)
```

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

In [93]:

```
A1=np.array(L1)
print(A1)
```

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

In [94]:

```
type(A1)
```

Out[94]:

```
numpy.ndarray
```

In [95]:

```
A2=np.array(L2)
print(A2)
```

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

In [96]:

```
A3=A1+A2
print(A3)
```

```
[12 14 16 18 20 22 24 26 28]
```

In [97]:

```
A1.ndim
```

Out[97]:

```
1
```

In [98]:

```
A1.shape
```

Out[98]:

```
(9,)
```

In [99]:

```
A4=A1.reshape(3,3)
print(A4)
```

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

In [100]:

```
A4.ndim
```

Out[100]:

```
2
```

In [102]:

```
#dir(A1)
```

In [104]:

```
A4.shape
```

Out[104]:

```
(3, 3)
```

In [106]:

```
A5=A4**2
print(A5)
```

```
[[ 1  4  9]
 [16 25 36]
 [49 64 81]]
```

In [107]:

```
A6=A5.T
print(A6)
```

```
[[ 1 16 49]
 [ 4 25 64]
 [ 9 36 81]]
```

In [109]:

```
AT=A4.T
print(AT)
```

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

In [110]:

```
A4.dtype
```

Out[110]:

```
dtype('int64')
```

In [111]:

```
A7=A4.astype(np.float32)
print(A7)
```

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

In [112]:

```
A7.dtype
```

Out[112]:

```
dtype('float32')
```

In [113]:

```
A7.itemsize
```

Out[113]:

```
4
```

In [114]:

```
A8=np.arange(0,15)
print(A8)
```

```
[ 0  1  2  3  4  5  6  7  8  9 10 11 12 13 14]
```

In [115]:

```
L4=['a','b','c','d',6,7,8,9]

A9=np.asarray(L4)
print(A9)
```

```
['a' 'b' 'c' 'd' '6' '7' '8' '9']
```

In [116]:

```
A9.dtype
```

Out[116]:

```
dtype('<U1')
```

In [117]:

```
L5=['a','b','c','d']
A10=np.array(L5)
print(A10)
```

```
['a' 'b' 'c' 'd']
```

In [118]:

```
A10.dtype
```

Out[118]:

```
dtype('<U1')
```

In [119]:

```
A11=np.arange(1,16)
print(A11)
```

```
[ 1  2  3  4  5  6  7  8  9 10 11 12 13 14 15]
```

In [120]:

```
A12=A11.reshape(3,5)
print(A12)
```

```
[[ 1  2  3  4  5]
 [ 6  7  8  9 10]
 [11 12 13 14 15]]
```

In [125]:

```
A13=A12[1:,2:4]
print(A13)
```

```
[[ 8  9]
 [13 14]]
```

In [130]:

```
A12[:,2:]
```

Out[130]:

```
array([[ 3,  4,  5],
       [ 8,  9, 10],
       [13, 14, 15]])
```

In [134]:

```
A12[::1,::2]
```

Out[134]:

```
array([[ 1,  3,  5],
       [ 6,  8, 10],
       [11, 13, 15]])
```

In [135]:

```
M1=np.matrix(A4)
print(M1)
```

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

In [137]:

```
print(A4)
```

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

In [139]:

```
print(A1)
M2=np.matrix(A1)
print(M2)
```

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

In [140]:

```
M2.ndim
```

Out[140]:

```
2
```

In [141]:

```
A1.ndim
```

Out[141]:

```
1
```

In [142]:

```
M3=M2.T
print(M3)
```

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

In [143]:

```
M1
```

Out[143]:

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

In [144]:

```
M1.T
```

Out[144]:

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

In [147]:

```
1 import unittest
2 dir(unittest)
```

Out[147]:

```
['BaseTestSuite',
 'FunctionTestCase',
 'SkipTest',
 'TestCase',
 'TestLoader',
 'TestProgram',
 'TestResult',
 'TestSuite',
 'TextTestResult',
 'TextTestRunner',
 '_TextTestResult',
 '__all__',
 '__builtins__',
 '__cached__',
 '__doc__',
 '__file__',
 '__loader__',
 '__name__',
 '__package__',
 '__path__',
 '__spec__',
 '__unittest',
 'case',
 'defaultTestLoader',
 'expectedFailure',
 'findTestCases',
 'getTestCaseNames',
 'installHandler',
 'load_tests',
 'loader',
 'main',
 'makeSuite',
 'registerResult',
 'removeHandler',
 'removeResult',
 'result',
 'runner',
 'signals',
 'skip',
 'skipIf',
 'skipUnless',
 'suite',
 'util']
```

In [ ]:

```
# %load testfun1.py

import unittest

def fun(n):
    '''fun Function'''
    return n+3
def add(x,y):
    '''addition of x and y'''
    return x+y

class A:
    '''Class A'''
    def __init__(self,age):
        self.age=age

    def display(self):
        return self.age

s="surendra"

class MyTest(unittest.TestCase):
    def test(self):
        self.assertEqual(fun(4),7)

    def testA(self):
        self.assertEqual(add(4,7),11)

    def testB(self):
        self.assertEqual(A(40).display(),40)


class EcoBankTest(unittest.TestCase):
    def testC(self):
        self.assertEqual(s.upper(),'SURENDRA')


unittest.main()
```

In [150]:

```
run testfun1.py -v
```

```
testC (__main__.EcoBankTest) ... ok
test (__main__.MyTest) ... ok
testA (__main__.MyTest) ... ok
testB (__main__.MyTest) ... ok

----------------------------------------------------------------------
--
Ran 4 tests in 0.004s

OK
```

In [151]:

In [151]:

```
1 import unittest
2
3 dir(unittest.TestCase)
```

Out[151]:

```
['__call__',
 '__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__',
 '_addExpectedFailure',
 '_addSkip',
 '_addUnexpectedSuccess',
 '_baseAssertEqual',
 '_classSetupFailed',
 '_deprecate',
 '_diffThreshold',
 '_feedErrorsToResult',
 '_formatMessage',
 '_getAssertEqualityFunc',
 '_truncateMessage',
 'addCleanup',
 'addTypeEqualityFunc',
 'assertAlmostEqual',
 'assertAlmostEquals',
 'assertCountEqual',
 'assertDictContainsSubset',
 'assertDictEqual',
 'assertEqual',
 'assertEquals',
 'assertFalse',
 'assertGreater',
 'assertGreaterEqual',
```

```
'assertIn',
'assertIs',
'assertIsInstance',
'assertIsNone',
'assertIsNot',
'assertIsNotNone',
'assertLess',
'assertLessEqual',
'assertListEqual',
'assertLogs',
'assertMultiLineEqual',
'assertNotAlmostEqual',
'assertNotAlmostEquals',
'assertNotEqual',
'assertNotEquals',
'assertNotIn',
'assertNotIsInstance',
'assertNotRegex',
'assertNotRegexpMatches',
'assertRaises',
'assertRaisesRegex',
'assertRaisesRegexp',
'assertRegex',
'assertRegexpMatches',
'assertSequenceEqual',
'assertSetEqual',
'assertTrue',
'assertTupleEqual',
'assertWarns',
'assertWarnsRegex',
'assert_',
'countTestCases',
'debug',
'defaultTestResult',
'doCleanups',
'fail',
'failIf',
'failIfAlmostEqual',
'failIfEqual',
'failUnless',
'failUnlessAlmostEqual',
'failUnlessEqual',
'failUnlessRaises',
'failureException',
'id',
'longMessage',
'maxDiff',
'run',
'setUp',
'setUpClass',
'shortDescription',
'skipTest',
'subTest',
'tearDown',
'tearDownClass']
```

In [ ]:

```
1
```