In [21]:

```
'''
@proerty
Contextmanager
specialmethods
'''
class Box:
    def __init__(self,a1):
        self.V=a1
    def f1(self):
        print(self.V)

obj1=Box("Test1")
print("-->{}".format(obj1))
obj1.f1()
obj1=Box("Test2")
print("-->{}".format(obj1))
obj1.f1()
```

```
--><__main__.Box object at 0x00000000050823A0>
Test1
--><__main__.Box object at 0x00000000050827F0>
Test2
```

In [7]:

```
class Box:
    def __init__(self,a1):
        self.V=a1
        print("Constructor")
    def f1(self):
        print("F1 is called")
        return self.V
    def f2(self,a1):
        print("F2 is called")
        self.V=a1
    rv=property(f1,f2)

obj=Box("Test1")
print(obj.rv)
obj.rv="Test2"
print(obj.rv)
```

```
Constructor
F1 is called
Test1
F2 is called
F1 is called
Test2
```

In [11]:

```
class Box:
    def __init__(self,a1):
        self.V=a1
        print("Constructor")
    @property
    def f1(self):
        print("F1 block")
        return self.V
    @f1.setter
    def f2(self,a1):
        print("F2 block")
        self.V=a1

obj=Box("Test1")
print(obj.f1)
obj.f2="Test2"
print(obj.f1)
```

```
Constructor
F1 block
Test1
F2 block
F1 block
Test2
```

In [14]:

```
class Box:
    def __init__(self,a1):
        self.V=a1
        print("Constructor")
    @property
    def f1(self):
        print("F1 block")
        return self.V
    @f1.setter
    def f1(self,a1):
        print("setter block")
        self.V=a1

    @f1.deleter
    def f1(self):
        print("Deleting value")
        del(self.V)

obj=Box("Test1")
print(obj.f1)
obj.f1="Test2"
print(obj.f1)
del(obj.f1)
```

```
Constructor
F1 block
Test1
setter block
F1 block
Test2
Deleting value
```

In [16]:

```
def f1(a):
    def f2():
        a()
    return f2

@f1
def fA():
    print("fA block")
@f1
def fB():
    print("fB block")
```

```
fB block
```

In [20]:

```
class Box:
    def __init__(self,port=0):
        self.port=port
    @property
    def f1(self):
        return self.port
    @f1.setter
    def f1(self,a1):
        self.port=a1

obj=Box(1000)
print("-->{}".format(obj))
print(obj.f1)
obj.f1=2000
print("-->{}".format(obj))
print(obj.f1)
obj.f1=3000
print("-->{}".format(obj))
print(obj.f1)
```

```
--><__main__.Box object at 0x0000000005082A30>
1000
--><__main__.Box object at 0x0000000005082A30>
2000
--><__main__.Box object at 0x0000000005082A30>
3000
```

In [27]:

```
# Block style - automatically block will end ->descriptor will closed
#
# resource - file,db
#
# __enter__()   __exit__(exception type,name,args)
#
class Box:
    def __enter__(self):
        print("ONE")

    def __exit__(self,a,b,c):
        print("END")


with Box() as obj:
    print("With block")
```

```
ONE
With block
END
```

In [28]:
```
with Box() as obj1:
    print("With Block-1")
```

```
ONE
With Block-1
END
```

In [30]:
```
import requests
# python -m pip install requests (Winx)
# pip3 install requests (non-winx)
r=requests.get("https://www.python.org")
r.status_code
```

Out[30]:
```
200
```

In [33]:
```
r=requests.get('http://www.invalidurl.com')
r.status_code
```

Out[33]:
```
404
```

In [ ]:
```
r=requests.get('http://www.invalidurl.com')
if r.status_code != 200:
    print("URL download is failed")
```

In [ ]:
```
GET request status code

200 - Okay
301 - redirecting different endpoint
400 - bad request
401 - not authenticated; api - require login crediential
404 - not found
503 - server is not ready to handle the request
```

In [ ]:
```
# html -> Page (webpage) page+data
# Enter emp id:|__________|  -->webserver --------> .....
#                 Name,place                           |
#             <html>
#            <head>
#            <title>titlename</title>
#             <body>
#             <p> ....</p>
#             <p><h2><b>Name:_____</p></h2></b>
#             .... Place:________</>
#             ..
#             </html>

# json --> Data only - not page
```

In [35]:
```
r=requests.get('https://www.python.org')
r.headers['Content-Type']
```

Out[35]:
```
'text/html; charset=utf-8'
```

In [37]:
```
r=requests.get('https://reqres.in/api/products/3')
r.status_code
r.headers['Content-Type']
```

Out[37]:
```
'application/json; charset=utf-8'
```

In [41]:
```
r=requests.get('https://www.google.com')
s=r.text
print(type(s),len(s))
```

```
<class 'str'> 15157
```

In [42]:
```
with open("D:\\test1.html","w") as WH:
        WH.write(s+"\n")
```

In [54]:
```
r=requests.get('https://reqres.in/api/products/3')
jd=r.text
type(jd)
```

Out[54]:
```
str
```

In [56]:
```
import json
pd=json.loads(jd)
type(pd)
print(pd)
```

```
{'data': {'id': 3, 'name': 'true red', 'year': 2002, 'color': '#BF1932', 'panto
ne_value': '19-1664'}, 'support': {'url': 'https://reqres.in/#support-heading',
'text': 'To keep ReqRes free, contributions towards server costs are appreciate
d!'}}
```

In [58]:
```
import pprint
pprint.pprint(pd)
```

```
{'data': {'color': '#BF1932',
          'id': 3,
          'name': 'true red',
          'pantone_value': '19-1664',
          'year': 2002},
 'support': {'text': 'To keep ReqRes free, contributions towards server costs '
                     'are appreciated!',
             'url': 'https://reqres.in/#support-heading'}}
```

In [60]:
```
pd['support']['url']='https://updated_url.com'
pprint.pprint(pd)
```

```
{'data': {'color': '#BF1932',
          'id': 3,
          'name': 'true red',
          'pantone_value': '19-1664',
          'year': 2002},
 'support': {'text': 'To keep ReqRes free, contributions towards server costs '
                     'are appreciated!',
             'url': 'https://updated_url.com'}}
```

In [49]:
```
emp={"NAME":"Arun","dept":"SALES","EID":1234}
jd=json.dumps(emp)
json.loads(jd)
```

Out[49]:
```
{'NAME': 'Arun', 'dept': 'SALES', 'EID': 1234}
```

In [50]:
```
s={"K":["V1","V2"]}
jd=json.dumps(s)
json.loads(jd)
```

Out[50]:
```
{'K': ['V1', 'V2']}
```

In [69]:
```
import requests
import json

r=requests.get("https://api.github.com/users/hadley/repos")
r.status_code
page=r.text

pd=json.loads(page)
type(pd)
len(pd)
type(pd[0]) # list of dict ->[{}]
len(pd[0].keys())
print(pd[0]['id'])
print(pd[0]['owner']['id'])
```

```
40423928
4196
```

In [70]:
```
emp={"NAME":"Arun","dept":["sales","QA"],"eid":1234}
with open("D:\\test1.json","w") as WH:
    json.dump(emp,WH)
```

In [71]:
```
with open("D:\\test1.json") as FH:
    pd=json.load(FH)
pd
```

Out[71]:
```
{'NAME': 'Arun', 'dept': ['sales', 'QA'], 'eid': 1234}
```

In [ ]:
```
import sqlite3
con=sqlite3.connect('')
st=con.cursor()
st.execute('select *from table')
r=st.fetchall() ->[(),()]

jd=json.dumps(r)
            # |__ python struct
#
#-- json array ->[index,index]
```

In [ ]:

```
>>> import json
>>>
>>> d={"K1":"V1","K2":"V2","K3":["V1","V2","V3"]}
>>>
>>> json.dumps(d)
'{"K1": "V1", "K2": "V2", "K3": ["V1", "V2", "V3"]}'
>>>
>>> json.dumps(d,indent=2)
'{\n  "K1": "V1",\n  "K2": "V2",\n  "K3": [\n    "V1",\n    "V2",\n    "V3"\n  ]
\n}'
>>> print(json.dumps(d,indent=2))
{
  "K1": "V1",
  "K2": "V2",
  "K3": [
    "V1",
    "V2",
    "V3"
  ]
}
>>> print(json.dumps(d)
... )
{"K1": "V1", "K2": "V2", "K3": ["V1", "V2", "V3"]}
>>>
>>> print(json.dumps(d))
{"K1": "V1", "K2": "V2", "K3": ["V1", "V2", "V3"]}
>>>
>>> print(json.dumps(d,indent=3))
{
   "K1": "V1",
   "K2": "V2",
   "K3": [
      "V1",
      "V2",
      "V3"
   ]
}
>>> print(json.dumps(d,indent=4))
{
    "K1": "V1",
    "K2": "V2",
    "K3": [
        "V1",
        "V2",
        "V3"
    ]
}
>>> print(json.dumps(d,indent=5))
{
     "K1": "V1",
     "K2": "V2",
     "K3": [
          "V1",
          "V2",
          "V3"
     ]
}
>>> print(json.dumps(d,indent=6))
{
      "K1": "V1",
      "K2": "V2",
      "K3": [
            "V1",
            "V2",
            "V3"
      ]
}
>>>
```

In [ ]:

```
v={'user':'root',
  'password':'passend'
  }
requests.get('url',params=v) - download
requests.post('url',params=v) - post
```

In [ ]:

```
CGI - programming
----------------------
Client  ----------Webserver-------------------ServerScripts(.py)
          (1)                     (2)         --------------- (3)
                                                    | ...
                                                    |

            ---------------<----------------
                             (4)

<html>                                             python ->DS+function+class
</html>

Login:input type="text" name="n1" value=""
       _____________
Login: |___root____ |  ===========================>{"n1":'root'}
                                                    ...
                                          print("Content-type:text/html\n")
                                    print("<h2><font color=green>Hello root</h2><
 Hello root

# install any webserver
# configure webserver - client - html
#                     - server - python file
# /cgi-bin/
# ------------ alias

# start a webserver daemon - R+
# open a broswer -> type IP address on the addressbar
# 192.140.45.25
---------------------
  index.html

----------------------

# 192.140.45.25/cgi-bin/p1.py{Enter}
# python response code
#
------------------------------------
# 192.140.45.25/login.html{Enter} -->client page
# ----------------------------------
# Name:|_________|
# passwd:|__________|
#
# (submit)  (cancel)
#   ^^^^
#     |----------------------->--------request --webserver ---->server scripts
#                                                                    |
#                                                                    |
#         ----------------------<---------------------------------
#    Response page
#
```

In [ ]:

```
Broswer(client) ---------PYthon---------DBModule-------DataBase
                  CGI       |
       |--------------------|-----------------------------|
                            |
                   ------------------
                  |                 |
                 json               FileStorage
```

In [ ]:

```
Enter a emp name,emp id,dept from Form --->PYthon---->DB (DB)

# file:enroll.html  - client -> /var/www/enroll.html
# ------------------
# <html>
# <head>
# <form action=/cgi-bin/e1.py method='post'>
#Emp name: <input type='text' name='n1' value=''>
#Emp dept: <input type='text' name='n2' value=''>
#Emp ID: <input type='text' name='n3' value=''>
# ...
# </html>

# file:e1.py  -- server script ->/usr/lib/cgi-bin/e1.py
# ---------
import cgi
import sqlite3

obj=cgi.FieldStorage()
def fx(en,eid,ed):
        con=sqlite3.connect('emp.db')
        sth=con.cursor()
        sth.execute("create table <tbname>.... ename,eid,edept")
        sth.execute("insert into <table> value (?,?,?)",(en,eid,ed))
for var in obj:
    if(obj.getvalue(var)):
        print("<h1>....Hey...you not filled ..details")
    else:
        ename=obj.getvalue('n1')
        edept=obj.getvalue('n2')
        eid=obj.getvalue('n3')
        fx(ename,edept,eid)

Enter emp id:_____  =======> python {"n3":123}
                                        |
                                    v=d['n3']
                            r=fy(v)
                            for v in r:
                                 print("<h2>{}\t{}\t{}\t{}".format(...))
    def fy(eid):
        r=sth.execute("select *from table where=?",(eid))
        return r
```

In [72]:

```
import socket
sobj=socket.socket(socket.AF_INET,socket.SOCK_STREAM)
server                             client
socket                             socket
bind                               connect
listen    -----<-------------------
accept ---------------------------->
           ------------<-----------------

send                              recive
sendall                            ..
```

In [ ]:

```
# Connect
client = SSHClient()
client.connect(host, port, username)
# Obtain session
session = client.get_transport().open_session()
# Forward local agent
AgentRequestHandler(session)
# Commands executed after this point will see the forwarded agent on
# the remote end.
session.exec_command("git clone https://my.git.repository/")
```

In [ ]:

```
file:server.log

client = SSHClient()
for var in open("server.log"):
    cobj=client.connect(var...)
    ..
    r=cobj.exec_command("command")
    (STDOUT,STDERR)

result=[]
client=SSHClient()
client.connect('host03.example.com','root'..)
for v in [.....]:
    r=client.exec_command(v)
    result.append(r[0])

with open("/var/log/result.log","w") as WH:
    for v in result:
        WH.write(v)

json.dumps(result) # convert to json
```

In [77]:

```
import os
```

In [78]:

```
import openpyxl
obj=openpyxl.Workbook()
sh=obj.active
sh['A1']='Data1'
sh['A2']='Data2'
sh['B1']=100
sh['B2']=100+200
sh['C5']="result1"
sh['C6']="result2"
sh['F5']=os.popen("whoami").read().strip()
obj.save("D:\\t1.xlsx")
```

In [83]:

```
import openpyxl
obj=openpyxl.Workbook()
sh=obj.active
sh['A1']="Data1"
sh.cell(row=2,column=2).value="Data2"
sh.cell(row=2,column=3).value="Data3"
sh.cell(row=5,column=3).value="Data4"
records=((100,200,300,400),(10,20,30,40),(300,400,500,600))
for var in records:
    sh.append(var)
obj.save("D:\\t1.xlsx")
```

In [86]:

```
import openpyxl
wb=openpyxl.load_workbook('D:\\t1.xlsx')
sh=wb.active
V1=sh['A1']
V2=sh['A2']
V3=sh.cell(row=2,column=2)
print(V1,V2,V3)
print(V1.value,V2.value,V3.value)
```

```
<Cell 'Sheet'.A1> <Cell 'Sheet'.A2> <Cell 'Sheet'.B2>
Data1 None Data2
```

In [94]:

```
import openpyxl
wb=openpyxl.load_workbook('D:\\t1.xlsx')
sh=wb.active

for var in sh['A1':'D8']:
    for v in var:
        if(v.value == None):
            continue
        else:
            print(v.value)
```

```
Data1
Data2
Data3
Data4
100
200
300
400
10
20
30
40
300
400
500
600
```

In [97]:

```
import openpyxl
obj=openpyxl.Workbook()
sh=obj.active
sh['A1']="Data1"
sh.cell(row=2,column=2).value="Data2"
sh.cell(row=2,column=3).value="Data3"
sh.cell(row=5,column=3).value="Data4"
records=((100,200,300,400),(10,20,30,40),(300,400,500,600))
for var in records:
    sh.append(var)
for v in sh.iter_rows(min_row=1,min_col=1,max_row=5,max_col=4):
    for cell in v:
        print(cell.value)
    print("")
obj.save("D:\\t1.xlsx")
```

```
Data1
None
None
None

None
Data2
Data3
None

None
None
None
None

None
None
None
None

None
None
Data4
None
```

In [98]:

```
import openpyxl
obj=openpyxl.Workbook()
sh=obj.active
sh['A1']="Data1"
sh.cell(row=2,column=2).value="Data2"
sh.cell(row=2,column=3).value="Data3"
sh.cell(row=5,column=3).value="Data4"
records=((100,200,300,400),(10,20,30,40),(300,400,500,600))
for var in records:
    sh.append(var)
for v in sh.iter_cols(min_row=1,min_col=1,max_row=5,max_col=4):
    for cell in v:
        print(cell.value)
    print("")
obj.save("D:\\t1.xlsx")
```

```
Data1
None
None
None
None

None
Data2
None
None
None

None
Data3
None
None
Data4

None
None
None
None
None
```

In [102]:

```
from functools import reduce
```

In [103]:

```
wb=openpyxl.load_workbook("D:\\t1.xlsx")
sh=wb.active
L=list()
for v in sh.iter_cols(min_row=1,min_col=5,max_row=10,max_col=5):
    for v1 in v:
        print(v1.value)
        L.append(v1.value)

total=reduce(lambda a,b:a+b,L)
print(total)
```

```
100
150
200
250
300
350
400
450
500
550
3250
```

In [106]:

```
from openpyxl import Workbook

from openpyxl.chart import (
    PieChart,
    ProjectedPieChart,
    Reference
)
from openpyxl.chart.series import DataPoint

data = [
    ['HD1', 'Sold'],
    ['HD2', 150],
    ['HD3', 3560],
    ['HD4', 100],
    ['HD5', 4000],
    ['HD6',450]
]

wb = Workbook()
ws = wb.active

for row in data:
    ws.append(row)

pie = PieChart()
labels = Reference(ws, min_col=1, min_row=2, max_row=5)
data = Reference(ws, min_col=2, min_row=1, max_row=5)
pie.add_data(data, titles_from_data=True)
pie.set_categories(labels)
pie.title = "Pies sold by category"

# Cut the first slice out of the pie
slice = DataPoint(idx=0, explosion=20)
pie.series[0].data_points = [slice]

ws.add_chart(pie, "D1")


ws = wb.create_sheet(title="Projection")

data = [
    ['Page', 'Views'],
    ['Search', 95],
    ['Products', 4],
    ['Offers', 0.5],
    ['Sales', 0.5],
]

for row in data:
    ws.append(row)

projected_pie = ProjectedPieChart()
projected_pie.type = "pie"
projected_pie.splitType = "val" # split by value
labels = Reference(ws, min_col=1, min_row=2, max_row=5)
data = Reference(ws, min_col=2, min_row=1, max_row=5)
```

```
projected_pie.add_data(data, titles_from_data=True)
projected_pie.set_categories(labels)

ws.add_chart(projected_pie, "A10")

from copy import deepcopy
projected_bar = deepcopy(projected_pie)
projected_bar.type = "bar"
projected_bar.splitType = 'pos' # split by position

ws.add_chart(projected_bar, "A27")

wb.save("pie.xlsx")
```

In [108]:

```
import bs4
bs4.BeautifulSoup("<html></html>")
```

Out[108]:

```
<html></html>
```

In [113]:

```
import requests
import bs4
r=requests.get("https://www.google.com")
page=r.text

obj=bs4.BeautifulSoup(page)
obj.find("b")
```

Out[113]:

```
<b class="gb1">Search</b>
```

In [115]:

```
print(obj.prettify())
```

```
<!DOCTYPE html>
<html itemscope="" itemtype="http://schema.org/WebPage" lang="en-IN">
 <head>
  <meta content="text/html; charset=utf-8" http-equiv="Content-Type"/>
  <meta content="/images/branding/googleg/1x/googleg_standard_color_128dp.pn
g" itemprop="image"/>
  <title>
   Google
  </title>
  <script nonce="MSmatZd2M0umpiShA7yiAQ==">
   (function(){window.google={kEI:'GsU4YNHjKPLFz7sPzJ-0oAk',kEXPI:'0,1302440,
56969,954,5105,206,2414,2390,2316,383,246,5,1354,4920,330,1666,1991,2728,1116
131,1232,1196531,519,328985,51224,16114,28684,9188,8384,4858,1362,9290,3026,3
892,850,8000,4841,4020,978,13228,2054,920,873,3599,593,6430,1141,6291,1221,58
74,4521,2774,919,2277,8,85,4304,1279,2212,530,149,1103,840,517,1522,4258,109,
203,1132,4,3,2048,621,2023,1777,520,4269,328,1284,8789,2269,1,957,2845,7,477
4,7580,5096,7877,4928,108,3407,908,2,3555,2397,7470,3275,3,576,1014,1,820,7,4
618,148,5990,5333,991,1661,4,1528,2304,1240,1141,4658,74,1717,266,1107,1519,4
60,1555,4067,5634,1426,374,2110,1714,1297,1753,2658,4242,57,462,912,564,1118,
```

In [122]:

```
for var in obj.find_all("meta"):
    print(var)
print("")
for var in obj.find_all("meta"):
    print(var.get("content"))
```

```
<meta content="text/html; charset=utf-8" http-equiv="Content-Type"/>
<meta content="/images/branding/googleg/1x/googleg_standard_color_128dp.png" it
emprop="image"/>

text/html; charset=UTF-8
/images/branding/googleg/1x/googleg_standard_color_128dp.png
```

In [130]:

```
print(obj.title)
print(obj.title.name)
print(obj.title.string)
print(obj.title.parent.name)
print("")
print(obj.title.parent)
```

```
<title>Google</title>
title
Google
head

<head><meta content="text/html; charset=utf-8" http-equiv="Content-Type"/><me
ta content="/images/branding/googleg/1x/googleg_standard_color_128dp.png" ite
mprop="image"/><title>Google</title><script nonce="MSmatZd2M0umpiShA7yiAQ==">
(function(){window.google={kEI:'GsU4YNHjKPLFz7sPzJ-0oAk',kEXPI:'0,1302440,569
69,954,5105,206,2414,2390,2316,383,246,5,1354,4920,330,1666,1991,2728,111613
1,1232,1196531,519,328985,51224,16114,28684,9188,8384,4858,1362,9290,3026,389
2,850,8000,4841,4020,978,13228,2054,920,873,3599,593,6430,1141,6291,1221,587
4,4521,2774,919,2277,8,85,4304,1279,2212,530,149,1103,840,517,1522,4258,109,2
03,1132,4,3,2048,621,2023,1777,520,4269,328,1284,8789,2269,1,957,2845,7,4774,
7580,5096,7877,4928,108,3407,908,2,3555,2397,7470,3275,3,576,1014,1,820,7,461
8,148,5990,5333,991,1661,4,1528,2304,1240,1141,4658,74,1717,266,1107,1519,46
0,1555,4067,5634,1426,374,2110,1714,1297,1753,2658,4242,57,462,912,564,1118,3
2,3854,1809,2466,3285,2214,1592,713,638,1494,605,2,1483,1886,8,1,1,1600,2361,
55,3008,4369,11,731,665,323,1378,444,3209,464,2539,479,138,373,2,1962,1140,1
7,50,99,1905,1067,6,908,3,107,1220,2036,178,1,651,391,2,2732,2241,538,444,162
7,253,423,1143,1008,2808,449,1187,369,258,38,245,38,662,272,3374,1940,618,126
0,238,956,2,42,17,1015,432,2792,432,3282,287,77,89,50,1555,2,77,749,2,61,448,
57,877,95,444,144,311,2,51,161,186,182,353,1337,229,28,121,2,1182,8,1847,20,4
85,1511,76,440,92,3251,312,163,559,391,942,4,75,32,226,164,2,2,608,290,123,56
88454,1873,1997,35,63,38,5997203,2800707,549,333,444,1,2,80,1,900,896,1,9,2,2
551,1,748,141,801,557,1,4265,1,1,2,1331,3299,248,595,1,2608,155,17,13,72,162,
12,8,1,92,21,36,109,3,17,12,1,23956922,2776205,1234067,267,732,2,148,2,775,25
335',kBL:'iaOK'};google.sn='webhp';google.kHL='en-IN';})();(function(){
var f=[];google.getEI=function(a){for(var b;a&&(!a.getAttribute||!(b=a.getAtt
ribute("eid")));)a=a.parentNode;return b||google.kEI};google.getLEI=function
(a){for(var b=null;a&&(!a.getAttribute||!(b=a.getAttribute("leid")));)a=a.par
entNode;return b};google.ml=function(){return null};google.log=function(a,b,
c,d,g){if(c=google.logUrl(a,b,c,d,g)){a=new Image;var e=f.length;f[e]=a;a.one
rror=a.onload=a.onabort=function(){delete f[e]};a.src=c}};google.logUrl=funct
ion(a,b,c,d,g){var e="";c||-1!=b.search("&ei=")||(e="&ei="+google.getEI(d),-1
==b.search("&lei=")&&(d=google.getLEI(d))&&(e+="&lei="+d));d="";!c&&window._c
shid&&-1==b.search("&cshid=")&&"slh"!=a&&(d="&cshid="+window._cshid);c=c|
|"/"+(g||"gen_204")+"?atyp=i&ct="+a+"&cad="+b+e+"&zx="+Date.now()+d;/^http:/
i.test(c)&&"https:"==window.location.protocol&&(google.ml(Error("a"),!1,{src:
c,glmm:1}),c="");return c};}).call(this);(function(){google.y={};google.x=fun
ction(a,b){if(a)var c=a.id;else{do c=Math.random();while(google.y[c])}google.
y[c]=[a,b];return!1};google.lm=[];google.plm=function(a){google.lm.push.apply
(google.lm,a)};google.lq=[];google.load=function(a,b,c){google.lq.push([[a],
b,c])};google.loadAll=function(a,b){google.lq.push([a,b])};google.bx=!1;googl
e.lx=function(){};}).call(this);google.f={};(function(){
document.documentElement.addEventListener("submit",function(b){var a;if(a=b.t
arget){var c=a.getAttribute("data-submitfalse");a="1"==c||"q"==c&&!a.element
s.q.value?!0:!1}else a=!1;a&&(b.preventDefault(),b.stopPropagation())},!0);do
cument.documentElement.addEventListener("click",function(b){var a;a:{for(a=b.
target;a&&a!=document.documentElement;a=a.parentElement)if("A"==a.tagName){a
="1"==a.getAttribute("data-nohref");break a}a=!1}a&&b.preventDefault()},!
0);}).call(this);
var a=window.location,b=a.href.indexOf("#");if(0<=b){var c=a.href.substring(b
+1);/(^|&)q=/.test(c)&&-1==c.indexOf("#")&&a.replace("/search?"+c.replace(/(^
|&)fp=[^&]*/g,"")+"&cad=h")};</script><style>#gbar,#guser{font-size:13px;padd
ing-top:1px !important;}#gbar{height:22px}#guser{padding-bottom:7px !importan
t;text-align:right}.gbh,.gbd{border-top:1px solid #c9d7f1;font-size:1px}.gbh
{height:0;position:absolute;top:24px;width:100%}@media all{.gb1{height:22px;m
argin-right:.5em;vertical-align:top}#gbar{float:left}}a.gb1,a.gb4{text-decora
tion:underline !important}a.gb1,a.gb4{color:#00c !important}.gbi .gb4{color:#
dd8e27 !important}.gbf .gb4{color:#900 !important}
</style><style>body,td,a,p,.h{font-family:arial,sans-serif}body{margin:0;over
flow-y:scroll}#gog{padding:3px 8px 0}td{line-height:.8em}.gac_m td{line-heigh
t:17px}form{margin-bottom:20px}.h{color:#1558d6}em{font-weight:bold;font-styl
e:normal}.lst{height:25px;width:496px}.gsfi,.lst{font:18px arial,sans-serif}.
gsfs{font:17px arial,sans-serif}.ds{display:inline-box;display:inline-block;m
argin:3px 0 4px;margin-left:4px}input{font-family:inherit}body{background:#ff
f;color:#000}a{color:#4b11a8;text-decoration:none}a:hover,a:active{text-decor
ation:underline}.fl a{color:#1558d6}a:visited{color:#4b11a8}.sblc{padding-to
p:5px}.sblc a{display:block;margin:2px 0;margin-left:13px;font-size:11px}.lsb
b{background:#f8f9fa;border:solid 1px;border-color:#dadce0 #70757a #70757a #d
adce0;height:30px}.lsbb{display:block}#WqQANb a{display:inline-block;margin:0
12px}.lsb{background:url(/images/nav_logo229.png) 0 -261px repeat-x;border:no
ne;color:#000;cursor:pointer;height:30px;margin:0;outline:0;font:15px arial,s
ans-serif;vertical-align:top}.lsb:active{background:#dadce0}.lst:focus{outlin
e:none}</style><script nonce="MSmatZd2M0umpiShA7yiAQ=="></script></head>
```

In [133]:

```python
# How to extract all the URLs from python.org portal?
r=requests.get("https://www.python.org")
page=r.text

obj=bs4.BeautifulSoup(page)
for var in obj.find_all("a"):
    print(var.get("href"))
```

```
#content
#python-network
/
/psf-landing/
https://docs.python.org (https://docs.python.org)
https://pypi.org/ (https://pypi.org/)
/jobs/
/community-landing/
#top
/
https://psfmember.org/civicrm/contribute/transact?reset=1&id=2 (https://psfme
mber.org/civicrm/contribute/transact?reset=1&id=2)
#site-map
#
javascript:;
javascript:;
javascript:;
#
https://www.facebook.com/pythonlang?fref=ts (https://www.facebook.com/pythonl
ang?fref=ts)
```

In [134]:
```
import re
# How to extract all the URLs from python.org portal?
r=requests.get("https://www.python.org")
page=r.text

obj=bs4.BeautifulSoup(page)
for var in obj.find_all("a"):
    if(re.search("^https|^http",var.get("href"))):
        print(var.get("href"))
```

```
https://docs.python.org (https://docs.python.org)
https://pypi.org/ (https://pypi.org/)
https://psfmember.org/civicrm/contribute/transact?reset=1&id=2 (https://psfme
mber.org/civicrm/contribute/transact?reset=1&id=2)
https://www.facebook.com/pythonlang?fref=ts (https://www.facebook.com/pythonl
ang?fref=ts)
https://twitter.com/ThePSF (https://twitter.com/ThePSF)
http://brochure.getpython.info/ (http://brochure.getpython.info/)
https://docs.python.org/3/license.html (https://docs.python.org/3/license.htm
l)
https://wiki.python.org/moin/BeginnersGuide (https://wiki.python.org/moin/Beg
innersGuide)
https://devguide.python.org/ (https://devguide.python.org/)
https://docs.python.org/faq/ (https://docs.python.org/faq/)
http://wiki.python.org/moin/Languages (http://wiki.python.org/moin/Languages)
http://python.org/dev/peps/ (http://python.org/dev/peps/)
https://wiki.python.org/moin/PythonBooks (https://wiki.python.org/moin/Python
Books)
https://wiki.python.org/moin/ (https://wiki.python.org/moin/)
http://planetpython.org/ (http://planetpython.org/)
http://pyfound.blogspot.com/ (http://pyfound.blogspot.com/)
http://pycon.blogspot.com/ (http://pycon.blogspot.com/)
https://wiki.python.org/moin/PythonEventsCalendar#Submitting_an_Event (http
s://wiki.python.org/moin/PythonEventsCalendar#Submitting_an_Event)
http://docs.python.org/3/tutorial/introduction.html#using-python-as-a-calcula
tor (http://docs.python.org/3/tutorial/introduction.html#using-python-as-a-ca
lculator)
https://www.jetbrains.com/lp/python-developers-survey-2020/ (https://www.jetb
rains.com/lp/python-developers-survey-2020/)
https://docs.python.org (https://docs.python.org)
https://blog.python.org (https://blog.python.org)
http://feedproxy.google.com/~r/PythonSoftwareFoundationNews/~3/v6DGTOxHRDQ/py
thon-developers-survey-2020-results.html (http://feedproxy.google.com/~r/Pyth
onSoftwareFoundationNews/~3/v6DGTOxHRDQ/python-developers-survey-2020-result
s.html)
http://feedproxy.google.com/~r/PythonInsider/~3/neAiV_Bq2Ck/python-392-and-38
8-are-now-available.html (http://feedproxy.google.com/~r/PythonInsider/~3/neA
iV_Bq2Ck/python-392-and-388-are-now-available.html)
http://feedproxy.google.com/~r/PythonInsider/~3/K1v58uqKOsI/python-392rc1-and
-388rc1-are-now.html (http://feedproxy.google.com/~r/PythonInsider/~3/K1v58uq
KOsI/python-392rc1-and-388rc1-are-now.html)
http://feedproxy.google.com/~r/PythonInsider/~3/NfEAmX9_8yk/python-3710-and-3
613-security-updates.html (http://feedproxy.google.com/~r/PythonInsider/~3/Nf
EAmX9_8yk/python-3710-and-3613-security-updates.html)
http://feedproxy.google.com/~r/PythonSoftwareFoundationNews/~3/HTf9a0pMcBI/we
```

```
lcoming-google-as-visionary-sponsor.html (http://feedproxy.google.com/~r/Pyth
onSoftwareFoundationNews/~3/HTf9a0pMcBI/welcoming-google-as-visionary-sponso
r.html)
http://www.djangoproject.com/ (http://www.djangoproject.com/)
http://www.pylonsproject.org/ (http://www.pylonsproject.org/)
http://bottlepy.org (http://bottlepy.org)
http://tornadoweb.org (http://tornadoweb.org)
http://flask.pocoo.org/ (http://flask.pocoo.org/)
http://www.web2py.com/ (http://www.web2py.com/)
http://wiki.python.org/moin/TkInter (http://wiki.python.org/moin/TkInter)
https://wiki.gnome.org/Projects/PyGObject (https://wiki.gnome.org/Projects/Py
GObject)
http://www.riverbankcomputing.co.uk/software/pyqt/intro (http://www.riverbank
computing.co.uk/software/pyqt/intro)
https://wiki.qt.io/PySide (https://wiki.qt.io/PySide)
https://kivy.org/ (https://kivy.org/)
http://www.wxpython.org/ (http://www.wxpython.org/)
http://www.scipy.org (http://www.scipy.org)
http://pandas.pydata.org/ (http://pandas.pydata.org/)
http://ipython.org (http://ipython.org)
http://buildbot.net/ (http://buildbot.net/)
http://trac.edgewall.org/ (http://trac.edgewall.org/)
http://roundup.sourceforge.net/ (http://roundup.sourceforge.net/)
http://www.ansible.com (http://www.ansible.com)
http://www.saltstack.com (http://www.saltstack.com)
https://www.openstack.org (https://www.openstack.org)
http://brochure.getpython.info/ (http://brochure.getpython.info/)
https://docs.python.org/3/license.html (https://docs.python.org/3/license.htm
l)
https://wiki.python.org/moin/BeginnersGuide (https://wiki.python.org/moin/Beg
innersGuide)
https://devguide.python.org/ (https://devguide.python.org/)
https://docs.python.org/faq/ (https://docs.python.org/faq/)
http://wiki.python.org/moin/Languages (http://wiki.python.org/moin/Languages)
http://python.org/dev/peps/ (http://python.org/dev/peps/)
https://wiki.python.org/moin/PythonBooks (https://wiki.python.org/moin/Python
Books)
https://wiki.python.org/moin/ (https://wiki.python.org/moin/)
http://planetpython.org/ (http://planetpython.org/)
http://pyfound.blogspot.com/ (http://pyfound.blogspot.com/)
http://pycon.blogspot.com/ (http://pycon.blogspot.com/)
https://wiki.python.org/moin/PythonEventsCalendar#Submitting_an_Event (http
s://wiki.python.org/moin/PythonEventsCalendar#Submitting_an_Event)
https://devguide.python.org/ (https://devguide.python.org/)
https://bugs.python.org/ (https://bugs.python.org/)
https://mail.python.org/mailman/listinfo/python-dev (https://mail.python.org/
mailman/listinfo/python-dev)
https://github.com/python/pythondotorg/issues (https://github.com/python/pyth
ondotorg/issues)
https://status.python.org/ (https://status.python.org/)
```

In [135]:
```
import re
# How to extract all the URLs from python.org portal?
r=requests.get("https://www.python.org")
page=r.text

obj=bs4.BeautifulSoup(page)
for var in obj.find_all("a"):
    if(re.search("(^https|^http).*(org$|com$)",var.get("href"))):
        print(var.get("href"))
```

```
https://docs.python.org (https://docs.python.org)
https://docs.python.org (https://docs.python.org)
https://blog.python.org (https://blog.python.org)
http://bottlepy.org (http://bottlepy.org)
http://tornadoweb.org (http://tornadoweb.org)
http://www.scipy.org (http://www.scipy.org)
http://ipython.org (http://ipython.org)
http://www.ansible.com (http://www.ansible.com)
http://www.saltstack.com (http://www.saltstack.com)
https://www.openstack.org (https://www.openstack.org)
```

In [137]:
```
type(requests.get("https://docs.python.org").text)
```

Out[137]:
```
str
```

In [138]:
```
# Extracting all the text from page:
print(obj.get_text())
```

In [ ]:

```
import bs4
soup=bs4.BeautifulSoup("webpage")
soup.find("<tagname>") ->''
soup.find_all("<tagname>")->[]

# nestedtag
# ---------
#  |_open a outertag ->att=value -->{att:value} //dict logic
#  |--------------------------------------------------------

# Regx(import re)
```

In [159]:

```
from bs4 import BeautifulSoup
obj=BeautifulSoup("<h1><b>Sample</b></h1>")
t1=obj.h1
t2=obj.b
print(t1)
print(t2)
print(t1.name)
t1.name="HEAD1"
print("-->",t1)
print(t1.name)
```

```
<h1><b>Sample</b></h1>
<b>Sample</b>
h1
--> <HEAD1><b>Sample</b></HEAD1>
HEAD1
```

In [163]:

```
obj=BeautifulSoup('<b class="bold">SAMPLE</b>')
t=obj.b
r=t.attrs
print(r)
d={'href':['url1']}
d['href'][0]
```

```
{'class': ['bold']}
```

Out[163]:

```
'url1'
```

In [165]:

```
obj=BeautifulSoup('<a href="https://www.python.org">')
obj.a.attrs['href']
```

Out[165]:

```
'https://www.python.org'
```

In [166]:

```
obj=BeautifulSoup(page)
obj.a.attrs
```

Out[166]:

```
{'href': '#content', 'title': 'Skip to content'}
```

In [170]:

```
# Navigating using tag names
# obj.head # <head>...</head>
# obj.body # <body>...</body>
obj.title
```

Out[170]:

```
<title>Welcome to Python.org</title>
```

In [171]:

```
obj.head.contents
```

Out[171]:

```
['\n',
 <meta charset="utf-8"/>,
 '\n',
 <meta content="IE=edge" http-equiv="X-UA-Compatible"/>,
 '\n',
 <link href="//ajax.googleapis.com/ajax/libs/jquery/1.8.2/jquery.min.js" rel
="prefetch"/>,
 '\n',
 <link href="//ajax.googleapis.com/ajax/libs/jqueryui/1.12.1/jquery-ui.min.j
s" rel="prefetch"/>,
 '\n',
 <meta content="Python.org" name="application-name"/>,
 '\n',
 <meta content="The official home of the Python Programming Language" name="m
sapplication-tooltip"/>,
 '\n',
 <meta content="Python.org" name="apple-mobile-web-app-title"/>,
 '\n',
 <meta content="yes" name="apple-mobile-web-app-capable"/>,
 '\n',
```

In [180]:
```
r=requests.get("http://yum.oracle.com/oracle-linux-7.html")
page=r.text
soup=BeautifulSoup(page)
soup.find_all(re.compile("img*"))
# (or)
for v in soup.find_all("a"):
    print(v.get('href'))
```

```
http://oracle.com (http://oracle.com)
index.html
http://www.oracle.com/us/technologies/linux/support/overview/ (http://www.oracle.com/us/technologies/linux/support/overview/)
http://www.oracle.com/technetwork/server-storage/linux/documentation/ (http://www.oracle.com/technetwork/server-storage/linux/documentation/)
http://www.oracle.com/technetwork/server-storage/linux/downloads/index.html (http://www.oracle.com/technetwork/server-storage/linux/downloads/index.html)
http://www.facebook.com/oraclelinux (http://www.facebook.com/oraclelinux)
http://www.twitter.com/OracleLinux (http://www.twitter.com/OracleLinux)
http://www.linkedin.com/groups?gid=120238 (http://www.linkedin.com/groups?gid=120238)
http://www.youtube.com/oraclelinuxchannel (http://www.youtube.com/oraclelinuxchannel)
http://blogs.oracle.com/linux (http://blogs.oracle.com/linux)
index.html
/repo/OracleLinux/OL7/latest/x86_64/index.html
/repo/OracleLinux/OL7/latest/x86_64/index_src.html
```

In [ ]:
```
Framework   - Django=project+Apps
-----------         -------
                       |
                   configurations
        <project>
            |__manage.py..
            |__myapp1/....
            |__myapp2/....

    root@host~]# django-admin startproject <projectNAME> # nonwinx

    C:\>python -m django startproject <projectNAME> # winxd

```