

In []: Recap

```
-----  
python native types ->Operators ->Conditional Statement ->Looping statements  
python containers ->list,tuple,dict  
-----  
user defined function ->def <functionName>():....  
function Call arguments ->Scope of the block ->global Vs return  
File Handling  
-----  
->OOPs style code => class object method inheritance  
| ----- =====  
| class className:  
| ...  
| obj = className() <== constructor  
| obj.methodCall() -> def methodName(self):
```


In []:

```

DBI
|-> DataBase Connection ->Query
    -----(1)----- ....

class DBI:
    def connect(self,....):
        '''db connection'''

    def method1(self):
        '''method1'''

    def method2(self):
        '''method2'''

obj = DBI()           obj = DBI()
obj.connect()  Vs      obj.method1() - valid methodCall ->Operation is failed
obj.method1()

In Python any variablename/function_name
starts with __<name>__ =>special attribute (or) python built in attributes

>>> class box:
...     def f1(self):
...         print("This is f1 block - nonconstructor")
...
>>> box()
<__main__.box object at 0x000002535AE206E0>
>>>
>>> obj = box()
>>> obj.f1()
This is f1 block - nonconstructor
>>>
>>> class box:
...     def __init__(self):
...         print("This is constructor block")
...
>>> box()
This is constructor block
<__main__.box object at 0x000002535AE20830>
>>> obj = box()
This is constructor block
>>>
>>> class box:
...     def __init__(self,a1,a2=0,*a3,**a4):
...         print(a1,a2,a3,a4)
...
>>> obj = box(10,20,30,40,db="mysql",user="root")
10 20 (30, 40) {'db': 'mysql', 'user': 'root'}
>>> objX = box()
Traceback (most recent call last):
  File "<python-input-13>", line 1, in <module>
    objX = box()
TypeError: box.__init__() missing 1 required positional argument: 'a1'
>>>
>>> objX
Traceback (most recent call last):
  File "<python-input-15>", line 1, in <module>

```

```
    objX  
NameError: name 'objX' is not defined. Did you mean: 'obj'?  
>>>
```

```
In [1]: class Enrollment:  
        '''This is Enrollment class'''  
        def __init__(self,ename,edob):  
            self.ename = ename  
            self.edob = edob  
            print('Enrollment is done')  
        def display(self):  
            print(f'About {self.ename} details:-')  
            print(f'Name:{self.ename} DOB:{self.edob}')
```

```
In [2]: eobj1 = Enrollment('Arun','1st Jan')  
eobj2 = Enrollment('Vijay','2nd Feb')
```

```
Enrollment is done  
Enrollment is done
```

```
In [3]: eobj1.display()
```

```
About Arun details:-  
Name:Arun DOB:1st Jan
```

```
In [4]: eobj2.display()
```

```
About Vijay details:-  
Name:Vijay DOB:2nd Feb
```

```
In [5]: import openpyxl
```

```
In [6]: wobj = openpyxl.Workbook()  
sh = wobj.active  
sh['A1']=100  
sh['A2']=200  
sh['B1']='Data1'  
wobj.save('E:\\test1.xlsx')
```

```
In [7]: wobj
```

```
Out[7]: <openpyxl.workbook.workbook.Workbook at 0x219db208690>
```

```
In [8]: import time
```

```
In [9]: from openpyxl import Workbook
wobj = Workbook()
sh = wobj.active
sh['F1']=time.ctime()
sh['D2']=10+34.32
wobj.save('E:\\test2.xlsx')
```

```
In [10]: class p1:
    pid = 100

class p2:
    pcost = 1000

obj = p2()
obj.pcost
```

Out[10]: 1000

```
In [11]: obj.pid
```

AttributeError Traceback (most recent call last)
Cell In[11], line 1
----> 1 obj.pid
AttributeError: 'p2' object has no attribute 'pid'

```
In [ ]: [parent]
      | <== ISA relationship
[child]

class parent:
    <attribute>

class child(parent_className): <= Inheritance
    <attributes>
```

```
In [12]: class p1:
    pid = 101
class p2(p1): ##### Vs def f2(args): <== function/method
    pcost = 1230

obj = p2()
obj.pid
```

Out[12]: 101

```
In [14]: class p1:  
    pid = 101  
    pname = 'matplotlib'  
class p2(p1):  
    pcost = 1235  
    pname = 'prometheus'  
  
obj = p2()  
print(obj.pid,obj.pcost)  
print(obj.pname)
```

```
101 1235  
prometheus
```

```
In [15]: class p1:  
    pid = 101  
    pname = 'matplotlib'  
class p2(p1):  
    pcost = 1235  
    pname = 'prometheus'  
    pvar = p1.pname # parent attributes to UDV  
  
obj = p2()  
print(obj.pid,obj.pcost)  
print(obj.pname)  
print(obj.pvar)
```

```
101 1235  
prometheus  
matplotlib
```

```
In [16]: class p1:  
    def f1(self):  
        print('parent class method')  
  
class p2(p1):  
    def f2(self):  
        print('f2 method')  
    def f1(self):  
        print('p2 - child class method')  
  
obj = p2()  
obj.f1()
```

```
p2 - child class method
```

```
In [17]: class p1:
    def f1(self):
        print('parent class method')

class p2(p1):
    def f2(self):
        print('f2 method')
    def f1(self):
        print('p2 - child class method')
        super().f1() # invoke parent class f1 method

obj = p2()
obj.f1()
```

p2 - child class method
parent class method

```
In [ ]: Python programming
|->direct access - procedure style code -> var=10 ; functionName()
|->object based access - oops style -> obj.var=10 ; obj.function()
|->functional style process - SingleLine Code -
      (this is not a block style code)
```

```
In [25]: L = []
def fx(a):
    return a+100

for var in [10,20,30,40,50]:
    r = fx(var)
    L.append(r)

print(L)
```

[110, 120, 130, 140, 150]

```
In [26]: list(map(lambda a:a+100,[10,20,30,40,50]))
```

Out[26]: [110, 120, 130, 140, 150]

```
In [ ]: 1. lambda
         2. list comprehension
         3. generator
         4. map,filter,enumerator <==
```

```
lambda
  - unnamed function
  - function call with arguments and return <someValue>
    |
    def fx(a):
      return <value>

  - lambda <list of args>:<basic Operation>
    | ->expression;functionCall;methodCall
```

```
In [27]: lambda a1,a2:a1+a2
```

```
Out[27]: <function __main__.lambda>(a1, a2)
```

```
In [28]: f1 = lambda a1,a2:a1+a2
         f1(10,20)
```

```
Out[28]: 30
```

```
In [29]: def f2(a1,a2):
          return a1+a2

          f2(10,20)
```

```
Out[29]: 30
```

```
In [30]: f3 = lambda a1: a1 >100
         f3(150)
```

```
Out[30]: True
```

```
In [31]: f3(50)
```

```
Out[31]: False
```

```
In [32]: f4 = lambda a1:'sales' in a1
         f4('raj,sales,pune')
```

```
Out[32]: True
```

```
In [33]: f5 = lambda a:a.upper()
         f5('abc')
```

```
Out[33]: 'ABC'
```

```
In [34]: def F5(a):
    return a.upper()

F5('abc')
```

Out[34]: 'ABC'

```
In [35]: def fx(a):
    if(a >50 and a<60):
        return a+100
    elif(a>100 and a<200):
        return a+200
    else:
        return a+300
```

```
In [36]: f6 = lambda a1:fx(a1)
f6(55)
```

Out[36]: 155

```
In [37]: f6(150)
```

Out[37]: 350

```
In [38]: f6(250)
```

Out[38]: 550

```
In [39]: # List comprehension
# -----//List Append Operation
#
# [finalValue for <iterable>]
#      -----(1)----
# -----(2)----

L = []
for var in range(1,7):
    r=var+100
    L.append(r)
L
```

Out[39]: [101, 102, 103, 104, 105, 106]

```
In [41]: [var+100 for var in range(1,7)]
```

Out[41]: [101, 102, 103, 104, 105, 106]

```
In [42]: L = []
for var in range(1,15):
    if(var >10):
        r=var+100
        L.append(r)
    else:
        r=var+500
        L.append(r)
L
```

```
Out[42]: [501, 502, 503, 504, 505, 506, 507, 508, 509, 510, 111, 112, 113, 114]
```

```
In [44]: [var+100 if var >10 else var+500 for var in range(1,15)]
```

```
Out[44]: [501, 502, 503, 504, 505, 506, 507, 508, 509, 510, 111, 112, 113, 114]
```

```
In [ ]: generator
-----
| ->Function - returns an iterator(address)
===== yield
| ->open this address - get value
    1) next(generator) ->Value...StopIteration
    2) use for loop
    3) typecast to list
```

```
In [47]: def f1():
    return 10

def f2():
    yield 10

print(type(f1),type(f2))

print(type(f1()),type(f2()))
```

```
<class 'function'> <class 'function'>
<class 'int'> <class 'generator'>
```

```
In [48]: def f1():
    return 10 # exit from f1 block
    print("OK") # python won't interprete this line

f1()
```

```
Out[48]: 10
```

```
In [49]: def f2():
    yield 10
    print("OK")
    yield 10+20
    yield "D1", "D2"
    yield "D1", ["F1", 'F2', 'F3'], ['F4', 'F5']

f2()
```

Out[49]: <generator object f2 at 0x00000219DC782CF0>

```
In [54]: gobj = f2()
```

```
In [55]: print(next(gobj))
print(next(gobj))
print(next(gobj))
print(next(gobj))
```

```
10
OK
30
('D1', 'D2')
('D1', ['F1', 'F2', 'F3'], ['F4', 'F5'])
```

```
In [56]: print(next(gobj))
```

```
StopIteration                                     Traceback (most recent call last)
Cell In[56], line 1
      1 print(next(gobj))
```

```
StopIteration:
```

```
In [57]: geobj = f2()
```

```
for var in geobj:
    print(var)
```

```
10
OK
30
('D1', 'D2')
('D1', ['F1', 'F2', 'F3'], ['F4', 'F5'])
```

```
In [58]: genobj = f2()
list(genobj) # we can typecast generator object to list
```

```
OK
```

Out[58]: [10, 30, ('D1', 'D2'), ('D1', ['F1', 'F2', 'F3'], ['F4', 'F5'])]

```
In [59]: class box:  
    def method1(self):  
        yield "Value1"  
        yield "D1", "D2"  
        yield "Dx", ["F1", "F2"]  
        yield "Dy", ["f1", "f2", "f3"], ['f4', 'f5', 'f6']  
  
obj = box()  
print(type(obj.method1()))  
  
<class 'generator'>
```

```
In [60]: list(obj.method1())
```

```
Out[60]: ['Value1',  
          ('D1', 'D2'),  
          ('Dx', ['F1', 'F2']),  
          ('Dy', ['f1', 'f2', 'f3'], ['f4', 'f5', 'f6'])]
```

```
In [62]: #help(enumerate)  
  
s='python'  
enumerate(s)
```

```
Out[62]: <enumerate at 0x219dc7cbb50>
```

```
In [63]: for var in enumerate(s):  
    print(var)  
  
(0, 'p')  
(1, 'y')  
(2, 't')  
(3, 'h')  
(4, 'o')  
(5, 'n')
```

```
In [64]: for var in enumerate(s,15):  
    print(var)  
  
(15, 'p')  
(16, 'y')  
(17, 't')  
(18, 'h')  
(19, 'o')  
(20, 'n')
```

```
In [65]: for var in enumerate(open('E:\\emp.csv')):
    print(var)
```

```
(0, 'eid,ename,edept,eplace,ecost\n')
(1, '101,raj,sales,pune,1000\n')
(2, '102,leo,prod,bglore,2000\n')
(3, '103,paul,HR,chennai,3000\n')
(4, '104,anu,hr,hyderabad,4000\n')
(5, '456,kumar,sales,bglore,3000\n')
(6, '105,zion,Hr,mumbai,5000\n')
(7, '106,bibu,sales,bglore,1450\n')
(8, '107,theeb,sales,noida,4590\n')
(9, '108,bibu,sales,bglore,5000\n')
(10, '113,kumar,prod,hyderabad,5423DATA')
```

```
In [66]: for var in enumerate(open('E:\\emp.csv'),1):
    print(var)
```

```
(1, 'eid,ename,edept,eplace,ecost\n')
(2, '101,raj,sales,pune,1000\n')
(3, '102,leo,prod,bglore,2000\n')
(4, '103,paul,HR,chennai,3000\n')
(5, '104,anu,hr,hyderabad,4000\n')
(6, '456,kumar,sales,bglore,3000\n')
(7, '105,zion,Hr,mumbai,5000\n')
(8, '106,bibu,sales,bglore,1450\n')
(9, '107,theeb,sales,noida,4590\n')
(10, '108,bibu,sales,bglore,5000\n')
(11, '113,kumar,prod,hyderabad,5423DATA')
```

```
In [67]: for line,data in enumerate(open('E:\\emp.csv'),1):
    print(f'{line}\t{data.strip()}')
```

```
1      eid,ename,edept,eplace,ecost
2      101,raj,sales,pune,1000
3      102,leo,prod,bglore,2000
4      103,paul,HR,chennai,3000
5      104,anu,hr,hyderabad,4000
6      456,kumar,sales,bglore,3000
7      105,zion,Hr,mumbai,5000
8      106,bibu,sales,bglore,1450
9      107,theeb,sales,noida,4590
10     108,bibu,sales,bglore,5000
11     113,kumar,prod,hyderabad,5423DATA
```

```
In [68]: for line,data in enumerate(open('E:\\emp.csv'),1):
    if(line >=5 and line <8):
        print(f'{line}\t{data.strip()}')
```

```
5      104,anu,hr,hyderabad,4000
6      456,kumar,sales,bglore,3000
7      105,zion,Hr,mumbai,5000
```

```
In [ ]: map(function,<collection>) -> generator
        | --<--<--<----|
        |
        lambda
```

```
In [69]: L = []

def f1(a):                      # Lambda a: a+100 -> fx=Lambda a:a+100 ->fx(\n    return a+100\n\n    for var in [10,20,30,40,50]:      # <collection>\n        r = f1(var)\n        L.append(r)\n\nL
```

Out[69]: [110, 120, 130, 140, 150]

```
In [71]: list(map(lambda a:a+100,[10,20,30,40,50]))
```

Out[71]: [110, 120, 130, 140, 150]

```
In [73]: d={}
d['K1']=list(map(lambda a:a+100,[10,20,30,40,50]))
L=[]
L.append(list(map(lambda a:a+100,[10,20,30,40,50])))
print(d)
L
```

{'K1': [110, 120, 130, 140, 150]}

Out[73]: [[110, 120, 130, 140, 150]]

```
In [ ]: map(function,<Collection>) -> generator
        | - arithmetic, relational, membership etc.,\n        |
        filter(function,<Collection>) -> generator
        |->filter True value - not supports arithmetic operation
```

```
In [74]: list(map(lambda a:a >100,[120,110,50,60,130]))
```

Out[74]: [True, True, False, False, True]

```
In [75]: list(map(lambda a:a +100,[120,110,50,60,130]))
```

Out[75]: [220, 210, 150, 160, 230]

```
In [76]: list(filter(lambda a:a >100,[120,110,50,60,130]))
```

Out[76]: [120, 110, 130]

```
In [77]: list(filter(lambda a:a +100,[120,110,50,60,130]))
```

```
Out[77]: [120, 110, 50, 60, 130]
```

```
In [78]: print("One")
print("Two")
print("Three")
print("FOur")
```

```
Cell In[78], line 3
    print("Three")
    ^

```

```
SyntaxError: unterminated string literal (detected at line 3)
```

```
In [79]: print("One")
print("Two")
print(Three)
print("FOur")
print("Five")
```

```
One
Two
```

```
NameError
```

```
Traceback (most recent call last)
```

```
Cell In[79], line 3
    1 print("One")
    2 print("Two")
----> 3 print(Three)
    4 print("FOur")
    5 print("Five")
```

```
NameError: name 'Three' is not defined
```

```
In [ ]: try:
    <code>
except <classname>:
    <Handle this error>
else:
    <There is no Error>
finally:
    <Always running block>
```

```
In [80]: try:  
    print(va)  
except Exception:  
    print("Exception is occurred")  
else:  
    print("There is no error")  
finally:  
    print("Thank you")
```

Exception is occurred
Thank you

```
In [81]: try:  
    va=10  
except Exception:  
    print("Exception is occurred")  
else:  
    print("There is no error")  
finally:  
    print("Thank you")
```

There is no error
Thank you

```
In [83]: try:  
    print(vA)  
except Exception as eobj:  
    print("Exception is occurred")  
    print(eobj)  
else:  
    print("There is no error")  
finally:  
    print("Thank you")
```

Exception is occurred
name 'vA' is not defined
Thank you

```
In [84]: import sys  
try:  
    print(vA)  
except Exception:  
    print("Exception is occurred")  
    print(sys.exc_info())  
else:  
    print("There is no error")  
finally:  
    print("Thank you")
```

Exception is occurred
(<class 'NameError'>, NameError("name 'vA' is not defined"), <traceback object at 0x00000219DC541C40>)
Thank you

```
In [85]: import sys
print("one")
print("two")
print("three")
try:
    print(four)
except Exception:
    print(sys.exc_info())

for var in range(5):
    print(var)

print("End of the line")
```

```
one
two
three
(<class 'NameError'>, NameError("name 'four' is not defined"), <traceback object at 0x00000219DA430F80>)
0
1
2
3
4
End of the line
```

```
In [88]: fobj = open('E:\\emp.csv')
L = fobj.readlines()
fobj.close()

for var in L:
    var=var.strip()
    if('sales' in var):
        L1=var.split(",")
        print(f'{L1[1].title()}\\t{L1[-1]}')
```

```
Raj      1000
Kumar   3000
Bibu    1450
Theeb   4590
Bibu    5000
```

```
In [89]: list(filter(lambda a:'sales' in a,open('E:\\emp.csv')))
```

```
Out[89]: ['101,raj,sales,pune,1000\\n',
          '456,kumar,sales,bglore,3000\\n',
          '106,bibu,sales,bglore,1450\\n',
          '107,theeb,sales,noida,4590\\n',
          '108,bibu,sales,bglore,5000\\n']
```

```
In [90]: list(map(lambda a:a.split(","),filter(lambda a:'sales' in a,open('E:\\emp.csv'))))
```

```
Out[90]: [['101', 'raj', 'sales', 'pune', '1000\n'],
           ['456', 'kumar', 'sales', 'bglore', '3000\n'],
           ['106', 'bibu', 'sales', 'bglore', '1450\n'],
           ['107', 'theeb', 'sales', 'noida', '4590\n'],
           ['108', 'bibu', 'sales', 'bglore', '5000\n']]
```

```
In [92]: list(map(lambda a:a.split(",")[1].title(),filter(lambda a:'sales' in a,
                                                       open('E:\\emp.csv'))))
```

```
Out[92]: ['Raj', 'Kumar', 'Bibu', 'Theeb', 'Bibu']
```

```
In [93]: list(map(lambda a:a.split(",")[-1],filter(lambda a:'sales' in a,
                                                       open('E:\\emp.csv'))))
```

```
Out[93]: ['1000\n', '3000\n', '1450\n', '4590\n', '5000\n']
```

```
In [94]: # python 2.x
          # reduce(Lambda a,b:a+b,<inputCollection>) ->SingleValue

L=[10,20,30,40,50]
t=0
for var in L:
    t=t+var
print(t)
```

```
150
```

```
In [95]: # python 3.x
          # functools package

from functools import reduce
reduce(lambda a,b:a+b,L)
```

```
Out[95]: 150
```

```
In [97]: reduce(lambda a,b:int(a)+int(b),map(lambda a:a.split(",")[-1],
                                             filter(lambda a:'sales' in a,
                                                   open('E:\\emp.csv'))))
```

```
Out[97]: 15040
```

In []: Regular Expression

|->Search |->Substitute |->Split

```
import re <== module
```

```
search
```

|-> re.search() => re.search('input_pattern','input_String',re.I) -><ack>/Nor
|-> re.findall() => re.findall('input_pattern','input_string',re.I)->[result],

In [99]:

```
import re
s='101,raj,sales,pune,1000'
re.search('sales',s)
```

Out[99]: <re.Match object; span=(8, 13), match='sales'>

In [100]:

```
re.search('prod',s)
```

In [101]:

```
bool(re.search('prod',s))
```

Out[101]: False

In [102]:

```
re.findall('sales',s)
```

Out[102]: ['sales']

In [103]:

```
re.findall('prod',s)
```

Out[103]: []

In [104]:

```
bool(re.findall('prod',s))
```

Out[104]: False

In [106]:

```
if(re.search('sales',s,re.I)):
    print(s)
else:
    print('Sorry Given Pattern is not matched')
```

101,raj,sales,pune,1000

In []: Reg Chars

```
-----
^ ==> ^pattern - line starts with pattern
$ ==> pattern$ - line ends with pattern
^pattern$ => Pattern only
. -----> It match any single chars
.* -----> List of all
[] -----> Character class_ - character based search
[a-zA-Z]
[A-Z]
[a-z]
[0-9] <or> \d
[a-zA-Z0-9] <or> \w
\s <== space
^[]
[]$
[^] <== Not matching chars
^$ <== Empty line

| -> pattern1|pattern2 <== any one pattern is matched ->OK
() -> (pattern1)(pattern2) <== both pattern should match same order
+
{}
```

In [107]: re.findall('sales','101,raj,sales,pune')

Out[107]: ['sales']

In [108]: re.findall('^sales','101,raj,sales,pune')

Out[108]: []

In [110]: re.findall('sales\$','asfasfsd sales details')

Out[110]: []

In [111]: re.findall('^sales\$','sales')

Out[111]: ['sales']

In [112]: re.findall('^sales\$','sales,')

Out[112]: []

In [113]: re.findall('^sales:\$','sales')

Out[113]: []

```
In [115]: re.findall('^sales...','sales,101,pune')
#           -----<CHAR><CHAR><CHAR>
```

```
Out[115]: ['sales,10']
```

```
In [118]: re.findall('.....$','python programming language3.13')
```

```
Out[118]: ['e3.13']
```

```
In [119]: re.findall('sales.*','list of emp sales dept details code number')
```

```
Out[119]: ['sales dept details code number']
```

```
In [120]: re.findall('sales.*code','list of emp sales dept details code number')
```

```
Out[120]: ['sales dept details code']
```

```
In [121]: re.findall('sales.*code$', 'list of emp sales dept details code number')
```

```
Out[121]: []
```

```
In [ ]: # [] <== Character class_ [] <= 1 char [][] <== 2 chars
```

```
[Aa]run -> Arun arun //OK
[aAk][rR]un
-----
arun
aRun
Arun
ARun
krun
kRun
-----//OK
```

```
In [122]: msg='''Click the URL provided by the organizer. If a URL was not included
in the meeting invite, click Join a Zoom Meeting and enter the meeting details
when prompted.Meeting Code:JK-1314 dated on 26th Nov 2024 Sno 5 details'''
```

```
In [123]: re.findall('[0-9]',msg)
```

```
Out[123]: ['1', '3', '1', '4', '2', '6', '2', '0', '2', '4', '5']
```

```
In [124]: re.findall('\d',msg)
```

```
Out[124]: ['1', '3', '1', '4', '2', '6', '2', '0', '2', '4', '5']
```

```
In [126]: re.findall('[2-4]',msg)
```

```
Out[126]: ['3', '4', '2', '2', '2', '4']
```

```
In [127]: s='531,LEO,pune'
re.findall('[0-9]',s) # match any digits
```

```
Out[127]: ['5', '3', '1']
```

```
In [128]: re.findall('^[0-9]',s) # starts with any single digits
```

```
Out[128]: ['5']
```

```
In [129]: re.findall('^[^0-9]',s) # NOT Matching any digits
```

```
Out[129]: [',', 'L', 'E', '0', ',', 'p', 'u', 'n', 'e']
```

```
In [130]: s="userA@host.com,12%DATA(repo )"
```

```
re.findall('^[^0-9A-Za-z]',s) # match - specialchars only
```

```
Out[130]: ['@', '.', ',', '%', '(', ' ', ')']
```

```
In [131]: re.findall('^[^0-9A-Za-z\s]',s) # match - specialchars only
```

```
Out[131]: ['@', '.', ',', '%', '(', ')']
```

```
In [132]: re.findall('^\w\s',s)
```

```
Out[132]: ['@', '.', ',', '%', '(', ')']
```

```
In [133]: msg='python supports flask and fastAPI for fullstack development'
```

```
re.findall('django|flask|rails',msg)
```

```
Out[133]: ['flask']
```

```
In [134]: re.findall('^[A-Za-z].*[0-9]$|[a-z].[A-Z]$|[a-z].*API.*[a-z]$',msg)
```

```
Out[134]: ['python supports flask and fastAPI for fullstack development']
```

```
In [135]: re.findall('(flask)(fastAPI)',msg)
```

```
Out[135]: []
```

```
In [136]: re.findall('(flask).*(fastAPI)',msg)
```

```
Out[136]: [('flask', 'fastAPI')]
```

```
In [137]: re.findall('(fastAPI).*(flask)',msg)
```

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

```
In [ ]: # <pattern>+  => 1 or more times
ab+c
|
abc
abbc
aaaaaaaaaaaaaaaaaaaa //OK
```

```
In [ ]: re.findall('\s+',inputString)
|
1 or more space is matched ->OK

<Pattern>{n} -> n times
ab{3}c => abbc //OK

ab{3,}c ==> abbc aaaaaaaaaaaaaaaaaaaaa //OK

ab{3,5}c ==> abbc abbbbc abbbbc //OK

re.sub('oldPattern', 'replacement_str', 'input_string', count, re.I) -> str ->(A)
(A) -> replaced string if old pattern is matched
      -> input_string if old pattern is not matched
```

```
In [139]: for var in open('e:\\emp.csv'):
    var=var.strip()
    print(re.sub('sales.', '', var))
```

```
eid,ename,edept,eplace,ecost
101,raj,pune,1000
102,leo,prod,bglore,2000
103,paul,HR,chennai,3000
104,anu,hr,hyderabad,4000
456,kumar,bglore,3000
105,zion,Hr,mumbai,5000
106,bibu,bglore,1450
107,theeb,noida,4590
108,bibu,bglore,5000
113,kumar,prod,hyderabad,5423DATA
```

```
In [141]: for var in open('e:\\emp.csv'):
    var=var.strip()
    if(re.search('sales',var)):
        print(re.sub('sales.', '', var))
```

```
101,raj,pune,1000
456,kumar,bglore,3000
106,bibu,bglore,1450
107,theeb,noida,4590
108,bibu,bglore,5000
```

```
In [144]: import os
for var in os.listdir("E:\\"):
    if(re.search("log$|pdf$",var)):
        print(var)
```

```
ansible_playbook_essentials.pdf
Billing_info.log
DAY-3.pdf
ICR.log
IP.log
IP1.log
pin_history.log
products.log
r1.log
```

```
In [145]: s='root:x:bin:bash'
s.split(':')
```

```
Out[145]: ['root', 'x', 'bin', 'bash']
```

```
In [147]: s='root:x,bin~bash'
print(s.split(':'),s.split("~"))
```

```
['root', 'x,bin~bash'] ['root:x,bin', 'bash']
```

```
In [148]: # re.split('RegXPattern','inputString') ->List
re.split('[^\w\s]',s)
```

```
Out[148]: ['root', 'x', 'bin', 'bash']
```

```
In [ ]: prodID,prodName, vendor, cost, salescount
101,prodA,V1,1000,45
102:prodB|V2:2000-55
103:prodC~V3:3000:32
104,prodD|V4:,10
105:prodX|V5:1000-55
106:prodY~V6:,20
```

```
In [149]: for var in open('unformatted_IP.txt'):
    var=var.strip()
    L=re.split('[^\w\s]',var)
    print(L)
```

```
['prodID', 'prodName', 'vendor', 'cost', 'salescount']
['101', 'prodA', 'V1', '1000', '45']
['102', 'prodB', 'V2', '2000', '55']
['103', 'prodC', 'V3', '3000', '32']
['104', 'prodD', 'V4', '', '10']
['105', 'prodX', 'V5', '1000', '55']
['106', 'prodY', 'V6', '', '20']
```

```
In [150]: for var in open('unformatted_IP.txt'):
    var=var.strip()
    L=re.split('[^\w\s]',var)
    print(f'{L[0]}\t{L[1]}')
```

```
prodID prodName
101 prodA
102 prodB
103 prodC
104 prodD
105 prodX
106 prodY
```

```
In [ ]: read emp.csv file
|
filter/search -> list sales emp's - search
|           - substitute
current working city in pune or noida -> HYDERABAD
|           | split
                           EmpName,Dept,City
```

```
In [151]: for var in open('emp.csv'):
    var=var.strip()
    if(re.search('sales',var)):
        print(var)
```

```
101,raj,sales,pune,1000
106,bibu,sales,bglore,1450
107,theeb,sales,noida,4590
```

```
In [152]: for var in open('emp.csv'):
    var=var.strip()
    if(re.search('sales',var)):
        s=re.sub('pune|noida','HYDERABAD',var)
        print(s)
```

```
101,raj,sales,HYDERABAD,1000
106,bibu,sales,bglore,1450
107,theeb,sales,HYDERABAD,4590
```

```
In [154]: for var in open('emp.csv'):
    var=var.strip()
    if(re.search('sales',var)):
        s=re.sub('pune|noida','HYDERABAD',var)
        print(s)
```

```
101,raj,sales,HYDERABAD,1000
106,bibu,sales,bglore,1450
107,theeb,sales,HYDERABAD,4590
```

```
In [157]: for var in open('emp.csv'):
    var=var.strip() # remove \n chars
    if(re.search('sales',var)): # filter sales dept
        s=re.sub('pune|noida','HYDERABAD',var) # substitute
        if(re.search('hyderabad',s,re.I)): # search a pattern
            L=re.split('[^\w\s]',s) # split single string into multiple values
            print(f'{L[1]}\t{L[2]}\t{L[3]}')
```

```
raj      sales   HYDERABAD
theeb    sales   HYDERABAD
```

```
In [158]: s='120GB'
# How to delete/remote GB -> '120'
re.findall('[A-Za-z]',s)
```

```
Out[158]: ['G', 'B']
```

```
In [159]: re.findall('[A-Za-z]+',s)
```

```
Out[159]: ['GB']
```

```
In [160]: re.sub('[A-Za-z]+','',s)
```

```
Out[160]: '120'
```

```
In [161]: L=['120GB','150GB','135Gb','GB100','200',130]
#
# ->Calculate sum of used size

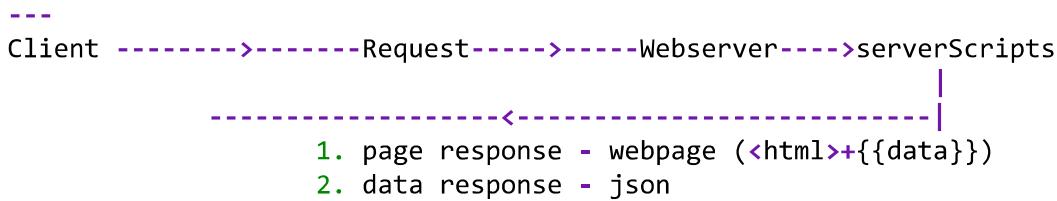
for var in L:
    print(re.sub('[a-zA-Z]+','',str(var)))
```

```
120
150
135
100
200
130
```

```
In [162]: L=['120GB','150GB','135Gb','GB100','200',130]
#
# ->Calculate sum of used size
t=0
for var in L:
    t=t+int(re.sub('[a-zA-Z]+','',str(var)))
else:
    print(f'Sum of Used size:{L} is: {t} GB')
```

```
Sum of Used size:['120GB', '150GB', '135Gb', 'GB100', '200', 130] is: 835 GB
```

In []: Web



In [163]: `import requests
requests.get('https://www.google.com')`

Out[163]: <Response [200]>

In [164]: `r = requests.get('https://www.google.com')
r.status_code`

Out[164]: 200

In [165]: `r.headers`

Out[165]: `{'Date': 'Tue, 26 Nov 2024 10:55:30 GMT', 'Expires': '-1', 'Cache-Control': 'private, max-age=0', 'Content-Type': 'text/html; charset=ISO-8859-1', 'Content-Security-Policy-Report-Only': "object-src 'none';base-uri 'self';script-src 'nonce-i785rquoP4WUxy28HI9i02A' 'strict-dynamic' 'report-sample' 'unsafe-eval' 'unsafe-inline' https: http://report-uri (http://report-uri) https://csp.withgoogle.com/csp/gws/other-hp", ('https://csp.withgoogle.com/csp/gws/other-hp',) 'Accept-CH': 'Sec-CH-Prefers-Color-Scheme', 'P3P': 'CP="This is not a P3P policy! See g.co/p3phelp for more info."', 'Content-Encoding': 'gzip', 'Server': 'gws', 'X-XSS-Protection': '0', 'X-Frame-Options': 'SAMEORIGIN', 'Set-Cookie': 'AEC=AZ6Zc-U5kNqRnnW0UM6t6AufvSpc06HJ0sa-LDGaVsb95c4CwXinzly2w; expires=Sun, 25-May-2025 10:55:30 GMT; path=/; domain=.google.com; Secure; HttpOnly; SameSite=lax, NID=519=3gaYXkPXM4fH_vv2mIZwUPdDUp9RhtaowR7mYq5ATWCg04FITHXxdSRAZFGH2CP3YfeZ8mkmyvZPTioY9c8YmZtH9uAODbdbdM4VQmF_6kqsw7wCd1OPtJzyrZq_d_9priUKBayGQ8wyHeL7_l-mHZA29-t32ySwqPFz5CjsVeodZ02mHOQkFH5Ms7LBnf1gpuw; expires=Wed, 28-May-2025 10:55:30 GMT; path=/; domain=.google.com; HttpOnly', 'Alt-Svc': 'h3=":443"; ma=2592000,h3-29=":443"; ma=2592000', 'Transfer-Encoding': 'chunked'}`

In [166]: `r.headers['Content-Type']`

Out[166]: `'text/html; charset=ISO-8859-1'`

In [167]: `s=r.text
print(type(s),len(s))`

`<class 'str'> 21153`

In [168]: `v = r.content
type(v),len(v)`

Out[168]: `(bytes, 21153)`

```
In [169]: wobj =open('E:\\test.html','w')
wobj.write(s)
wobj.close()
```

```
In [171]: #print(s)
```

```
In [172]: import requests
url = 'https://www.python.org'
r = requests.get(url)
if r.status_code != 200:
    print(f'requests from {url} is failed')
    exit() # exit from execution

if 'text/html' in r.headers['Content-Type']:
    web_page = r.text
```

```
In [173]: import bs4
```

```
In [174]: bs4.BeautifulSoup('<h1>Test message</h1>')
```

```
Out[174]: <html><body><h1>Test message</h1></body></html>
```

```
In [ ]: bs4_obj.<htmlTag> ->data ==>(1)
----- =====

bs4_obj.find('<htmlTag>') -> str_data =>(2)
Vs
bs4_obj.find_all('<htmlTag>') ->list_data =>(3)
```

```
In [175]: bs4_obj = bs4.BeautifulSoup('<h1>Test message</h1>')

bs4_obj.h1
```

```
Out[175]: <h1>Test message</h1>
```

```
In [176]: bs4_obj.find('h1')
```

```
Out[176]: <h1>Test message</h1>
```

```
In [177]: bs4_obj.find_all('h1')
```

```
Out[177]: [<h1>Test message</h1>]
```

```
In [178]: import requests
url = 'https://www.python.org'
r = requests.get(url)
if r.status_code != 200:
    print(f'requests from {url} is failed')
    exit() # exit from execution

if 'text/html' in r.headers['Content-Type']:
    web_page = r.text
```

```
In [179]: sobj = bs4.BeautifulSoup(web_page)
sobj.title
```

```
Out[179]: <title>Welcome to Python.org</title>
```

```
In [180]: sobj.title.name
```

```
Out[180]: 'title'
```

```
In [181]: sobj.title.contents
```

```
Out[181]: ['Welcome to Python.org']
```

```
In [182]: sobj.title.string
```

```
Out[182]: 'Welcome to Python.org'
```

```
In [183]: sobj.p
```

```
Out[183]: <p><strong>Notice:</strong> While JavaScript is not essential for this website, your interaction with the content will be limited. Please turn JavaScript on for the full experience. </p>
```

```
In [184]: sobj.h1
```

```
Out[184]: <h1 class="site-headline">
<a href="/"></a>
</h1>
```

```
In [185]: ## sobj.find(<htmlTag>)
```

```
sobj.find('h1')
```

```
Out[185]: <h1 class="site-headline">
<a href="/"></a>
</h1>
```

In [186]: `## sobj.find_all(<htmlTag>) -> list_output`

```
sobj.find_all('h1')
```

Out[186]: [`<h1 class="site-headline">`
``
`</h1>,`
`<h1>Functions Defined</h1>,`
`<h1>Compound Data Types</h1>,`
`<h1>Intuitive Interpretation</h1>,`
`<h1>All the Flow You'd Expect</h1>,`
`<h1>Quick & Easy to Learn</h1>]`

In [188]: `for var in sobj.find_all('h1'):`
 `print(var.string) # extract data/string`

```
None
Functions Defined
Compound Data Types
Intuitive Interpretation
All the Flow You'd Expect
Quick & Easy to Learn
```

In [191]: `sobj.find('h1')`

Out[191]: `<h1 class="site-headline">`
``
`</h1>`

In [192]: `sobj.find('h1')['class']`

Out[192]: `['site-headline']`

In [193]: `sobj.find('h1')['href']`

```
-----
KeyError
Cell In[193], line 1
----> 1 sobj.find('h1')['href']
```

Traceback (most recent call last)

```
File C:\ProgramData\anaconda3\Lib\site-packages\bs4\element.py:1573, in Tag.__getitem__(self, key)
1570     def __getitem__(self, key):
1571         """tag[key] returns the value of the 'key' attribute for the Tag,
1572         and throws an exception if it's not there."""
-> 1573     return self.attrs[key]
```

KeyError: 'href'

```
In [195]: d={'K1':'V1'}  
d['K1']  
d['Kx']
```

KeyError

Traceback (most recent call last)

```
Cell In[195], line 3  
  1 d={'K1':'V1'}  
  2 d['K1']  
----> 3 d['Kx']
```

KeyError: 'Kx'

```
In [196]: print(d.get('K1'),d.get('Kx'))
```

V1 None

```
In [197]: sobj.find('h1').get('href')
```

```
In [198]: sobj.find('a')
```

```
Out[198]: <a href="#content" title="Skip to content">Skip to content</a>
```

```
In [200]: sobj.find('a').get('href')
```

```
Out[200]: '#content'
```

```
In [201]: sobj.p.parent
```

```
Out[201]: <div class="do-not-print" id="nojs">  
<p><strong>Notice:</strong> While JavaScript is not essential for this website, your interaction with the content will be limited. Please turn JavaScript on for the full experience. </p>  
</div>
```

In [202]: `sobj.title.parent`

Out[202]:

```
<head>
<!-- Google tag (gtag.js) -->
<script async="" src="https://www.googletagmanager.com/gtag/js?id=G-TF35YF9CVH"></script>
<script>
    window.dataLayer = window.dataLayer || [];
    function gtag(){dataLayer.push(arguments);}
    gtag('js', new Date());
    gtag('config', 'G-TF35YF9CVH');
</script>
<!-- Plausible.io analytics -->
<script data-domain="python.org" defer="" src="https://plausible.io/js/script.js"></script>
<meta charset="utf-8"/>
<meta content="IE=edge" http-equiv="X-UA-Compatible"/>
<link href="//ajax.googleapis.com/ajax/libs/jquery/1.8.2/jquery.min.js" rel="prefetch"/>
<link href="//ajax.googleapis.com/ajax/libs/jqueryui/1.12.1/jquery-ui.min.js" rel="prefetch"/>
<link href="//ajax.googleapis.com/ajax/libs/jqueryui/1.12.1/jquery-ui.min.js" rel="prefetch"/>
```

In [203]:

```
# Tasks
# -----
# Get list of all href links from google.com

r = requests.get('https://www.google.com')
if r.status_code != 200:
    exit()

google_webpage = r.text
```

In [204]:

```
gpage = bs4.BeautifulSoup(google_webpage)
gpage.find('a')
```

Out[204]:

```
<a class="gb1" href="https://www.google.com/imghp?hl=en&tab=wi">Images</a>
```

In [205]: `gpage.find_all('a')`

Out[205]:

```
<a class="gb1" href="https://www.google.com/imghp?hl=en&tab=wi">Images</a>,
<a class="gb1" href="https://maps.google.co.in/maps?hl=en&tab=w1">Maps</a>,
<a class="gb1" href="https://play.google.com/?hl=en&tab=w8">Play</a>,
<a class="gb1" href="https://www.youtube.com/?tab=w1">YouTube</a>,
<a class="gb1" href="https://news.google.com/?tab=wn">News</a>,
<a class="gb1" href="https://mail.google.com/mail/?tab=wm">Gmail</a>,
<a class="gb1" href="https://drive.google.com/?tab=wo">Drive</a>,
<a class="gb1" href="https://www.google.co.in/intl/en/about/products?tab=wh" style="text-decoration:none"><u>More</u> ></a>,
<a class="gb4" href="http://www.google.co.in/history/optout?hl=en">Web History</a>,
<a class="gb4" href="/preferences?hl=en">Settings</a>,
<a class="gb4" href="https://accounts.google.com/ServiceLogin?hl=en&passive=true&continue=https://www.google.com/&ec=GAZAAQ" id="gb_70" target="_top">Sign in</a>,
<a href="/advanced_search?hl=en-IN&authuser=0">Advanced search</a>,
<a href="https://www.google.com/setprefs?sig=0_XFMPqIqaBMTsgDIX-F0n29bui4I%3D&hl=hi&source=homepage&sa=X&ved=0ahUKEwjMgeCD8_mJAxXgSwwGHX28Eo0Q2ZgBCAY">ହିନ୍ଦୀ</a>,
<a href="https://www.google.com/setprefs?sig=0_XFMPqIqaBMTsgDIX-F0n29bui4I%3D&hl=bn&source=homepage&sa=X&ved=0ahUKEwjMgeCD8_mJAxXgSwwGHX28Eo0Q2ZgBCAc">ବାଂଲା</a>,
<a href="https://www.google.com/setprefs?sig=0_XFMPqIqaBMTsgDIX-F0n29bui4I%3D&hl=te&source=homepage&sa=X&ved=0ahUKEwjMgeCD8_mJAxXgSwwGHX28Eo0Q2ZgBCAg">ଓଡ଼ିଆ</a>,
<a href="https://www.google.com/setprefs?sig=0_XFMPqIqaBMTsgDIX-F0n29bui4I%3D&hl=mr&source=homepage&sa=X&ved=0ahUKEwjMgeCD8_mJAxXgSwwGHX28Eo0Q2ZgBCAk">ମରାଠୀ</a>,
<a href="https://www.google.com/setprefs?sig=0_XFMPqIqaBMTsgDIX-F0n29bui4I%3D&hl=ta&source=homepage&sa=X&ved=0ahUKEwjMgeCD8_mJAxXgSwwGHX28Eo0Q2ZgBCAo">ତମିଳୁ</a>,
<a href="https://www.google.com/setprefs?sig=0_XFMPqIqaBMTsgDIX-F0n29bui4I%3D&hl=gu&source=homepage&sa=X&ved=0ahUKEwjMgeCD8_mJAxXgSwwGHX28Eo0Q2ZgBCAs">ગુજરાતી</a>,
<a href="https://www.google.com/setprefs?sig=0_XFMPqIqaBMTsgDIX-F0n29bui4I%3D&hl=kn&source=homepage&sa=X&ved=0ahUKEwjMgeCD8_mJAxXgSwwGHX28Eo0Q2ZgBCAw">କେନ୍ଦ୍ରୀଆ</a>,
<a href="https://www.google.com/setprefs?sig=0_XFMPqIqaBMTsgDIX-F0n29bui4I%3D&hl=ml&source=homepage&sa=X&ved=0ahUKEwjMgeCD8_mJAxXgSwwGHX28Eo0Q2ZgBCA0">ଓପାଷଣା</a>,
<a href="https://www.google.com/setprefs?sig=0_XFMPqIqaBMTsgDIX-F0n29bui4I%3D&hl=pa&source=homepage&sa=X&ved=0ahUKEwjMgeCD8_mJAxXgSwwGHX28Eo0Q2ZgBCA4">ପେନ୍ଜାବୀ</a>,
<a href="/intl/en/ads/">Advertising</a>,
<a href="http://www.google.co.in/services/">Business Solutions</a>,
<a href="/intl/en/about.html">About Google</a>,
<a href="https://www.google.com/setprefdomain?prefdom=IN&prev=https://www.google.co.in/&sig=K_xCCckhVoLHRu9k1Vzxd_yhHPuts%3D">Google.co.in</a>,
<a href="/intl/en/policies/privacy/">Privacy</a>,
<a href="/intl/en/policies/terms/">Terms</a>]
```

```
In [206]: for var in gpage.find_all('a'):
    print(var.get('href'))
```

<https://www.google.com/imghp?hl=en&tab=wi> (<https://www.google.com/imghp?hl=en&tab=wi>)
<https://maps.google.co.in/maps?hl=en&tab=w1> (<https://maps.google.co.in/maps?hl=en&tab=w1>)
<https://play.google.com/?hl=en&tab=w8> (<https://play.google.com/?hl=en&tab=w8>)
<https://www.youtube.com/?tab=w1> (<https://www.youtube.com/?tab=w1>)
<https://news.google.com/?tab=wn> (<https://news.google.com/?tab=wn>)
<https://mail.google.com/mail/?tab=wm> (<https://mail.google.com/mail/?tab=wm>)
<https://drive.google.com/?tab=wo> (<https://drive.google.com/?tab=wo>)
<https://www.google.co.in/intl/en/about/products?tab=wh> (<https://www.google.co.in/intl/en/about/products?tab=wh>)
<http://www.google.co.in/history/optout?hl=en> (<http://www.google.co.in/history/optout?hl=en>)
[/preferences?hl=en](#)
<https://accounts.google.com/ServiceLogin?hl=en&passive=true&continue=https://www.google.com/&ec=GAZAAQ> (<https://accounts.google.com/ServiceLogin?hl=en&passive=true&continue=https://www.google.com/&ec=GAZAAQ>)
[/advanced_search?hl=en-IN&authuser=0](#)
https://www.google.com/setprefs?sig=0_XFMPqIqaBMTsgDIx-F0n29bui4I%3D&hl=hi&source=homepage&sa=X&ved=0ahUKEwjMgeCD8_mJAxXgSwwGHX28Eo0Q2ZgBCAY (https://www.google.com/setprefs?sig=0_XFMPqIqaBMTsgDIx-F0n29bui4I%3D&hl=hi&source=homepage&sa=X&ved=0ahUKEwjMgeCD8_mJAxXgSwwGHX28Eo0Q2ZgBCAY)
https://www.google.com/setprefs?sig=0_XFMPqIqaBMTsgDIx-F0n29bui4I%3D&hl=bn&source=homepage&sa=X&ved=0ahUKEwjMgeCD8_mJAxXgSwwGHX28Eo0Q2ZgBCAc (https://www.google.com/setprefs?sig=0_XFMPqIqaBMTsgDIx-F0n29bui4I%3D&hl=bn&source=homepage&sa=X&ved=0ahUKEwjMgeCD8_mJAxXgSwwGHX28Eo0Q2ZgBCAc)
https://www.google.com/setprefs?sig=0_XFMPqIqaBMTsgDIx-F0n29bui4I%3D&hl=te&source=homepage&sa=X&ved=0ahUKEwjMgeCD8_mJAxXgSwwGHX28Eo0Q2ZgBCAg (https://www.google.com/setprefs?sig=0_XFMPqIqaBMTsgDIx-F0n29bui4I%3D&hl=te&source=homepage&sa=X&ved=0ahUKEwjMgeCD8_mJAxXgSwwGHX28Eo0Q2ZgBCAg)
https://www.google.com/setprefs?sig=0_XFMPqIqaBMTsgDIx-F0n29bui4I%3D&hl=mr&source=homepage&sa=X&ved=0ahUKEwjMgeCD8_mJAxXgSwwGHX28Eo0Q2ZgBCAk (https://www.google.com/setprefs?sig=0_XFMPqIqaBMTsgDIx-F0n29bui4I%3D&hl=mr&source=homepage&sa=X&ved=0ahUKEwjMgeCD8_mJAxXgSwwGHX28Eo0Q2ZgBCAk)
https://www.google.com/setprefs?sig=0_XFMPqIqaBMTsgDIx-F0n29bui4I%3D&hl=ta&source=homepage&sa=X&ved=0ahUKEwjMgeCD8_mJAxXgSwwGHX28Eo0Q2ZgBCAo (https://www.google.com/setprefs?sig=0_XFMPqIqaBMTsgDIx-F0n29bui4I%3D&hl=ta&source=homepage&sa=X&ved=0ahUKEwjMgeCD8_mJAxXgSwwGHX28Eo0Q2ZgBCAo)
https://www.google.com/setprefs?sig=0_XFMPqIqaBMTsgDIx-F0n29bui4I%3D&hl=gu&source=homepage&sa=X&ved=0ahUKEwjMgeCD8_mJAxXgSwwGHX28Eo0Q2ZgBCAs (https://www.google.com/setprefs?sig=0_XFMPqIqaBMTsgDIx-F0n29bui4I%3D&hl=gu&source=homepage&sa=X&ved=0ahUKEwjMgeCD8_mJAxXgSwwGHX28Eo0Q2ZgBCAs)
https://www.google.com/setprefs?sig=0_XFMPqIqaBMTsgDIx-F0n29bui4I%3D&hl=kn&source=homepage&sa=X&ved=0ahUKEwjMgeCD8_mJAxXgSwwGHX28Eo0Q2ZgBCAw (https://www.google.com/setprefs?sig=0_XFMPqIqaBMTsgDIx-F0n29bui4I%3D&hl=kn&source=homepage&sa=X&ved=0ahUKEwjMgeCD8_mJAxXgSwwGHX28Eo0Q2ZgBCAw)
https://www.google.com/setprefs?sig=0_XFMPqIqaBMTsgDIx-F0n29bui4I%3D&hl=ml&source=homepage&sa=X&ved=0ahUKEwjMgeCD8_mJAxXgSwwGHX28Eo0Q2ZgBCA0 (https://www.google.com/setprefs?sig=0_XFMPqIqaBMTsgDIx-F0n29bui4I%3D&hl=ml&source=homepage&sa=X&ved=0ahUKEwjMgeCD8_mJAxXgSwwGHX28Eo0Q2ZgBCA0)
https://www.google.com/setprefs?sig=0_XFMPqIqaBMTsgDIx-F0n29bui4I%3D&hl=pa&source=homepage&sa=X&ved=0ahUKEwjMgeCD8_mJAxXgSwwGHX28Eo0Q2ZgBCA4 (https://www.google.com/setprefs?sig=0_XFMPqIqaBMTsgDIx-F0n29bui4I%3D&hl=pa&source=homepage&sa=X&ved=0ahUKEwjMgeCD8_mJAxXgSwwGHX28Eo0Q2ZgBCA4)
[/intl/en/ads/](#)
<http://www.google.co.in/services/> (<http://www.google.co.in/services/>)
[/intl/en/about.html](#)

https://www.google.com/setprefdomain?prefdom=IN&prev=https://www.google.co.in/&sig=K_xCCckhVoLHRu9klVzxd_yhHPuts%3D (https://www.google.com/setprefdomain?prefdom=IN&prev=https://www.google.co.in/&sig=K_xCCckhVoLHRu9klVzxd_yhHPuts%3D)
</intl/en/policies/privacy/>
</intl/en/policies/terms/>

```
In [207]: for var in gpage.find_all('a'):
    if(re.search('^http|https.*',var.get('href'))): # filter line starts with
        print(var.get('href'))
```

<https://www.google.com/imghp?hl=en&tab=wi> (<https://www.google.com/imghp?hl=en&tab=wi>)
<https://maps.google.co.in/maps?hl=en&tab=w1> (<https://maps.google.co.in/maps?hl=en&tab=w1>)
<https://play.google.com/?hl=en&tab=w8> (<https://play.google.com/?hl=en&tab=w8>)
<https://www.youtube.com/?tab=w1> (<https://www.youtube.com/?tab=w1>)
<https://news.google.com/?tab=wn> (<https://news.google.com/?tab=wn>)
<https://mail.google.com/mail/?tab=wm> (<https://mail.google.com/mail/?tab=wm>)
<https://drive.google.com/?tab=wo> (<https://drive.google.com/?tab=wo>)
<https://www.google.co.in/intl/en/about/products?tab=wh> (<https://www.google.co.in/intl/en/about/products?tab=wh>)
<http://www.google.co.in/history/optout?hl=en> (<http://www.google.co.in/history/optout?hl=en>)
<https://accounts.google.com/ServiceLogin?hl=en&passive=true&continue=https://www.google.com/&ec=GAZAAQ> (<https://accounts.google.com/ServiceLogin?hl=en&passive=true&continue=https://www.google.com/&ec=GAZAAQ>)
https://www.google.com/setprefs?sig=0_XFMPqIqaBMTsgDIx-F0n29bui4I%3D&hl=hi&source=homepage&sa=X&ved=0ahUKEwjMgeCD8_mJAxXgSwwGHX28Eo0Q2ZgBCAY (https://www.google.com/setprefs?sig=0_XFMPqIqaBMTsgDIx-F0n29bui4I%3D&hl=hi&source=homepage&sa=X&ved=0ahUKEwjMgeCD8_mJAxXgSwwGHX28Eo0Q2ZgBCAY)
https://www.google.com/setprefs?sig=0_XFMPqIqaBMTsgDIx-F0n29bui4I%3D&hl=bn&source=homepage&sa=X&ved=0ahUKEwjMgeCD8_mJAxXgSwwGHX28Eo0Q2ZgBCAc (https://www.google.com/setprefs?sig=0_XFMPqIqaBMTsgDIx-F0n29bui4I%3D&hl=bn&source=homepage&sa=X&ved=0ahUKEwjMgeCD8_mJAxXgSwwGHX28Eo0Q2ZgBCAc)
https://www.google.com/setprefs?sig=0_XFMPqIqaBMTsgDIx-F0n29bui4I%3D&hl=te&source=homepage&sa=X&ved=0ahUKEwjMgeCD8_mJAxXgSwwGHX28Eo0Q2ZgBCAg (https://www.google.com/setprefs?sig=0_XFMPqIqaBMTsgDIx-F0n29bui4I%3D&hl=te&source=homepage&sa=X&ved=0ahUKEwjMgeCD8_mJAxXgSwwGHX28Eo0Q2ZgBCAg)
https://www.google.com/setprefs?sig=0_XFMPqIqaBMTsgDIx-F0n29bui4I%3D&hl=mr&source=homepage&sa=X&ved=0ahUKEwjMgeCD8_mJAxXgSwwGHX28Eo0Q2ZgBCAk (https://www.google.com/setprefs?sig=0_XFMPqIqaBMTsgDIx-F0n29bui4I%3D&hl=mr&source=homepage&sa=X&ved=0ahUKEwjMgeCD8_mJAxXgSwwGHX28Eo0Q2ZgBCAk)
https://www.google.com/setprefs?sig=0_XFMPqIqaBMTsgDIx-F0n29bui4I%3D&hl=ta&source=homepage&sa=X&ved=0ahUKEwjMgeCD8_mJAxXgSwwGHX28Eo0Q2ZgBCAo (https://www.google.com/setprefs?sig=0_XFMPqIqaBMTsgDIx-F0n29bui4I%3D&hl=ta&source=homepage&sa=X&ved=0ahUKEwjMgeCD8_mJAxXgSwwGHX28Eo0Q2ZgBCAo)
https://www.google.com/setprefs?sig=0_XFMPqIqaBMTsgDIx-F0n29bui4I%3D&hl=gu&source=homepage&sa=X&ved=0ahUKEwjMgeCD8_mJAxXgSwwGHX28Eo0Q2ZgBCAs (https://www.google.com/setprefs?sig=0_XFMPqIqaBMTsgDIx-F0n29bui4I%3D&hl=gu&source=homepage&sa=X&ved=0ahUKEwjMgeCD8_mJAxXgSwwGHX28Eo0Q2ZgBCAs)
https://www.google.com/setprefs?sig=0_XFMPqIqaBMTsgDIx-F0n29bui4I%3D&hl=kn&source=homepage&sa=X&ved=0ahUKEwjMgeCD8_mJAxXgSwwGHX28Eo0Q2ZgBCAw (https://www.google.com/setprefs?sig=0_XFMPqIqaBMTsgDIx-F0n29bui4I%3D&hl=kn&source=homepage&sa=X&ved=0ahUKEwjMgeCD8_mJAxXgSwwGHX28Eo0Q2ZgBCAw)
https://www.google.com/setprefs?sig=0_XFMPqIqaBMTsgDIx-F0n29bui4I%3D&hl=ml&source=homepage&sa=X&ved=0ahUKEwjMgeCD8_mJAxXgSwwGHX28Eo0Q2ZgBCA0 (https://www.google.com/setprefs?sig=0_XFMPqIqaBMTsgDIx-F0n29bui4I%3D&hl=ml&source=homepage&sa=X&ved=0ahUKEwjMgeCD8_mJAxXgSwwGHX28Eo0Q2ZgBCA0)
https://www.google.com/setprefs?sig=0_XFMPqIqaBMTsgDIx-F0n29bui4I%3D&hl=pa&source=homepage&sa=X&ved=0ahUKEwjMgeCD8_mJAxXgSwwGHX28Eo0Q2ZgBCA4 (https://www.google.com/setprefs?sig=0_XFMPqIqaBMTsgDIx-F0n29bui4I%3D&hl=pa&source=homepage&sa=X&ved=0ahUKEwjMgeCD8_mJAxXgSwwGHX28Eo0Q2ZgBCA4)
<http://www.google.co.in/services/> (<http://www.google.co.in/services/>)
https://www.google.com/setprefdomain?prefdom=IN&prev=https://www.google.co.in/&sig=K_xCCckhVoLHRu9k1Vzxd_yhHPuts%3D (https://www.google.com/setprefdomain?prefdom=IN&prev=https://www.google.co.in/&sig=K_xCCckhVoLHRu9k1Vzxd_yhHPuts%3D)

```
n?prefdom=IN&prev=https://www.google.co.in/&sig=K_xCCckhVoLHRu9klVzxd_yhHPut  
s%3D)
```

```
In [208]: d={'K1':['V1','V2','V3'],'K2':{'K1':['P1','P2','P3']}, 'K3':[{'K1':[1,2,3]}]}
```

```
In [209]: import json  
json.dumps(d) # convert to json
```

```
Out[209]: '{"K1": ["V1", "V2", "V3"], "K2": {"K1": ["P1", "P2", "P3"]}, "K3": [{"K1": [1, 2, 3]}]}
```

```
In [210]: jd = json.dumps(d)  
  
json.loads(jd) # convert to python
```

```
Out[210]: {'K1': ['V1', 'V2', 'V3'],  
           'K2': {'K1': ['P1', 'P2', 'P3']},  
           'K3': [{'K1': [1, 2, 3]}]}
```

```
In [212]: wobj = open('E:\\test.json','w')  
json.dump(d,wobj) # structure ->json_file  
wobj.close()
```

```
In [213]: fobj = open('E:\\test.json')  
python_data = json.load(fobj)  
fobj.close()
```

```
In [214]: python_data
```

```
Out[214]: {'K1': ['V1', 'V2', 'V3'],  
           'K2': {'K1': ['P1', 'P2', 'P3']},  
           'K3': [{'K1': [1, 2, 3]}]}
```

```
In [215]: r = requests.get('https://api.github.com/users/hadley/orgs')  
r.status_code
```

```
Out[215]: 200
```

In [216]: r.headers

```
Out[216]: {'Date': 'Tue, 26 Nov 2024 11:49:04 GMT', 'Content-Type': 'application/json; charset=utf-8', 'Cache-Control': 'public, max-age=60, s-maxage=60', 'Vary': 'Accept,Accept-Encoding, Accept, X-Requested-With', 'ETag': 'W/"61f25e2a55611f7eaf75a20ed5fe501245c5af99bc383854f5442975318fad26"', 'X-GitHub-Media-Type': 'github.v3; format=json', 'x-github-api-version-selected': '2022-11-28', 'Access-Control-Expose-Headers': 'ETag, Link, Location, Retry-After, X-GitHub-OTP, X-RateLimit-Limit, X-RateLimit-Remaining, X-RateLimit-Used, X-RateLimit-Resource, X-RateLimit-Reset, X-OAuth-Scopes, X-Accepted-OAuth-Scopes, X-Poll-Interval, X-GitHub-Media-Type, X-GitHub-SSO, X-GitHub-Request-Id, Deprecation, Sunset', 'Access-Control-Allow-Origin': '*', 'Strict-Transport-Security': 'max-age=31536000; includeSubdomains; preload', 'X-Frame-Options': 'deny', 'X-Content-Type-Options': 'nosniff', 'X-XSS-Protection': '0', 'Referrer-Policy': 'origin-when-cross-origin, strict-origin-when-cross-origin', 'Content-Security-Policy': "default-src 'none'", 'Content-Encoding': 'gzip', 'Server': 'github.com', 'X-RateLimit-Limit': '60', 'X-RateLimit-Remaining': '58', 'X-RateLimit-Reset': '1732625260', 'X-RateLimit-Resource': 'core', 'X-RateLimit-Used': '2', 'Accept-Ranges': 'bytes', 'Content-Length': '1008', 'X-GitHub-Request-ID': '047F:344C37:47132A:4E28B5:6745B5B0'}
```

In [217]: r.headers['Content-Type']

```
Out[217]: 'application/json; charset=utf-8'
```

In [218]: jd = r.text

In [219]: pd = json.loads(jd) # convert to python
print(type(pd), len(pd))

```
<class 'list'> 10
```

In [220]: pd[0]

```
Out[220]: {'login': 'ggobi',
'id': 423638,
'node_id': 'MDEyOk9yZ2FuaXphdGlvbjQyMzYzOA==',
'url': 'https://api.github.com/orgs/ggobi',
'repos_url': 'https://api.github.com/orgs/ggobi/repos',
'events_url': 'https://api.github.com/orgs/ggobi/events',
'hooks_url': 'https://api.github.com/orgs/ggobi/hooks',
'issues_url': 'https://api.github.com/orgs/ggobi/issues',
'members_url': 'https://api.github.com/orgs/ggobi/members{/member}',
'public_members_url': 'https://api.github.com/orgs/ggobi/public_members{/member}',
'avatar_url': 'https://avatars.githubusercontent.com/u/423638?v=4',
'description': ''}
```

In [221]: pd[0]['node_id']

```
Out[221]: 'MDEyOk9yZ2FuaXphdGlvbjQyMzYzOA=='
```

```
In [222]: import requests
url = 'https://www.python.org'
r = requests.get(url)
if r.status_code != 200:
    print(f'requests from {url} is failed')
    exit() # exit from execution

if 'text/html' in r.headers['Content-Type']:
    web_page = r.text
    # do webscraping - bs4 module
elif 'application/json' in r.headers['Content-Type']:
    jd = r.text
    # convert to python - json module
```