```
import pprint

d = {}
d['K1'] = 'V1'
d['K2'] = [10,20,30,40]
d['K3'] = [('record1'),('record2')]
d['K4'] = {'Kx':'Vx','Ky':{'url':'https://www.google.com'},'Kz':{'url':'https://www.python.org'}}

pprint.pprint(d)
```

```
{'K1': 'V1',
 'K2': [10, 20, 30, 40],
 'K3': ['record1', 'record2'],
 'K4': {'Kx': 'Vx',
        'Ky': {'url': 'https://www.google.com'},
        'Kz': {'url': 'https://www.python.org'}}}
```

```
for var in d:
  print(var) # list of keys
```

```
K1
K2
K3
K4
```

```
pprint.pprint(d['K3'][1])
```

```
'record2'
```

```
pprint.pprint(d['K4'])
print('\n')
print(d['K4']['Ky']['url']) # dict of dict
```

```
{'Kx': 'Vx',
 'Ky': {'url': 'https://www.google.com'},
 'Kz': {'url': 'https://www.python.org'}}


https://www.google.com
```

```
d = {}
d['K1'] = 'V1'
d['K2'] = [10,20,30,40]
d['K3'] = [('record1',),('record2',)]
d['K4'] = {'Kx':'Vx','Ky':{'url':'https://www.google.com'},'Kz':{'url':'https://www.python.org'}}
print(d)
```

```
{'K1': 'V1', 'K2': [10, 20, 30, 40], 'K3': [('record1',), ('record2',)], 'K4': {'Kx': 'Vx', 'Ky': {'url': 'https://www.goog
```

```
pprint.pprint(d)
```

```
{'K1': 'V1',
 'K2': [10, 20, 30, 40],
 'K3': [('record1',), ('record2',)],
 'K4': {'Kx': 'Vx',
        'Ky': {'url': 'https://www.google.com'},
        'Kz': {'url': 'https://www.python.org'}}}
```

```
### File Handling
'''
File - Regular File - ASCII / TEXT file
Keyboard(<STDIN>) -----<------input()------------[ Python ] --------------print()----->----- Monitor(STDOUT)
                                              ==============
                                                   |  |
                                                   | read/write
                                                   |  |
                                                Storage Device (FILE)

1. Reading data from <FILE> --> Python Program --> Diplay to monitor ( Not reading from Keyboard)

2. Read data from <STDIN> ---> Python program -->Create a newFile and Write data to FILE  ( Not displaying to Monitor)

3. Reading data from <FILE-1> --> Python Program --> Create a newFile and Write data to FILE (Not using IO)

open a file   ----> open()  ==>  open('filename',mode)  -> fileObject # mode - operation - read mode 'r' write 'w' append
read/create                      fileObject.read() --> str
                                 fileObject.readlines() -->list
                                 fileObject.write(str(......))
close a file              ==>    fileObject.close()
```

```
1. Reading data from <FILE> --> Python Program --> Diplay to monitor ( Not reading from Keyboard)

 fobj = open('filename','r')
 s = fobj.read()
 fobj.close()

  <or>
 fobj = open('filename','r')
 L = fobj.readlines()
 fobj.close()

'''
```

```
fobj = open('emp.csv','r')
s = fobj.read()
fobj.close()
print(type(s),len(s))
```

```
<class 'str'> 253
```

```
print(s)
```

```
eid,ename,edept,ecity,ecost
101,raj,sales,pune,1000
102,leo,prod,bglore,2301
230,raj,prod,pune,2300
450,shan,sales,bglore,3401
542,anu,HR,mumbai,4590
321,bibu,sales,hyd,5419
651,ram,hr,bglore,3130
541,leo,admin,chennai,4913
652,karthik,sales,bglore,3405
```

```
fobj = open('emp.csv','r')
L = fobj.readlines()
fobj.close()
print(type(L),len(L))
```

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

```
print(L)
```

```
['eid,ename,edept,ecity,ecost\n', '101,raj,sales,pune,1000\n', '102,leo,prod,bglore,2301\n', '230,raj,prod,pune,2300\n', '45
```

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

```
eid,ename,edept,ecity,ecost

101,raj,sales,pune,1000

102,leo,prod,bglore,2301

230,raj,prod,pune,2300

450,shan,sales,bglore,3401

542,anu,HR,mumbai,4590

321,bibu,sales,hyd,5419

651,ram,hr,bglore,3130

541,leo,admin,chennai,4913

652,karthik,sales,bglore,3405
```

```
L[-3:] # last 3 lines
```

```
['651,ram,hr,bglore,3130\n',
 '541,leo,admin,chennai,4913\n',
 '652,karthik,sales,bglore,3405']
```

```
for var in L:
  if 'hr' in var:
    print(var)
```

```
651,ram,hr,bglore,3130
```

```
'sales' in '101,raj,sales,pune'  # <== input is string
```

```
True
```

```
'sales' in ['101,raj,sales,pune'] # <== input is list
```

```
False
```

```
'sales' in ['101,raj,sales,pune','sales'] # <== input is list
```

```
True
```

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

```
eid,ename,edept,ecity,ecost

101,raj,sales,pune,1000

102,leo,prod,bglore,2301

230,raj,prod,pune,2300

450,shan,sales,bglore,3401

542,anu,HR,mumbai,4590

321,bibu,sales,hyd,5419

651,ram,hr,bglore,3130

541,leo,admin,chennai,4913

652,karthik,sales,bglore,3405
```

```
s='data\n'
s.strip()
```

```
'data'
```

```
for var in L:
  print(var.strip())
```

```
eid,ename,edept,ecity,ecost
101,raj,sales,pune,1000
102,leo,prod,bglore,2301
230,raj,prod,pune,2300
450,shan,sales,bglore,3401
542,anu,HR,mumbai,4590
321,bibu,sales,hyd,5419
651,ram,hr,bglore,3130
541,leo,admin,chennai,4913
652,karthik,sales,bglore,3405
```

```
'''
2. Read data from <STDIN> ---> Python program -->Create a newFile and Write data to FILE  ( Not displaying to Monitor)

wobj = open('NewFile','w')
wobj.write('Single String\n')
...
wobj.close()
C:\Users\karth>mkdir DAY3

C:\Users\karth>cd DAY3

C:\Users\karth\DAY3>python
Python 3.14.2 (tags/v3.14.2:df79316, Dec  5 2025, 17:18:21) [MSC v.1944 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>>
>>> open('r1.log','w')
<_io.TextIOWrapper name='r1.log' mode='w' encoding='cp1252'>
>>>
>>> open('r1.log','w')
<_io.TextIOWrapper name='r1.log' mode='w' encoding='cp1252'>
>>>
>>>
>>> wobj = open('r1.log','w')
>>>
```

```
>>> wobj.write("data-1\n") # wobj.write('Single String\n')
7
>>> wobj.write(456.32)
Traceback (most recent call last):
  File "<python-input-9>", line 1, in <module>
    wobj.write(456.32)
    ~~~~~~~~~~^^^^^^^^
TypeError: write() argument must be str, not float
>>> wobj.write(str(456.32))
6
>>> wobj.write("Data1","Data2")
Traceback (most recent call last):
  File "<python-input-11>", line 1, in <module>
    wobj.write("Data1","Data2")
    ~~~~~~~~~~^^^^^^^^^^^^^^^^^
TypeError: TextIOWrapper.write() takes exactly one argument (2 given)
>>>
>>> wobj.write("Data1"+"Data2"+"\n")
11
>>> ename="raj"
>>> eid=345
>>> ecost=534234.32
>>>
>>> wobj.write(f'Emp name is:{ename} Emp id:{eid} Cost is:{ecost}\n')
45
>>>
>>> wobj.write(f'{ename},{eid},{ecost}\n')
18
>>> wobj.write(f'{ename}:{eid}:{ecost}\n')
18
>>> wobj.close()
>>>
>>> wobj = open('r1.log','w') # overwrite Vs  wobj=open('r1.log','a') # Append
>>> wobj.write('This is sample data\n')
20
>>> import time
>>> wobj.write(f'data written time is:{time.ctime()}\n')
46
>>> time.ctime()
'Wed Feb 11 11:10:03 2026'
>>>
>>> wobj.write('End of the line\n')
16
>>> wobj.close()
>>>
>>> wobj = open('r1.log','a') # append
>>> wobj.write(str(1233.3434)+"\n")
10
>>> wobj.write(f'Data updated time is:{time.ctime()}\n')
46
>>> wobj.write('-'*15+'\n')
16
>>> wobj.close()
>>>
'''
```

```
wobj = open('r1.log','w')
wobj.write("data-1\n")
wobj.write("Data-2\n")
wobj.write("Product Name is:pA Cost is:456.32\n")
wobj.write("------- End of the line ------\n")
wobj.close()
```

```
fobj = open('r1.log','r')
s = fobj.read()
fobj.close()
print(s)
```

```
data-1
Data-2
Product Name is:pA Cost is:456.32
------- End of the line ------
```

```
# 3. Reading data from <FILE-1> --> Python Program --> Create a newFile and Write data to FILE (Not using IO)

fobj = open('r1.log','r')
```

```
wobj = open('r2.log','w')

s = fobj.read()

wobj.write(s)

fobj.close()
wobj.close()
```

```
# Write a python program:
#  read data from emp.csv file
#  filter sales dept details -----<-- use membership operator
#        |-> Write sales dept details to external file

FH = open('emp.csv','r')
WH = open('sales.log','a')

L = FH.readlines()
for var in L:
  if('sales' in var):
    WH.write(var) # Writing data(sales dept) to file(sales.log)

FH.close()
WH.close()
```

```
print(type(WH))
```

```
<class '_io.TextIOWrapper'>
```

```
net_info = {}
net_info['Type'] = 'Ethernet'
net_info['Interface'] = 'eth0'
net_info['onboot'] = 'No'
net_info['bootproto'] = 'dhcp'
net_info
'''
1. Take this net_info dict as input
2. Create a new config file (ex: network.cfg)
3. Iterate input dict
4.   Write dict data - INI format
         Type=Ethernet
         Interface=eth0
         ..
5. close your file operation
|
6. open network.cfg file
```

```
{'Type': 'Ethernet', 'Interface': 'eth0', 'onboot': 'No', 'bootproto': 'dhcp'}
```

```
for var in net_info:
  print(f'{var} = {net_info[var]}')
```

```
Type = Ethernet
Interface = eth0
onboot = No
bootproto = dhcp
```

```
net_info = {}
net_info['Type'] = 'Ethernet'
net_info['Interface'] = 'eth0'
net_info['onboot'] = 'No'
net_info['bootproto'] = 'dhcp'

wobj = open('network.cfg','w')
for var in net_info:
  wobj.write(f'{var} = {net_info[var]}\n')
wobj.close()
```

```
# 2. Read data from <STDIN> ---> Python program -->Create a newFile and Write data to FILE
 #                                                    ( Not displaying to Monitor)

wobj = open('result.log','w')
for var in range(5):
  myvar = input('Enter a server name:')  # interface to <STDIN>
  wobj.write(f'Input Server name:{myvar}\t')
  myIP = input(f'Enter {myvar} IP:')      # interface to <STDIN>
  wobj.write(f'IP Address:{myIP}\n')
  wobj.write("-"*20+"\n")
wobj.close()
```

```
Enter a server name:OL5
Enter OL5 IP:10.20.30.40
Enter a server name:OL6
Enter OL6 IP:10.33.32.44
Enter a server name:DEB14
Enter DEB14 IP:130.323.33.33
Enter a server name:DEB16
Enter DEB16 IP:19.34.33.22
Enter a server name:Winx
Enter Winx IP:192.161.3.33
```

```
# with as - keywords
>>>
>>> fobj = open('r1.log','r')
>>> L = fobj.readlines()
>>> for var in L:
...     print(var.strip())
...
This is sample data
data written time is:Wed Feb 11 11:09:46 2026
End of the line
1233.3434
Data updated time is:Wed Feb 11 11:11:32 2026
---------------
>>> fobj.close()
>>>
>>> with open("r1.log","r") as fobj:
...     L = fobj.readlines()
...     for var in L:
...         print(var.strip())
...
This is sample data
data written time is:Wed Feb 11 11:09:46 2026
End of the line
1233.3434
Data updated time is:Wed Feb 11 11:11:32 2026
---------------
>>>
>>> with open('r2.log','w') as wobj:
...     wobj.write('data-1\n')
...     for var in range(5):
...         wobj.write(f'data-{var+1}\n')
...         wobj.write(var+100+'\n')
...
7
7
Traceback (most recent call last):
  File "<python-input-8>", line 5, in <module>
    wobj.write(var+100+'\n')
               ~~~~~~~^~~~~
TypeError: unsupported operand type(s) for +: 'int' and 'str'
>>>
>>>
>>> with open('r2.log','w') as wobj:
...     wobj.write('data-1\n')
...     for var in range(5):
...         wobj.write(f'data-{var+1}\n')
...         wobj.write(str(var+100)+'\n')
...

### obj.close() is not required when using with as - keywords
```

```
with open('emp.csv','r') as fobj:
  for var in fobj.readlines():
    if('sales' in var): # filter sales details
      print(var.strip())
```

```
101,raj,sales,pune,1000
450,shan,sales,bglore,3401
321,bibu,sales,hyd,5419
652,karthik,sales,bglore,3405
```

```
with open('e1.csv','w') as wobj:
  for var in range(3):
```

```
    pname = input('Enter product Name:')
    pid = input(f'Enter {pname} product ID:')
    pcost = input(f'Enter {pname} Cost:')
    GST = float(pcost) * 0.18
    TOTAL = GST + float(pcost)
    wobj.write(f'{pid},{pname},{pcost},{GST},{TOTAL}\n')
```

```
Enter product Name:Laptop
Enter Laptop product ID:L1
Enter Laptop Cost:50000
Enter product Name:Mobile
Enter Mobile product ID:M1
Enter Mobile Cost:15000
Enter product Name:SSD
Enter SSD product ID:S1
Enter SSD Cost:23459.32
```

```
fobj = open('e1.csv','r')
print(fobj.read())
```

```
L1,Laptop,50000,9000.0,59000.0
M1,Mobile,15000,2700.0,17700.0
S1,SSD,23459.32,4222.6776,27681.9976
```

```
## Recap
L=[10,20,30,40,50]
total=0
for var in L:
  total = total+var
print(total)
print('next section of code')
d={'K1':'V1','K2':'V2'}
for var in d:
  print(f'{var} - {d[var]}')

d['K2'] = 'Value-2' # updating an existing value
d['K3'] = 'Value-3' # create new value

for var in d:
  print(f'{var} - {d[var]}')

L.append(50)
L.append(60)
total=0
for var in L:
  total = total+var
print(total)

fobj = open('r1.log','r')
s = fobj.read()
fobj.close()
# ..
# ..
fobj = open('r2.log','r')
s = fobj.read()
fobj.close()
```

```
fobj = open('r1.log','r')
s = fobj.read()
print(s)
fobj.close()

fobj = open('e1.csv','r')
s = fobj.read()
print(s)
fobj.close()

fobj = open('r2.log','r')
s = fobj.read()
print(s)
fobj.close()
```

```
def display():
  '''function definition'''
  total = 0
  for var in [10,20,30,40]:
    total = total + var
  print('Sum value is:',total)

display()
```

```
Sum value is: 100
```

```
# Code reusability
# ------------------------
def file_read(a1):
  print(f"File name is:{a1}")
  fobj = open(a1,'r')
  s = fobj.read()
  print(s)
  fobj.close()
  print(f"File {a1} read operation is done!!!\n")

file_read('r1.log') # functionCall with argument
file_read('e1.csv') # functionCall with argument
file_read('r2.log') # functionCall with argument
```

```
File name is:r1.log
data-1
Data-2
Product Name is:pA Cost is:456.32
------- End of the line ------

File r1.log read operation is done!!!

File name is:e1.csv
L1,Laptop,50000,9000.0,59000.0
M1,Mobile,15000,2700.0,17700.0
S1,SSD,23459.32,4222.6776,27681.9976

File e1.csv read operation is done!!!

File name is:r2.log
data-1
Data-2
Product Name is:pA Cost is:456.32
------- End of the line ------

File r2.log read operation is done!!!
```

```
Write a python program:
- modify the pin example
|
create a new file pin_history.log - append mode
|->user input pin - Success/Failed - update to pin_history.log
                                   - update date/time use time module

import time
time.ctime()

Success - Pin Entry time is:....
Failed - Invalid pin :33333 Pin Entry time is:
```

```
import time
wobj = open('pin_history.log','a')
pin = 1234
count = 0
while(count < 3):
  p = input('Enter a pin number:')
  count = count + 1
  if(int(p) == pin):
    print(f'Success - pin is matched - {count}')
    wobj.write(f'Success - pin is matched-{count}-Entry time:{time.ctime()}\n')
    break
  else:
    wobj.write(f'Failed - User input pin is:{p} - Entry time:{time.ctime()}\n')

if(int(p) != pin):
  wobj.write(f'Sorry your pin is blocked. - Entry time:{time.ctime()}\n')
  print('Sorry your pin is blocked')

wobj.close()
```

```
Enter a pin number:12
Enter a pin number:22
Enter a pin number:3345
Sorry your pin is blocked
```

```
def pin_Test():
  wobj = open('pin_history.log','a')
  pin = 1234
  count = 0
  while(count < 3):
    p = input('Enter a pin number:')
    count = count + 1
    if(int(p) == pin):
      print(f'Success - pin is matched - {count}')
      wobj.write(f'Success - pin is matched-{count}-Entry time:{time.ctime()}\n')
      break
    else:
      wobj.write(f'Failed - User input pin is:{p} - Entry time:{time.ctime()}\n')
  if(int(p) != pin):
    wobj.write(f'Sorry your pin is blocked. - Entry time:{time.ctime()}\n')
    print('Sorry your pin is blocked')
  wobj.close()
  print('Test is done!!!')
```

```
pin_Test()
```

```
Enter a pin number:34223
Enter a pin number:3243242
Enter a pin number:1234
Success - pin is matched - 3
Test is done!!!
```

```
p = input('Enter a pinNumber:')
if(len(p) != 4):
  print('Usage:pin number must be 4 digits')
  #exit()
```

```
Enter a pinNumber:43
Usage:pin number must be 4 digits
```

```
L=[]
# L.append('Data1')
# L.append("D2","D3")
# L.append()
L=[10,20,30,40,50]
L.pop()  # OK
L.pop(1) # OK
# L.pop(2,3) TypeError: pop expected at most 1 argument, got 2
```

Show hidden output

Next steps: Explain error

```
def f1(a1,a2): # Required arguments
  print('a1=',a1)
  print("a2=",a2)

# f1() TypeError: f1() missing 2 required positional arguments: 'a1' and 'a2'
# f1(10) TypeError: f1() missing 1 required positional argument: 'a2'
# f1(10,20,30) TypeError: f1() takes 2 positional arguments but 3 were given
```

```
def f2(a1=100,a2='data-1'): # Default arguments
  print("a1=",a1)
  print("a2=",a2)

f2()
f2(10)
f2('r1.csv','10.20.30.40')
# f2(1,2,3) TypeError: f2() takes from 0 to 2 positional arguments but 3 were given
```

```
a1= 100
a2= data-1
a1= 10
a2= data-1
a1= r1.csv
a2= 10.20.30.40
```

```
def f3(a1,a2,a3,a4=True,a5=0.0): # 3required arguments + 2default args
  print(f'a1={a1} a2={a2} a3={a3}')
  print(f'a4={a4} a5={a5}')

# f3() TypeError: f3() missing 3 required positional arguments: 'a1', 'a2', and 'a3'
f3(10,20,30)
f3("D1","D2","D3","D4")
f3("F1","F2","F3","F4","F5")
```

```
a1=10 a2=20 a3=30
a4=True a5=0.0
a1=D1 a2=D2 a3=D3
a4=D4 a5=0.0
a1=F1 a2=F2 a3=F3
a4=F4 a5=F5
```

```
def fx(a1,a2,a3=0,a4):
  print("OK")
```

```
  File "/tmp/ipython-input-2753258766.py", line 1
    def fx(a1,a2,a3=0,a4):
                      ^
SyntaxError: parameter without a default follows parameter with a default
```

Next steps: Explain error

```
# we can define N no.of requrired arguments followed by we can define N no.of default args

L=[]
# L.append('Data1')
# L.append("D2","D3")
# L.append()
L=[10,20,30,40,50]
L.pop()  # OK
L.pop(1) # OK
# L.pop(2,3) TypeError: pop expected at most 1 argument, got 2

def append(Value):  <===
  ...
def pop(Value=-1):  <===
```

```
# Variable length args - tuple -> immutable

def f1(*args):          # Vs  def f1(args): <-- required args
  print(type(args))
  print("-->",args)

f1()
f1(10,20,30,40)
f1(1,2,3,4,'D1','D2','D3','D4','d5')
f1(['F1','F2'],('F3','F4'),{'K1':'V1'})
```

```
<class 'tuple'>
--> ()
<class 'tuple'>
--> (10, 20, 30, 40)
<class 'tuple'>
--> (1, 2, 3, 4, 'D1', 'D2', 'D3', 'D4', 'd5')
<class 'tuple'>
--> (['F1', 'F2'], ('F3', 'F4'), {'K1': 'V1'})
```

```
f1(DB="emp.db",port=3343) # functionCall
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
/tmp/ipython-input-3474631143.py in <cell line: 0>()
----> 1 f1(DB="emp.db",port=3343)

TypeError: f1() got an unexpected keyword argument 'DB'
```

Next steps: Explain error

```
def f4(**a1):  # keyword arguments
  print(type(a1))
  print(a1)


f4()
f4(DB='emp.db',port=3366,IP='127.0.0.1')
```

```
<class 'dict'>
{}
<class 'dict'>
{'DB': 'emp.db', 'port': 3366, 'IP': '127.0.0.1'}
```

```
f4(10,20,30)
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
/tmp/ipython-input-4081917241.py in <cell line: 0>()
----> 1 f4(10,20,30)

TypeError: f4() takes 0 positional arguments but 3 were given
```

Next steps: Explain error

```
def functioName(a1,a2,a3,a4,a5,a6=Value,a7=Value,a8=value,*a9,**a10):
                                                             ===========
```

```
simple function call
call with arguments
              |->required,defaultArguments,VariableLengthArgs,KeywordArgs

scope of variable
```

```
def f1(fname): # Required Argument
  fobj = open(fname,'r')
  s = fobj.read()
  fobj.close()
  print(s)
  print('End of the function-f1 definition\n')

f1('emp.csv') # OK
# f1() ->Error
# f1('emp.csv','r1.log') # -->Error
```

```
def f2(fname='emp.csv'): # Default args
  fobj = open(fname,'r')
  s = fobj.read()
  fobj.close()
  print(s)
  print('End of the function-f2 definition\n')

f2()
f2('r1.log')
```

```
eid,ename,edept,ecity,ecost
101,raj,sales,pune,1000
102,leo,prod,bglore,2301
230,raj,prod,pune,2300
450,shan,sales,bglore,3401
542,anu,HR,mumbai,4590
321,bibu,sales,hyd,5419
651,ram,hr,bglore,3130
541,leo,admin,chennai,4913
652,karthik,sales,bglore,3405
End of the function-f2 definition

data-1
Data-2
Product Name is:pA Cost is:456.32
------- End of the line ------

End of the function-f2 definition
```

```
def f2(*fname): # Variable args
  print(fname)
  if(len(fname) == 0):
    print("Empty arguments")
    return # break ->exit from loop ; return ->exit from function definition
  for var in fname: # fname - tuple
      fobj = open(var,'r')
      s = fobj.read()
      fobj.close()
      print(s)
  print('End of the function-f2 definition\n')

f2()
```

```
f2('r1.log')
f2('emp.csv','e1.csv','network.cfg')
```

```
()
Empty arguments
('r1.log',)
data-1
Data-2
Product Name is:pA Cost is:456.32
------- End of the line ------

End of the function-f2 definition

('emp.csv', 'e1.csv', 'network.cfg')
eid,ename,edept,ecity,ecost
101,raj,sales,pune,1000
102,leo,prod,bglore,2301
230,raj,prod,pune,2300
450,shan,sales,bglore,3401
542,anu,HR,mumbai,4590
321,bibu,sales,hyd,5419
651,ram,hr,bglore,3130
541,leo,admin,chennai,4913
652,karthik,sales,bglore,3405
L1,Laptop,50000,9000.0,59000.0
M1,Mobile,15000,2700.0,17700.0
S1,SSD,23459.32,4222.6776,27681.9976

Type = Ethernet
Interface = eth0
onboot = No
bootproto = dhcp

End of the function-f2 definition
```

```
def f3(**kwargs):
  if(len(kwargs) == 0):
    print('Empty args')
    return
  fobj = open(kwargs['fname'],'r')
  s = fobj.read()
  print(s)
  fobj.close()


f3()
print('')
f3(fname="emp.csv")
```

```
Empty args

eid,ename,edept,ecity,ecost
101,raj,sales,pune,1000
102,leo,prod,bglore,2301
230,raj,prod,pune,2300
450,shan,sales,bglore,3401
542,anu,HR,mumbai,4590
321,bibu,sales,hyd,5419
651,ram,hr,bglore,3130
541,leo,admin,chennai,4913
652,karthik,sales,bglore,3405
```

```
def app():
  app_name = 'flask'
  app_port = 5000
  print(f'App name is:{app_name}')
  print(f'Running port number:{app_port}')


app()
# print(app_name) - NameError: name 'app_name' is not defined
```

```
App name is:flask
Running port number:5000
```

```
# global keyword
def app():
  global app_name,app_port
  app_name = 'flask'
  app_port = 5000
  print(f'App name is:{app_name}')
  print(f'Running port number:{app_port}')
```

```
app()
```

```
App name is:flask
Running port number:5000
```

```
print(app_name,app_port)
```

```
flask 5000
```

```
def fx():
  print(app_name,app_port)

fx()
```

```
flask 5000
```

```
def fy():
  myvar = 100
  return myvar # exit from function block - exit with this value

fy()
```

```
100
```

```
rv = fy()
print(f'rv value:{rv}')
```

```
rv value:100
```

```
def fx():
  var = 450
  return 10,20,30,var # tuple

fx()
```

```
(10, 20, 30, 450)
```

```
>>> var=10
>>> type(var)
<class 'int'>
>>>
>>> var=10,
>>> var
(10,)
>>> type(var)
<class 'tuple'>
>>>
>>> L=[("R1"),("R2"),("R3")]
>>> L
['R1', 'R2', 'R3']
>>>
>>> "R1"
'R1'
>>> "R1",
('R1',)
>>>
>>> L=[("R1",),("R2",),("R3",)]
>>> L
[('R1',), ('R2',), ('R3',)]
>>>
>>>
```

```
def f1():
  var = 10

# default return will be None

f1()
```

```
rv = f1()
print(rv)
```

```
None
```

```
def f1(a1):
  if(a1 > 100):
    return a1+100
```

```
L = []

for var in [110,120,30,140,50]:
  rv = f1(var)
  L.append(rv)

print(L)
print(L[-3:])
```

```
[210, 220, None, 240, None]
[None, 240, None]
```

```
relational operators
logical operators
memebership operators
-------------------> expression returns ->bool

obj.methodcall() ----->bool
```

```
s="python"
s.islower()
```

```
True
```

```
s.isupper()
```

```
False
```

```
if(s.isupper()):
  print(f'Yes - Given String {s} is uppercase')
else:
  print(f'No - Given String {s} is not uppercase')
```

```
No - Given String python is not uppercase
```

```
# bool(AnyValue) ->True/False  if() <== bool
print(bool(0),bool(0.0),bool(''),bool([]),bool(()),bool({}),bool(None))
```

```
False False False False False False False
```

```
bool(-1)
```

```
True
```

```
login_name = input('Enter a login name:')
if(len(login_name) == 0):
  print(f'Input is is empty')
else:
  print(f'Hello {login_name}')
```

```
Enter a login name:
Input is is empty
```

```
login_name = input('Enter a login name:')
if(login_name):
  print(f'Hello..{login_name}')
else:
  print('Input is empty')
```

```
Enter a login name:
Input is empty
```

```
login_name = input('Enter a login name:')
if(login_name):
  print(f'Hello..{login_name}')
else:
  print('Input is empty')
```

```
Enter a login name:root
Hello..root
```

```
'''
file: myfile.py                                  file: p1.py
-------------------                              ---------
def display():                                    import myfile
  '''display customer records'''                  myfile.display() # OK
```

```
                                                  print("Hello")
def process():                                   -----------------
  '''display process details'''

def files():
  '''display list of files'''
'''   -------------------------//module

module
 - existing python code
 - reusability

 import <moduleName>
 <moduleName>.<member>
                  |--->variable,function,class,...
```

```
import sys
print(sys.version)
```

```
3.12.12 (main, Oct 10 2025, 08:52:57) [GCC 11.4.0]
```

```
print(sys.platform)
```

```
linux
```

```
print(sys.argv)
help(sys)
```

```
import os
os.system("uname -a")
```

```
0
```

```
os.system("sleep 5")
```

```
0
```

```
os.system("command") <-- python
                          | | 0 <== status Code - Success
                        subshell (commandline terminal)
                          | |
                          command - execute
```

```
os.getcwd()
```

```
'/content'
```

```
os.listdir(".")
```

```
['.config',
 'r2.log',
 'r1.log',
 'result.log',
 'emp.csv',
 'pin_history.log',
 'e1.csv',
 'network.cfg',
 '.ipynb_checkpoints',
 'sales.log',
 'sample_data']
```

```
# os - os commands
# sys - python related info
# re - regular expression
# json - json data
# ...  - ...
import sys
sys.modules
sys.path
```

```
['/content',
 '/env/python',
 '/usr/lib/python312.zip',
 '/usr/lib/python3.12',
 '/usr/lib/python3.12/lib-dynload',
 '',
 '/usr/local/lib/python3.12/dist-packages',
 '/usr/lib/python3/dist-packages',
 '/usr/local/lib/python3.12/dist-packages/IPython/extensions',
 '/root/.ipython']
```

```
>>> import os
>>> os.system("whoami")
paka\karth
0
>>>
>>> r = os.system("whoami")
paka\karth
>>>
>>> r
0
>>> os.popen("whoami")
<os._wrap_close object at 0x0000023CC7BDC6E0>
>>>
>>> os.popen("whoami").read()
'paka\\karth\n'
>>>
>>> os.popen("whoami").read().strip()
'paka\\karth'
>>>
>>> login_name = os.popen("whoami").read().strip()
>>>
>>> login_name
'paka\\karth'
>>>
>>> os.popen("whoami").readlines()
['paka\\karth\n']

pip install <moduleName>
```

```
file: E:\> myfile.py
------------------------
def app():
    return "<h1>Welcome</h1>"

code = 201
-------------------------
E:\> python myfile.py {Enter}
<empty>

file: E:\> p1.py
------------------
import myfile
print(myfile.app())
print(myfile.code)
if(myfile.code > 100):
    return True
else:
    return False
---------------------------
E:\>python p1.py {Enter}

E:\>
 |--->myfile.py
 |--->p1.py
 |
 |-->D1/
      |-->p2.py
          ======
      import myfile
      print(myfile.code)
      ===============
      python p2.py {Enter}
      ModuleNotFound Error - myfile


import <moduleName>
    ..........
    1st - directory - refer sys.path ->list
                    |-> found the directory
                              |--->myfile.py
                              |--->myfile.pyc <-- pvm
                              |
                              | --> __pycache__/ <==
                    |-> python (pvm) interpreat code line by line


E:\>
```

```
 |--->myfile.py
 |--->p1.py
 |
 |-->D1/
      |-->p2.py
          ======
      import myfile
      print(myfile.code)
      ===============
        |
        open this code editor
        import sys
        sys.path.append("E:\\")
        |
        import myfile
        print(myfile.code) -> OK

      python p2.py {Enter}
      ...
     ------------------------

env variable => PYTHONPATH


Linux
--------
~/.bashrc
  |
 export PYTHONPATH="/home/user/project/"

Winx
------
env variable => PYTHONPATH -> /home/user/project


>>> import os
>>> os.system("whoami")
paka\karth
0
>>>
>>> r = os.system("whoami")
paka\karth
>>>
>>> r
0
>>> os.popen("whoami")
<os._wrap_close object at 0x0000023CC7BDC6E0>
>>>
>>> os.popen("whoami").read()
'paka\\karth\n'
>>>
>>> os.popen("whoami").read().strip()
'paka\\karth'
>>>
>>> login_name = os.popen("whoami").read().strip()
>>>
>>> login_name
'paka\\karth'
>>>
>>> os.popen("whoami").readlines()
['paka\\karth\n']
```

```
############## End of the Day3 Session ##############
```