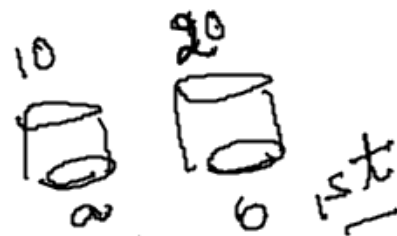
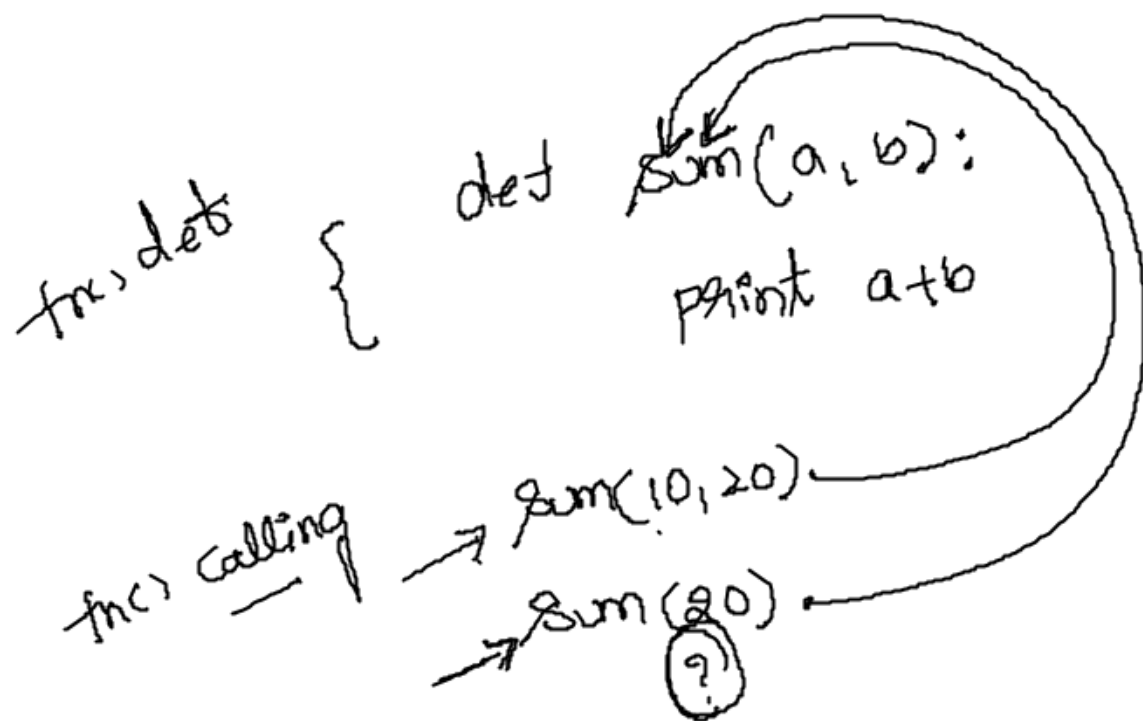


Ex

Python

functions ③ default arguments | parameters
diff values



20 ??

def sum(a=10, b=20):

print a+b

sum()

sum(30)

sum(40, 50)

→ o/p

o/p

o/p

20

50

90

40

30

10



50

20

20 ⇒ default



func1 with variable no. of arguments

```
def myargs (var1, var2, var3, var4, ... var, n):
    print myvar1, myvar2, myvar3
```

myargs (args)
~~args~~
n. no. of args

def myvars (var, *myvar):

print var,

print myvar, type(myvar)

myvars(10, 20, 30, 40, 50, 60, 70, 80, 90, 100)

tuple

tuple

working with files :-

object / reference

object₁

=

[open("myfile.txt", "a")]

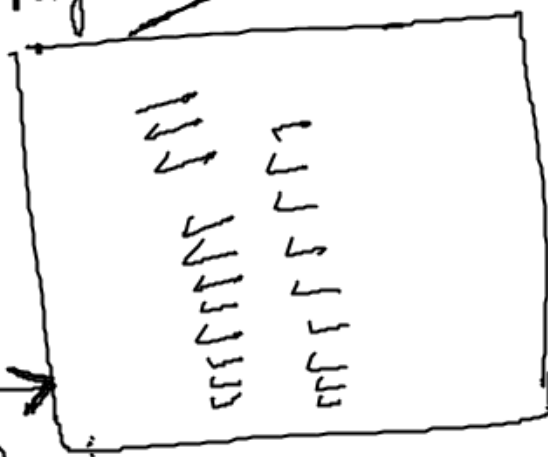
object₁ . readlines()

" . writelines()

" . delete lines()

✓ read (r)
✓ write (w)
✓ append (a)

myfile.txt



(address of ob)

open → (r)

only read data ⇒ note:-
file should exist

open → (w)

read, write

note:- If file exists then it will nullify the file content

If file not exists then it creates file

open → (a)

read, write

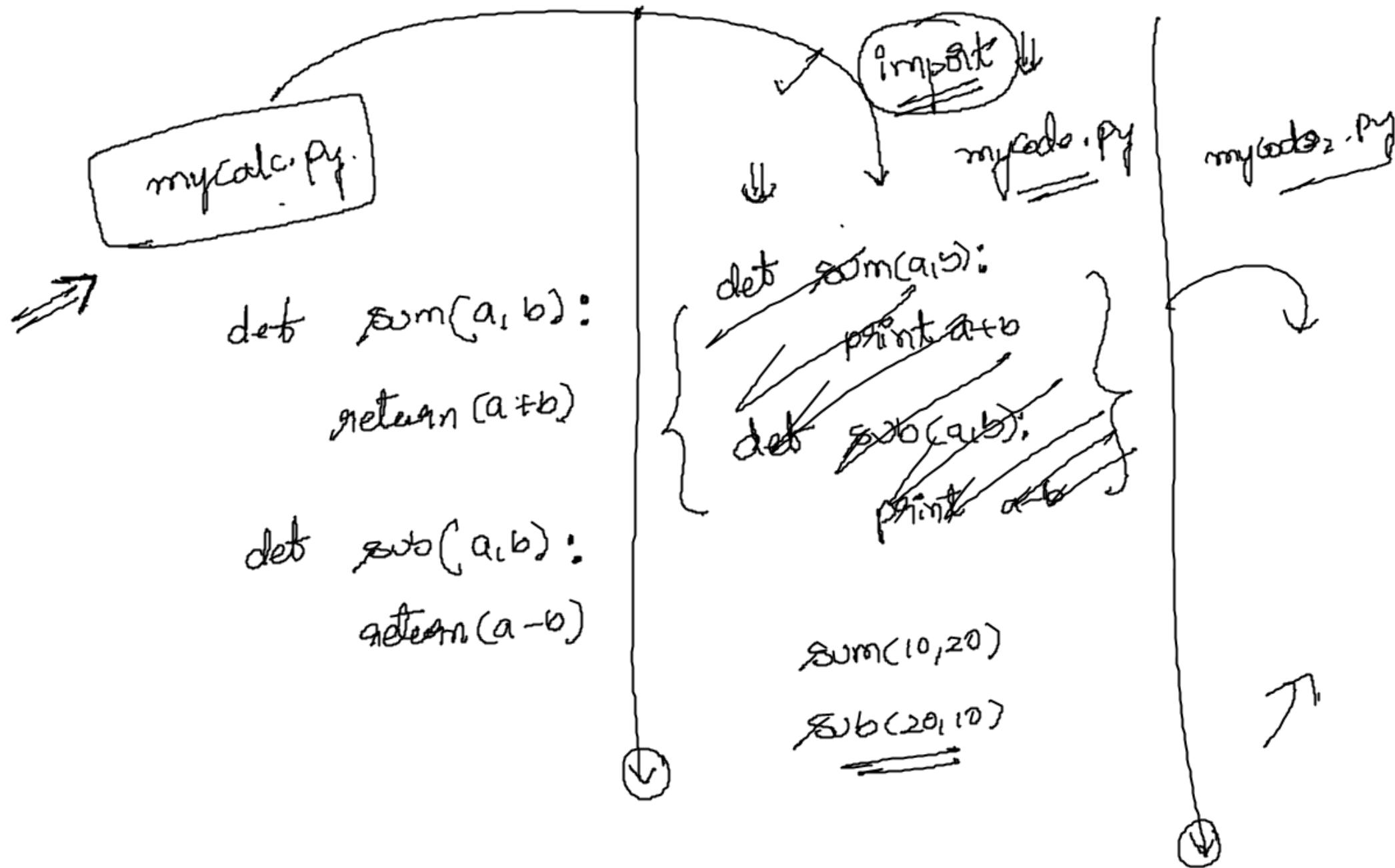
note:- If file not exists then it creates file

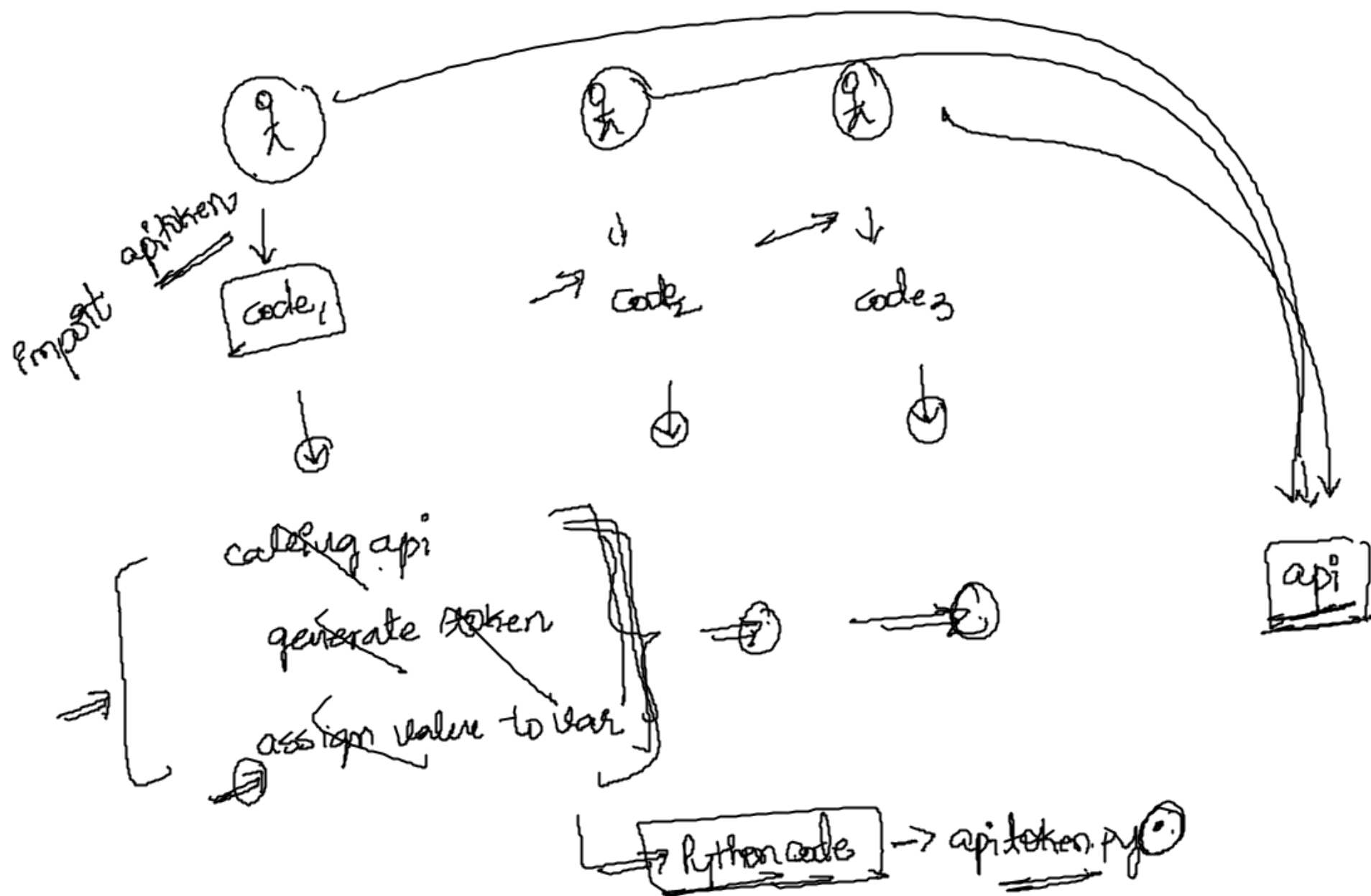
If file exists then it will write data at the end of the file

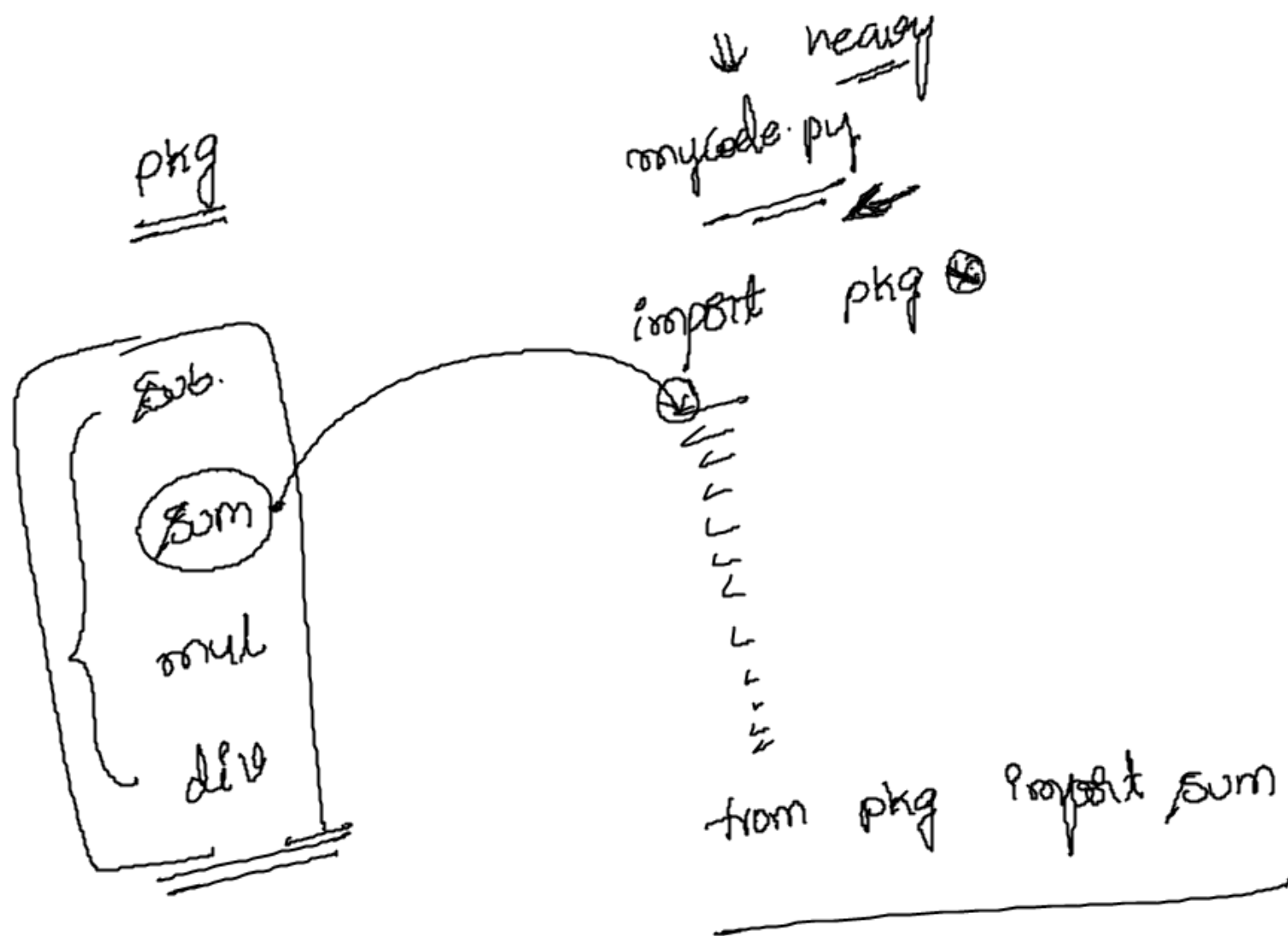
Regular expressions



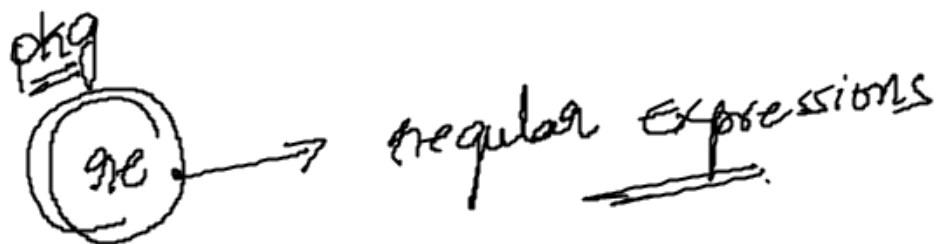
import[⊙]
{ from[⊙] }
=







10 pkg's



Export re

Working with OS

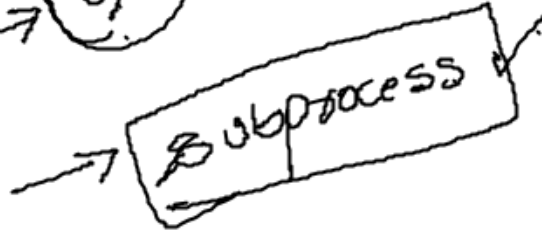
shell

{ uname -a
df -HT }

=



✓

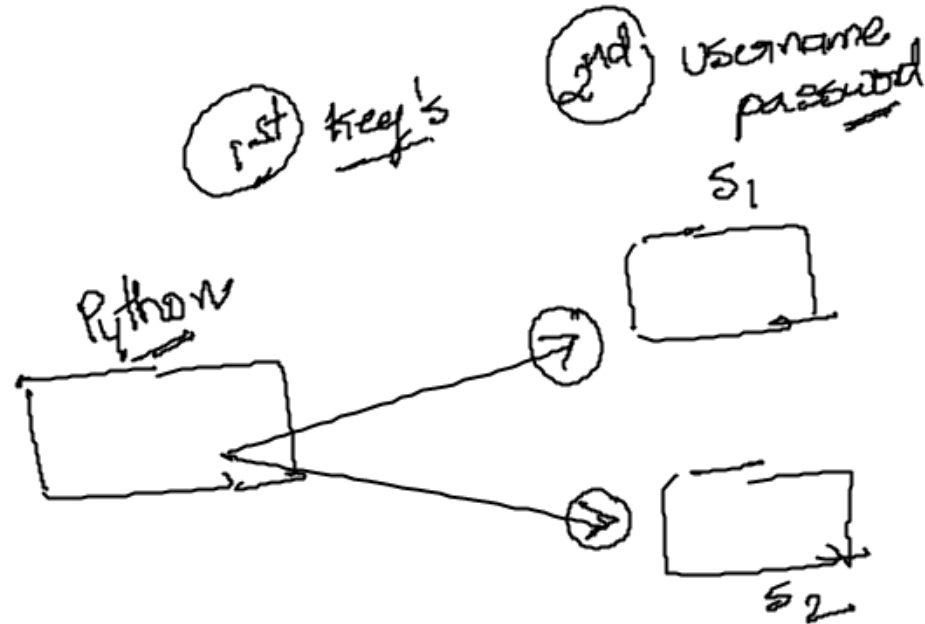


os.popen(command).read()

working with remote servers

Python → password
→ keybased

[paramiko
netmiko]



```
#!/usr/bin/python
import paramiko
k = paramiko.RSAKey.from_private_key_file("/tmp/ssh/class_devops.pem")
ssh = paramiko.SSHClient()
ssh.set_missing_host_key_policy(paramiko.AutoAddPolicy())
print "connecting"
ssh.connect( hostname = "52.56.74.235", username = "ec2-user", pkey = k )
print "connected"
commands = [ "/sbin/ifconfig eth0", "df -hT" ]
for command in commands:
    print "Executing {}".format( command )
    stdin, stdout, stderr = ssh.exec_command(command)
    if stderr:
        print "Command execution failed"

    print stdout.read()
    print( "Errors")
    print stderr.read()
ssh.close()
```

Ctrl