Selenium----> library------>collection module(component)  
module-----> class,def,variable  
3rd party library----------> not present--- -> install---->2 install ???? pip  
open source  
Automation---> UI,Database,API  
version-----> 3.x now 4.x  
python script---->selenium----->AUT----> application under test  
concept----->  
webapplication----->???  
RR cycel---->Request and response  
GUI----- presentation  
database-----> backend  
API---->  
json----> import json  
'key':value ---->str  
dict----> key:value ----> dict ----> key is immutable  
2 methods----->  
loads()---------> json object to python object  
dumps() -----> python object to json object  
webdriver------>???-----> browser  
100%----->  
what disadvan--->??  
barcode ----->manual  
payment ----->manual  
capcha -----> manual  
Selenium------>  
webdriver-----> class---->??  
class----> enchapsulation  
locator -----> variables  
methods -----> actions  
desktop ----->  
No reporting capacuty------>unittest,pytest  
from selenium import webdriver  
requirements.txt  
pip install requirements.txt-- 200+ ------>  
virtual env------>  
DeprecationWarning:  
4.x ----> 3.x----> 12+  
python2 dict ----> unorder  
python3.6+----> order dict  
  
https://www.selenium.dev/documentation/webdriver/getting\_started/upgrade\_to\_selenium\_4/#firefox-legacy  
  
title=??  
if title =="Upgrade to Selenium 4 | Selenium" ----> pass/fail

*#day1***from** selenium **import** webdriver  
**import** chromedriver\_autoinstaller  
chromedriver\_autoinstaller.install()  
driver=webdriver.Chrome()  
driver.get(**'https://www.selenium.dev/documentation/webdriver/getting\_started/upgrade\_to\_selenium\_4/#firefox-legacy'**)  
**if** driver.title==**"Upgrade to Selenium 4 | Selenium"**:  
 print(**'pass'**)  
**else**:  
 print(**'Fail'**)  
driver.close()

Types of frameworks----->selenium----  
1 Modular Testing Framework------>used--->resume  
 every tc consider in sepreate----->test\_login.py  
 ----> test\_dash.py  
 -----> test\_resig.py  
 all the modules are independed to each  
 seprete copy of file(.py)----> is created  
 from module import class----> user file  
  
2 Data driven framework----> USED  
 tc data is present inside outside of tc  
 we are going to use that data--->  
 test\_login.py----> n user-----> test manager  
 username---->  
 password---->  
 used data---->  
 module--->openpyxl,mysql-connector or pymysql, csv----> pip install pymysql  
 exel(100),database,csv  
 mockey testing-----> test engineer  
  
3 Keyword driven framework---->modular and data driver  
 web elements(locators),description,keyword----> excel,action(optional)  
  
4 Hybrid framwork---->combination of all the framwework----> used  
hybrid framwork with POM(page object module)----->yes  
page-----> is collection of web elements(locator)  
object----->yes----> based on object we call the method  
module---->yes----> test\_\*.py or \*\_test.py  
  
abc.py---->  
 import openpyxl  
 import login\_page  
 class Test\_login:  
 def test\_enter\_usename():  
 locator----------------> page.py(all the locators)  
 openpyxl----> usename/ database  
  
 def test\_enter\_password():  
 openpyxl----> usename/database  
  
 def test\_click\_on\_sub():  
 click----> function/action  
  
l1=login()  
l1.test\_enter\_name()  
100----> 5min----> 500min

framwork structure---> directory structure  
module----> compont  
1---> login--->3 file---->  
 login\_page.py ------> all locator define  
 login\_test.py -------> test cases---->  
 login\_helper.py ------>setup(open my brower) and teardown(close brower) ----> unitest/pytest  
  
driver.close()---->current brower will close  
driver.quite()-----> all brower ----> chrome  
  
  
how finding locators---->  
tools----->extention----------> no one takking----->  
 chropath------>  
 sectorhub------>id,css,xpath,class  
  
  
\*\*locators--->web elements----> locator on brower  
id ----->strong----> unique in nature----> unique-identify  
----------------------------------------------------------  
  
class ---->mutlple on web browsers----> radio button  
 --->list--->li ul  
 ----> checkbox  
 ----> dropdwon  
-----------------------------------------------------------------------------  
name ----> radio button,checkbox----> can be different in same group  
  
input type="radio" name='Male'  
input type="radio" name='Female'  
  
---------------------------------------------------------------  
  
linktext---->  
 full linktext-----> we specfy all the context of link  
 Register  
 partial linktext---->we specfy all the substring of link  
 Reg or ster  
----------------------------------------------------------  
css---->\*\* in deatils ---->  
customise css selector  
 tag and id  
 tag and class  
 tag and attribute  
 tag class and attribute  
--------------------------------------------------------------------  
xpath----> slow---->last  
 DOM---> document object model  
abs xpath---> full path---> root node  
/html/body/div[6]/div[1]/div[2]/div[2]/form/input  
  
rel xpath---->partial----> midle node  
//\*[@id="small-searchterms"]  
---------------------------------------------------------------  
  
types----->user ----> time cosumming  
  
css----> casting style sheet----> web developement  
  
  
selenium3---->  
a=find\_element\_by\_id('')  
str--------------->immutable----> new location  
b=find\_elements\_by\_class('')  
list------>[]-----> python list----> mutable--->CRUD  
  
  
selenium4----->  
from selenium.webdriver.common.by import By  
driver.find\_element(By.ID,'id of element')  
driver.find\_element(By.CLASS,'class of element')  
driver.find\_element(By.NAME,'NAME of element')  
driver.find\_element(By.XPATH,'XPATH of element')  
driver.find\_element(By.PARTIAL\_TEXT,'id of element')  
driver.find\_element(By.LINK\_TEXT,'id of element')  
driver.find\_element(By.CSS,'id of element')  
  
sleep()  
Chrome()  
get()  
maximize\_window()-----> maximize  
close()  
send\_keys('input')  
  
WAIT()---->  
selenium 2 types ---->  
implicit wait()---->if wait on brower level  
explicit wait()---->if wait at locator or any  
  
  
  
how to handle multiple windows----> 3 function  
windrow\_handler()--->collection of all the windows  
current\_window\_handle()---->unique id of current open window  
switch\_to\_window()----->switching betweens  
l1=driver.windrow\_handler()  
l2=current\_window\_handle()  
for l2 in l1:  
 if l2.  
 swith\_to\_window()

**import** time  
**from** selenium.webdriver.common.by **import** By  
**from** selenium **import** webdriver  
**import** chromedriver\_autoinstaller  
chromedriver\_autoinstaller.install()  
driver=webdriver.Chrome()  
driver.get(**'https://demo.nopcommerce.com/'**)  
driver.maximize\_window()  
time.sleep(2)  
*#driver.find\_element(By.ID,'small-searchterms').send\_keys('Lenovo')  
#driver.find\_element(By.NAME,'q').send\_keys('Lenovo')  
#driver.find\_element(By.XPATH,"//button[@class='button-1 search-box-button']").click()  
#driver.find\_element(By.LINK\_TEXT,'Register').click()*driver.find\_element(By.PARTIAL\_LINK\_TEXT,**'Reg'**).click()  
time.sleep(3)  
driver.close()

MVT-----> django  
M----->Model----> model.py-----> DATABASE in application  
V----->View-----> bis----> processing the data----> CRUD  
T----->Template--->html,css,jquery,javascript----> frontend  
  
Commands in selenium----->  
1 application commands----->variable  
 get(url)--->  
 title----->  
 current\_url---->  
 page\_source---->  
2 Conditional statement---->\*\*\*  
is\_displayed() -----> locator displayed ---> True  
is\_enabled() ----> if locator is clicable or taking input ---> True  
is\_selected() -----> radio  
  
3 Browser Commands--->  
close()----> single browser close  
quite()----> open browser close  
submit() ----> enter from keyboard  
  
4 Navigation commands--->  
forword() ----> next browser page  
back()------> previous browser page  
refresh() -----> page refresh  
  
5 WAIT---->  
implicit wait---->sec  
 if we want to apply wait for whole application  
 all the locator  
 driver.implicitly\_wait(5) # 5sec  
explicit wait---->  
 perticular codition,element  
 WebDriverWait----> selenium.webdriver.  
 #5sec  
  
from selenium.webdriver.support.wait import WebDriverWait  
from selenium.webdriver.support import expected\_conditions as EC  
locator.until(EC.presence\_of\_element\_located(By.NAME,'q'))  
expected wait is 5sec----> defect/bug  
konsa wait()----->  
time.sleep() ---->  
TC---->componet / Product ----> explication  
FAIL-----> elementNOtprsent expection  
 ---> elementnotdisploye  
 ---> elementnotvisible  
POM-----> page object module  
  
type='checkbox' and contains(@id,'day')  
Monday  
Tuesday  
Wednesday  
Thursday  
Friday  
Saturday  
Sunday  
  
  
alr--->  
3 type ---> accept,dismiss,input  
select---> dowpdown  
multiple windows---> window,tab how handle--->  
\*\*\* DATA Driven FRAMEWORk----> mini---> framework worked frame developed  
1---> mini project/ Python mini /manual s  
database----> 4-5hr  
APIS ---> 2hr  
realtime projects--->3-4hr  
automitive---->  
insurance domain/ossbss/photoshop management/hospital automatic system

**from** selenium **import** webdriver  
**import** chromedriver\_autoinstaller  
chromedriver\_autoinstaller.install()  
driver=webdriver.Chrome()  
driver.get(**'https://www.facebook.com/'**)  
driver.maximize\_window()  
*#print(driver.title)  
#rint(driver.current\_url)*var1=driver.current\_url *#'https://www.facebook.com/***for** i **in** var1:  
 print(i)  
  
**import** sys  
  
sys.exit(0)  
**import** time  
**from** selenium.webdriver.common.by **import** By  
**from** selenium **import** webdriver  
**import** chromedriver\_autoinstaller  
chromedriver\_autoinstaller.install()  
driver=webdriver.Chrome()  
driver.get(**'http://automationpractice.com/index.php'**)  
driver.maximize\_window()  
*#var=driver.find\_elements(By.CLASS\_NAME,'homeslider-container')  
#print(len(var))*var=driver.find\_elements(By.TAG\_NAME,**'a'**)  
  
**for** i **in** var:  
 print(i)  
  
time.sleep(2)  
  
*#find\_element or find\_elements  
#a=10 len(a)  
#a=[1,2,3,4]--->*

*#day4***import** time  
**from** selenium.webdriver.support.wait **import** WebDriverWait  
**from** selenium **import** webdriver  
**from** selenium.webdriver.common.by **import** By  
**import** chromedriver\_autoinstaller  
chromedriver\_autoinstaller.install()  
driver=webdriver.Chrome()  
driver.get(**'https://demo.nopcommerce.com/register'**)  
driver.maximize\_window()  
*#elment=driver.find\_element(By.XPATH,"//input[@id='small-searchterms']")  
#print('display the element',elment.is\_displayed()) #True  
#print('enable the element',elment.is\_enabled()) #True  
#driver.close()*print(**'\*\*No element is selected\*\*'**)  
male\_locator=driver.find\_element(By.XPATH,**"//input[@id='gender-male']"**)  
female\_locator=driver.find\_element(By.XPATH,**"//input[@id='gender-female']"**)  
print(male\_locator.is\_selected()) *#false*print(female\_locator.is\_selected()) *#false*print(**'\*\*male is selected\*\*'**)  
male\_locator.click() *#select*time.sleep(2)  
print(male\_locator.is\_selected()) *#True*print(female\_locator.is\_selected()) *#false*print(**'\*\*female is selected\*\*'**)  
female\_locator.click() *#select*time.sleep(2)  
print(male\_locator.is\_selected()) *#False*print(female\_locator.is\_selected()) *#True*driver.close()

**import** time  
**from** selenium.webdriver.support.wait **import** WebDriverWait  
**from** selenium **import** webdriver *#browser***from** selenium.webdriver.support **import** expected\_conditions **as** EC  
**from** selenium.webdriver.common.by **import** By  
**import** chromedriver\_autoinstaller  
chromedriver\_autoinstaller.install()  
driver=webdriver.Chrome()  
  
driver.get(**'https://www.google.com/'**)  
  
driver.implicitly\_wait(6)  
*#driver.maximize\_window()*w1=WebDriverWait(driver,6)  
  
search=driver.find\_element(By.NAME,**'q'**)  
w1.until(EC.presence\_of\_element\_located(By.NAME,**'q'**))  
  
search.send\_keys(**'hi msg'**)  
search.submit()  
driver.close()

**import** time  
**import** pdb  
**from** selenium **import** webdriver  
**from** selenium.webdriver.common.by **import** By  
**import** chromedriver\_autoinstaller  
chromedriver\_autoinstaller.install()  
driver=webdriver.Chrome()  
driver.get(**'https://itera-qa.azurewebsites.net/home/automation'**)  
driver.maximize\_window()  
*#single checkbox select  
#mon=driver.find\_element(By.XPATH,"//input[@id='monday']")  
#mon.click()  
  
#all the checkbox select*list\_checkbox=driver.find\_elements(By.XPATH,**"//input[@type='checkbox' and contains(@id,'day')]"**)  
*#print(len(list\_checkbox)) 7  
#for i in range(len(list\_checkbox)):# 0 to 6  
 # list\_checkbox[i].click()  
#5,6 ---> click  
#for i in range(len(list\_checkbox)-2,len(list\_checkbox)):# 7-2=5 to 7  
 #list\_checkbox[i].click()  
  
#for i in range(len(list\_checkbox)-4,len(list\_checkbox)-2): #range(3,5) ---->3,4  
 #list\_checkbox[i].click()  
  
#for i in range(0,len(list\_checkbox)-5):  
 #list\_checkbox[i].click()*time.sleep(5)  
*#for i in list\_checkbox:  
 # i.click()*driver.close()

**import** time  
**from** selenium.webdriver.common.by **import** By  
**from** selenium **import** webdriver  
**import** chromedriver\_autoinstaller  
chromedriver\_autoinstaller.install()  
driver=webdriver.Chrome()  
driver.get(**'https://www.google.com/'**) *#facebook*driver.maximize\_window()  
driver.find\_element(By.NAME,**'q'**).send\_keys(**'facebook'**)  
time.sleep(5)  
driver.find\_element(By.NAME,**'q'**).submit() *#enter from keyboard*time.sleep(5)  
  
  
  
  
**import** sys  
  
sys.exit(0)  
**import** time  
**from** selenium **import** webdriver  
**import** chromedriver\_autoinstaller  
chromedriver\_autoinstaller.install()  
driver=webdriver.Chrome()  
driver.get(**'https://www.facebook.com/'**) *#facebook*driver.maximize\_window()  
time.sleep(3)  
driver.get(**'https://www.amazon.in/mobile-phones/b/?ie=UTF8&node=1389401031&ref\_=nav\_cs\_mobiles'**) *#amazon*time.sleep(3)  
driver.back() *#facebook*time.sleep(3)  
driver.forward() *#amazon*time.sleep(3)  
driver.refresh()

**import** sys  
  
  
  
sys.exit(0)  
**import** time  
**from** selenium.webdriver.common.by **import** By  
**from** selenium **import** webdriver  
**import** chromedriver\_autoinstaller  
chromedriver\_autoinstaller.install()  
driver=webdriver.Chrome()  
driver.get(**'https://demo.nopcommerce.com/'**)  
driver.maximize\_window()  
time.sleep(2)  
driver.find\_element(By.LINK\_TEXT,**'Register'**).click()  
time.sleep(5)  
driver.back()  
time.sleep(5)  
driver.forward()  
time.sleep(5)  
driver.refresh()  
  
time.sleep(3)  
driver.close()

**import** sys  
sys.exit(0)  
**from** selenium **import** webdriver  
**import** chromedriver\_autoinstaller  
chromedriver\_autoinstaller.install()  
driver=webdriver.Chrome()  
  
driver.get(**'https://www.facebook.com/'**)  
driver.maximize\_window()  
*#print(driver.title)  
#rint(driver.current\_url)*print(driver.page\_source)  
  
  
  
**import** sys  
  
  
sys.exit(0)  
var1=driver.current\_url *#'https://www.facebook.com/*l1=var1.split(**'.'**)  
print(l1[1])  
  
  
**import** time  
**from** selenium.webdriver.common.by **import** By  
**from** selenium **import** webdriver  
**import** chromedriver\_autoinstaller  
chromedriver\_autoinstaller.install()  
driver=webdriver.Chrome()  
driver.get(**'http://automationpractice.com/index.php'**)  
driver.maximize\_window()  
*#var=driver.find\_elements(By.CLASS\_NAME,'homeslider-container')  
#print(len(var))*var=driver.find\_elements(By.TAG\_NAME,**'a'**)  
l1=var.split()  
print(l1)  
  
time.sleep(2)

**import** time  
**from** selenium **import** webdriver  
**from** selenium.webdriver.common.by **import** By  
**import** chromedriver\_autoinstaller  
chromedriver\_autoinstaller.install()  
driver=webdriver.Chrome()  
**from** selenium.webdriver.support.select **import** Select *#dropdown*driver.get(**'https://www.opencart.com/index.php?route=account/register'**)  
driver.maximize\_window()  
  
drp\_contry=driver.find\_element(By.XPATH,**"//select[@id='input-country']"**)  
*#print(drp\_contry.is\_displayed())*drp=Select(drp\_contry) *#dropdown is activate  
  
#3 methods---->  
#select\_by\_visible\_text(' ')----> visible text on UI  
#select\_by\_index() -----> 0 to len-1 ----> first(0) menu to last menu(len-1)  
#select\_by\_value() -----> value front engineer  
  
#drp.select\_by\_visible\_text('Aruba')#user ko jo visible  
#drp.select\_by\_value("4")  
  
#drp.select\_by\_index(10)*r=drp.options  
*#for i in r:  
 # print(i)  
#print(len(r))*time.sleep(4)  
driver.close()

*'''  
Alerts /PopUp---->  
driver.switch\_to.alert -----> our context we move to alert window  
text ----> text of alert  
accept()----> ok click  
dismiss() ----> cancel click  
  
//button[normalize-space()='Click for JS Prompt']  
  
Authentication alert -----> ???  
user/pass ---> alert???  
  
  
'''***import** time  
**from** selenium **import** webdriver  
**from** selenium.webdriver.common.by **import** By  
**import** chromedriver\_autoinstaller  
chromedriver\_autoinstaller.install()  
driver=webdriver.Chrome()  
  
driver.get(**'https://the-internet.herokuapp.com/javascript\_alerts'**)  
driver.maximize\_window()  
  
*#pop1=driver.find\_element(By.XPATH,"//button[normalize-space()='Click for JS Prompt']")  
#pop1=driver.find\_element(By.XPATH,"//button[normalize-space()='Click for JS Confirm']")*pop1=driver.find\_element(By.XPATH,**"//button[normalize-space()='Click for JS Alert']"**)  
pop1.click()  
time.sleep(5)  
var1=driver.switch\_to.alert  
  
*#var1.send\_keys('Hi i am input box')*var1.accept() *# alert window click on OK buttom  
#var1.dismiss()*time.sleep(5)  
driver.close()

*'''  
https://the-internet.herokuapp.com/basic\_auth  
  
  
authencation---->  
  
https://username:password@url  
  
  
'''***import** time  
**from** selenium **import** webdriver  
**from** selenium.webdriver.common.by **import** By  
**import** chromedriver\_autoinstaller  
chromedriver\_autoinstaller.install()  
driver=webdriver.Chrome()  
  
driver.get(**'https://admin:admin@the-internet.herokuapp.com/basic\_auth'**)  
driver.maximize\_window()  
  
time.sleep(5)  
driver.close()

**import** time  
**from** selenium **import** webdriver  
**from** selenium.webdriver.common.by **import** By  
**import** chromedriver\_autoinstaller  
chromedriver\_autoinstaller.install()  
driver=webdriver.Chrome()  
  
driver.get(**'http://automationpractice.com/index.php'**)  
driver.maximize\_window()  
*#driver.switch\_to.new\_window('tab') #to open new tab---> on same browser*driver.switch\_to.new\_window(**'window'**) *#new browser open for next urls---> different browser*driver.get(**'https://www.selenium.dev/selenium/web/web-form.html'**)  
  
  
time.sleep(5)  
  
driver.close()

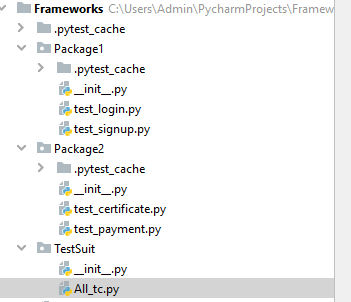
**import** time  
*#ActionChains***from** selenium **import** webdriver  
**from** selenium.webdriver **import** ActionChains  
**from** selenium.webdriver.common.by **import** By  
**import** chromedriver\_autoinstaller  
chromedriver\_autoinstaller.install()  
  
drive=webdriver.Chrome()  
drive.maximize\_window()  
drive.get(**'http://seleniumpractise.blogspot.com/2016/08/how-to-perform-mouse-hover-in-selenium.html'**)  
  
at=drive.find\_element(By.XPATH,**"//button[normalize-space()='Automation Tools']"**)  
act=ActionChains(drive)  
act.move\_to\_element(at).perform()  
time.sleep(3)  
drive.find\_element(By.XPATH,**"//a[text()='TestNG']"**).click()  
time.sleep(3)  
drive.close()  
  
**import** sys  
sys.exit(0)  
**import** time  
**from** selenium **import** webdriver  
**from** selenium.webdriver **import** ActionChains  
**from** selenium.webdriver.common.by **import** By  
**import** chromedriver\_autoinstaller  
chromedriver\_autoinstaller.install()  
  
driver=webdriver.Chrome()  
driver.maximize\_window()  
driver.get(**'http://seleniumpractise.blogspot.com/2016/08/how-to-perform-mouse-hover-in-selenium.html'**)  
time.sleep(2)  
automationTool=driver.find\_element(By.XPATH,**"//button[normalize-space()='Automation Tools']"**)  
act=ActionChains(driver)  
act.move\_to\_element(automationTool).pause(30).click(driver.find\_element(By.XPATH,**"//a[text()='Appium']"**)).perform()  
  
  
**import** sys  
sys.exit(0)  
**import** time  
**from** selenium **import** webdriver  
**from** selenium.webdriver **import** ActionChains  
**from** selenium.webdriver.common.by **import** By  
**import** chromedriver\_autoinstaller  
chromedriver\_autoinstaller.install()  
  
driver=webdriver.Chrome()  
driver.maximize\_window()  
driver.get(**'http://seleniumpractise.blogspot.com/2016/08/how-to-perform-mouse-hover-in-selenium.html'**)  
time.sleep(2)  
automationTool=driver.find\_element(By.XPATH,**"//button[normalize-space()='Automation Tools']"**)  
act=ActionChains(driver)  
act.move\_to\_element(automationTool).perform()  
time.sleep(2)  
driver.find\_element(By.XPATH,**"//button[normalize-space()='Automation Tools']"**).click()  
time.sleep(2)  
print(driver.current\_url)

**''' AUtomation/framweork developement  
How to handle multiple windows---->  
window----> unique id   
  
1. window\_handles ----->  
 all the windows id in 1 list--->[A,B,C,D]  
 list[0]----> parent window ----> automation context  
 list[1:]----> child windows   
   
2. current\_window\_handle ---->  
 unique id of current window----> parent----> l1[0]  
   
3. switch\_to.window(windowID) ---->  
 from one window to next window  
 context will switch from 1 to provide windowID  
  
Programs----->A,B,C,D  
l1[0]--->A  
l1[1:]--->B,C,D  
  
l1=driver.window\_handles  
c\_window\_id=driver.current\_window\_handle(id)  
c\_window\_id=driver.current\_window\_handle(id\_id\_67832D)  
  
l1=[id\_0109A,id\_78978B,id\_67816C,id\_67832D]  
c\_window\_id=id\_id\_67832D  
  
for w in l1:  
 if w==c\_window\_id: #last  
 driver.switch\_to.window(c\_window\_id)  
 driver.close()  
  
  
parent window--->  
child window1---->  
child window 2--->  
child window3---->  
  
context---> is located on parent window  
driver.close()--->single browser close--->  
konsa window close hoga??----> automation through----> webdriver  
  
driver.quite()----->all the open browsers  
konsa all windows close hoga??-----> automation through ----> webdriver  
  
  
Selenium all senois---->  
locatr=driver.find\_element(By.XPATH,'')  
locator.send\_keys('file\_path') # start to end (file name)  
  
  
screen shot--->   
 save\_screenshot()  
 get\_screenshot\_as\_file()  
   
data driven---> openpyxl  
database----> pymysql---> sqlworkbench  
api ----> postman,requests  
framework----> unitest,pytest**

**import** openpyxl  
  
*# we need to install 3rd party library----> python present but we use  
#file---> workbook--->sheets--->rows--->cells(r,c)*file1=**"E:\Book1.xlsx"**workbook=openpyxl.load\_workbook(file1)  
sheet=workbook.active *#parent sheet*row=sheet.max\_row *#5*column=sheet.max\_column *#3***for** r **in** range(2,row+1): *#range(6)---> #0(title) to 5* **for** c **in** range(1,column+1):  
 **if** r==2 **and** c==2:  
 sheet.cell(r,c).value=**'Mumbai'**workbook.save(file1)  
  
  
  
**import** sys  
sys.exit(0)  
*#all row and column se data read***for** r **in** range(2,row+1): *#range(6)---> #0(title) to 5* **for** c **in** range(1,column+1):  
 print(sheet.cell(r,c).value,end=**' '**)  
 print()  
  
  
  
  
**import** sys  
  
sys.exit(0)  
v=sheet.cell(4,2).value *#Mumbai*print(v)  
v=sheet.cell(1,2).value *#city*print(v)  
v=sheet.cell(1,1).value *#name*print(v)  
v=sheet.cell(2,3).value *#89545614*print(v)

*#ddf functions***import** openpyxl  
**def** get\_row\_count(file,sheetname):  
 workbook=openpyxl.load\_workbook(file)  
 sheet=workbook[sheetname]  
 **return** sheet.max\_row  
  
**def** get\_column\_count(file,sheetname):  
 workbook = openpyxl.load\_workbook(file)  
 sheet = workbook[sheetname]  
 **return** sheet.max\_column  
  
**def** read\_Data(file,sheetname,rowno,colno):  
 workbook=openpyxl.load\_workbook(file)  
 sheet=workbook[sheetname]  
 **return** sheet.cell(rowno,colno).value  
  
**def** write\_data(file,sheetname,rowno,colno,data):  
 workbook=openpyxl.load\_workbook(file)  
 sheet=workbook[sheetname]  
 sheet.cell(rowno,colno).value=data  
 workbook.save(file)

**import** time  
**import** function\_for\_ddl  
**from** selenium **import** webdriver  
**import** chromedriver\_autoinstaller  
**from** selenium.webdriver.common.by **import** By  
chromedriver\_autoinstaller.install()  
driver=webdriver.Chrome()  
driver.get(**'https://www.selenium.dev/selenium/web/web-form.html'**)  
driver.maximize\_window()  
  
file1=**'E:\VB.xlsx'**row=function\_for\_ddl.get\_row\_count(file1,**'Sheet1'**)  
  
**for** r **in** range(1,row+1):  
 usrname=function\_for\_ddl.read\_Data(file1,**'Sheet1'**,r,1)  
 password=function\_for\_ddl.read\_Data(file1,**'Sheet1'**,r,2)  
 print(usrname,password)  
 *#inputbox=driver.find\_element(By.XPATH,"//input[@id='my-text-id']")  
 #inputbox.send\_keys(usrname)  
 #time.sleep(5)  
 #passwordbox=driver.find\_element(By.XPATH,"//input[@name='my-password']")  
 #passwordbox.send\_keys(password)  
 #time.sleep(5)  
 #driver.find\_element(By.XPATH,"//button[normalize-space()='Submit']").click()  
 #time.sleep(5)*

2 framework----  
unittest or pytest -----> integration for all framework  
import unittest -------->  
import pytest ------>  
pytest,unitest----> OS----> linux  
\*\*\* working people---->\*\*  
  
\_\_name\_\_ -----> present modulename in python  
name==main ----> current module execute  
other module name==modulename  
  
if \_\_name\_\_ == '\_\_main\_\_' :  
 current module code  
   
\_\_init\_\_.py -----> interpreted--->compiler  
  
unittest ----->inbuild framework  
import unittest  
unitest.TestCase -----> class  
 D/I-----> unittest  
  
test case name unittest-----> aplh  
  
assert statement---->actual expected  
assertequal() ---->pass  
assertnotequal() ----> pass  
  
defect----> developer  
fixtures in unittest and pytest---->  
  
UI testing----->  
setup--->open brower  
testcases  
teardown---->close brower,quite  
  
database testing----->  
setup--->connection to data  
testcases  
teardown---->connection close  
data driver testing/keyword driven----->  
setup--->file-->woorkbook--->sheets--->row--->cell  
testcases ----> samplefile.writeadd()  
teardown---->save()  
  
API testing  
setup---> r=request.get()  
testcases----> v=r.response  
 assertequal(v,200)  
teardown---->  
  
samplefile----> 4 functions  
maxrow,maxcolumn,read,write  
  
  
fixtures----> setup(prerequist) and cleanup activity  
setup(self)  
teardown(self)  
@classmethod  
setUpClass(cls)  
@classmethod  
tearDownClass(cls)  
  
  
skip---->  
@unittest.skip ----> decorator ---> perticular tc will skip  
  
  
  
Ok  
failure  
error  
  
  
  
1 display our result on console we can our result in log file  
2 all the Case will be execute based on Aplhabical  
3 as part of batch(testsuit,job) execution, all the tc from testcase class will be execute  
4 if tc name is same----> python support---> only recent tc  
 method overloading----> in same class method name same  
5 fixture---> 4method---> setup,teardown,setupclass,teardownclass  
6 grouping TC---is not supported in Unittest  
  
Test Suit--->Unittest  
suit ---> collection of all the tc  
Master suit--->(700+)---->all tc type  
stable master suite------> 90 % 10%---> payment,bar,scanner,rect or angular  
 proctator tool----> agular js,react  
  
5 tc---->  
failed at 3??  
  
  


*#test\_signup.py***import** unittest  
**class** SignUpTest(unittest.TestCase):  
 **def** test\_signupbygmail(self):  
 print(**'this tc is for gmail signup-'**)  
 **def** test\_signupbyphone(self):  
 print(**'this tc is for phone signup'**)  
 **def** test\_signupbyfacebook(self):  
 print(**'this tc is for facebook signup'**)  
**if** \_\_name\_\_ == **"\_\_main\_\_"**:  
 unittest.main()

*#test\_login.py***import** unittest  
**class** LoginTest(unittest.TestCase):  
 **def** test\_loginbygmail(self):  
 print(**'this tc is for gmail login'**)  
 **def** test\_loginbyphone(self):  
 print(**'this tc is for phone login'**)  
 **def** test\_loginbyfacebook(self):  
 print(**'this tc is for facebook login'**)  
**if** \_\_name\_\_== **"\_\_main\_\_"**:  
 unittest.main()

*#test\_cerrtificate***import** unittest  
**class** CerficateTest(unittest.TestCase):  
 **def** test\_certificatebyname(self):  
 print(**'this tc is for name of vendor'**)  
 **def** test\_certificatebyformate(self):  
 print(**'this tc is for pdf formate'**)  
  
**if** \_\_name\_\_ == **"\_\_main\_\_"**:  
 unittest.main()

*#test\_payment***import** unittest  
**class** PaymentTest(unittest.TestCase):  
 **def** test\_paymentbydollor(self):  
 print(**'this tc is for payment dollor'**)  
 **def** test\_paymentbyrupees(self):  
 print(**'this tc is for payment by rupees'**)  
  
**if** \_\_name\_\_ == **"\_\_main\_\_"**:  
 unittest.main()

*#test\_suit file***import** unittest  
**from** Package1.test\_login **import** LoginTest  
**from** Package1.test\_signup **import** SignUpTest  
**from** Package2.test\_payment **import** PaymentTest  
**from** Package2.test\_certificate **import** CerficateTest  
  
t1=unittest.TestLoader().loadTestsFromTestCase(LoginTest)  
t2=unittest.TestLoader().loadTestsFromTestCase(SignUpTest)  
t3=unittest.TestLoader().loadTestsFromTestCase(PaymentTest)  
t4=unittest.TestLoader().loadTestsFromTestCase(CerficateTest)  
  
santitysuit=unittest.TestSuite([t1,t2])  
functionalsuit=unittest.TestSuite([t3,t4])  
mastersuit=unittest.TestSuite([t1,t2,t3,t4])  
  
unittest.TextTestRunner().run(santitysuit)