**1. What is range and types of range ?**

The range() Function

To loop through a set of code a specified number of times, we can use the range() function,

The range() function returns a sequence of numbers, starting from 0 by default, and increments by 1 (by default), and ends at a specified number.

The range() function defaults to increment the sequence by 1, however it is possible to specify the increment value by adding a third parameter: range(2, 30, **3**):

for x in range(2, 30, 3):  
  print(x)

**2.Multiple Loops using for loops:**

adj = ["red", "big", "tasty"]

fruits = ["apple", "banana", "cherry"]

for x in adj:

for y in fruits:

print(x, y)

red apple  
red banana  
red cherry  
big apple  
big banana  
big cherry  
tasty apple  
tasty banana  
tasty cherry

3. What are the class and Objects ?

Classes are nothing but user defined blueprint or prototype.

We provide the blueprint or prototype so that others will add the rest of the methods.

Wrapping of or binding the class members and member methods.

4. What are the Instance and Class Variables ?

Instance variables will call using self method.

Class Variables will call using class names.

5. What is static?

Static methods will call using class names.

6.How to declare static methods in Python ?

7. What are the methods ?

Methods are reusable components , perform certain set of operations.

Instance methods.

We call variables in instance using self key word.

Where as static variables we call using using class name,

8.Construtor

We declare constructor using the \_\_int()\_\_ method.

9. How to declare a class ?

Obj = classname()

S1 = classname()

10. super key word 🡪 which is used to call the parent class constructor.

super().\_\_init\_\_(a, b)

or classname.\_\_int\_\_(a,b)

11. How to reverse a string ?

mytxt = **"priya"**mytxt = mytxt[1:3] *# ri*mytxt = mytxt[len(mytxt):0] *# ri*mytxt = **"I wonder how this text looks like backwards"**mytxt = mytxt[::-4]  
print(mytxt)

In this particular example, the slice statement [::-1] means start at the end of the string and end at position 0, move with the step -1, *negative* one, which means one

step backwards.

12. File Handling operations:

**with** open(**'test.txt'**,**'a'**) **as** reader:  
 reader.write(**"This is lovely Priya Bharathi"**)

We are writing here in a code optimized way.

file = open(**"test.txt"**)  
*#print(file.read()) # readlines  
#print(file.readline())  
#print(file.readlines())*content = file.readlines()  
**for** i **in** content:  
 print(i)  
file.close()

**with** open(**'test.txt'**,**'w'**) **as** reader:  
 reader.write(**"This is lovely Priya Bharathi"**)

* w -- writing
* a -- append

12.**Reverse a string.**

**def** my\_function(x):  
 **return** x[::-1]  
  
mytxt = my\_function(**"I wonder how this text looks like backwards"**)  
print(mytxt)  
mytxt = my\_function(**"priya"**)  
print(mytxt)  
mytxt = **"priya"**mytxt = mytxt[1:3] *# ri*mytxt = mytxt[len(mytxt):0] *# ri*mytxt = **"I wonder how this text looks like backwards"**mytxt = mytxt[::-4] .., Last 4 digits   
print(mytxt)

**13. Reverse a list and append the same file.**

ReadLines procedure returns the List , EachLine is one value..

**Step1**: Reverse the List

**Step2**: Using loop , append to the text file.

**with** open(**'test.txt'**,**'r'**) **as** reader:  
 *#print(reader.read())* content1 = list(reversed(reader.readlines()))  
 print(**"{}{}"**.format(**"Reverse of the string:: "**,content1))  
  
 l2 = [**'Priya'**,**'Bharathi'**,**'Karnati'**]  
 content = list(reversed(l2))  
 print(**"{}{}"**.format(**"Reverse of the string:: "**,content))  
  
 **with** open(**'test1.txt'**,**'w'**) **as** writer:  
 **for** i **in** range(0,len(content1)):  
 writer.write(content1[i])

14.How to install pytest procedure in Python ?

pip install pytest

15. Try, Except, Finally.

Even though error comes it doesnot stop the execution. The flow will happen as it is. Finally : Closing the connections .., any clean up activities will happen here.

a= 2  
**try** :  
 **if** 2/0 == 0:  
 print(**"even number"**)  
 **else**:  
 print(**"odd number"**)  
**except** Exception **as** e:  
 print(**"{}{}"**.format(**"Testcase has failed due to the error : "**,e))  
  
Output: Testcase has failed due to the error : division by zero

16. What is pass keyword In Python ?

It skips the execution of the class.

**class** PyTest:  
 **pass**

**17.**

Import loggin

Import inspect

logger.setLevel(logging.DEBUG)

18. Jenkins : **Continuous Integration Server**

Jenkins Download.

Deploy the code \ Opening the Jenkins:

Java -jar -jenkins.war -httpport -9090

**Jenkins** is an open source automation tool written in Java with plugins built for Continuous Integration purpose. **Jenkins** is **used** to build and test your software projects continuously making it easier for developers to integrate changes to the project, and making it easier for users to obtain a fresh build

NewItem.. Executing project code through Jenkins. Build Now..Which browser is executing at run time.

See output in console window.

Jenkins

Parameters:

Py.test –browser\_name “$browsername” /workspace/report/report.html.

20.

@pytest.fixture()  
**def** setup():  
 print(**'hello'**)  
 driver = webdriver.Chrome(**"C:\\crome\chromedriver.exe"**)  
 driver.get(**"https://rahulshettyacademy.com/angularpractice/"**)  
 driver.implicitly\_wait(3)

21.

def test\_loggingDemo():

logger = logging.getLogger(\_\_name\_\_)

fileHandler = logging.FileHandler('logfile.log')

formatter = logging.Formatter("%(asctime)s :%(levelname)s : %(name)s :%(message)s")

fileHandler.setFormatter(formatter)

logger.addHandler(fileHandler) #filehandler object

logger.setLevel(logging.CRITICAL)

logger.debug("A debug statement is executed")

logger.info("Information statement")

logger.debug("A debug statement is executed")

logger.warning("Something is in warning mode")

logger.error("A Major error has happend")

logger.critical("Critical issue")

22. Selenium , it is open source web application tool.

This can be written in any language and in any operating system.

23. Different ways of identifying the locators:

Name

ID

ClassName

CSS Selector

XPath

Text

LinkText

CSS Selector:

Input[attribute\_name=attribute\_value]

Input[attribute\_name\* = attribute\_value]

XPath:

//Input[@attribute\_name=attribute\_value]

//contains(@text()=’Sign in’)

TagName.#ID

TagName.classname

XPath based on text()

//tagname[text() = ‘text’]

//input[text() = ‘abc’]

Parent to Child Transfer mechanism

XPATH : Parent Tag/Child Tag

CSS : Parent Tag Child Tag

Through XPath and CSS Selectors.

Select Parent Locator from Child using XPath:

XPath/parent::tagname.

24. POM:

For each web page in the application, there should be corresponding page class. This Page class will find the WebElements of that web page and also contains Page methods which perform operations on those WebElements.

25.In this way it is code is more readable , maintainable.

**class** HomePage:  
  
 **def** \_\_init\_\_(self, driver):  
 self.driver = driver  
  
 shop = (By.CSS\_SELECTOR, **"a[href\*='shop']"**)

**def** shopItems(self):  
 self.driver.find\_element(\*HomePage.shop).click()  
 checkOutPage = CheckOutPage(self.driver)  
 **return** checkOutPage  
  
**def** getName(self):  
 **return** self.driver.find\_element(\*HomePage.name)

26.

**from** selenium.webdriver.common.by **import** By  
**from** selenium.webdriver.support.wait **import** WebDriverWait  
**from** selenium.webdriver.support **import** expected\_conditions **as** EC  
**from** selenium.webdriver.support.select **import** Select

**def** verifyLinkPresence(self, text):  
 element = WebDriverWait(self.driver, 10).until(  
 EC.presence\_of\_element\_located((By.LINK\_TEXT, text)))  
  
**def** selectOptionByText(self,locator,text):  
 sel = Select(locator)  
 sel.select\_by\_visible\_text(text)

27.

assert count == 3

buttons = driver.find\_elements\_by\_xpath("//div[@class='product-action']/button")

#//div[@class='product-action']/button/parent::div/parent::div/h4

for button in buttons:

list.append(button.find\_element\_by\_xpath("parent::div/parent::div/h4").text)

button.click()

print(list)

17. How to execute the specific class using Pytest ?

C:\Users\kbharathi\Desktop\MDM\Phython\PyTEST\pytests>py.test test\_s.py -v -s

================================================= test session starts =================================================

platform win32 -- Python 3.8.1, pytest-5.4.1, py-1.8.1, pluggy-0.13.1 -- c:\users\kbharathi\appdata\local\programs\python\python38-32\python.exe

cachedir: .pytest\_cache

rootdir: C:\Users\kbharathi\Desktop\MDM\Phython\PyTEST\pytests

collected 3 items

test\_s.py::test\_p1 Program1..

PASSED

test\_s.py::test\_p2 Program2..

PASSED

test\_s.py::test\_p3 Program3..

FAILED

C:\Users\kbharathi\Desktop\MDM\Phython\PyTEST\pytests>py.test -v -s

=========================================================================== test session starts ===========================================================================

platform win32 -- Python 3.8.1, pytest-5.4.1, py-1.8.1, pluggy-0.13.1 -- c:\users\kbharathi\appdata\local\programs\python\python38-32\python.exe

cachedir: .pytest\_cache

rootdir: C:\Users\kbharathi\Desktop\MDM\Phython\PyTEST\pytests

collected 5 items

test\_s.py::test\_p1 Program1..

PASSED

test\_s.py::test\_p2 Program2..

PASSED

test\_s.py::test\_p3 Program3..

FAILED

test\_s1.py::test\_p4 Program4..

PASSED

test\_s1.py::test\_p5 Program5..

PASSED

**18. How to execute the program which has p4 keyword ?**

**C:\Users\kbharathi\Desktop\MDM\Phython\PyTEST\pytests>py.test -k p4 -v -s**

**=========================================================================== test session starts ===========================================================================**

**platform win32 -- Python 3.8.1, pytest-5.4.1, py-1.8.1, pluggy-0.13.1 -- c:\users\kbharathi\appdata\local\programs\python\python38-32\python.exe**

**cachedir: .pytest\_cache**

**rootdir: C:\Users\kbharathi\Desktop\MDM\Phython\PyTEST\pytests**

**collected 5 items / 4 deselected / 1 selected**

**test\_s1.py::test\_p4 Program4..**

**PASSED**

**import** pytest  
  
@pytest.mark.smoke  
**def** test\_p1():  
 print(**"Program1.."**)  
  
**def** test\_p2():  
 print(**"Program2.."**)  
  
**def** test\_p3():  
 print(**"Program3.."**)  
 **assert "Hello"**== **"Hi"**,**"Test failed because values didnt match"**

**import** pytest  
  
**def** test\_p4():  
 print(**"Program4.."**)  
  
@pytest.mark.smoke  
**def** test\_p5():  
 print(**"Program5.."**)

**import** pytest  
  
**def** test\_p4():  
 print(**"Program4.."**)  
  
@pytest.mark.xfail  
**def** test\_p5():  
 print(**"Program5.."**)

**C:\Users\kbharathi\Desktop\MDM\Phython\PyTEST\pytests>py.test test\_s1.py -v -s**

**================================================= test session starts =================================================**

**platform win32 -- Python 3.8.1, pytest-5.4.1, py-1.8.1, pluggy-0.13.1 -- c:\users\kbharathi\appdata\local\programs\python\python38-32\python.exe**

**cachedir: .pytest\_cache**

**rootdir: C:\Users\kbharathi\Desktop\MDM\Phython\PyTEST\pytests**

**collected 2 items**

**test\_s1.py::test\_p4 Program4..**

**PASSED**

**test\_s1.py::test\_p5 Program5..**

**XPASS**

@pytest.mark.skip  
**def** test\_p5():  
 print(**"Program5.."**)

**C:\Users\kbharathi\Desktop\MDM\Phython\PyTEST\pytests>py.test test\_s1.py -v -s**

**================================================= test session starts =================================================**

**platform win32 -- Python 3.8.1, pytest-5.4.1, py-1.8.1, pluggy-0.13.1 -- c:\users\kbharathi\appdata\local\programs\python\python38-32\python.exe**

**cachedir: .pytest\_cache**

**rootdir: C:\Users\kbharathi\Desktop\MDM\Phython\PyTEST\pytests**

**collected 2 items**

**test\_s1.py::test\_p4 Program4..**

**PASSED**

**test\_s1.py::test\_p5 SKIPPED**

**import** pytest  
  
@pytest.fixture()  
**def** setup():  
 print(**"i am executing first.."**)  
 **yield** print(**"i am executing last.."**)  
  
**def** test\_p1(setup):  
 print(**"program1"**)

**19.Display data line by line using readLine method.**

**with** open(**'test.txt'**,**'r'**) **as** reader:  
 **with** open(**'test2.txt'**, **'a'**) **as** writer:  
 line = reader.readline()  
 writer.write(line)  
 **while** line!=**''**:  
 line = reader.readline()  
 writer.write(line)  
 print(line)

20. what is Conftest.py file , Test Fixture in Pytest ?

Common code which applies the all the testcases.

All the prerequisites code will be placed.(It is just like teardown and setup method in py.test)

Common code like setup,tear down.

Fixture return the data in Tuple format and return the data.

Fixture provides the different set of parameters.

@pytest.fixture(scope=**"class"**)

Def setup():

Driver.get(‘’)

Yield:

Driver.close()

21. Data Driven testing ?

Same testcase will be executed multiple times with different set of data.

23.Pip install Pytest-html

pytest --html=report.html

24. CSS Selector : Selenium scans the element from top to left.. whichever comes first it accepts that.

Input[Attribute=value]

CSS Selector , XPath that can be build by us it does not depend on particular attribute.

XPath and CSS can be constructed based on the property..

25. Difference between relative and absolute path ?

Relative ,Absoulte XPath-- > Transfering from the Parent element

26. identifying the static dropdown which has select tag ?

Creating a class

S = select (driver.find\_element\_by\_id(‘id=’))

It applies if we have dropdown with select Tagname

Select\_by\_value 🡪 Value tag in the HTML Property

Select\_by\_Index

Select\_by\_Visble\_text 🡪 Visisble text as per the

Form.

27. What are assertions ?

Assertions “success” in message (or)

Assert “success” == “abc”

Assert 2>3

Assert always expects true.

29. Parent Child Transfer mechanism using XPATH

[\\div[@id=”id1”]\label](file:///\\div[@id=)

[\\div[@id=”id1”]\label[1]\label](file:///\\div[@id=)

XPath identify using text which will be used to identify linkText

//a[@text()==’Forgot Password’]

Or

Driver.find\_element\_by\_linkText(‘Forgot Password’)

Driver.find\_element\_by\_PartiallinkText(‘Forgot Password’)

* Nothing but Regular expression.

CSS Selector ,XPATH 🡪 CSS Selector is bit faster than XPATH

ID -> driver.find\_Elemnt\_by\_id(‘’)

NAME

CLASSNAME

LINKTEXT

XPATH

//input[@id =’id1’]

Regexpath 🡪 //contains[@aler\_success =’altersuccess’]

CSS SELECTOR

Input[id = ‘id1’]

\*Input[id = ‘id1’] 🡪 Doubt

Input#id

Input.TagName

Parent Child mechanism

CSSS,XPath

Click()

.text() 🡪 Extracting text

Send\_keys

.get 🡪 get the URL

.back

.close.quit

Difference between quit(Close all Windows) and close (Close current window)

CheckBox

RadioButton

Java,Javascript

Sleep

Len

Autosuggestive Dropdowns

30.

1. Retrieve the values of one column from excel to dataframe and convert into Excel.

**import** pandas **as** pd  
  
data = pd.read\_excel(**'DataDriven.xlsx'**)  
data = pd.DataFrame(data)  
print(data.columns)

TotalnumberofRows = list(data.shape)  
print(**"total number of records"** + str(TotalnumberofRows[0]))  
i = 0  
print(data[**'SQLQuery'**][1])  
print(TotalnumberofRows[0])  
  
**for** i **in** range(0,TotalnumberofRows[0]):  
 print(data[**'SQLQuery'**][i]) 🡪 We should specify the key value else it will throw Key error.

listofvalues = []  
**for** i **in** range(0,TotalnumberofRows[0]):  
 print(data[**'SQLQuery'**][i])  
 listofvalues.insert(i,data[**'SQLQuery'**][i])  
  
print(listofvalues)

<https://www.techbeamers.com/python-program-convert-lists-dictionary/>

3. concatenate the columns of a data frame.

listofvalues = []  
**for** i **in** range(0,TotalnumberofRows[0]):  
 print(str(data[**'SQLQUERY'**][i])+ **' '**+ str(data[**'SNO'**][i]))  
  
str = (data.apply(**lambda** row: **"'"**+str(row[**"SQLQUERY"**]) + **"','"** + str(row[**"SNO"**])+**"'"**, axis = 1))  
print(str)

4. Program on Parent and Child class.

Parent Class:

**class** employee:  
 empid = **''** empname =**''  
  
 def** \_\_init\_\_(self,empid,empname):  
 employee.empid=empid  
 employee.empname=empname  
 *# print("constructor class")* **def** display(self):  
 print(**"{}{}{}"**.format(**"Employeeid and employee name"**,employee.empid,employee.empname))  
  
*#e = employee('1210','Priya')  
#e.display()*

Child Class:

**import** OOPS.Classprogram  
  
**class** EmployeeAddress(OOPS.Classprogram.employee):  
 address = **"L.B.Nagar"  
  
 def** \_\_init\_\_(self,a,b):  
 super().\_\_init\_\_(a, b)  
 *#print("Child class construcntor")* **def** display(self):  
 print(self.empid,self.empname,self.address)  
  
  
ea = EmployeeAddress(**"1210"**,**"priya"**)  
ea.display()

4.

listofvalues = []  
**for** i **in** range(0,TotalnumberofRows[0]):  
 print(str(data[**'SQLQUERY'**][i])+ **' '**+ str(data[**'SNO'**][i]))  
  
str = (data.apply(**lambda** row: **"'"**+str(row[**"SQLQUERY"**]) + **"','"** + str(row[**"SNO"**])+**"'"**, axis = 1))  
print(str)

5. It display the data for every row and all the columns.   
**for** i,j **in** data.iterrows():  
 print(i,j)

It display the data for every row and for each column separately.   
**for** i,j **in** data.iteritems():  
 print(i,j)

It display the data for every row and and for each column separately.   
  
**for** i,j **in** data.items():  
 print(i,j)