1. What are generators example for generators (print starting date -todays to 5 days)Using  generators concepts

Generators are functions in python but in generator instead of using return keyword we use yield keyword.

Import datetime

def date(n):  
 for i in range(n):  
 y = datetime.date.today()+datetime.timedelta(days=i)

yield y

1. What are decorators example for decorators

Decorators python functions that are used to modify the existing functionality of the function without changing the initial function code.

def printStar(func):  
 def innerFunc():  
 print("\*"\*10)  
 func()  
 return innerFunc  
  
@printStar  
def pri():  
 return "hii"  
  
print(pri())

1. count the number of words present in the paragraph given below

“The horizontal offset of the shadow.    A positive value puts the shadow on the right side of t he box, a.    negative value puts the shadow on the left side of the box "

lis = """" !"#$%&'()\*+,-./:;<=>?@[\]^\_`{|}~"""  
a = "The horizontal offset of the shadow. A positive value puts the shadow on the right side of t he box, a.negative value puts the shadow on the left side of the box"  
  
b = a.split(" ")  
count = 0  
for i in b:  
 if i in lis:  
 continue  
 else:  
 count += 1  
print(count)  
print(b)

1. Write a query to find the max marks from the student table

Select max(marks)

From student

1. Write a query to find the second highest marks from student table

Select max(marks)

From student

Where marks < (Select max(marks)

From student);

1. Li = [0,1,0,1,0,1] Count number of 1 in list without using count () method and if- else?

li = [0,1,0,1,0,1]  
i = 0  
sz = len(li)  
count = 0  
while i < sz:  
 while i<sz and li[i] == 1:  
 count+=1  
 i+=1  
 i += 1  
print(count)

1. [0,1,1,1,1,1,1,0,0,0,0,0] find number of 1 and number of 0 from List without using if.. else, module, library

li = [0,1,1,1,1,1,1,0,0,0,0,0]  
i = 0  
sz = len(li)  
count = 0  
while i < sz:  
 while i<sz and li[i] == 1:  
 count+=1  
 i+=1  
 i += 1  
print("no of 1 = ",count)  
print("no of zero = ",len(li)-count)

1. Li = [4,5,6,2,7,1]

Target = 8

After adding the element, You should Target value

Ans = [6,2 , 7,1]

Li = [4,5,6,2,7,1]  
  
lis = []  
tar = 8  
for i in range(len(Li)):  
 for j in range(len(Li)):  
 if Li[i]!=Li[j] and Li[i]+Li[j]==tar:  
 if Li[j] in lis:  
 continue  
 else:  
 lis.append(Li[i])  
 lis.append(Li[j])  
print(lis)

1. What is architecture of Django?

Django follows MVT architecture where M stands for model V stands for views T stands for template.

Model is a class which contains certain fields which are related to fields in database.

View is function in django which process request and response from web based on the request it gives proper response.

Template is the HTML file which is used to differentiate between python and HTML code and it is used for generating dynamic pages.

1. I have two tables Author, Article and I have the name of Author I have to get all the Articles that are written by that particular Author, how can I do this?

How can I do the same in a single query?

Select article

From article

Where auther\_id = (select id

From author

Where name = “”):

Select article

From article inner join auther on article.author\_id = author.id

Where author\_name = “”:

1. Example code for Python List, dictionary, tuple comprehension.

List comprehension

List1 = [1,2,3,4]

List2 = [X for x in list! If x%2 == 0]

Dictionary comprehension

Dic! = {'a': 1, 'b': 2, 'c': 3, 'd': 4, 'e': 5}

Dic2 = {k:v\*\*2 for k,v in dic1.items()}

Tuple comprehension

Tup1 = (1,2,3,4)

Tup2 = (x for x in tup1 if x%2!=0)

1. Fibonacci with python

l1 = [1, 2]  
  
  
def gen(a, b):  
 for i in range(2, b):  
 total = a[i - 1] + a[i - 2]  
 a.append(total)  
 yield total  
  
  
for g in gen(l1, 10):  
 print(g)  
print(l1)

1. explain decorator with example

def Square(n):

return n\*n

add decorator to this function output should be 2x of Square

def decor(func):  
 def inner(n):  
 return 2\*func(n)  
 return inner  
@decor  
def squ(n):  
 return n\*n  
  
print(squ(4))

1. their are 3 table

1.student

id name

2.Subjevts

subject ,marks

3.StudentScore (in this table we have foregin key only)

find name of topper using sql and orm

select name

from student

where id = ( select student\_id

from studentscore

where subject\_id = (select id

from subjects

where marks = (select max(marks)

from subjects)

)

):

1. TableA has student\_id, student\_name, TableB has student\_id, marks: Find maximum marks, student\_name from the above tables (Note:Foregn key relationship)

Select student , marks

From TableA inner join TableB on

TableA.student\_id = TableB.student\_id

Where marks = (select max(marks)

From TableB):

1. Palindrom for a string and it's substrings

nums = "google"  
lists = []  
palindrom = []  
for i in range(len(nums) + 1):  
 for j in range(i):  
 lists.append(nums[j: i])  
new\_lis = [word[::-1] for word in lists]  
n = len(lists)  
for i in range(n):  
 if lists[i] == new\_lis[i] and lists[i] not in palindrom:  
 palindrom.append(lists[i])  
print(palindrom)

1. change the format of date using decorator

import datetime  
def deco(func):  
 def inner():  
 return func().strftime("%x")  
 return inner  
@deco  
def date():  
 return datetime.date.today()  
  
print(date())

1. implement stack

class Stack:  
 def \_\_init\_\_(self):  
 self.stack = []  
  
 def push(self,item):  
 self.stack.append(item)  
  
 def pop(self):  
 self.stack.pop()  
  
 def isEmpty(self):  
 if len(self.stack) == 0:  
 return "Stack in Empty"  
  
 def displayStack(self):  
 return self.stack  
  
 def isOrdered(self):  
 for i in range(len(self.stack)):  
 swap = False  
 for j in range(0,len(self.stack)-i-1):  
 if self.stack[j]>self.stack[j+1]:  
 self.stack[j],self.stack[j + 1]=self.stack[j+1],self.stack[j]  
 swap = True  
 if swap == False:  
 break  
 return self.stack

1. Implement immutable stack without using any module

class ImutebleStack:  
 def \_\_init\_\_(self):  
 self.stack = ()  
  
 def push(self,item):  
 list(self.stack).append(item)  
  
 def pop(self):  
 list(self.stack).pop()  
  
 def isEmpty(self):  
 if len(self.stack) == 0:  
 return "Stack in Empty"  
  
 def displayStack(self):  
 return self.stack  
  
 def isOrdered(self):  
 a = list(self.stack)  
 for i in range(len(a)):  
 swap = False  
 for j in range(0,len(a)-i-1):  
 if a[j]>a[j+1]:  
 a[j],a[j + 1]=a[j+1],a[j]  
 swap = True  
 if swap == False:  
 break  
 return tuple(a)