

17/1/2020

Write a function to check whether a no. is prime or not.

```
def isprime(a):
    for i in range(2, a/2 + 1):
        if (a % i == 0):
            return False
    return True
```

Lists : - built-in data structure

1. list
2. tuples
3. dictionaries
4. sets.

syntax

list-variable = []

- List elements can be of heterogeneous

eg: list1 = ['physics', 'chemistry', 1997, 2000]

l1 = [1, 2, 3] , l2 = [4, 5]

l3 = l1 + l2

l3 = [1, 2, 3, 4, 5]

l1 = [0] * 5 \Rightarrow l1 = [0, 0, 0, 0, 0]

- negative indexing is possible on lists.

marks = [23, 25, 19, 20, 18]

marks[0] \Leftarrow marks[-5]

marks[4] \Leftarrow marks[-1]

20/7/2020

built in funct's.

- len(list)
- len(max) max(list)
- len min(list)
- sort(reverse = True)
- sorted(l) : will return the sorted list, by does not effecting a original list.

To read n elements and define the function that returns mean of a list

```
def mean(l1):  
    total = 0  
    for i in l1:  
        total += i  
    return total/len(l1)
```

```
n = int(input("Enter the no. of elements:"))  
for lst = [] → print("Enter " + n + " elements")  
for i in range(n):  
    ele  
    lst[i] = int(input())  
    lst.append(ele)  
m = mean(lst)  
print("Mean = ", mean)
```

function that reads a string as a parameter and returns a list containing words that are no longer 4 characters.

```
def prepareList(str) :  
    lst = str.split()  
    list = []  
    for i in lst :  
        if (len(i) <= 4) :  
            list.append(i)  
    return list
```

```
s = input("Enter string")
```

```
t = prepareList(s)  
print(t)
```

function that takes a list as a parameter and return a new list containing elements of the original list with duplicates removed.

```
def removeDuplicate(lst) :  
    lst = []  
    for ele in list :  
        if ele not in lst :  
            lst.append(ele)  
    return lst
```

```
list = [1, 4, 2, 5, 1, 6, 4, 8, 2, 7]
newlist = removeDuplicate(list)
print ("Unique elements are :")
for elem in newlist
    print (elem)

import copy
union = copy.deepcopy(newlist) → to create a new object
of union instead of
referring to the object
newlist.

If it is union = newlist → it performs shallow copy
then if union changes, newlist also changes.
```

24/1/20.

Tuples.

- Tuple is sequence of values much like a list
- Tuple is immutable.

Creating tuple :

tup1 = ()

tup2 = (1, 2, 3, 'physics')

tup3 = (10,)

Accessing tuple :

print(tup1[0])

print(tup2[1:5])

Methods on tuple

count(x), index(x)

Functions :

len(), max(), min(), sorted()

↑
returns list not tuple.

sum(), tuple(), all(), any().

slicing :

s[-1::-1] returns reverse tuple.

To count the elements in a list until an element is a tuple.

list = [4, 6, 2, (8, 3), ...]

for i in list:

if ~~isinstance~~(~~i~~, tuple) != True
break

count = count + 1

~~print(count)~~

```
list = [ ]
```

```
Count = 0
```

```
for elem in list :
```

```
    if isinstance(elem, tuple) :
```

```
        break
```

```
    Count += 1
```

```
print(Count)
```

27/1/2020

Dictionary

- Known as associative array (also known as hash or key-value pair)
- Eg: $d = \{ \underbrace{\text{'Karnataka'}}_{\text{key}} : \text{'Kannada'}, \underbrace{\text{'Maharashtra'}}_{\text{key}} : \text{'Marathi'} \}$

Methods:

clear()

copy() - returns copy

items() - return view of the dictionary items
(key:value)

keys() - " " " " " keys

values() - " " " " " values

WAPP that first creates a telephone directories having phone no and name as key:value. And given a phone no display the corresponding name if exists and appropriate message otherwise.

d = {}

n = int(input('Enter no. of items in dictionary:'))

for i in range(n):

 key = int(input('Enter phone no: '))

 value = input('Enter name: ')

 d[key] = value

K = int(input('Enter phone no. to search: '))

for i in d.keys():

 if(K == i):

 print("Name:", d[i])

 break

 else:

 print('Phone no. does not exists')

if K not in d.keys():

 print('Name:', d[i])

else:

 print('No exists')

popitem() - removes and returns element from dictionary.

pop() - removes and returns element of given

Key

fromkeys() → d.fromkeys(Key, value)

a.update(b)

- Write a program to create a dictionary containing the frequency of occurrence of each character contained in a string.

s = input("Enter string :")

d = {}

lst = []

lst = str.split(" ") X

for i in s :

d[i] = d.get(i, 0) + 1

for key, value in d

print("Key", "appears", value, "times")

31/1/2020

Write a program that reads a file and displays the character count

```
fh = open(" ", "r")
count = 0
for line in fh:
    count += len(line.replace("\n", ""))
print(count)
```

Write a python program that reads a input text from a user and writes it to a file, terminate the program when user enters quit.

```
fh = open("xyz.txt", "w")
```

3/2/2020

write a program to create a file from a list
and copy the file to another file.

```
fh = open("xyz.txt", "w")
lst = ["This is a test\n"]
for ch in lst:
    fh.write(ch + '\n')
fh.close()
fh1 = open("xyz.txt", "r")
fh2 = open("abc.txt", "w")
for line in fh1.readlines():
    fh2.write(line)
fh1.close()
fh2.close()
```

with open(" ", " ") as file:

- block of statements.

WAP that creates a csv file with Itemcode, ItemName, Qty, Price and create a list of items whose qty has fallen below the reorder level, accept the reorder level from the user

```
import csv  
itemlst = [[ "I001", "XYZ", "15", "250"],  
           :  
           :  
           ]
```

```
with open("item.csv", "w", newline="") as fh:  
    w = csv.writer(fh)  
    w.writerows(itemlst)
```

```
lst = []  
reorder = int(input("Enter qty to be reordered:"))  
with open("item.csv", "r", newline="") as fh:  
    r = csv.reader(fh)  
    for row in r:  
        if (row[2] < reorder):  
            lst.append(row[1])  
print(lst)
```

7/2/2020

Exception Handling:

try :

~~else~~ for i in range(1, 2) <
except exception 1 :

except exception 2 :

else :

finally

Write a Python program that reads an arithmetic expression from the user and displays the result. Handle all the possible exceptions that could be raised.

exp = input("Enter exp : ")

try

r = eval(exp)

UNIT - 3

Class Student :

rollno = 100

def __init__(self)

rollno = rollno + 1 → Student.roll

def initialize(self, name, sgpa)

self

self.rollno = rollno

self.name = name

self.sgpa = sgpa.

def display(self)

print(self.rollno, self.name, self.sgpa)

StuList = []

s1 = Student()

s1.initialize("SPD", 9.58)

s1.display()

Define a base class called Shape

Define three derived classes viz.,

- Rectangle.
- Circle and
- Square
- The classes should initialize the instance variables and compute area of the shapes.

2-03-2020

GUI

```
import tkinter as tk
```

```
+ - Layout managers : pack(), grid(), place()
```

```
class
```

```
root = tk.Tk()
```

```
widget
```

```
root.title("My first GUI")
```

```
the
```

```
root.geometry("400x400")
```

doubleVar

```
l1 = tk.Label(root, text = "Name :") intVar
```

```
t1 = tk.Entry(root) — name = tk.StringVar()
```

```
l1.pack()
```

textvariable = name
tkinter variable

```
t1.pack()
```

```
root.mainloop()
```

```
b1 = tk.Button(root, text = "Click Me")
```

```
b1.pack()
```

Command = clicked

```
def clicked():
```

```
    print("Button was clicked")
```

```
    print(name.get())
```

```
from tkinter import messagebox
```

instead of print we

```
messagebox.showinfo("Greeting", "Welcome" +  
name.get())
```

pack(fill=tk.X) →

pack(fill=tk.X, padx=10) →

place(x= , y=)

place(x= , y= , width= , height=)

grid(row= , column=)

```
frame = tk.Frame(root)
import tkinter as tk
root = tk.Tk()
root.title("Simple Calculator")
l1 = tk.Label(root, text="First Number")
num1 = tk.IntVar()
t1 = tk.Entry(root, variable=num1)
l2 = tk.Label(root, text="Second Number")
t2 = tk.Entry(root, variable=num2)
l1.pack() l1.grid(row=0, column=0)
t1.pack() t1.grid(row=0, column=1)
l2.pack() l2.grid(row=1, column=0)
t2.pack() t2.grid(row=1, column=1)
b1 = tk.Button(frame, text="Add", command=add)
b2 = tk.Button(frame, text="Sub", command=sub)
b3 = tk.Button(frame, text="Mul", command=mul)
b4 = tk.Button(frame, text="Div", command=div)
l3 = tk.Label(root, text="Result")
res = tk.IntVar()
l4 = tk.Label(root, textvariable=res, state='disabled')
First No
Second No
Add Sub Mul Div
Result
```

```
def add():
    res.set(num1.get() + num2.get())
```

```
def sub():
    res.set(num1.get() - num2.get())
```

```
def mul():
    res.set(num1.get() * num2.get())
```

```
def div():
    if num2.get() != 0
        res.set(num1.get() / num2.get())
    else
```

```
    messagebox.showerror("ERROR", "Invalid
                           division")
```

6-3-2020.

My GUI

- □ ×

Enter number

o Find factorial o find sum of
natural no.

Click

Display result here.

```
import tkinter as tk  
root = tk.Tk()  
root.title("My GUI")  
l1 = tk.Label(root, text="Enter a number")  
num = tk.IntVar()  
t1 = tk.Entry(root, variable=num)  
→ gen = tk.IntVar()  
r1 = tk.Radiobutton(root, text="Find factorial",  
variable=gen, value=1)  
r2 = tk.Radiobutton(root, text="sum of natural nos.",  
variable=gen, value=2)  
l1.grid(row=0, column=0)  
t1.grid(row=0, column=1)  
r1.grid(row=1, column=0)  
r2.grid(row=1, column=1)  
txt = tk.StringVar()  
l2 = tk.Label(root, textvariable=txt)  
l2.grid(row=3, column=1)  
b1 = tk.Button(root, text="Click", command=click)  
root.mainloop()
```

```
def click():
    if gen.get() == 1
        res = 1
        for (i=2 ; i<=num ; i++)
            get()
            res = res * i
        txt.set("Factorial is " + str(res))
```

```
elif gen.get() == 2
    sum = 0
    for (i=1 ; i<=num.get() ; i++)
        sum = sum + i
    txt.set("sum is " + str(sum))
```

Check Box:

```
t1 = tk.Label(root, text="Preferred languages")
v1 = tk.IntVar()
c1 = tk.Checkbutton(root, text="Java", variable=v1)
v2 = tk.IntVar()
c2 = tk.Checkbutton(root, text="Python", variable=v2)
b1 = tk.Button(root, text="Submit", command=process)
t1.grid(row=0)
c1.grid(row=1)
c2.grid(row=2)
root.mainloop() → b1.grid(row=3)
```

```
def process():
    lang = ''
    if v1.get() == 1:
        lang += "Java\n"
    if v2.get() == 1:
        lang += "Python\n"
    messagebox.showinfo("Submission Data", lang)
```

Application form

Name :

Gender : Male Female

Email ID :

Languages Kan Marathi Hindi English

submit

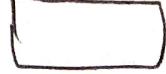
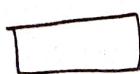
Ch

Select color

Red

Green

Blue



```
l3.config(bg="red")
```

```
c1 = tk.Checkbutton(root, text="red", command=ColorIt)
```

```
def ColorIt():
```

```
    if (v1.get() == 1)
```

```
        l3.config(bg="red")
```

```
else
```

```
    " " (" " = "white")
```