Practical 1

Aim: create national flag using python

Code:

import turtle

from turtle import\*

#screen for output

screen = turtle.Screen()

# Defining a turtle Instance

t = turtle.Turtle()

speed(0)

# initially penup()

t.penup()

t.goto(-400, 250)

t.pendown()

# Orange Rectangle

#white rectangle

t.color("orange")

t.begin\_fill()

t.forward(800)

t.right(90)

t.forward(167)

t.right(90)

t.forward(800)

t.end\_fill()

t.left(90)

t.forward(167)

# Green Rectangle

t.color("green")

t.begin\_fill()

t.forward(167)

t.left(90)

t.forward(800)

t.left(90)

t.forward(167)

t.end\_fill()

# Big Blue Circle

t.penup()

t.goto(70, 0)

t.pendown()

t.color("navy")

t.begin\_fill()

t.circle(70)

t.end\_fill()

# Big White Circle

t.penup()

t.goto(60, 0)

t.pendown()

t.color("white")

t.begin\_fill()

t.circle(60)

t.end\_fill()

# Mini Blue Circles

t.penup()

t.goto(-57, -8)

t.pendown()

t.color("navy")

for i in range(24):

    t.begin\_fill()

    t.circle(3)

    t.end\_fill()

    t.penup()

    t.forward(15)

    t.right(15)

    t.pendown()

# Small Blue Circle

t.penup()

t.goto(20, 0)

t.pendown()

t.begin\_fill()

t.circle(20)

t.end\_fill()

# Spokes

t.penup()

t.goto(0, 0)

t.pendown()

t.pensize(2)

for i in range(24):

    t.forward(60)

    t.backward(60)

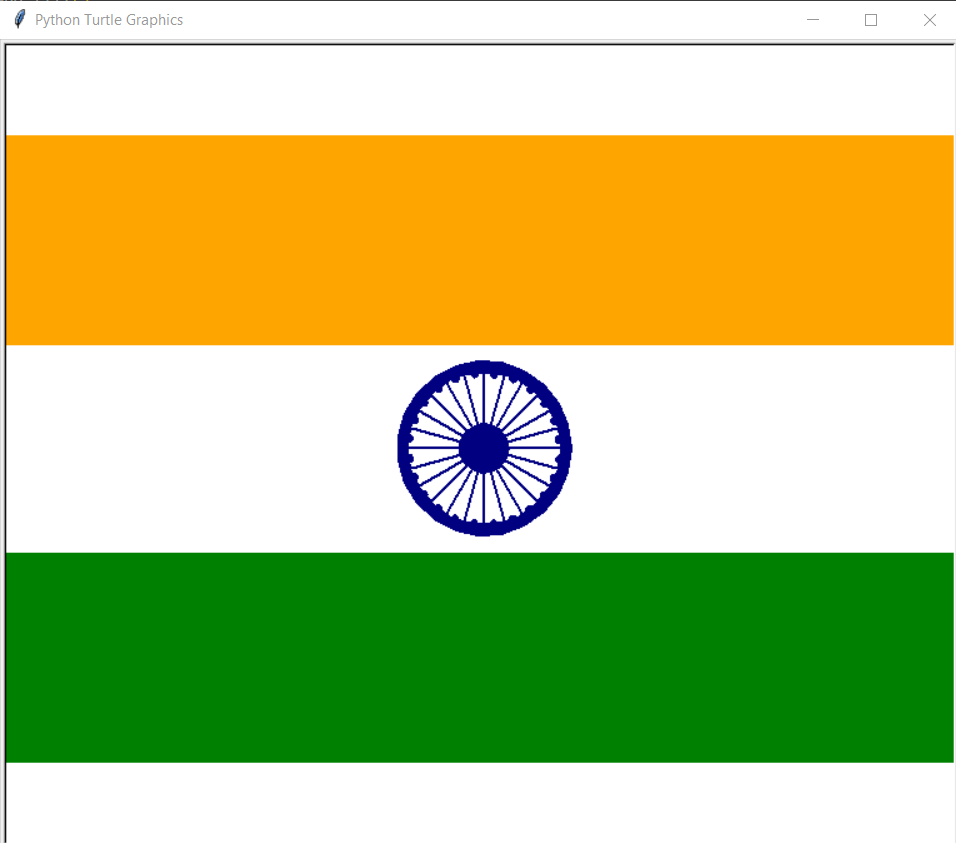
    t.left(15)

#to hold the

#output window

turtle.done()

Output:



Practical 2

Aim: create Instagram logo using python

Code:

from turtle import \* #importing the module

def backFrame():

    COLOR = (0.60156, 0, 0.99218)  # (154, 0, 254)

    TARGET = (0.86328, 0.47656, 0.31250)  # (221, 122, 80)

    screen = Screen()

    screen.tracer(False)

    WIDTH, HEIGHT = screen.window\_width(), screen.window\_height() #defining some useful variables

    deltas = [(hue - COLOR[index]) / HEIGHT for index, hue in enumerate(TARGET)]

    turtle = Turtle()

    turtle.color(COLOR)

    turtle.penup()

    turtle.goto(-WIDTH/2, HEIGHT/2)

    turtle.pendown()

    direction = 1

    #the gradient background

    for distance, y in enumerate(range(HEIGHT//2, -HEIGHT//2, -1)):

        turtle.forward(WIDTH \* direction)

        turtle.color([COLOR[i] + delta \* distance for i, delta in enumerate(deltas)])

        turtle.sety(y)

        direction \*= -1

    screen.tracer(True)

def main():

    ## setting all the pre-defined values to be used in the program

    pen\_color = 'white'

    width\_val = 23

    round\_coordA, round\_coordB = 34, 90

    circleAx, circleAy = 80, 360

    circleBx, circleBy = 7, 360

    gotoAx, gotoAy = 85, 30

    gotoBx, gotoBy = 160, -100

    pencolor(pen\_color) #defining the color of the pen

    width(width\_val) #defining the thickness of the pen

    penup() #start the draw

    goto(gotoBx, gotoBy) #set the origin for the graphics

    pendown() #stop the draw temporarily

    left(90) #turn at an angle

    for i in range(4): #loop for 4 times for a square-type shape

        forward(250)

        circle(round\_coordA, round\_coordB) #for border-radius

    penup() # for the big circle

    goto(gotoAx, gotoAy) # defining the new origin for the pen

    pendown() # stopping the draw

    circle(circleAx, circleAy) #built-in function for a circle in the middle

    penup()

    goto(110,130) # setting the new origin

    pendown()

    circle(circleBx, circleBy) # built-in function for a circle at the top-right

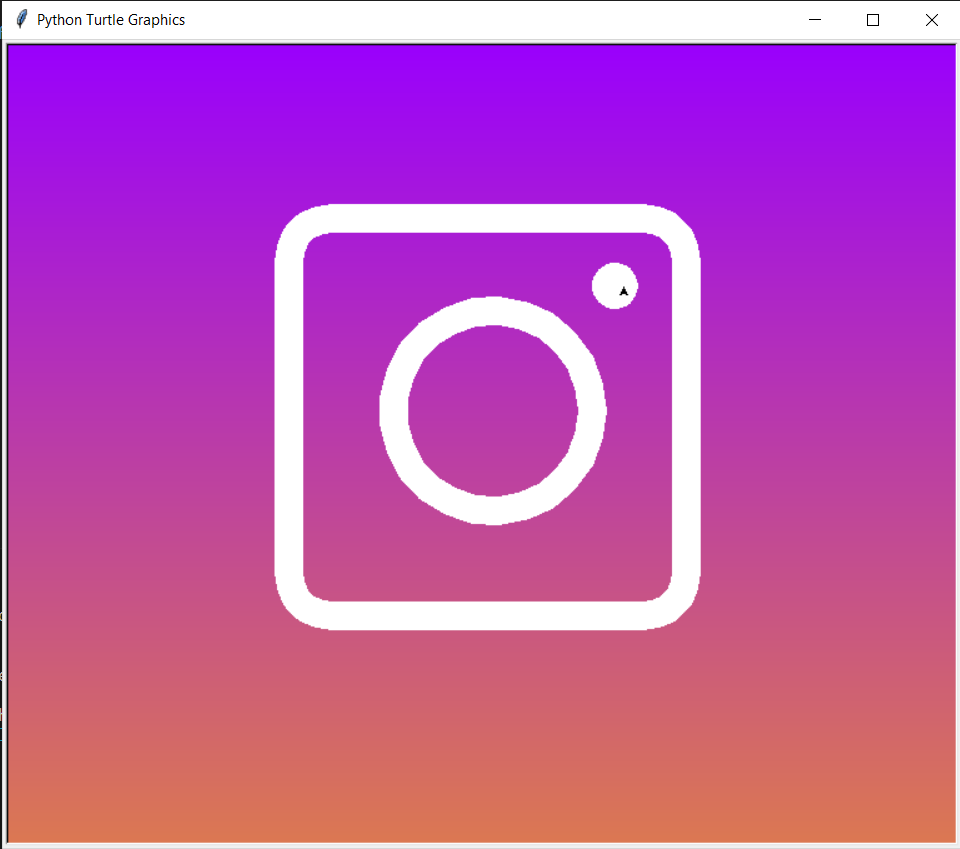
    done() # end the program

if \_\_name\_\_ == "\_\_main\_\_":

    backFrame()

    main()

Output:



Practical 3

Aim:plotting a graph using python

Code:

# importing the required module

>>> import matplotlib.pyplot as plt

>>>

>>> # x axis values

>>> x = [1,2,3]

>>> # corresponding y axis values

>>> y = [2,4,1]

>>>

>>> # plotting the points

>>> plt.plot(x, y)

[<matplotlib.lines.Line2D object at 0x7f4e064c6910>]

>>>

>>> # naming the x axis

>>> plt.xlabel('x - axis')

Text(0.5, 0, 'x - axis')

>>> # naming the y axis

>>> plt.ylabel('y - axis')

Text(0, 0.5, 'y - axis')

>>>

>>> # giving a title to my graph

>>> plt.title('My first graph!')

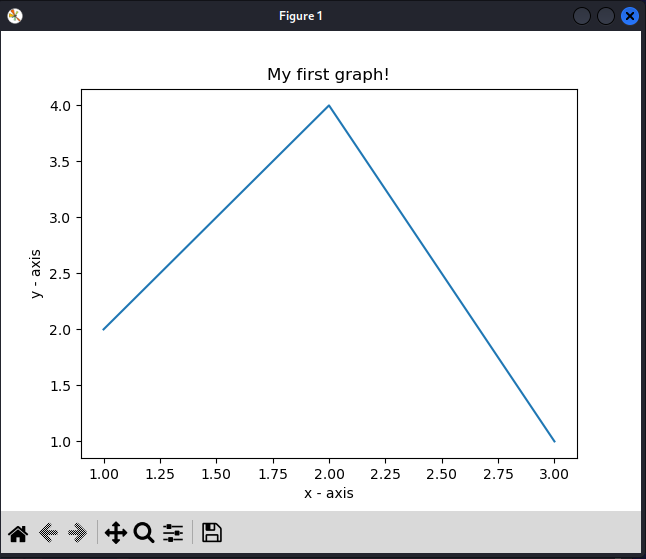
Text(0.5, 1.0, 'My first graph!')

>>>

>>> # function to show the plot

>>> plt.show()

Output:



1. bar graph

Code:

import matplotlib.pyplot as plt

>>>

>>> # x-coordinates of left sides of bars

>>> left = [1, 2, 3, 4, 5]

>>>

>>> # heights of bars

>>> height = [10, 24, 36, 40, 5]

>>>

>>> # labels for bars

>>> tick\_label = ['one', 'two', 'three', 'four', 'five']

>>>

>>> # plotting a bar chart

>>> plt.bar(left, height, tick\_label = tick\_label,

... width = 0.8, color = ['red', 'green'])

<BarContainer object of 5 artists>

>>>

>>> # naming the x-axis

>>> plt.xlabel('x - axis')

Text(0.5, 0, 'x - axis')

>>> # naming the y-axis

>>> plt.ylabel('y - axis')

Text(0, 0.5, 'y - axis')

>>> # plot title

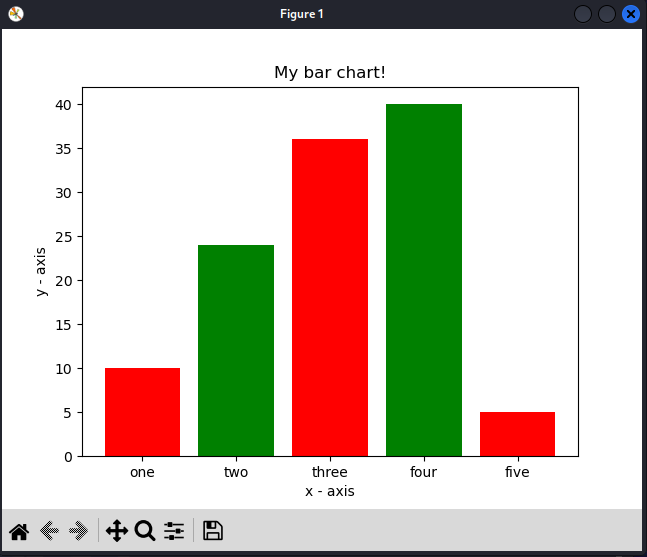
>>> plt.title('My bar chart!')

Text(0.5, 1.0, 'My bar chart!')

>>>

>>> # function to show the plot

>>> plt.show()



**Practical 4**

**Aim:** make a calculator using python

Code:

# Python program for simple calculator

# Function to add two numbers

def add(num1, num2):

    return num1 + num2

# Function to subtract two numbers

def subtract(num1, num2):

    return num1 - num2

# Function to multiply two numbers

def multiply(num1, num2):

    return num1 \* num2

# Function to divide two numbers

def divide(num1, num2):

    return num1 / num2

print("Please select operation -\n" \

        "1. Add\n" \

        "2. Subtract\n" \

        "3. Multiply\n" \

        "4. Divide\n")

# Take input from the user

select = int(input("Select operations form 1, 2, 3, 4 :"))

number\_1 = int(input("Enter first number: "))

number\_2 = int(input("Enter second number: "))

if select == 1:

    print(number\_1, "+", number\_2, "=",

                    add(number\_1, number\_2))

elif select == 2:

    print(number\_1, "-", number\_2, "=",

                    subtract(number\_1, number\_2))

elif select == 3:

    print(number\_1, "\*", number\_2, "=",

                    multiply(number\_1, number\_2))

elif select == 4:

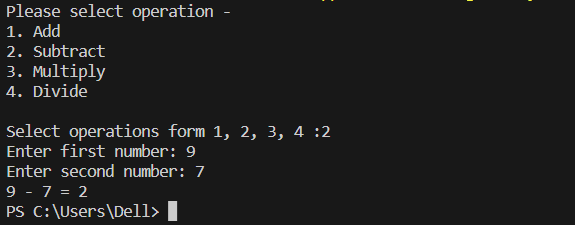
    print(number\_1, "/", number\_2, "=",

                    divide(number\_1, number\_2))

else:

    print("Invalid input")

Output:



Gui calculator using python

Code:

# pip install tkinter

import tkinter as tk

import tkinter.messagebox

from tkinter.constants import SUNKEN

window = tk.Tk()

window.title('Calculator-GeeksForGeeks')

frame = tk.Frame(master=window, bg="skyblue", padx=10)

frame.pack()

entry = tk.Entry(master=frame, relief=SUNKEN, borderwidth=3, width=30)

entry.grid(row=0, column=0, columnspan=3, ipady=2, pady=2)

def myclick(number):

    entry.insert(tk.END, number)

def equal():

    try:

        y = str(eval(entry.get()))

        entry.delete(0, tk.END)

        entry.insert(0, y)

    except:

        tkinter.messagebox.showinfo("Error", "Syntax Error")

def clear():

    entry.delete(0, tk.END)

button\_1 = tk.Button(master=frame, text='1', padx=15,

                    pady=5, width=3, command=lambda: myclick(1))

button\_1.grid(row=1, column=0, pady=2)

button\_2 = tk.Button(master=frame, text='2', padx=15,

                    pady=5, width=3, command=lambda: myclick(2))

button\_2.grid(row=1, column=1, pady=2)

button\_3 = tk.Button(master=frame, text='3', padx=15,

                    pady=5, width=3, command=lambda: myclick(3))

button\_3.grid(row=1, column=2, pady=2)

button\_4 = tk.Button(master=frame, text='4', padx=15,

                    pady=5, width=3, command=lambda: myclick(4))

button\_4.grid(row=2, column=0, pady=2)

button\_5 = tk.Button(master=frame, text='5', padx=15,

                    pady=5, width=3, command=lambda: myclick(5))

button\_5.grid(row=2, column=1, pady=2)

button\_6 = tk.Button(master=frame, text='6', padx=15,

                    pady=5, width=3, command=lambda: myclick(6))

button\_6.grid(row=2, column=2, pady=2)

button\_7 = tk.Button(master=frame, text='7', padx=15,

                    pady=5, width=3, command=lambda: myclick(7))

button\_7.grid(row=3, column=0, pady=2)

button\_8 = tk.Button(master=frame, text='8', padx=15,

                    pady=5, width=3, command=lambda: myclick(8))

button\_8.grid(row=3, column=1, pady=2)

button\_9 = tk.Button(master=frame, text='9', padx=15,

                    pady=5, width=3, command=lambda: myclick(9))

button\_9.grid(row=3, column=2, pady=2)

button\_0 = tk.Button(master=frame, text='0', padx=15,

                    pady=5, width=3, command=lambda: myclick(0))

button\_0.grid(row=4, column=1, pady=2)

button\_add = tk.Button(master=frame, text="+", padx=15,

                    pady=5, width=3, command=lambda: myclick('+'))

button\_add.grid(row=5, column=0, pady=2)

button\_subtract = tk.Button(

    master=frame, text="-", padx=15, pady=5, width=3, command=lambda: myclick('-'))

button\_subtract.grid(row=5, column=1, pady=2)

button\_multiply = tk.Button(

    master=frame, text="\*", padx=15, pady=5, width=3, command=lambda: myclick('\*'))

button\_multiply.grid(row=5, column=2, pady=2)

button\_div = tk.Button(master=frame, text="/", padx=15,

                    pady=5, width=3, command=lambda: myclick('/'))

button\_div.grid(row=6, column=0, pady=2)

button\_clear = tk.Button(master=frame, text="clear",

                        padx=15, pady=5, width=12, command=clear)

button\_clear.grid(row=6, column=1, columnspan=2, pady=2)

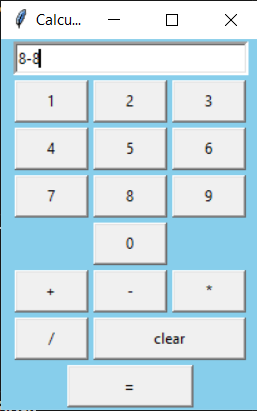
button\_equal = tk.Button(master=frame, text="=", padx=15,

                        pady=5, width=9, command=equal)

button\_equal.grid(row=7, column=0, columnspan=3, pady=2)

window.mainloop()

Output:



**Practical 5**

**Aim:** create a countdown timer using python  
**code:**

# import the time module

import time

# define the countdown func.

def countdown(t):

    while t:

        mins, secs = divmod(t, 60)

        timer = '{:02d}:{:02d}'.format(mins, secs)

        print(timer, end="\r")

        time.sleep(1)

        t -= 1

    print('Fire in the hole!!')

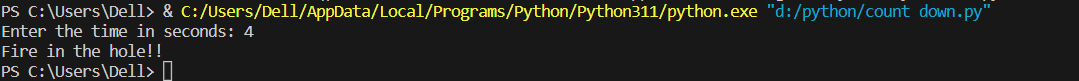
# input time in seconds

t = input("Enter the time in seconds: ")

# function call

countdown(int(t))

Output:



**Practical 6**

**Aim:** create a digital clock using python

**Code:**

# importing whole module

from tkinter import \*

from tkinter.ttk import \*

# importing strftime function to

# retrieve system's time

from time import strftime

# creating tkinter window

root = Tk()

root.title('Clock')

# This function is used to

# display time on the label

def time():

    string = strftime('%H:%M:%S %p')

    lbl.config(text=string)

    lbl.after(1000, time)

# Styling the label widget so that clock

# will look more attractive

lbl = Label(root, font=('calibri', 40, 'bold'),

            background='purple',

            foreground='white')

# Placing clock at the centre

# of the tkinter window

lbl.pack(anchor='center')

time()

mainloop()

**Output:**

