

## -:Python Alarm Clock Project :-

Step by step :

1]install required models

- 1)pip install tk                      from tkinter import \*
- 2)pip install datetime              import datetime
- 3)pip install python-time           import time
- 4)winsound (it's build in module) import winsound

2]Create a while loop :

```
def alarm(set_alarm_timer):  
    while True:  
        time.sleep(1)  
        current_time = datetime.datetime.now()  
        now = current_time.strftime("%H:%M:%S")  
        date = current_time.strftime("%d/%m/%Y")  
        print("The Set Date is:",date)  
        print(now)  
        if now == set_alarm_timer:  
            print("Time to Wake up")  
            winsound.PlaySound("sound.wav",winsound.SND_ASYNC)  
            break  
  
def actual_time():  
    set_alarm_timer = f"{hour.get()}:{min.get()}:{sec.get()}"  
    alarm(set_alarm_timer)
```

Explanation:

Define a function named as alarm() which takes the argument of (set\_alarm\_timer).It contains a while loop with a Boolean function True which makes the program automatic to work.

time.sleep(1) halts the execution of the further commands given until we get the time value from the user later in the code and returns the background thread of the clock time going on at a regular interval.

Get the current time using current\_time which takes the argument of datetime.datetime.now().

now is used to print the time and date is used to print the current date by string conversion using strftime()

Define another function here named actual\_time() which takes in the user value for setting the alarm in the string format. The same argument of (set\_alarm\_timer) as alarm before to execute the while loop which we further use while making GUI.

If loop suggests that if the user input time set\_alarm\_timer matches with the while loop ongoing time now, the message is printed as "Time to Wake up".

winsound.SND\_ASYNC plays the system generated sound as soon the condition satisfies, acting as a reminder for the alarm clock.

4]Create GUI (Graphical user interface):

```
clock = Tk()
```

```
clock.title("DataFlair Alarm Clock")
```

```
clock.geometry("400x200")
```

```
time_format=Label(clock, text= "Enter time in 24 hour format!", fg="red",bg="black",font="Arial").place
```

```

e (x=60,y=120)
addTime = Label(clock,text = "Hour  Min  Sec",font=60).place(x = 110)
setYourAlarm = Label(clock,text = "When to wake you up",fg="blue",relief = "solid",font= ("Helvetica",7,"
bold")).place(x=0, y=29)

# The Variables we require to set the alarm(initialization):
hour = StringVar()
min = StringVar()
sec = StringVar()

#Time required to set the alarm clock:
hourTime= Entry(clock,textvariable = hour,bg = "pink",width = 15).place(x=110,y=30)
minTime= Entry(clock,textvariable = min,bg = "pink",width = 15).place(x=150,y=30)
secTime = Entry(clock,textvariable = sec,bg = "pink",width = 15).place(x=200,y=30)

#To take the time input by user:
submit = Button(clock,text = "Set Alarm",fg="red",width = 10,command = actual_time).place(x =110,y=70)

clock.mainloop()
#Execution of the window.

```

### Explanation:

To Initialize tkinter, we pass a command under the name clock as Tk().

The dialog box has the title as DataFlair Alarm Clock with a geometry of (400x200). We pass on the heading to mention the time format for 24 hours using time\_format.

The second heading is given above the user input boxes for the labeling to be “Hour Min Sec” using addTime.

Just to make the dialog box look funkier, adding another label as “when to wake you up” using setYourAlarm.

As we have already converted the current time in the string before (actual time), the variables we initialize for the user input dialog boxes are in StringVar().

Finally make the input boxes such as hourTime, minTime, and secTime which takes the entry of the time the user wants to set the alarm on in 24-hour format.

Submit takes the command of the defined function actual\_time and executes the clock as it acts as a set button to start the program.

Clock.mainloop() is the basic and the last command was given to compile all the previous commands with their basic settings of color, font, width, axis, etc. and displays the window as soon as the program is run.