BASAVESHWAR ENGINEERING College (AUTONOMOUS) BAGALKOt



DEPARTMENT OF ELECTRONICS AND COMMUNICATION

REPORT

Stopwatch using python

Subject: Higher Programming Paradigm

Subject code : 21UEC308C

DIV: B

Presented By :

|  |  |  |
| --- | --- | --- |
| Name | USN | Roll no. |
| Pavan C Kumatole | 2BA21EC052 | 21 |
| Preetam B Kanal | 2BA21EC067 | 28 |

Faculty I/C: Dr. M.C.Aralimarad

**Create a stopwatch**

**Introduction:**

A stopwatch is a handheld timepiece designed to measure the amount of time elapsed from a particular time when it is activated to the time when the piece is deactivated. A large digital version of a stopwatch designed for viewing at a distance, as in a sports stadium, is called a stop clock. In manual timing, the clock is started and stopped by a person pressing a button. In fully automatic time, both starting and stopping are triggered automatically, by sensors. Now lets try to create a program using Tkinter module to create a stopwatch.

**Description:**A Stopwatch is created using tkinter and datetime libraries in this project. A basic GUI of stopwatch with start-stop and reset functions.  
  
**Libraries used in this project are:**  
 Installation of tkinter-(pip install python-tk)  
 Installation of datetime-(pip install DateTime)

**Tkinter :**Tkinter is the standard GUI library for Python. Python when combined with Tkinter provides a fast and easy way to create GUI applications. Tkinter provides a powerful object-oriented interface to the Tk GUI toolkit. It’s very easy to get started with Tkinter, here are some sample codes to get your hands on Tkinter in python.

**Explanation**

Libraries needed for this script are tkinter for the GUI and datetime for calculating the time.After executing the script a GUI with Start Stop and Reset will be displayed.

Start- By pressing on start stopwatch gets activated and the time runs from 00:00:00 hours minutes and seconds respectively. If there is a timer running then the start button will be inactive state.

Stop-To make the stop button activate the timer must be active i.e., we can access the stop button only after the start is initiated after the timer is running the stop button is used to stop or pause the timer.

Reset-It is used to reset the timer after it is started. It gets activated only if there is any activity or change in time which clarifies we can only reset the timer if there is a start initiated.

**Source Code :**

# Python program to illustrate a stop watch

# using Tkinter

#importing the required libraries

import tkinter as Tkinter

from datetime import datetime

counter = 66600

running = False

def counter\_label(label):

def count():

if running:

global counter

# To manage the initial delay.

if counter==66600:

display="Starting..."

else:

tt = datetime.fromtimestamp(counter)

string = tt.strftime("%H:%M:%S")

display=string

label['text']=display # Or label.config(text=display)

# label.after(arg1, arg2) delays by

# first argument given in milliseconds

# and then calls the function given as second argument.

# Generally like here we need to call the

# function in which it is present repeatedly.

# Delays by 1000ms=1 seconds and call count again.

label.after(1000, count)

counter += 1

# Triggering the start of the counter.

count()

# start function of the stopwatch

def Start(label):

global running

running=True

counter\_label(label)

start['state']='disabled'

stop['state']='normal'

reset['state']='normal'

# Stop function of the stopwatch

def Stop():

global running

start['state']='normal'

stop['state']='disabled'

reset['state']='normal'

running = False

# Reset function of the stopwatch

def Reset(label):

global counter

counter=66600

# If rest is pressed after pressing stop.

if running==False:

reset['state']='disabled'

label['text']='Welcome!'

# If reset is pressed while the stopwatch is running.

else:

label['text']='Starting...'

root = Tkinter.Tk()

root.title("Stopwatch")

# Fixing the window size.

root.minsize(width=250, height=70)

label = Tkinter.Label(root, text="Welcome!", fg="black", font="Verdana 30 bold")

label.pack()

f = Tkinter.Frame(root)

start = Tkinter.Button(f, text='Start', width=6, command=lambda:Start(label))

stop = Tkinter.Button(f, text='Stop',width=6,state='disabled', command=Stop)

reset = Tkinter.Button(f, text='Reset',width=6, state='disabled', command=lambda:Reset(label))

f.pack(anchor = 'center',pady=5)

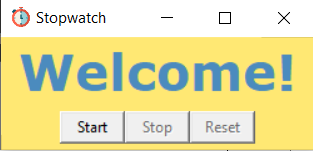
start.pack(side="left")

stop.pack(side ="left")

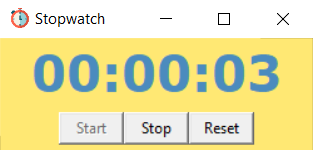
reset.pack(side="left")

root.mainloop()

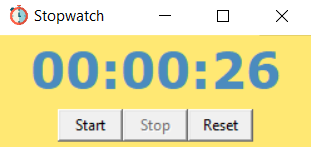
**Output:**



-After executing the script a stopwatch window will be opened.



-Stopwatch is activated by pressing Start.



-Stopwatch is paused by pressing the stop.



-Stopwatch will be reset by pressing the reset.

**Conclusion:**

So this is how we can write a program to create a stopwatch.A large digital version of a stopwatch designed for viewing at a distance, as in a sports stadium. The device is used when time period need to be measured precisely and with minimum complication.