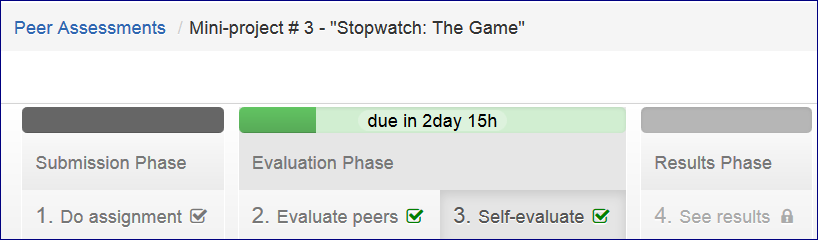
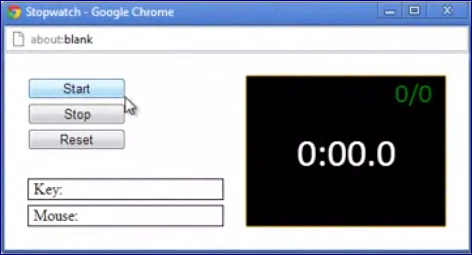
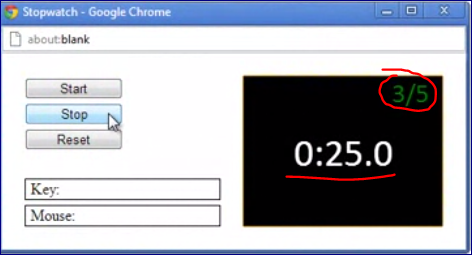
**MiniProject3\_StopWatch**

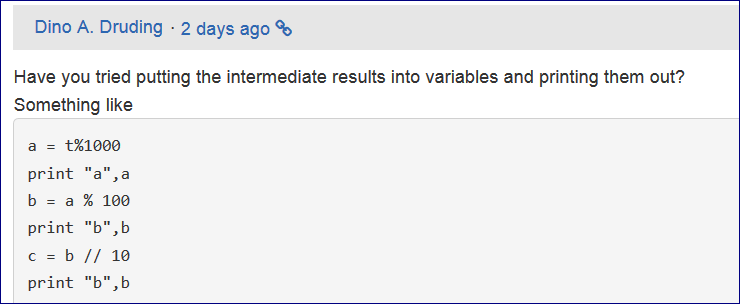


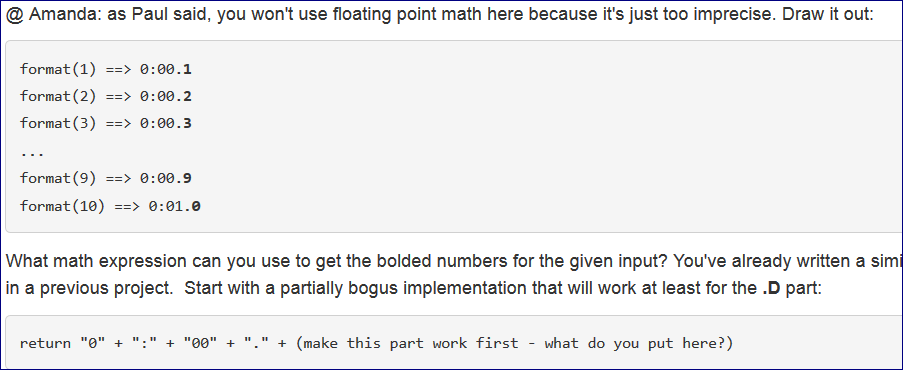
<http://www.codeskulptor.org/#user12_qZGLKSAnbP_0.py>

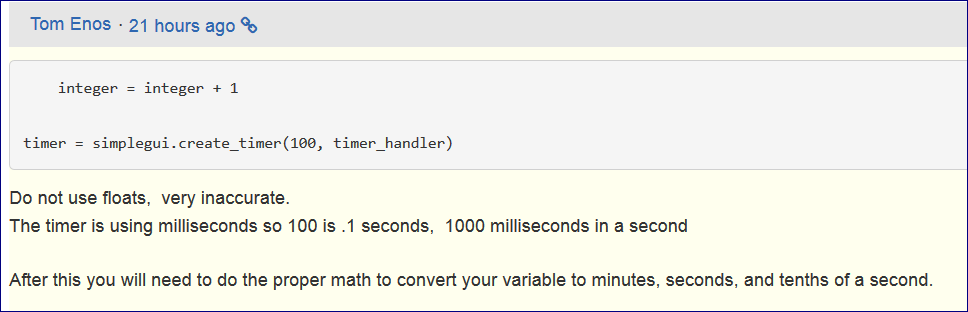




<https://class.coursera.org/interactivepython-002/forum/thread?thread_id=2731&post_id=13915#post-13915>







**Test the format function:**

<http://www.codeskulptor.org/#user12_CnA1Sr3Bi8_0.py>

+===============================================

Printing to the console milliseconds

import simplegui

import time

curTime = 0

def tick():

global curTime

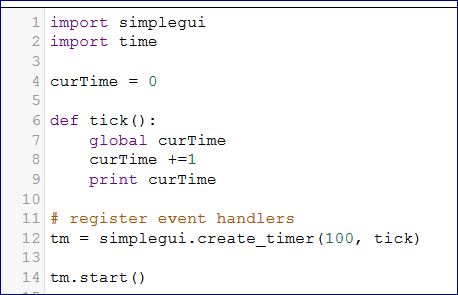
curTime +=1

print curTime

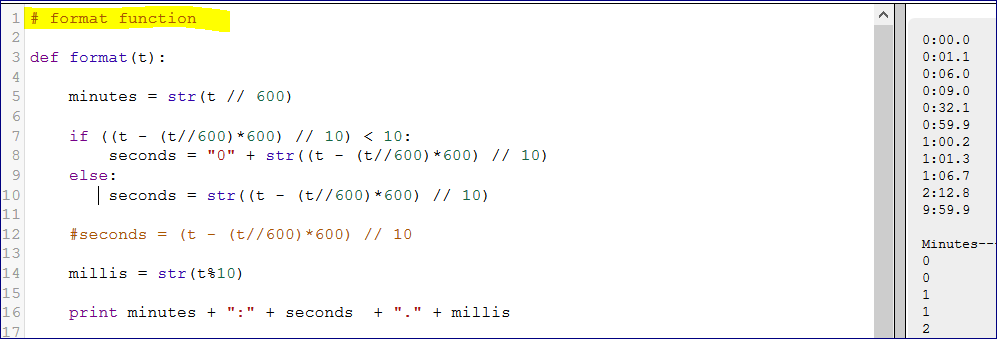
# register event handlers

tm = simplegui.create\_timer(100, tick)

tm.start()



def format(t):  
    minutes = t // 600  
    seconds = (t / 10) // 10  
    seconds\_c = (t / 10) % 10  
    tenths = (t / 10) // 10  
    return str(minutes) + ":" + str(seconds) + str(seconds\_c) + "." + str(tenths)



--=====================================================

Final Version

<http://www.codeskulptor.org/#user12_rkmq2xi8xX_3.py>

# template for "Stopwatch: The Game"

import simplegui

import time

# define global variables

curTime = 0

attempts = 0

success = 0

stopped = 0

# define helper function format that converts time

# in tenths of seconds into formatted string A:BC.D

def format(t):

minutes = str(t // 600)

if ((t - (t//600)\*600) // 10) < 10:

seconds = "0" + str((t - (t//600)\*600) // 10)

else:

seconds = str((t - (t//600)\*600) // 10)

millis = str(t%10)

conv = minutes + ":" + seconds + "." + millis

return conv

# define event handlers for buttons; "Start", "Stop", "Reset"

def startBtn():

global curTime, attempts, success, stopped

stopped = 1

tm.start()

def stopBtn():

tm.stop()

global curTime, attempts, success, stopped

if stopped !=0:

attempts +=1

#print format(curTime)

if curTime%10 == 0 and stopped != 0:

success +=1

stopped = 0

def resetBtn():

tm.stop()

global curTime, attempts, success

curTime = 0

attempts = 0

success = 0

# define event handler for timer with 0.1 sec interval

def tick():

global curTime, stopped

if stopped != 0:

curTime +=1

# define draw handler

def drawTime(canvas):

global curTime

canvas.draw\_text(format(curTime), (80, 120), 60, "White")

canvas.draw\_text(str(success) + "/" + str(attempts), (250, 30), 30, "Red")

# create frame

f = simplegui.create\_frame("Stopwatch", 300, 200)

# register event handlers

tm = simplegui.create\_timer(100, tick)

f.set\_draw\_handler(drawTime)

start = f.add\_button("Start", startBtn, 150)

stop = f.add\_button("Stop", stopBtn, 150)

reset = f.add\_button("Reset", resetBtn, 150)

# start frame

f.start()

tm.start()