```python
# -*- coding: utf-8 -*-
"""
Created on Mon Oct  1 00:42:25 2018

@author: maxhu
"""

import matplotlib.pyplot as plt
import numpy as np

# constants
L = 1.0
g = 9.8
theta_min = 5*np.pi/180
theta_max = np.pi/2
omega_0 = 0
t_0 = 0
delta_t = 10**(-3)

# lists to store data
thetas = np.linspace(theta_min, theta_max, 100)
T_values = []
T2_values = []


# nested loop
# for loop handles different lengths of pendulum
for n in range(len(thetas)):
    # set initial conditions for looping calculations
    flag = True
    t_i = t_0
    omega_i = omega_0
    theta_i = thetas[n]
    counter = 0

    # while loop handles omega and theta values
    while flag == True:
        omega_f = omega_i - (g/L) *np.sin(theta_i) * delta_t
        theta_f = theta_i + omega_f*delta_t
        t_f = t_i + delta_t

        omega_i = omega_f
        theta_i = theta_f
        t_i = t_f

        # when your current position is really close to the starting position
        # exit the while loop becuase you've gone a full period
        if (np.abs(theta_i - thetas[n]) < 10**(-3) and counter >= 50):
            flag = False

        counter = counter + 1


    T_values.append(t_i)
    T2_values.append( T_values[n]**2 )
```

```
fig = plt.figure(1)
my_fig = fig.add_subplot(1,1,1)
plt.grid(True)
plt.plot(thetas, T2_values, color = 'black', label ='L = 1.0m', linestyle = '-')

my_fig.set_xlabel('Starting Angle (rad)')
my_fig.set_ylabel('$T^2 (s^2)$')
my_fig.set_title('$T^2$ vs Varied Starting Angles')
plt.legend(loc = 'best')

plt.savefig('T2 v Theta.png', dpi=300)




#This code will plot a T^2 vs Varying string length with a small and large angle

L = 1.0
g = 9.8
theta_0 = 5*np.pi/180
omega_0 = 0
t_0 = 0
delta_t = 10**(-3)

# lists to store data
l = np.linspace(0.1,L,1000)
T_values = []
T2_values = []


# nested loop
# for loop handles different lengths of pendulum
for n in range(len(l)):
    # set initial conditions for looping calculations
    flag = True
    t_i = t_0
    omega_i = omega_0
    theta_i = theta_0
    counter = 0

    # while loop handles omega and theta values
    while flag == True:
        omega_f = omega_i - (g/l[n]) *np.sin(theta_i) * delta_t
        theta_f = theta_i + omega_f*delta_t
        t_f = t_i + delta_t

        omega_i = omega_f
        theta_i = theta_f
        t_i = t_f

        # when your current position is really close to the starting position
        # exit the while loop becuase you've gone a full period
        if (np.abs(theta_i - theta_0) < 10**(-4) and counter >= 20):
```

```
            flag = False
        #This counter ensures that the theta goes back to starting position and not just move a sm
        counter = counter + 1


    T_values.append(t_i)
    T2_values.append( T_values[n]**2 )



fig = plt.figure(2)
my_fig = fig.add_subplot(1,1,1)
plt.grid(True)
plt.plot(l, T2_values, color = 'black', label ='$5^\circ$ Starting Angle', linestyle = '-')

my_fig.set_xlabel('Length of String (m)')
my_fig.set_ylabel('$T^2 (s^2)$')
my_fig.set_title('$T^2$ vs Varied String Lengths')
plt.legend(loc = 'best')

plt.savefig('T2 v L Small angle.png', dpi=300)


L = 1.0
g = 9.8
theta_0 = 90*np.pi/180
omega_0 = 0
t_0 = 0
delta_t = 10**(-3)

# lists to store data
l = np.linspace(0.1,L,1000)
T_values = []
T2_values = []


# nested loop
# for loop handles different lengths of pendulum
for n in range(len(l)):
    # set initial conditions for looping calculations
    flag = True
    t_i = t_0
    omega_i = omega_0
    theta_i = theta_0
    counter = 0

    # while loop handles omega and theta values
    while flag == True:
        omega_f = omega_i - (g/l[n]) *np.sin(theta_i) * delta_t
        theta_f = theta_i + omega_f*delta_t
        t_f = t_i + delta_t

        omega_i = omega_f
        theta_i = theta_f
        t_i = t_f
```

```
        # when your current position is really close to the starting position
        # exit the while loop becuase you've gone a full period
        if (np.abs(theta_i - theta_0) < 10**(-3) and counter >= 25):
            flag = False
        #This counter ensures that the theta goes back to starting position and not just move a sma
        counter = counter + 1


    T_values.append(t_i)
    T2_values.append( T_values[n]**2 )



fig = plt.figure(3)
my_fig = fig.add_subplot(1,1,1)
plt.grid(True)
plt.plot(l, T2_values, color = 'black', label ='$90^\circ$ Starting Angle', linestyle = '-')

my_fig.set_xlabel('Length of String (m)')
my_fig.set_ylabel('$T^2 (s^2)$')
my_fig.set_title('$T^2$ vs Varied String Lengths')
plt.legend(loc = 'best')


plt.savefig('T2 v L Large angle.png', dpi=300)
```