CW3: Two Balls on a Spring [tuple, list, for]

I. A ball on a spring

```
from visual import *

g = 9.8                     # g = 9.8 m/s^2
size, m = 0.05, 0.2         # ball size =  0.05 m, ball mass = 0.2kg
r, k = 0.5, 15              # spring original length = 0.5m, force constant = 10 N/m

scene = display(width=800, height=800, center=(0, -0.2, 0), background=(0.5,0.5,0))       # open window
ceiling = box(length=0.8, height=0.005, width=0.8, color=color.blue)               # ceiling
ball = sphere(radius = size,  color=color.red)                         # ball
spring = helix(radius=0.02, thickness =0.01)                           # spring endpoint A at (0,0,0), 10 coils

ball.pos = vector(0, -r , 0)                                           # ball initial position
ball.v = vector(0, 0, 0)                                               # ball initial velocity

dt = 0.001
while True:
   rate(1000)
   spring.axis = ball.pos - spring.pos            # spring extended from spring endpoint A to ball

   spring_force = - k * (mag(spring.axis) - r) * spring.axis / mag(spring.axis)        # spring force vector
   ball.a = vector(0, - g, 0) + spring_force / m                       # ball acceleration = - g in y + spring force /m

   ball.v +=   ball.a*dt
   ball.pos += ball.v*dt
```

1. Tuple assignment

size, m = 0.05, 0.2   or   (size, m) = (0.05, 0.2)

*** a = (0.05, 0.2) is a tuple data type. When two tuples are at both sides of an equation, the value assignment is done term by term in order. Here ball size and ball mass are assigned in the same line. You can do this for many items in the same line as long as both sides of the equation has the same number of variables and values. The same for

r, k = 0.5, 15

2.

Draw a spring (spring = helix(radius=0.02, thickness =0.01)) as a helix object with radius = 0.02 and thickness = 0.01. Its endpoint A defaults to vector(0, 0, 0). It can be fixed later by spring.pos = vector(0,0,0) or while generating it by spring = helix(pos = vector(0,0,0), radius=0.02, thickness =0.01). More info is at http://vpython.org/contents/docs/helix.html. You can find the usage for its axis attribute, such as used in spring.axis = ball.pos - spring.pos, that draws the spring from spring.pos to ball.pos

3. Force vector by the spring,

spring_force = - k * (mag(spring.axis) - r) * spring.axis / mag(spring.axis)

mag(a vector) is a function that yields the magnitude of the vector. Can you figure out why this line yield the force generated by the spring?

4. Find the acceleration

ball.a = vector(0, - g, 0) + spring_force / m

Can you understand this line?

II. *** More about python:

1. List:

```
a = [1, 5 , 'a', 4.0]
print a[0]
print a[1]
print a[2]
for i in a:
   print i
```

Variable a is assigned to be a data type call list, which is an ordered collection of objects. In Python, objects can be any data type. These include integers, floats, strings, or even lists, or any generated objects, such as the "ball" used in our simulation. In the example, the 0th element of a is integer 1, the 1st is integer 5, the 2nd is string 'a', and so forth. Notice in python, the order starts at 0. And more important, unlike other coding languages, the elements of list can be mixed of any different types. The major difference between a list and a tuple is that for list, its length and elements can be changed anytime later, but tuple can NOT. For example,

```
a=[]
a.append(1)
a.append(2)
a.append(3.0)
a[0]= 5
```

After this, contained in list a will be 5, 2, 3.0

But

```
a = (1, 2, 3)
a[0] = 5.0
```

will yield an error.

2. for loop

```
for i in a:
   print i
```

is to let variable i be the value of each element in list a, then to execute the associate codes of "for" in order.

Homework Submission:

Now, modify the above program.

First generate two lists,

sizes = [0.05, 0.04]

ms = [0.2, 0.15]

Then use a list variable "balls" to contain two balls generated by sphere, such balls[0] has attribute radius = sizes[0] and mass=ms[0] and balls[1] has sizes[1] and ms[1].

Instead of hooking one ball on the spring to the ceiling, now let the two balls connect together horizontally by the spring and simulate their oscillation in x axis, with the position of balls[0] originally at vector(0,0,0) and the position of balls[1] originally at vector(-r-0.2, 0, 0).

At the end of each period of the oscillation, print the period of the oscillation and the averaged center of mass.