PHY 250L – Spring 2018
Python tutorial 5

Welcome back to the PHY 250L Python programming tutorial!

Things you should understand after week 5:

- numerical integration
- how to choose an appropriate integration step size
- the basics of VPython
- primitives in VPython
- animating motion given a force

Problems for 3.29.2018 One of the following problems should be completed and uploaded to Sakai by 09:45 on 3.29.2018. Each problem should correspond to its own python program (*i.e.*, each problem will correspond to a single file). The preferred names for the files are indicated in each problem.

1. `earth_sun.py`, 50 points
   Using VPython, write a simulation of the Earth's orbit around the Sun. Your simulation should calculate the gravitational forge between the two and update the Earth's velocity and position accordingly. (You can leave the Sun stationary.) You will need to look up or calculate the relevant physical parameters for this to work. Though the geometry of the system should be physically accurate, you may make the radii of the Earth and Sun larger than life so that you can see them. Note that the timescale here should be big! If you set $dt = 0.0001$ and use the SI-unit values of the physical quantities, your simulation will take a rather *long* time to produce any discernible motion! Your simulation should run for at least three years (of simulation time, not real-world time).

2. `dpends.py`, 50 points
   Using VPython, write a simulation of two coupled harmonic oscillators, m_1 and m_2, constrained to move along the x axis. Each has a mass of 1 kg. m_1 has an equilibrium point at $x_1 = -0.2$ m, to which it is coupled by a spring with spring constant $k = 12$ N/m. m_2 has an equilibrium point at $x_2 = +0.2$ m, to which it is coupled by a spring with spring constant $k = 12$ N/m. The two masses are coupled by a wimpy spring ($k = 0.01$ N/m) that has a relaxed length of 0.4 m. Begin the system with the m_1 at $x = -0.25$ m. Include in your simulation a plot of the total energy of each mass versus time (*i.e.*, two plots on the same axes).