Chesick Scholars - Intro to computer programming in STEM

Prof. Dan Grin

Deep Patel

When

Monday 8/22 1-2:30 PM

Tuesday 8/23 1-2:30 PM

Wednesday 8/24 11-11:30AM

Where

H204 - SCIENCE BUILDING

Mask please

Extra masks available at front of room

Today's agenda

- Download Mersive Solstice Client in windows on desktop and try it
- Login - go to your program with sphere calculations
- Program flow — loops, if / then
- Example(s) — falling ball — *your job, add acceleration and motion along table*
- Example- moving planet — *your job, add the moon or extra-solar planets*
- Mini-project — projectile motion

vpython – visual data types

```
box()
```

```
box(pos=vector(1,0,0),size=vector(.5,.3,.2),
    color=color.green)
```

mybox=box(length=0.5, height=0.5, width=0.5, color=color.yellow)

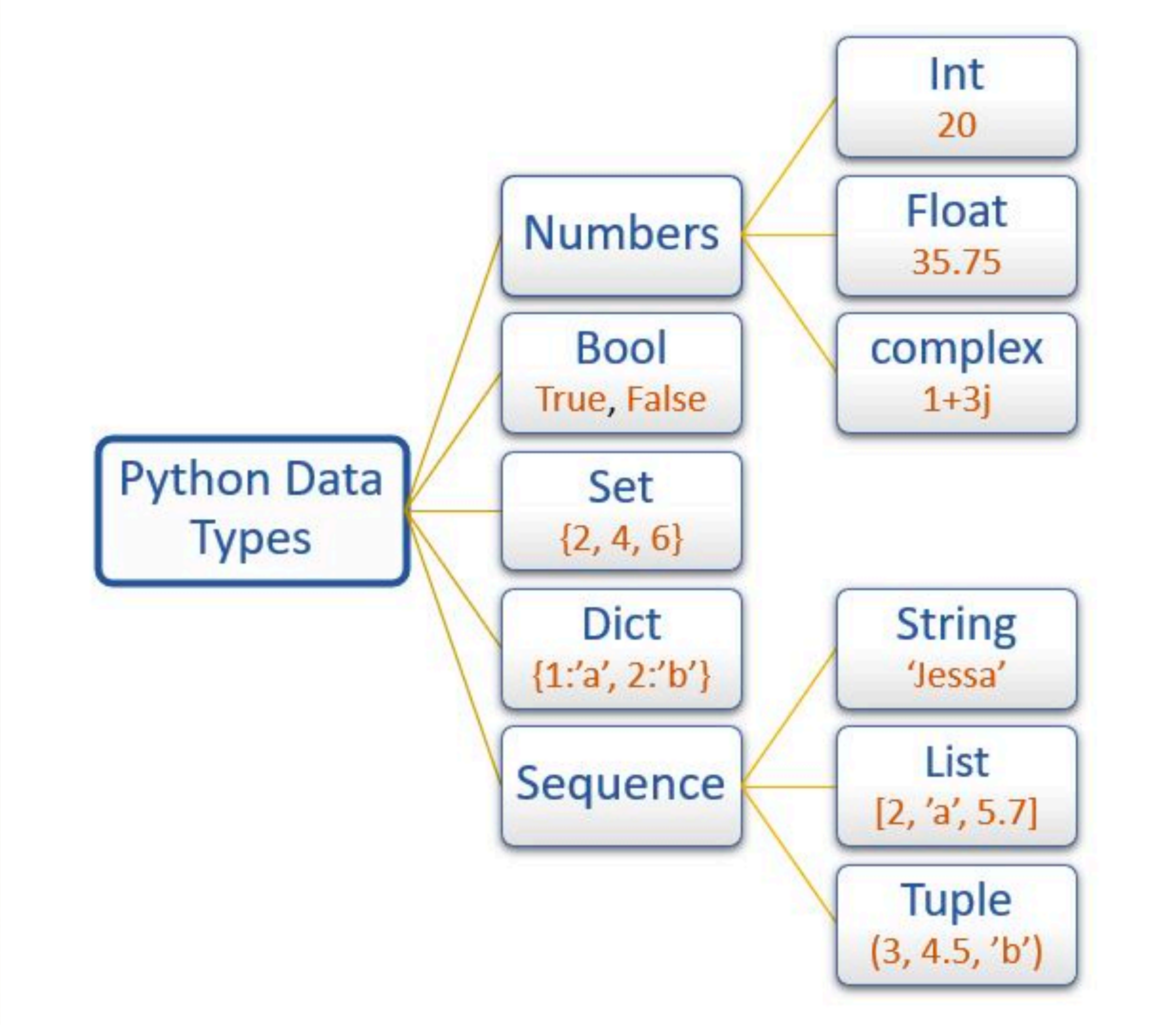
Computer takes care of world rendering

Try it! - now change it!

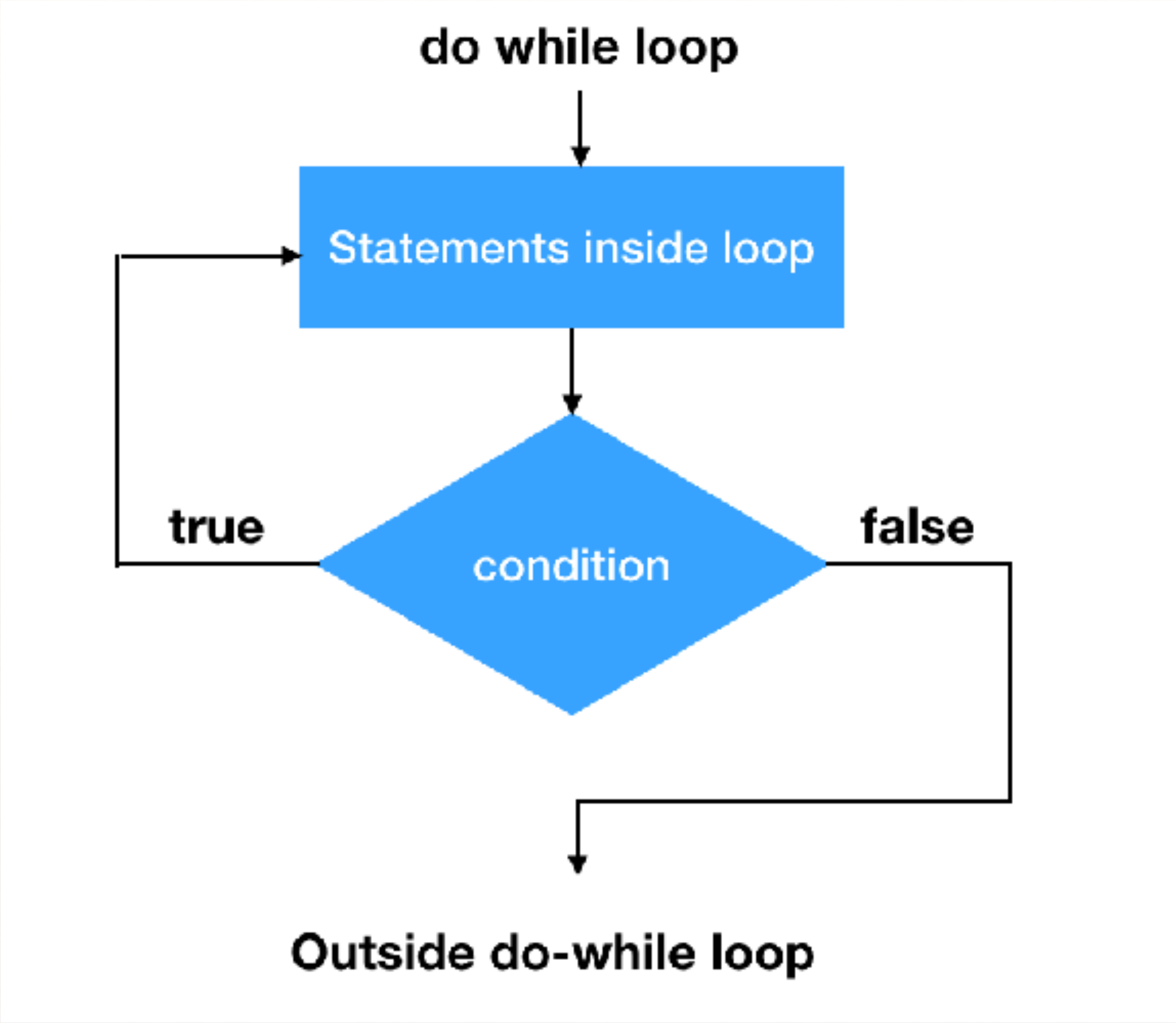
Calculating with python and using variables

```
R = 1

mysphere = sphere(radius = R, color = color.magenta)

volume = (4/3)*pi*R**3

print("The volume of the sphere is:", volume)
```

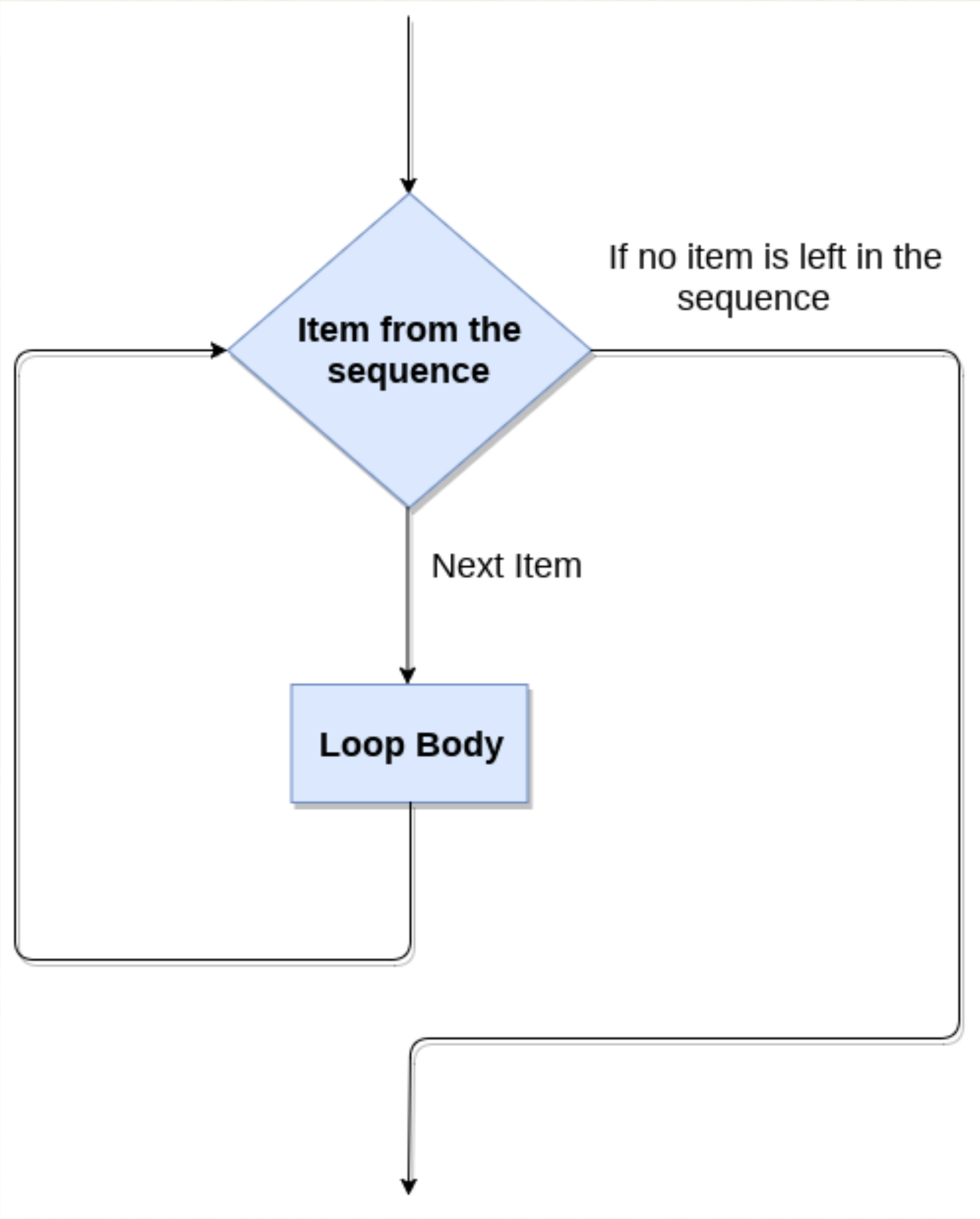
More Python data types



program flow



for loop



Loops, example

https://matter-interactions.trinket.io/00_welcome_to_vpython#/welcome-to-vpython/loops

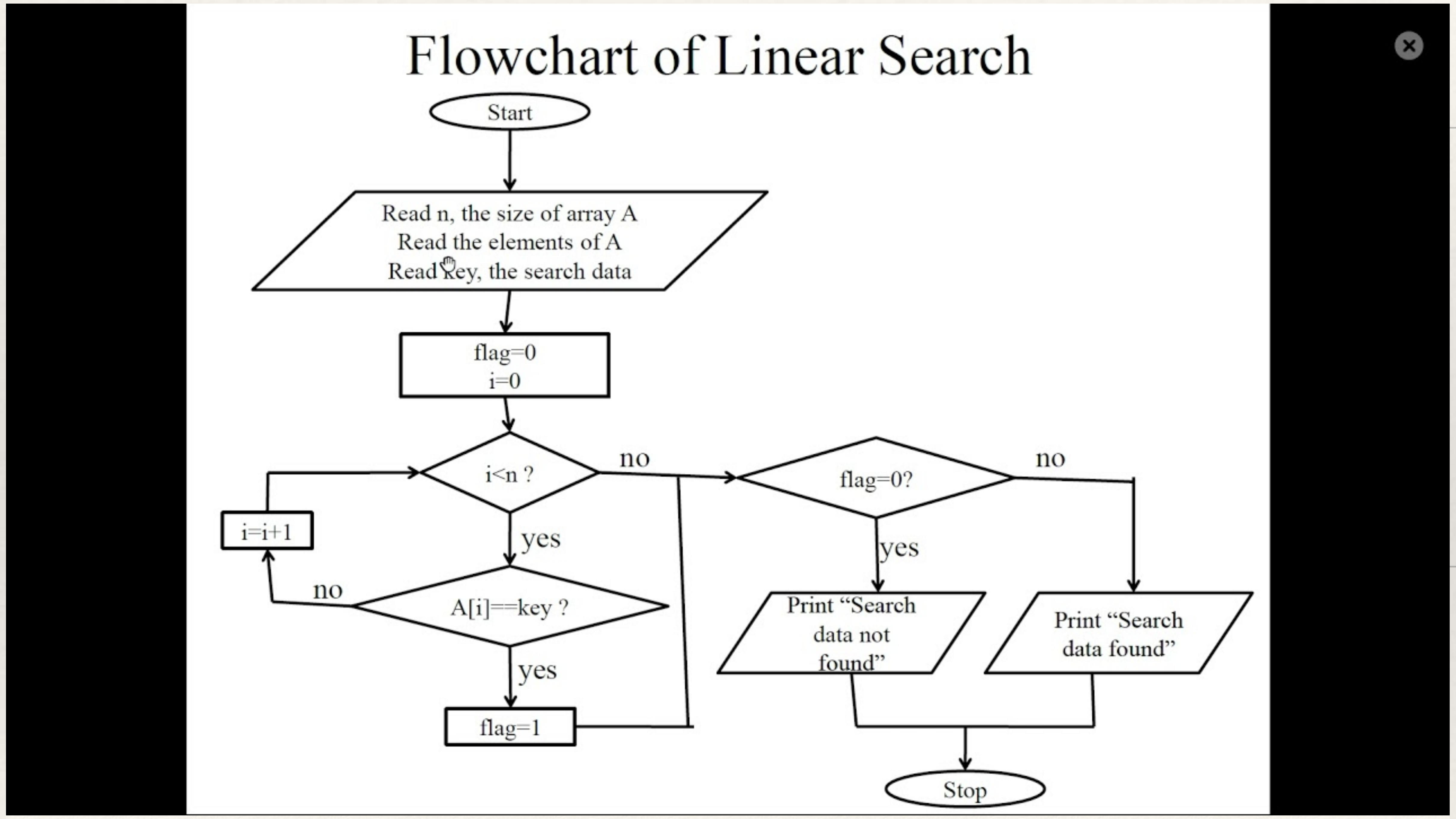
```
t = 0
print("t =", t)

t = t + 1
print("t =", t)

t = t + 1
print("t =", t)
```

```
t = 0
while t < 11:
  print("t=", t)
  t = t + 2

print("The loop is finished")
```

dg loop examples

Debugging

Plan your code



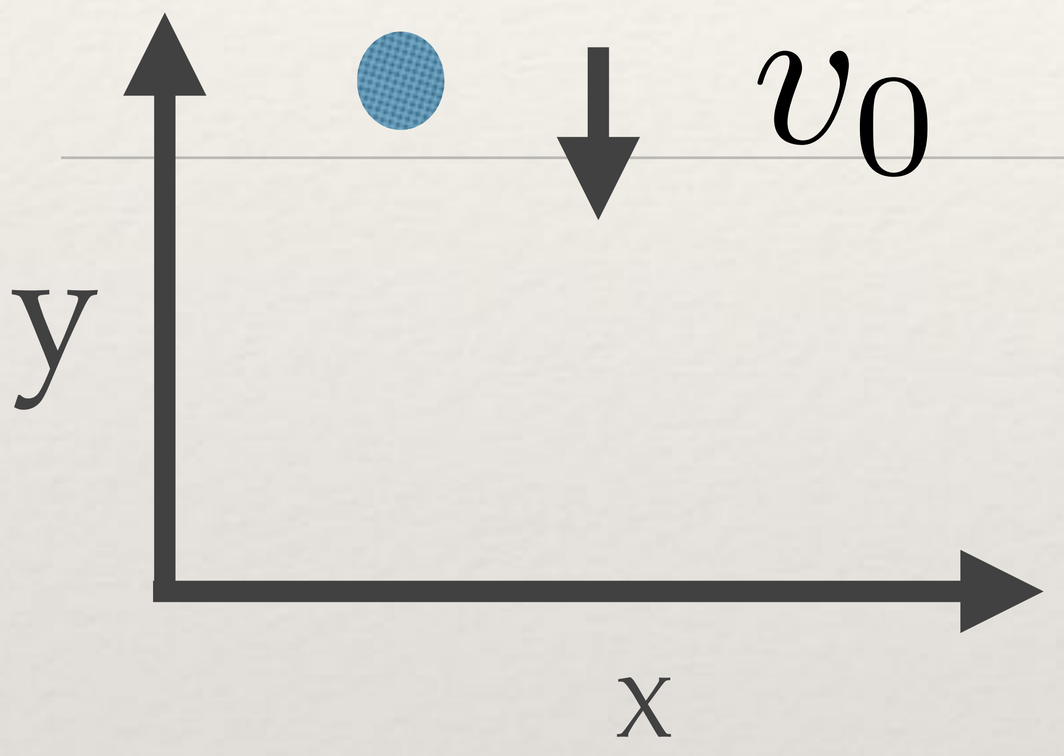
Work in pairs and plan out code on white board – check in with Deep or me before coding

Comment your code

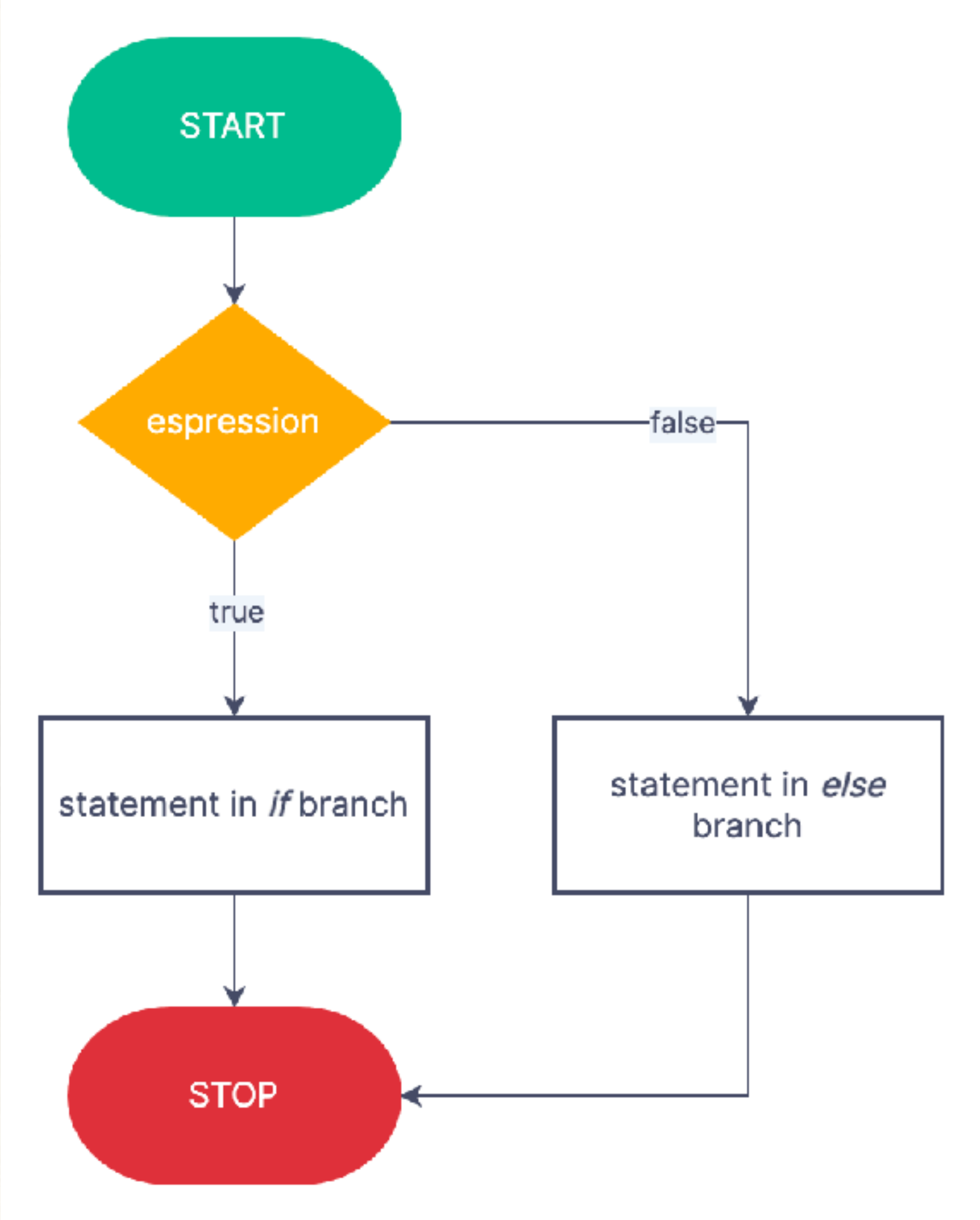
- Helps you remember what you were doing years later :)
- Helps colleagues expand and edit your code
- Helps debug (comment out and add back in code that is not working)

Moving ball example

$$y = y_0 + v_0 t$$



if/then program flow control



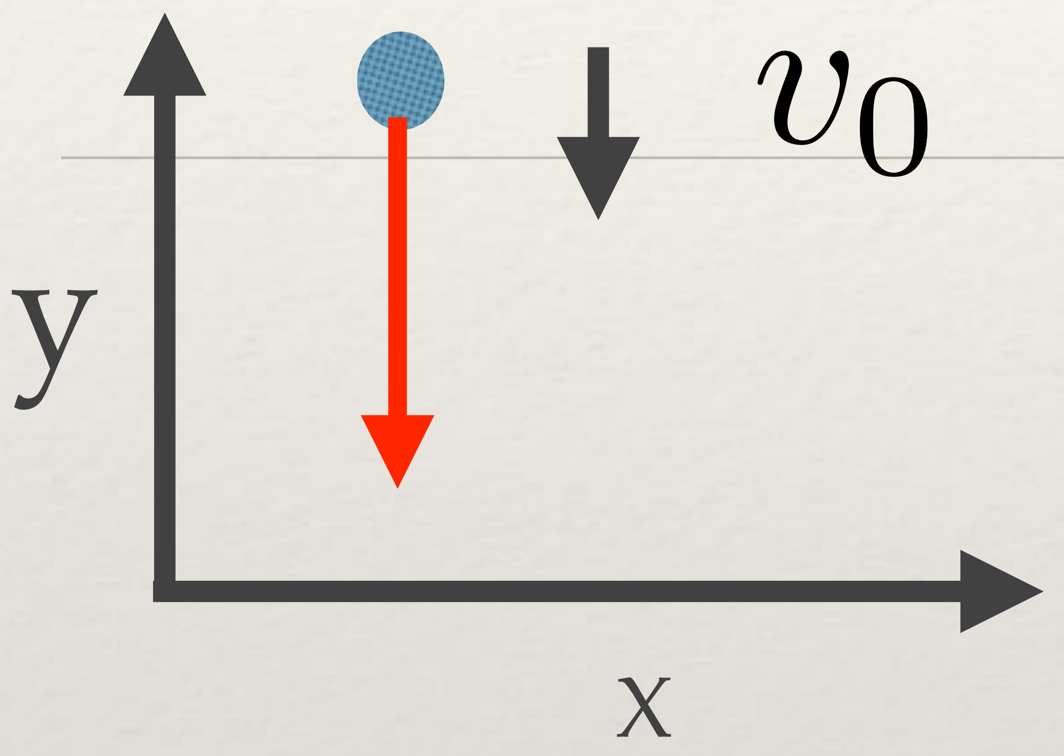
Use to impose table boundary

Your turn – make the ball roll to the right once it hits table

Falling ball with gravity— your turn! (Work in pairs)

$$y = y_0 + v_0 t - gt^2/2$$

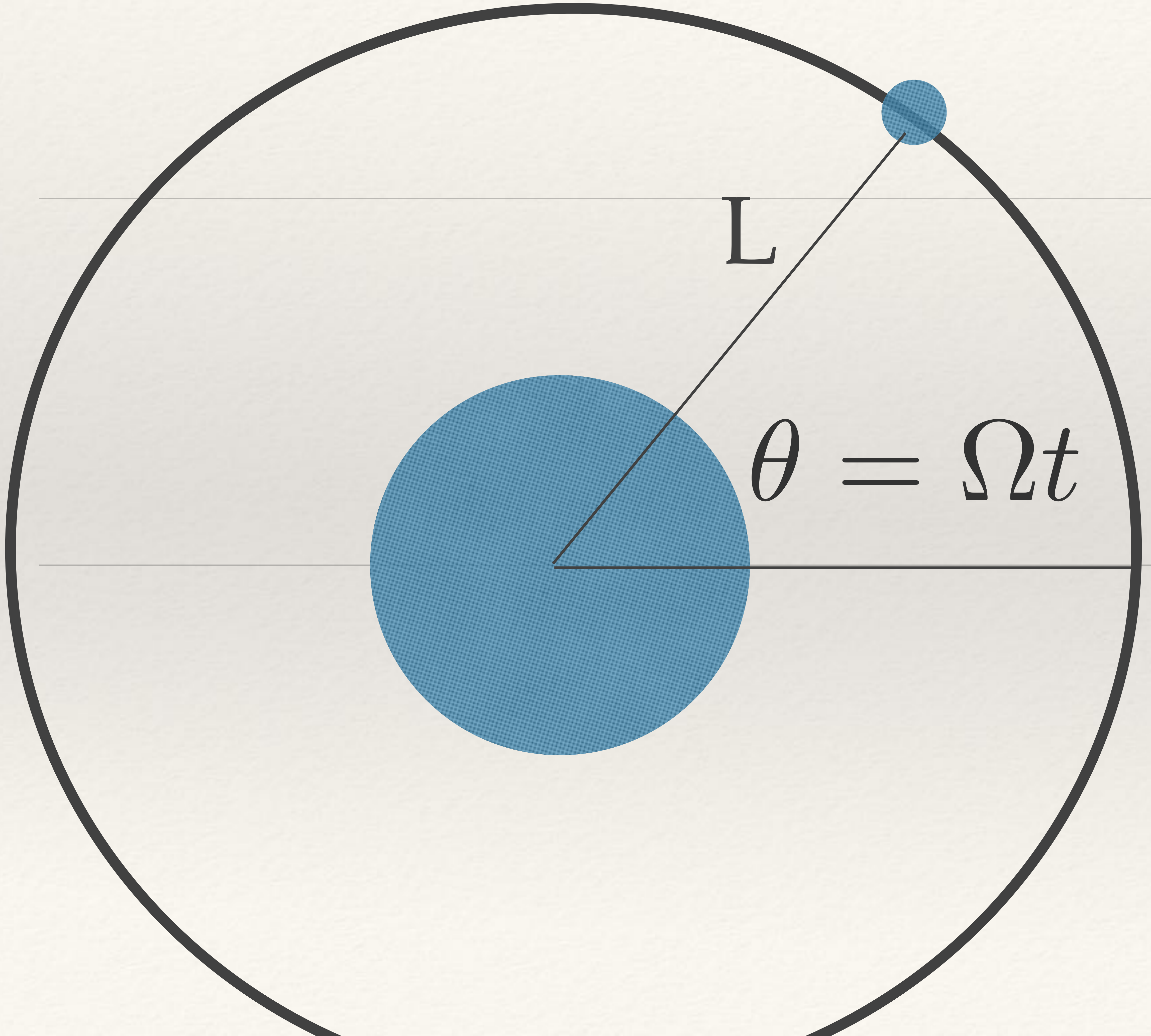
$$v = v_0 - gt$$



$$\vec{F} = m\vec{g}$$

Challenge – add a plot of v(t)

Example – orbits

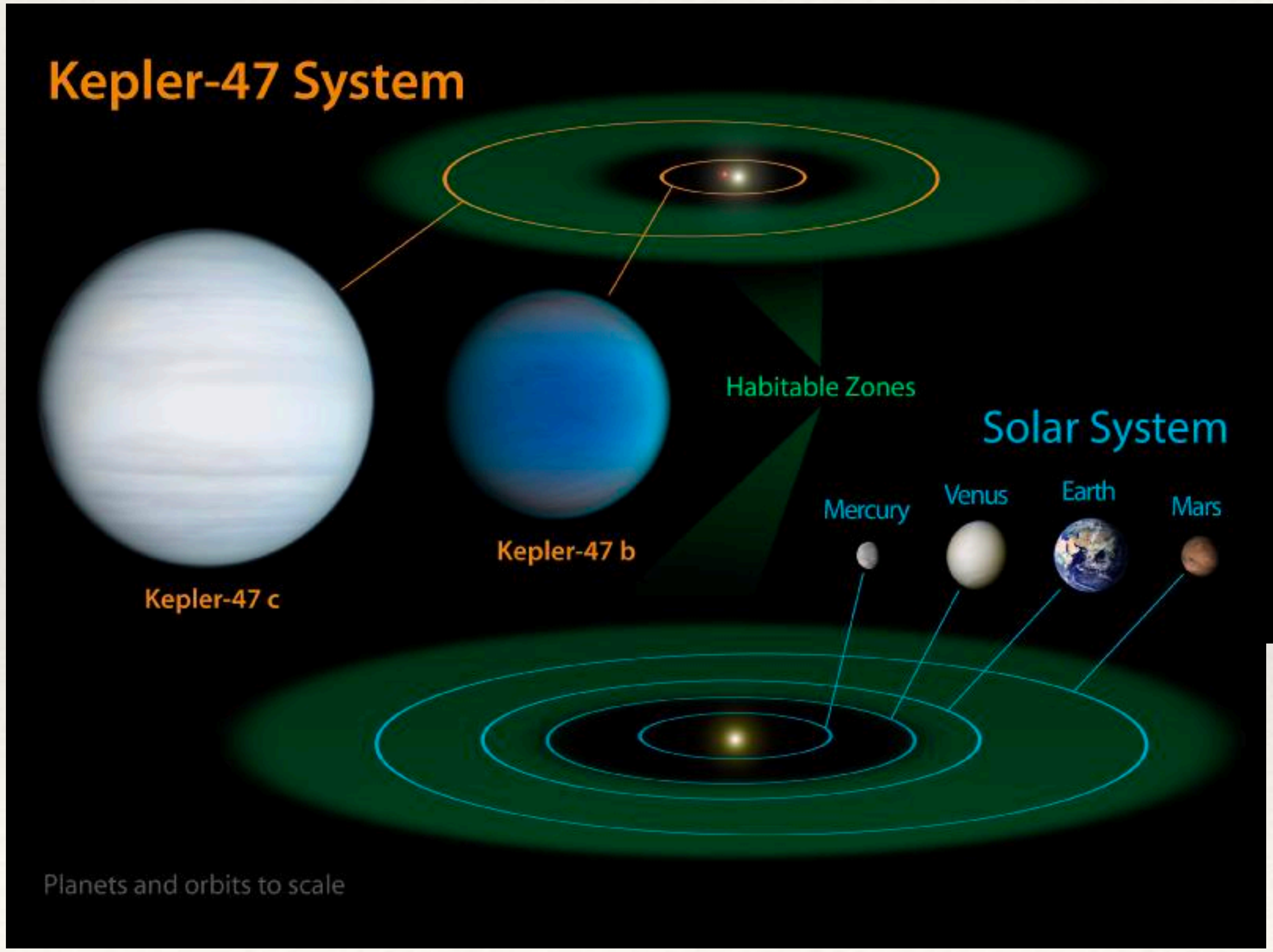


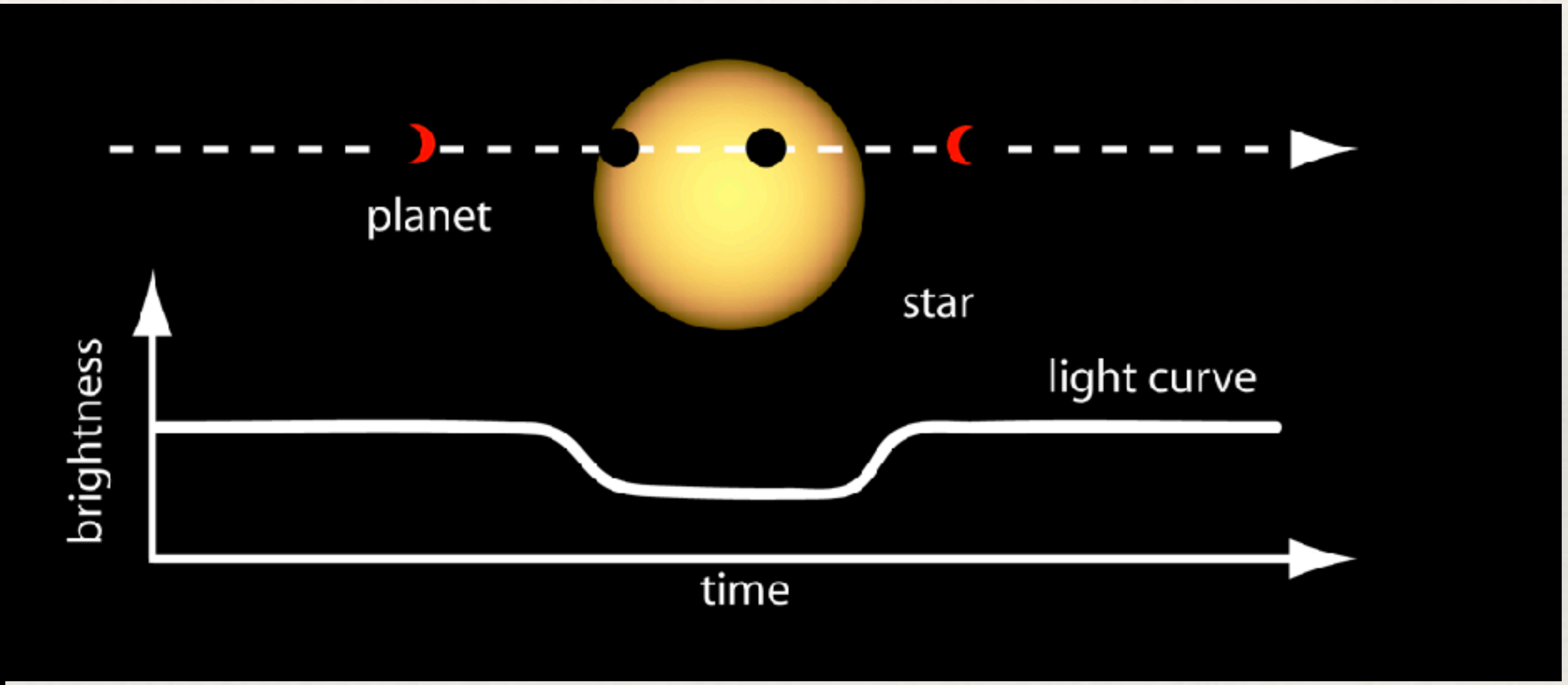
$$x = L\cos\left(\Omega \times t\right)$$
$$y = L\sin\left(\Omega \times t\right)$$

Your turn – add the moon

Your turn — extrasolar planets

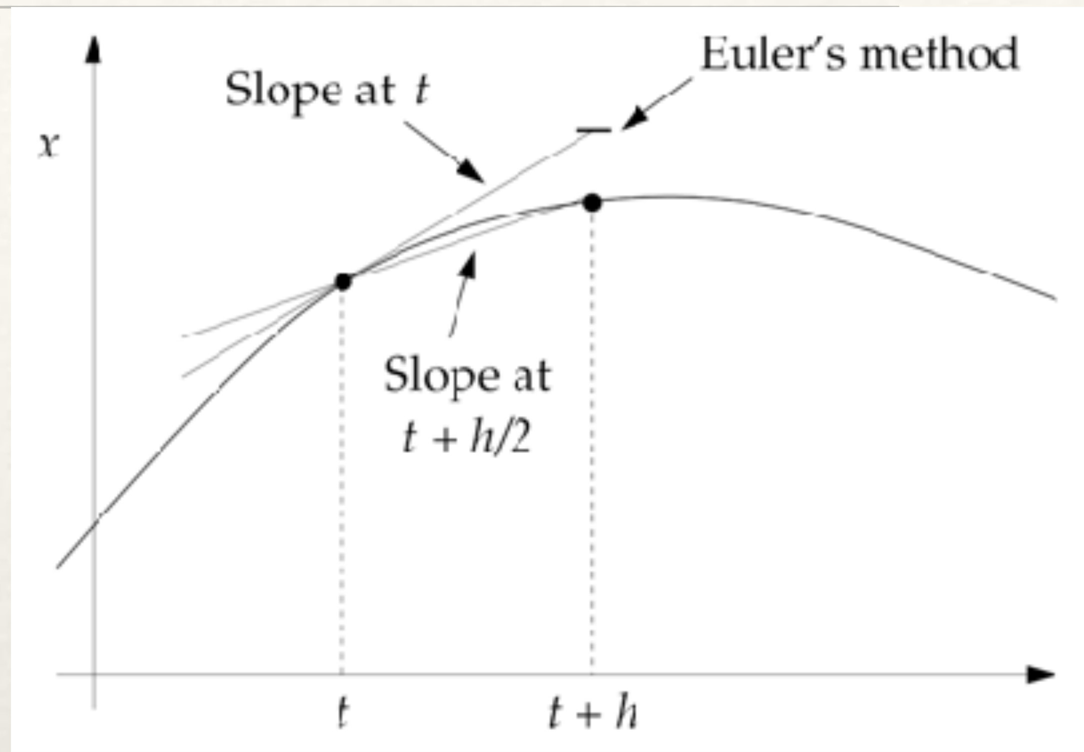
https://exoplanetarchive.ipac.caltech.edu/cgi-bin/TblView/nph-tblView?app=ExoTbls&config=PS





Velocity

$$v_{avg} = \frac{\Delta x}{\Delta t} = \frac{x_2 - x_1}{\Delta t}$$



$$x = x + v * dt$$

Chop motion into segments/**assign** new position

Euler's method

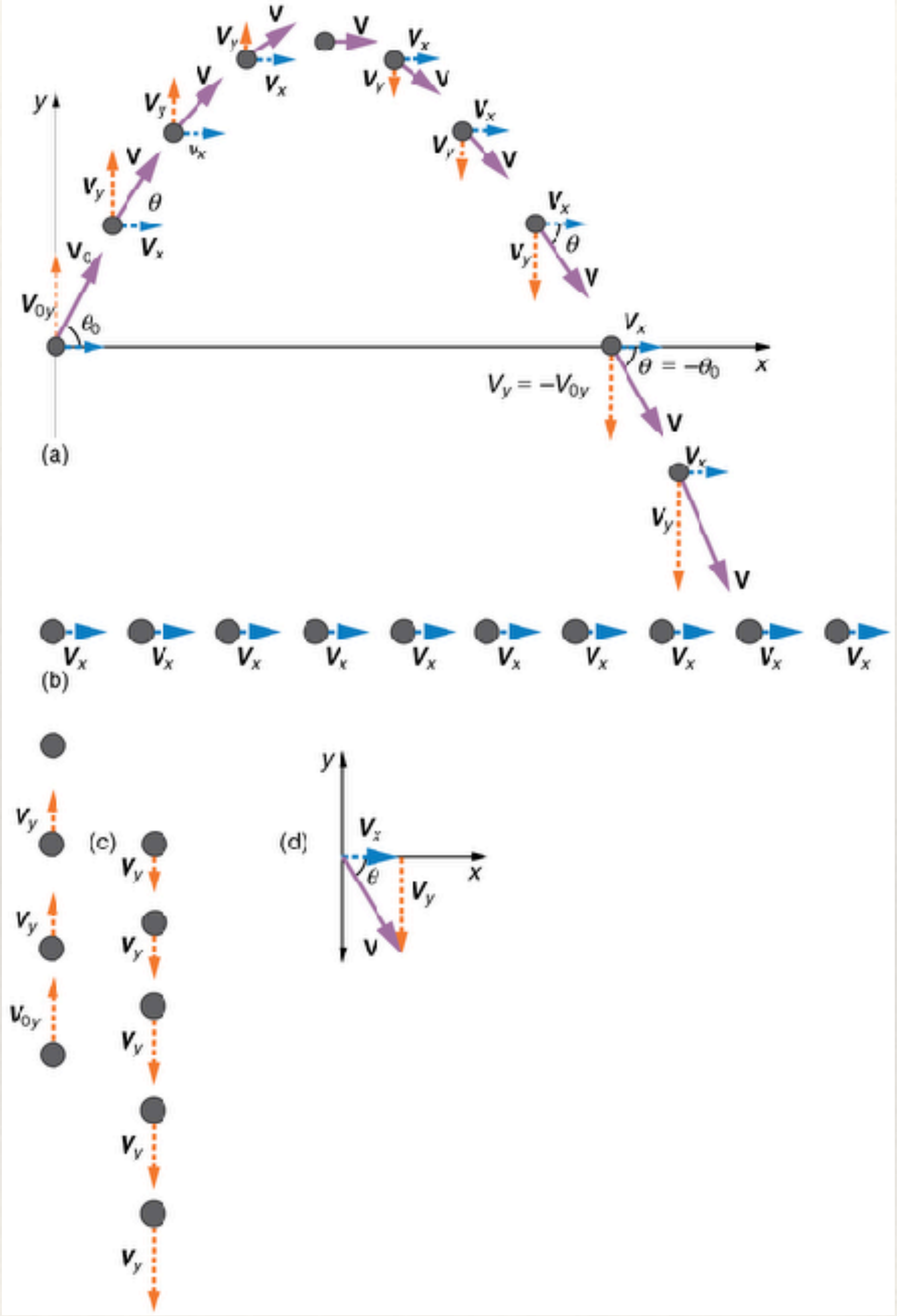
$$x_2 = x_1 + v\Delta t,$$

Animation/integration

```
GlowScript 2.7 VPython

# Defining the "graph"
g1=graph(width=400, height=250)
xDots=gdots(color=color.green, graph=g1)

# Defining the Object
obj=sphere(pos=vector(-1,0,0),radius=0.1,color=color.red)

# Setting initial conditions and step size, dt
t=0
dt=0.05
x=-3.0
v=2.0

# This is main part - the loop
while t<3:
    rate(10)
    obj.pos=vector(x,0,0)
    xDots.plot(t,x)
    # Updating the position
    x=x+v*dt
    t=t+dt
```

Integration rather than exact formula

Projectile motion



and

$$x = x_0 + v_{x0}t + \frac{1}{2}a_x t^2;$$

$$v_x = v_{x0} + a_x t;$$

$$v_x^2 = v_{x0}^2 + 2a_x(x - x_0).$$

Your turn (work in pairs, make sure to write pseudo code + lots of comments):

1) *Write a program to simulate this- make the projectile stop when it hits ground - is the range consistent with*

$$R = \frac{v_0^2 \sin 2\theta}{g} \quad ?$$

2) *Make the ground sloping [and make the ball stop when it hits the sloping ground]*

Last example - a more realistic method

$$x = x + v * dt$$

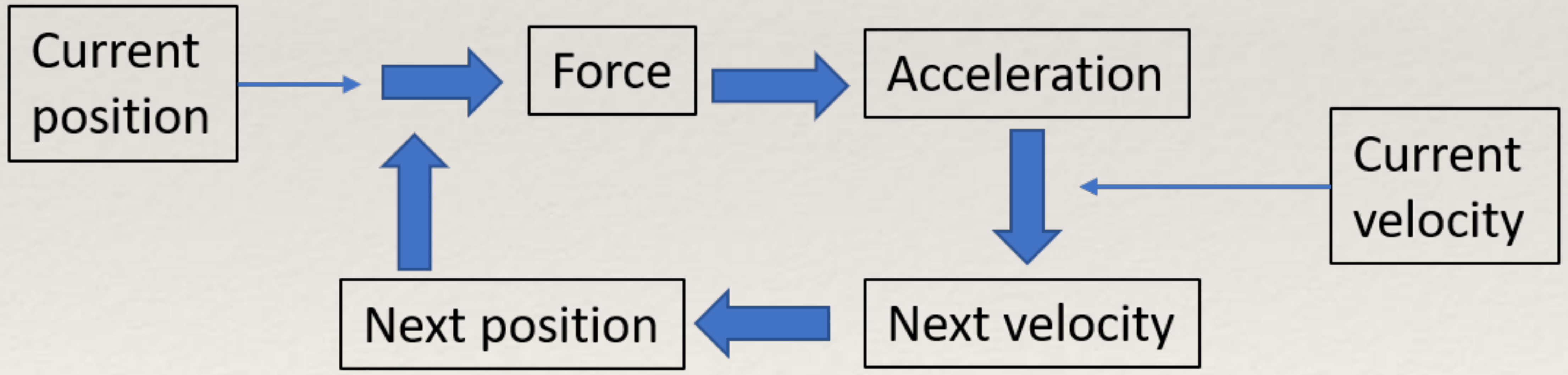
Assumed constant velocity

$$\vec{F} = m\vec{a} = \frac{\Delta \vec{v}}{\Delta t}$$

$$\vec{v} = \frac{\Delta \vec{x}}{\Delta t}$$

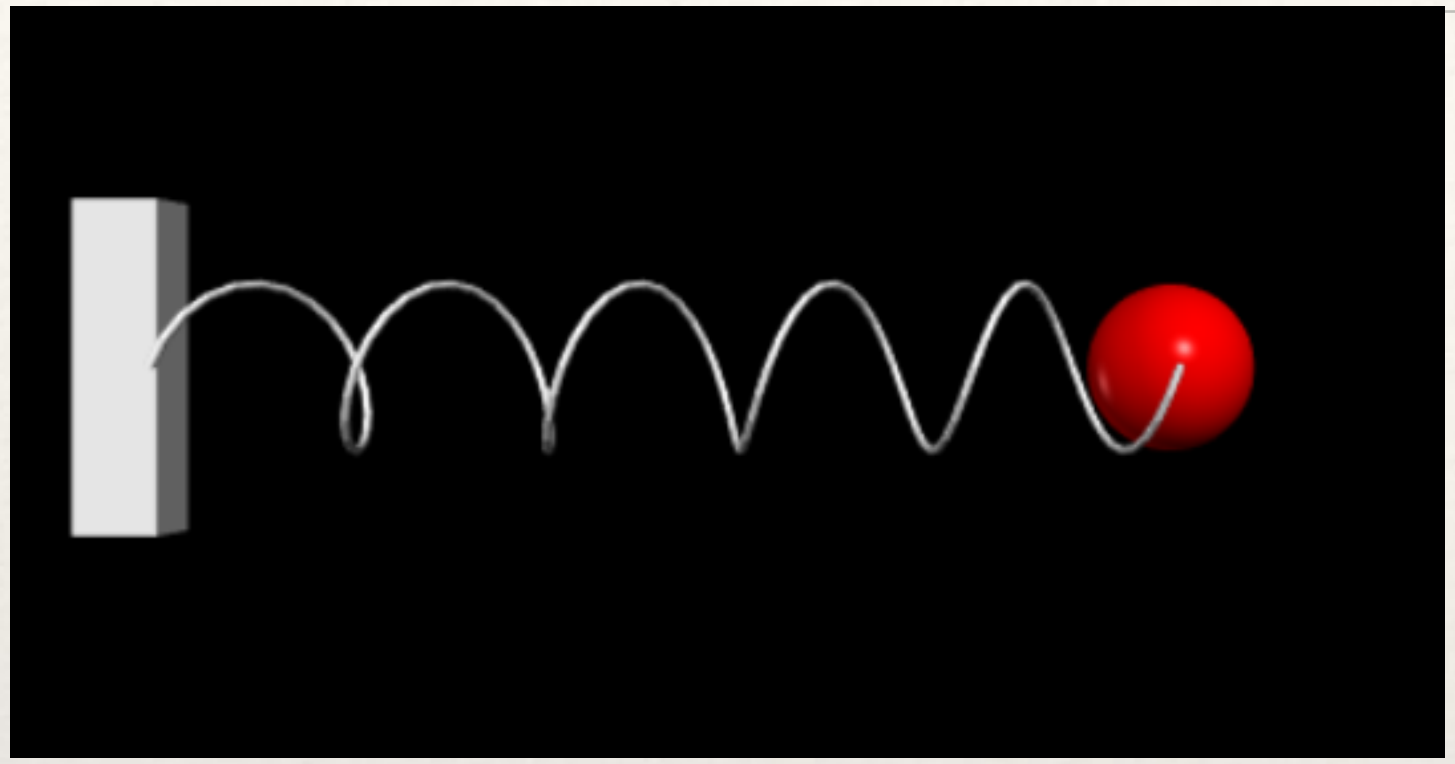
More realistic

```
F=-k*x
a=F/m
v=v+a*dt
x=x+v*dt
t=t+dt
```



Euler-Cromer method

Simple harmonic oscillator



To illustrate the method, let's get our feet wet with a relatively simple problem - a mass on a horizontal spring. All we have to remember is Hooke's law for ideal springs,

$$F(x) = -kx, \tag{6.5}$$

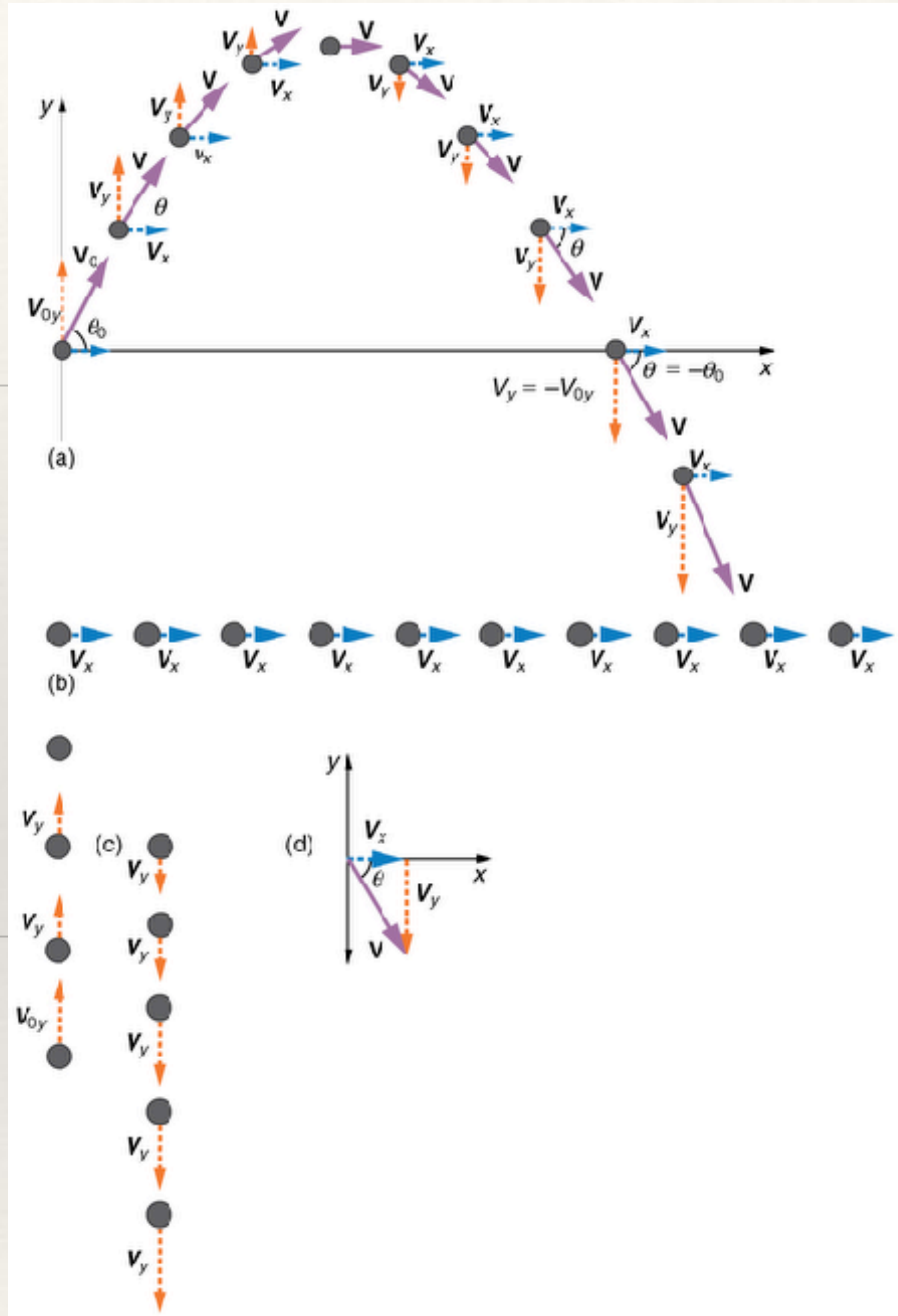
1) Try DG example code
2) Change the initial conditions of the mass, how does that change the graphs?
3) *Change k and m? How does quadrupling k or m change the period of oscillation?*
4) *Compare your results with the formula*

$$T = 2\pi\sqrt{\frac{m}{k}}$$

Your turn - projectile motion with drag.

1) *Redo projectile motion with Euler-Cromer method (check agreement with exact formulae)*
   work in pairs, make sure to write pseudo code + lots of comments

$$
\begin{aligned}
v_x &= v_x \\
v_y &= v_y + a_y t \\
x &= x + v_x t \\
y &= y + v_y t \\
a_y &= -g = -9.8 \text{ m s}^{-2}
\end{aligned}
$$



2) *Add drag force (air resistance)*

$$\vec{F}_D = \frac{1}{2}\rho A C_D v^2(-\hat{v}), \qquad \hat{v} = \vec{v}/|v|.$$

- **mag(A)** and **mag2(A)**, where the output yields the length and length-squared of that vector, respectively.
- **A.hat**, which produces from the vector **A** its unit vector (by dividing it by its length). This syntax treats the directionality (given by the unit vector) as a *property* of the vector. It is just like any other property of a vector, such as **A.x**, which yields the x-component of the vector **A**.

Projectile motion

- *How does the range quantitatively compare with before?*