Loops and Orbits-Week 1-Day 2- Computer Science

Let's review a bit. At the end of last time, we were just starting into comparison expressions which are very often used to control flow.

NEWBIE MISTAKE!!
Confusing == with =.
Comparison is not assignment!

Comparisons

>	read this as "is less than"
<	read this as "is greater than"
==	read this as "is equal to"
>=	read this as "is greater or equal to"
<=	read this as "is less than or equal to"
!=	read this as "is not equal to"

Example 5!=3 is read as:

"Five is not equal to three?"

Read comparison operators as questions. Your tone should rise up a little at the end, even if you are just listening to the voice in your head.

Control flow — While Loops

Now that we have comparisons, we can use them in code to have the program decide what to do next.

```
x = 27      ← position of ball in meters
v = 6       ← velocity of ball in meters/second
dt = 1      ← time step in seconds

t = 3738    ← initial time in seconds,
                  final time in seconds
while t < 3748:
    x = x + v * dt
    t = t + dt
                    at the end,
t, x ←              have the kernel return,
                    both t and x back
                    to the notebook
```

The only thing that is not like the example you did in worksheet 2 is that at every time step $v=6$ (meters/second). In Messi's actual run, $v_{0\to1} = 3$ m/s, $v_{1\to2} = v_{2\to3} = v_{3\to4} = v_{4\to5} = 6$ m/s, $v_{5\to6} = 9$ m/s, $v_{6\to7} = 21$ m/s, $v_{7\to8} = v_{8\to9} = 6$ m/s, and $v_{9\to10} = 9$ m/s. Next we will learn how to vary v.

Lists

Let's call $v_0 \to 1$ v0
and $v_1 \to 2$ v1
and $v_2 \to 3$ v2

⋮

and $v_9 \to 10$ v9

Python variables are allowed to have in digits them as long as they don't start with a digit

So we could put the ball velocities in like this

```
v0 = 3
v1 = 6
v2 = 6
v3 = 6
v4 = 6
v5 = 9
v6 = 21
v7 = 6
v8 = 6
v9 = 6
```

← lots of repetitive typing

However, that is going to involve a lot of unnecessary typing and we are failing to use the computer's ability to do repetitive work.

Instead we do this:

velocities = [3, 6, 6, 6, 6, 9, 21, 6, 6, 9]

Suppose we want the first velocity.

velocities[0]

the first one has index 0 — this will give 3

velocities[5]

the 6th one has index 5 — this will give 9

velocities[-1] this is how you get the last element in Python!

Here is how we write the code to get the ball position from the velocities:

```
x=21
velocities=[3, 6, 6, 6, 6, 9, 21, 6, 6 9]
dt=1
t=3738
while t < 3748:
    x= x + velocities[t] * dt
    t= t + dt

t, x
```

Suppose you want to know all the ball positions and all the times. Make them lists!

```
positions = [27]   # this list starts with just the initial position
times = [3738]     # this list starts with just the initial time
velocities=[3, 6,6, 6, 6, 9, 21, 6, 6 9]
dt=1
i=0   # a counter that will take on the 10 values 0, ..., 9
while i<10:
    x_before = positions[-1]
    t_before = times[-1]
    x_after = x_before + velocities[i]*dt
    t_after = t_before + dt
    positions.append[x_after]
    times.append[t_after]

times, positions
```

Summary of List Features

Let's just summarize the computer science concepts we have added.

* We know how to use conditionals to control the execution of while loops
* We have introduced lists of integers
    ** You can have lists of anything. For us it will be very common to have lists of floating point numbers rather than integers.
    ** We know how to access the first element (`my_list[0]`), and the last element (`my_list[-1]`), and anything in between.
    ** We know how to initialize lists
    ** We know how to append to lists
* By the way, pretty much all computer languages have lists. In almost all languages besides Python, what we are calling lists the other languages call arrays. Beats me, why Guido van Rossum was out of step on that.