First-Class Functions

Computational Thinking, Day 7

Agenda

- Dailies
- Indentation example
- PopularName.py
- First-class functions - definition
- FCFs thought experiment
- First-class functions - programming practice
    - (with file reading/writing review!)

Dailies

- I am confused about how indentation is read and would appreciate if we went over it in class.
- I'm very relieved that iterations seem a lot more simpler than recursions... at least so far they do.
- Struggled immensely with the "PopularName" code and dealing with carrying over information from one file to another.
- Loops are easy to understand the concept but I am having a hard time implementing them easily.
- things are definitely going better with Python, I think I am over the worst of the learning curve, although I wouldn't quite say that I know what I'm doing
- In general, I am feeling far more confident in my skills
- absolutely loved having a dog come into class

First-Class Objects

In Python, all objects are “first-class”. This means they can be:

- assigned to variables.
- passed as arguments to procedures.
- returned as values of procedures.
- incorporated into data structures

http://groups.csail.mit.edu/mac/classes/6.001/abelson-sussman-lectures/

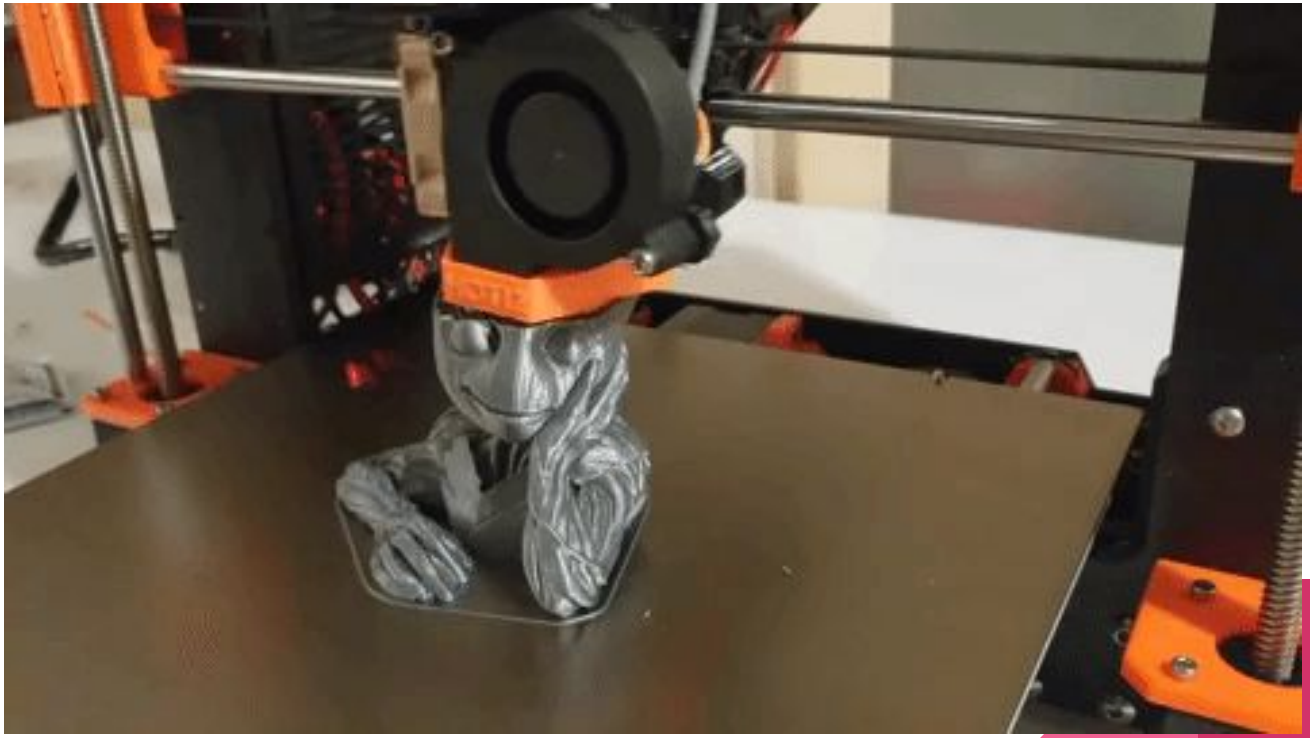
First-Class … Functions?

Saying that functions are first-class means… what?

-> Functions have all of the same privileges as on the prev. slide.

(assigned to varnames, passed to other functions, returned from other functions, incorporated into our own types)

3D Printer Analogy



3D Printer Analogy

3D Printer Analogy

What does this mean for us?

-> We can treat functions the way we've been treating other objects so far!

(let's see an example)

map(), filter(), and reduce()

Three functions which each take in:

1) Another function; and
2) A list.

- map(f, list) -> Apply *f* to each element of *list* and return the results
- filter(f, list) -> Apply *f* to each element of *list* and only keep elements that make *f* return True
- reduce(f, list) -> Combine elements of *list* using *f*