1. **What is the DRY principle?** DRY stands for Don’t Repeat Yourself. This means that when a chunk of code is repeating itself in your program, you could have just made a function so that it is only written once.
2. **You will often hear that functions should be short and do one thing. Does that mean each function should always contain 1 statement? Why or why not?** A function can have more than 1 statement. Functions doing one thing means that the function can be explained without having to use “and.” For example, in the exercise code, the function would reverse and borg a name. But after I changed it there was a function to reverse and a second function to borg. A function may take a few steps to properly be able to reverse a name.
3. **Imagine that you extract a unit of work into a well-named function, but that function is only called in one place. Your colleague argues, "Why do that? Now when I want to follow the code I have to jump from the call site to the function definition and then back, it breaks up my flow of reading. The function isn't being reused, so just replace the call with the actual statements so it's all *right there*." The argument has merit. But explain one or two reasons why having the function has its benefits.** The function may only be used once right now but if while adding more to the program it is needed again it is already in place. It can also make the program more readable for a new person. Instead of having to decipher code, all they have to do is read the name of the function. If the function was made right, the name should describe, for the most part, what it does. If the function name doesn’t describe enough, then the comments for the function will help.