Introducing functions

In Swift, functions are declared like this:

```
func functionName()
{

}
```

Look familiar? That's how the `initialize` function is declared.

Functions start off with the keyword `func`. This tells the computer that you want to declare a new function. Then you provide a function name, in this case it was `initialize`. After the function name comes a pair of parenthesis. Last comes the open and close curly braces. Anything in between those two curly braces is part of the function definition.

Why do I need functions?

We know that variables store information to use later, but what about functions? A function stores code that can be run by other parts of the program. Code that it is going to be used more than once or in different parts of a program probably belongs in a function.

Examining our set up function

Let's take a look at the code in the `setupMonsterStats()` function.

It is formatted just like the `initialize` function. It begins with the keyword `func` and is followed by the function name, `setupMonsterStats` and a set of parenthesis `()`:

```
func setupMonsterStats() {
    element = "Leaf"
    level = 1
}
```

We define two different variables in this function: the monster's `element` and its `level`.

Changing the stats

Right now, the variable `element` is set to "Leaf". Try changing the monster's element type to something else, either "Fire" or "Water."

Now run the code. Notice any changes?

Even though we changed the `element` variable, the code in the `setupMonsterStats` function is not actually getting run by the computer, because we never told the computer to look at the code in that function.

We could do tons of crazy stuff in that function, but unless we tell the computer to run the code in that function, nothing will change -- so your monster is still formless!

How to make a function run

Code inside a function will not get run by the computer unless a function is "called." That is how we tell the computer that it needs to pay attention to the code written in a function. To call a function, you need the function's name, followed by parentheses: `functionName()`.

When you ran the code after changing your monster's nickname, you saw that change reflected in the simulator -- that is because the `intialize` function is special, and it automatically gets called.

When we tried changing the information in `setupMonsterStats` without calling the function first, we didn't see any changes in the simulator.

Even though the `setupMonsterStats` function has always been declared underneath the `initialize` function, it is not until we actually call `setupMonsterStats` that the code in that function gets read.