Put the enemy to sleep

You are going to use your monster's powerful **Sing** move to damage your enemy by putting it to sleep. This is going to require the use of everything you have learned so far, so get ready!

The Sing move is only effective if your monster has a higher level than the enenmy monster -- otherwise pressing the Sing button will make your monster level up! You are going to use your monster's powerful Sing move to damage your enemy by putting it to sleep. This is going to require the use of everything you have learned so far, so get ready!

Add the following code between `// singButtonPressed goes below this` and `// singButtonPressed goes above this`:

```
function singButtonPressed {
    if level > enemyLevel
    {
        for counter in 1...(level-enemyLevel)
        {
            singMove()
        }
    }
    else {
        //YOUR CODE HERE
    }
 }
```

That is a lot to digest, so let's break it down into smaller chunks while we take a look at it.

Sing Button Handler

The very first part is the function declaration of the Sing Button Handler. `singButtonPressed()` is what automatically gets called by the computer when it notices someone pressed the Sing button.

If-then-else statement

Next is the if-then-else statment. This time, we are checking to see if our monsters level is greater than the enemy monster's level. The statement `level > enemyLevel` is like asking the question: "is `level` a bigger number than `enemyLevel`?" If the answer to that question is true, then the computer will run the code inside of the first pair of curly brackets, before the `else`.

Otherwise, if the answer to that question is false, then the computer will run the code inside the second pair of curly brackets, after the `else`.

For-loop with singMove()

The code that gets run if the answer to "is `level` a bigger number than `enemyLevel`?" is `YES` has a for-loop. This for-loop means that `singMove()` will get called from the **Attack Library** once for each level that your monster is above the enemy level. If your monster is level 7 and the enemy is level 4, then the code will get run 3 times (`level - enemyLevel = 7 - 4 = 3`).

```
for counter in 1...(level-enemyLevel)
    {
        singMove()
    }
```

Variable change with additon

Earlier, we mentioned that the computer is able to perform algebraic operations like addition and subtraction -- you are going to need to use this to complete the next step. When someone presses the Sing button, we want to make sure their level is higher than the enemy level -- otherwise, we want the monster to level up!

That means that in between the curly brackets after the `else`, where it says `//YOUR CODE HERE`, you need to write code that increases the monster's level. Remember, we stored the monster's level as an `Int` in the variable `level`.

Try your hand at leveling the monster up! If you need some help, head to the next step.