

# Introduction to Loops

Imagine you're building a video game and in this game, you want to add 15 aliens to the screen.

How do we use code to tell a computer this: "Create a variable and call the method 15 times"?

We could write it out 15 times:

```
AddAlien();  
AddAlien();  
AddAlien();  
AddAlien();  
...
```

...We'll spare you the rest. This approach takes a long time and it can easily lead to mistakes. Instead, let's give the instructions once and tell the computer how many times to repeat them:

```
for (int i = 0; i < 15; i++)  
{  
    AddAlien();  
}
```

*Loops* are structures that we can use to repeat instructions until a certain condition is met. That condition could be a change in state ("keep playing music until 10pm"), an end of a list ("say each name out loud"), or a number ("knock three times").

When we see repetition in our code, it's a good sign that we should probably use a loop. By not writing out instructions multiple times, we reduce our chance for errors and save ourselves some time.

There are several kinds of loops that we can use depending on the situation. This lesson will cover:

`while` loops

`do...while` loops

for loops

foreach loops

### ☒ Instructions

Click the shapes in the diagram to move through each step of the loop structure.  
What is the condition or count that must be met to stop looping?