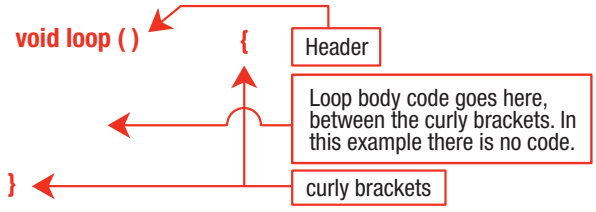
Name:
Date:

// Vocabulary: repetition, header, loop body, curly brackets

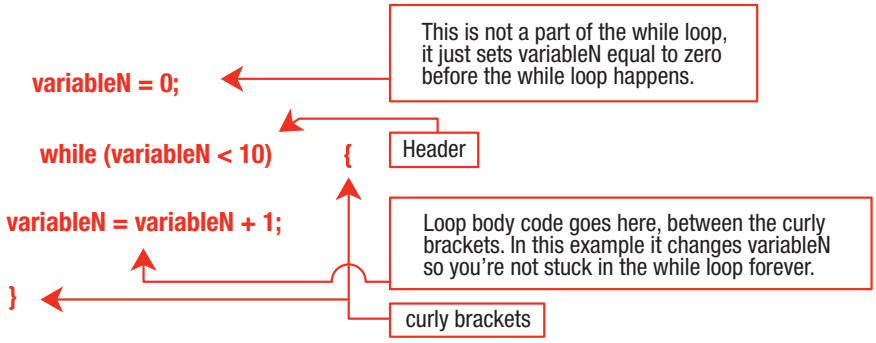
In computer programming **repetition** means repeating a portion of code. This can happen in a bunch of different ways, but the most important thing is to first understand how it happens, not all the different ways it can happen. There are really only two portions to any **repetition**, the **header** and the **loop body**. The **header** usually looks about the same, but the **loop body** can contain any kind of code depending on what you are programming. The **loop body** can even contain another **repetition**!

Repetition with the header, loop body, semicolons and curly brackets labeled:

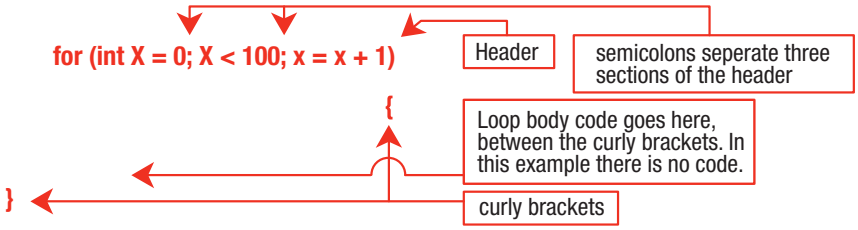
loop ( ):



while ( ):



for ( ):



Just so we're clear on the important concepts that we will use when we talk about each different kind of repetition, please fill in definitions or explanations of the terms below.

Repetition: _____

Header: _____

Loop body: _____

Curly brackets: _____

Name:
Date:

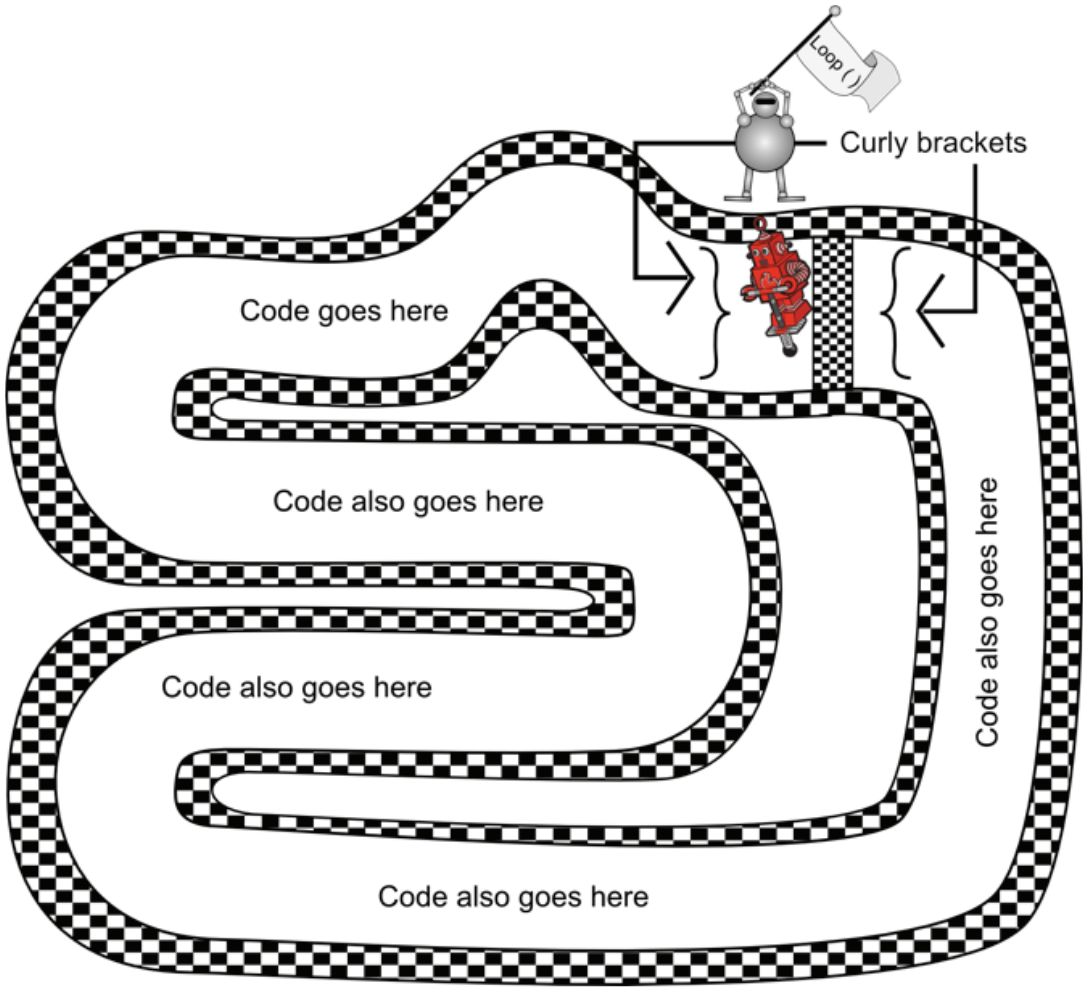
// Vocabulary: loop ( )

The most common form of iteration in Arduino is called the loop( ) function. It exists in all Arduino sketches and its whole purpose is to do all the code written inside of it once, then start over back at the beginning of the loop( ) function and do it all again. Pretty simple, right? The most important things to remember about the loop( ) function are that it is present in every single Arduino sketch, can only be used once per sketch, and it never ends. You will not find a single Arduino sketch that does not have a loop( )function in it and whenever anything happens in your sketch it is because of code inside the loop( ) function.

The loop( ) function looks like this:

```
void loop( ){
 // Lots (or just a little) of loop body code here between curly
brackets.
}
```

Pay attention to the header and the curly brackets which are at the beginning and end of the loop body code. The header is just void loop( ). Think of the loop( ) function as a racetrack. The loop( ) header portion is the flag that starts the computer going around the racetrack and the curly brackets are the beginning and end of the racetrack. Now imagine your computer, Arduino, or robot running around and around the racetrack. It's up to you, the programmer, to put If statements, variables and other code along the way around the racetrack.



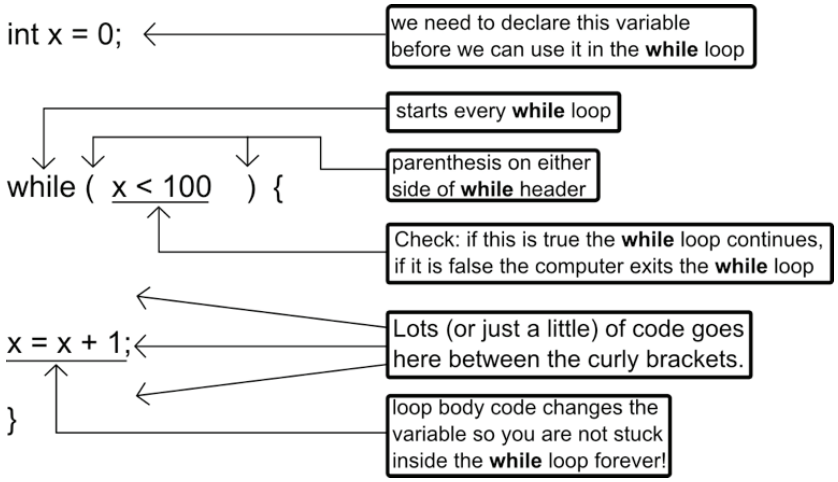
Name:
Date:

// Vocabulary: while, loop ( )

So, you just learned about loop( ), which is the simplest form of repetition, but there are many other forms of repetition in Arduino. Another very common form of repetition is the while loop. A while loop is used when you want the computer or Arduino to do some code while a statement is true. The while loop is usually found inside of the loop( ) function. The code of a while loop has two parts, the header and the loop body code. The header is the most important part to learn and always has the same structure. The code in the curly brackets below the header can be anything, it just depends on what you want to happen each time the computer goes around your while loop.

The header of a while loop has the word while and a statement inside of parenthesis. The while loop checks to see if the statement inside of the parenthesis is true and will repeat as long as that statement remains true. Pretty simple, right?

While loop example with variable declaration. Explanation of the while loop example.



What happens during the while loop above using our robot racetrack as an example:

At the beginning:	Later on, after 100 laps: