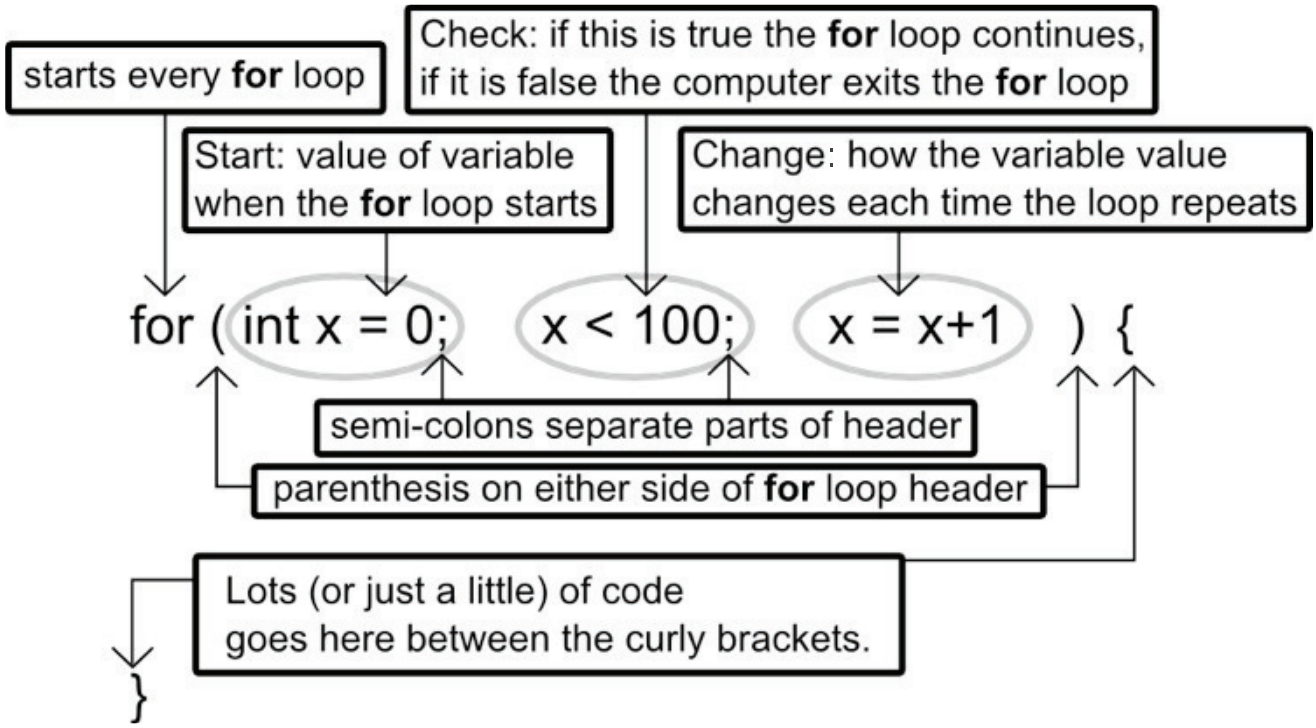
Name:
Date:

// Vocabulary: for, loop ( )

So, you just learned about while, which is a simple form of repetition, but there are many other loop functions. Another very common form of repetition is the for loop. A for loop is used when you want the computer or Arduino to change a variable each time through the loop and do code which often uses that variable. For loops are usually found inside of the loop( ) function. The code of a for loop has two parts, the header and the code inside the loop. The header is the most important part to learn and always looks about the same. The loop body 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 for loop.

The header of a for loop has the word for and in parenthesis three parts called start, check and change. Each of these parts have semicolons between them so you can tell them apart. These three parts (circled in gray below) are the most important parts to understand, they are the three simple parts you need to make a for loop work.



Start:
The first circled part is start, this happens before anything else, it's sort of like putting on running shoes before starting to run around the track. It is a simple declaration and assignment of a variable, in this case the variable is an integer named x.

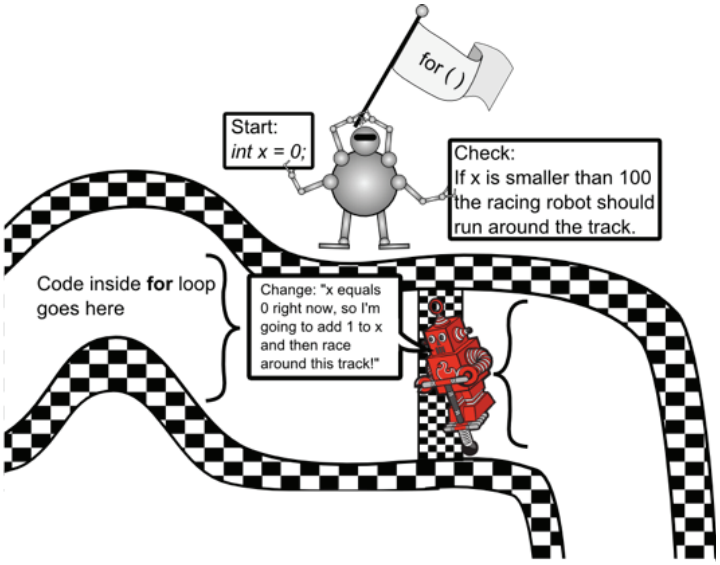
Check:
The second circled part is check. Every time the computer gets to the end of the for loop the computer will check to see if this part is true. The first time the for loop above checks, x is equal to zero, so the for loop continues, does change and then the code inside the curly brackets. It's kind of like checking how many laps a racer has completed to see if the racer has finished the race.

Change:
The third circled part is change, after the variable is checked it changes so that it is closer to making the check statement false so the for loop stops. For a racer this part of the for loop is like adding to (or updating) the number of laps or miles the racer has completed so far in the race.

Name:
Date:

// Vocabulary: for, loop ( )

Here is an example of what happens when the for loop on the previous page begins using our robot racetrack as an example:



The next time the racing robot makes its way around the track to the starting line it has to check again. It doesn't have to start again, but it does need to check to see if the race is over. The first time around the track, x will equal one and the check that x is smaller than 100 is still true. The robot changes the variable by adding one to x again (x now equals two) and then the robot runs around the track executing the loop body code between the curly brackets. The robot will continue to run around the racetrack until x equals 100 at which point the computer exits the for loop.