Just like IF statements for and while loops can be nested within each other. In the example below a FOR loop is nested inside of another FOR loop.

```
#!/usr/bin/env bash
# File: nestedloops.sh

for number in {1..3}
do
  for letter in a b
  do
    echo "number is $number, letter is $letter"
  done
done

```

Based on what we know about FOR loops try to predict what this program will print out before we run the program. Now that you've written down or typed out your prediction let's run it.

```
bash nestedloops.sh

## number is 1, letter is a
## number is 1, letter is b
## number is 2, letter is a
## number is 2, letter is b
## number is 3, letter is a
## number is 3, letter is b

```

Let's closely examine what's going on here. The outer most FOR loop starts iterating through the sequence generated by {1..3}. On the first pass through the loop, the inner loop iterates through the sequence a b which first prints number is 1, letter is a followed by number is 1, letter is b. The first iteration of the outer loop is then finished and the whole process starts over with number having a value of 2. This process continues going through the inner loop until the sequence for the outer loop is exhausted. I again strongly encourage you to pause for a moment and write some of your own nested loops based on the code above. Try to predict what your nested loop program will print before you run your program. If the printed result does not match your prediction trace your way through the program and try to figure out why. Don't just limit yourself to nested FOR loops, use nested WHILE loops, or FOR and WHILE loops in nested combinations.

Besides nesting loops within each other you can also nest loops within IF statements and IF statements within loops. Let's take a look at an example:

```
#!/usr/bin/env bash
# File: ifloop.sh

for number in {1..10}
do
  if [[ $number -lt 3 ]] || [[ $number -gt 8 ]]
  then
    echo $number
  fi
done

```

Before we run this example try once more to guess what the output will be.

```
bash ifloop.sh

## 1
## 2
## 9
## 10

```

For each iteration of the loop above, the value of number was checked in the IF statement, and the echo command was only run if number was outside the range from 3 to 8.

There are endless combinations for nesting IF statements and loops, but one good rule of thumb you should remember is that your nesting should never go more than two or possibly three layers deep. If you find yourself writing code with lots of nesting, you should consider restructuring your program. Deeply nested code is difficult to read and even more difficult to debug if your program contains mistakes.

Summary

- Loops allows you repeat sections of your program.
- FOR loops iterate through a sequence so that a variable that you assign takes on the value of every element of the sequence in every iteration of the loop.
- WHILE loops check a conditional statement at the beginning of every iteration. If the condition is equivalent to true then one iteration of the loop is executed and then the conditional statement is checked again. Otherwise the loop ends.
- IF statements and loops can be nested in order to make more powerful programming structures.

Mark as completed