Now that we've gotten a few FOR loops working let's move on to WHILE loops. The **WHILE loop** is truly the Reese's Peanut Butter Cup of programming structures, combining parts of the FOR loop and the IF statement. Let's take a look at an example WHILE loop so you can see what I mean:

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

count=3

while [[ $count -gt 0 ]]
do
  echo "count is equal to $count"
  let count=$count-1
done

```

The WHILE loop begins first with the while keyword followed by a conditional expression. As long as the conditional expression is equivalent to true when an iteration of the loop begins, then the code within the WHILE loop will continue to be executed. Based on the code for whileloop.sh what do you think will be printed to the console when we run this script? Let's find out:

```
bash whileloop.sh

## count is equal to 3
## count is equal to 2
## count is equal to 1

```

Before the WHILE the count variable is set to be 3, but then each time the WHILE loop is executed 1 is subtracted from the value of count. The loop then starts from the top again and the conditional expression is re-checked to see if it's still equivalent to true. After three iterations through the loop count is equal to 0 since 1 is subtracted from count in every iteration. Therefore the logical expression [[$count -gt 0]]is no longer equal to true and the loop ends. By changing the value of the variable in the logical expression inside of the loop we're able to ensure that the logical expression will eventually be equivalent to false, and therefore the loop will eventually end.

If the logical expression is never equivalent to false then we've created an *infinite loop*, so the loop never ends and the program runs forever. Obviously we would like for our programs to end eventually, and therefore creating infinite loops is undesirable. However let's create an infinite loop so we know what to do if we get into a situation where our program won't terminate. With a simple "typo" we can change the program above so that it runs forever but substituting the minus sign - with a plus sign + so that count is always greater than zero (and growing) after every iteration.

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

count=3

while [[ $count -gt 0 ]]
do
  echo "count is equal to $count"
  let count=$count+1              # We only changed this line!
done

## ...
## count is equal to 29026
## count is equal to 29027
## count is equal to 29028
## count is equal to 29029
## count is equal to 29030
## ...

```

If the program is working, then count is being incremented very rapidly and you're watching number wiz by in your terminal! Don't fret, you can terminate any program that's stuck in an infinite loop using Control + C. Use Control + C to get the prompt back so that we can continue.

When constructing WHILE loops, make absolutely sure that you've structured the program so that the loop will terminate! If the logical expression after while never becomes false then the program will run forever, which is probably not the kind of behavior you were planning for your program.

Mark as completed