Python

Because iteration is so common, Python provides several language features to make it easier. One is the *while* statement. Here is a version of *countdown* that uses a *while* statement:

```
def countdown(n):
    while n > 0:
        print n
        n = n-
    print 'Blastoff!'
```

You can almost read the *while* statement as if it were English. It means, "While n is greater than 0, display the value of n and then reduce the value of n by 1. When you get to 0, display the word Blastoff!"

The body of the loop should change the value of one or more variables so that eventually the condition becomes false and the loop terminates. Otherwise, the loop will repeat forever, which is called an *infinite loop*.

In the case of countdown, we can prove that the loop terminates because we know that the value of n is finite, and we can see that the value of n gets smaller each time through the loop, so eventually we have to get to 0. In other cases, it is not so easy to tell:

```
def sequence(n):
    while n != 1:
        print n,
    if n%2 == 0: # n is even
        n = n/2
    else: # n is odd
        n = n*3+1
```

The condition for this loop is n != 1 , so the loop will continue until n is 1, which makes the condition false.

Each time through the loop, the program outputs the value of n and then checks whether it is even or odd. If it is even, n is divided by 2. If it is odd, the value of n is replaced with n*3+1. For example, if the argument passed to sequence is 3, the resulting sequence is 3, 10, 5, 16, 8, 4, 2, 1.

Since n sometimes increases and sometimes decreases, there is no obvious proof that n will ever reach 1, or that the program terminates. For some particular values of n, we can prove termination. For example, if the starting value is a power of two, then the value of n will be even each time through the loop until it reaches 1. The previous example ends with such a sequence, starting with 16.

Example 1

```
iVal = 1
iMax = 12

while iVal < iMax:
    iVal = iVal + 3
print("Answer " + str(iVal))
```

Example 2

```
1 = 1
i2 = 0

while i1 < 8:
    while i2 < 6:
        print("Even Numbers " + str(i2))
        i2 += 2

        print("Odd Numbers " + str(i1))
        i1 += 2
print("Finished…")
```

Example 3

```
i = 5
print("Loop: ")
while i > 3:
    i -= 1
print("Loop finshed…" + str(i))
```