Loops

For vs. While Loops

for loops have a pre-specified range over which they run. They run for each thing in some set of things. e.g. for each number in a range of numbers, for each letter in a word

while loops have a condition that they check to determine if they should keep running. They run while some condition is true. e.g. while some total we are counting is less than some threshold, while a player has not yet lost the game

for loops are usually preferable when possible

for loops cannot run infinitely, since the set of things it runs over is known beforehand

while loops can run infinitely, if their condition never becomes false

Loops can be put in other loops

Syntax

See Code

Converting Loop Types

All for loops can be written as while loops
Not all while loops can be written as for loops
(e.g. end of loop is dependent on user input)

for loops are usually preferable when they can be used

See code for examples

range vs. xrange

range puts all the numbers in the range into memory, then selects them one at a time

xrange makes the numbers in the range one at a time. This is much more time and memory efficient
**range/xrange** take 1, 2, or 3 arguments:
stop - the number the iteration stops at, NOT INCLUDED
start - the number the iteration starts at, INCLUDED
step - the number that the value increases by each iteration (e.g. a step of 2 could be used to count just even numbers)
1 argument: range(stop) # Default start is 0, step is 1
2 arguments: range(start,stop) # Default step is 1
3 arguments: range(start,stop,step)

**break and continue**

**break** terminates the innermost loop completely. The program jumps to the code immediately after the loop.

**continue** ends only the current iteration of the loop. The program jumps to the top of the loop and continues with the next iteration of the loop.
- a **for loop** moves on to the next item in its list of things to iterate through
- a **while loop** checks its condition again and returns either to the top of the loop if it is True or to the code immediately after the loop if it is False

**Guess and Check Approximation**

Square Root Approximator
See Code

**Nested Loops & Conditionals**

Multiplication Tables
See Code

Greatest Common Denominator
See Code