**Python for Everyone: Notes**

**Chapter 5: Loop and iteration**

* Breaking out of a loop
  + The break statement ends the current loop and jumps to the statement immediately following the loop
  + It is like a loop test that can happen anywhere in the body of the loop
* Finishing an iteration with continue
  + The continue statement ends the current iteration and jumps to the top of the loop and starts the next iteration
* Indefinite loops
  + While loops are called indefinite loops because they keep going until a logical condition becomes false
  + The loops we have seen so far are pretty easy to examine to see if they will terminate or if they will be infinite loops
  + Sometimes it is little harder to be sure if a loop will terminate
* Definite loops
  + Iterating over a set of items
  + Often we have a list of items of the lines in a file – effectively a finite set of things
  + We can write a loop to run the loop once for each of the items in a set using the python for construct
  + These loops are called definite loops because they execute an excact number of times
  + We say that definite loops iterate through the members of a set
* A simple definite loop
  + Definite loops (for loops) have explicit iteration variables that change each time through the loop
  + These iteration variables move through the sequence or set
* Looking at in
  + The iteration variable iterates through the sequence
  + The block (body) of code is executed once for each value in the sequence
  + The iteration variable moves through the values in the sequence
* Loop idioms: What we do in loops
* Making smart loops
  + The trick is knowing something about the whole loop when you are stuck writing code that only sees one entry at a time
* Finding the largest variable
  + We make a variable that contains the largest value we have seen so far. If the current number we are looking at is larger, it is the new largest number we have seen so far
* Counting in a loop
  + To count how many times we execute a loop, we introduce a counter variable that starts at 0 and we add one to each time through the loop
* Summing in a loop
  + To add up a value we encounter in a loop, we introduce a sum variable that starts at 0 and we add the value to the sum each time through the loop
* Finding the average in a loop
  + An average just combines the counting and sum patterns and divides when the loop is done
* Filtering in a loop
  + We use an if statement in the loop to catch / filter the values we are looking for
* Search using a Boolean variable
  + If we just want to search and know if a value was found, we use a variable that starts at false and set it to true as soon as we find what we are looking for
* Finding the smallest value
  + We still have a variable that is the smallest so far. The first time through the loop smallest is None, so we take the first value to be the smallest
* The is an is not operators
  + Python has an is operator that can be used in logical expressions
  + Implies Is the same as
  + Similar to but stronger than ==
  + Is not also is a logical operator