Iterators and List Comprehensions

Iterator is an object which allows a programmer to traverse through all the elements of a collection, regardless of its specific implementation.

In Python 3, range is not a list, but called an iterator, which provides the functionality required by the for loop. The iter object is a container that gives you access to the next object for as long as it's valid. Python’s itertools library contains a count function that acts as an infinite range.

Enumerate iterate not only the values in an array, but also keep track of the index. You might be tempted to do things this way. Zip iterator zips together iterables with multiple lists. The map iterator takes a function and applies it to the values in an iterator. The filter iterator looks similar to map, except it only passes-through values for which the filter function evaluates to True.

The itertools module contains a whole host of useful iterators.

List comprehensions provide a concise way to create lists. It consists of brackets containing an expression followed by a for clause, then zero or more for or if clauses.