EdX 6.00x Notes

# Lecture 12:

* How plist.sort() works
  + Python uses the timsort algorithm for sorting sequences
    - Timsort – a highly-optimized combination of merge and insertion sorts that has a very good average case performance
    - The only knowledge needed about the objects being sorted is the result of a “less than” comparison between two objects
    - Python interpreter translates obj1 < obj2 into a method call on obj1 -> obj1.\_\_lt\_\_(obj2)
    - To enable sort operations on instances of a class, implement the \_\_lt\_\_ special method
* Inheritance:
  + When you allow a class to have access to all the characteristics of the superclass
* Substitution principle:
  + Important behaviors of superclass should be supported by all subclasses
* Note:
  + Be careful not to violate the data hiding aspect of an object, and exposing the internal representation.
  + Always try to separate collection of data from use of data.
* Generators:
  + Any procedure or method (procedure that belongs to a class) with a **yield** statement is called a **generator**
  + Generators have a next() method which starts/resumes execution of the procedure. Inside of generator:
    - **Yield** suspends execution and returns a value
    - Returning from a generator raises a **StopIteration** exception
* Why generators?
  + A generator separates the concept of computing a very long sequence of objects, from the actual process of computing them explicitly
  + Allow one to generate each new objects as needed as part of another computation (rather than computing a very long sequence, only to throw most of it away while you do something on an element, then repeating the process)