concurrent.futures module

The **concurrent.futures** module was added in Python 3.2. According to the Python documentation it *provides the developer with a high-level interface for asynchronously executing callables.* Basically concurrent.futures is an abstraction layer on top of Python's threading and multiprocessing modules that simplifies using them. However it should be noted that while the abstraction layer simplifies the usage of these modules, it also removes a lot of their flexibility, so if you need to do something custom, then this might not be the best module for you.

Concurrent.futures includes an abstract class called **Executor**. It cannot be used directly though, so you will need to use one of its two subclasses: **ThreadPoolExecutor** or **ProcessPoolExecutor**. As you've probably guessed, these two subclasses are mapped to Python's threading and multiprocessing APIs respectively. Both of these subclasses will provide a pool that you can put threads or processes into.

The term **future** has a special meaning in computer science. It refers to a construct that can be used for synchronization when using concurrent programming techniques. The **future** is actually a way to describe the result of a process or thread before it has finished processing. I like to think of them as a pending result.