《Core Python Applications Programming》 读书笔记

一．网络编程：

1. 1 socket 层将传输层上的TCP， UDP协议抽象，为应用层提供一个简单的接口；是客户端和服务器端的通信端点，网络应用程序必须创建套接字，否则无法进行通信

1.2 基于文件的套接字：主机上的一个程序与另一个程序进行通信，即进程间通信

（这些套接字是基于文件的，这意味着文件系统支持他们的底层基 础结构。 文件系统是一个运行在同一主机上的多个进程之前的共享常量）

面向网络的套接字：TCP，UDP

2.在一台计算机内，使用PID确定一个进程，而在互联网中，使用IP地址+协议+端口号来标示网络中的一个进程

1. 客户端/服务器架构：

硬件客户端/服务器架构，文件服务器就是一个例子，庞大存储容量的计算机允许客户端远程访问，客户端计算机会挂载服务器计算机上的磁盘，看起来磁盘好像就在本地计算机上一样

软件客户端/服务器架构，Web服务器，在一台或者多台计算机上安装希望提供给客户的Web页面和Web应用；数据库服务器

1. 多线程编程

适用于编程任务本质是异步的（执行的先后顺序没有影响），需要多个并发活动，每个活动的处理顺序是不确定的

Python支持多线程编程，但是需要取决于它所运行的操作系统

守护线程，一个等待客户端请求服务的服务器，如果没有客户端请求，守护线程就是空闲的，进程退出是不需要等待这个线程的执行完成

1. Web开发
2. Web浏览器是基本的web客户端，主要用于在web上查询或者下载文档，但还可以常见一些具有其他功能的web客户端，比如web爬虫，它可以为Google等大型搜索引擎创建索引，缓存web页面，节省再次访问web站点的下载时间
3. Web服务器无法创建动态内容， CGI（Common Gateway Interface)进程针对每个web请求进行创建，用完就抛弃。 创建大量的语言解释器很快就会让服务器停机，因此采用服务器集成，或者外部进程的方法。

WSGI（Web Server Gateway Interface)只是定义的一个几口，起目标是在web服务器和Web应用框架层之间提供一个通用的API标准

3.web框架之django，下载好之后有一个自带的开发用服务器，但是只能作为测试使用

、