1. Web服务器主机分为硬件和软件，软件指Web服务器应用，其上还有各种辅助应用
2. Web服务器不做的两件事：
   1. 动态内容
      1. Web服务器只提供**静态Web页面**，静态页面只是原封不动地待在目录中，服务器找到静态页面，并把它原样传给客户，每个客户看到的东西都是一样的！
      2. **动态网页**的动态指的是能与用户进行交互，比如登录时输入正确的用户名和密码，系统会提示登录成功。（这里动态的概念就是Java中反射动态的概念，动态创建一个在发出请求之前还不存在的页面）
   2. 在服务器上保存数据
      1. Web服务器处理动态页面需要服务器上另一个**辅助应用**来处理，然后返回静态响应给Web服务器应用，Web服务器应用再把响应传给客户端浏览器
      2. 用户提交表单数据时，Web服务器**通过辅助应用处理这个表单数据**，把参数移交给那个辅助应用，由这个应用来为客户生成一个响应
         1. 保存到一个文件或数据库中
         2. 用来生成响应页面
3. Web服务器辅助应用就是CGI程序，Servlet和CGI在Web服务器中都扮演者辅助应用的角色
   1. CGI（Common Gateway Interface）即公共网关接口
4. Servlet没有main方法，他们受限于另一个应用，这个应用被称为容器（Tomcat也是一个容器）
5. Web服务器应用得到一个指向Servlet的请求时，服务器将该请求交给部署该Servlet的容器，要由容器向Servlet提供HTTP请求和响应，而且由容器调用Servlet的方法（doGet()、doPost()等）
6. 使用部署描述文件(DD)把URL映射到servlet（在tomcat中是web.xml）
7. 容器能提供什么？
   1. 通信支持
      1. 建立ServerSocket、创建流
      2. 容器知道自己与Web服务器之间的协议，所以servlet不必担心Web服务器和Web代码之间的API，你只需考虑如何在servlet中实现业务逻辑
   2. 生命周期管理
      1. 容器负责加载类、实例化和初始化servlet、调用servlet方法，以及使servlet实例能够被垃圾回收
   3. 多线程支持
      1. 容器自动地为它接受的每个servlet请求创建一个新的Java线程。如果servlet已经完成相应的HTTP服务方法，这个线程就会结束
      2. 然而，还是会遇到同步问题的！线程可能不安全
   4. 声明方式实现安全
      1. 利用容器，使用XML部署描述文件来配置安全性，而不必硬编码写到servlet类代码中
   5. JSP支持
      1. 容器负责把JSP翻译成servlet