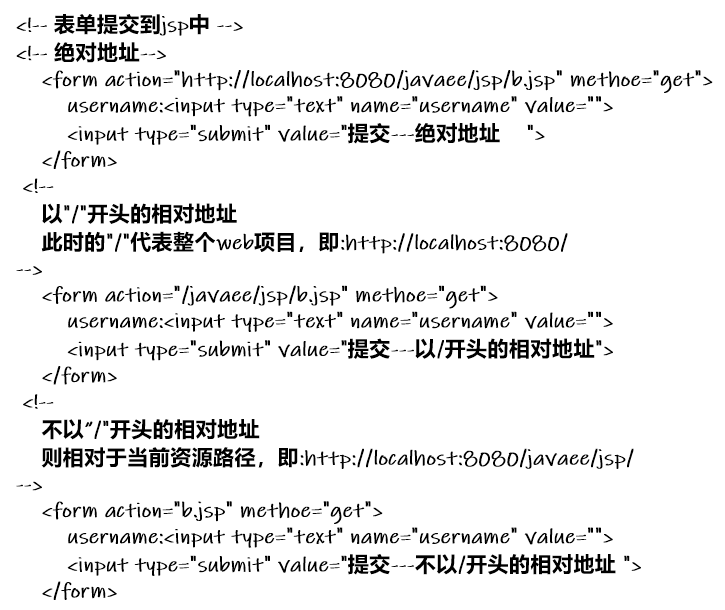
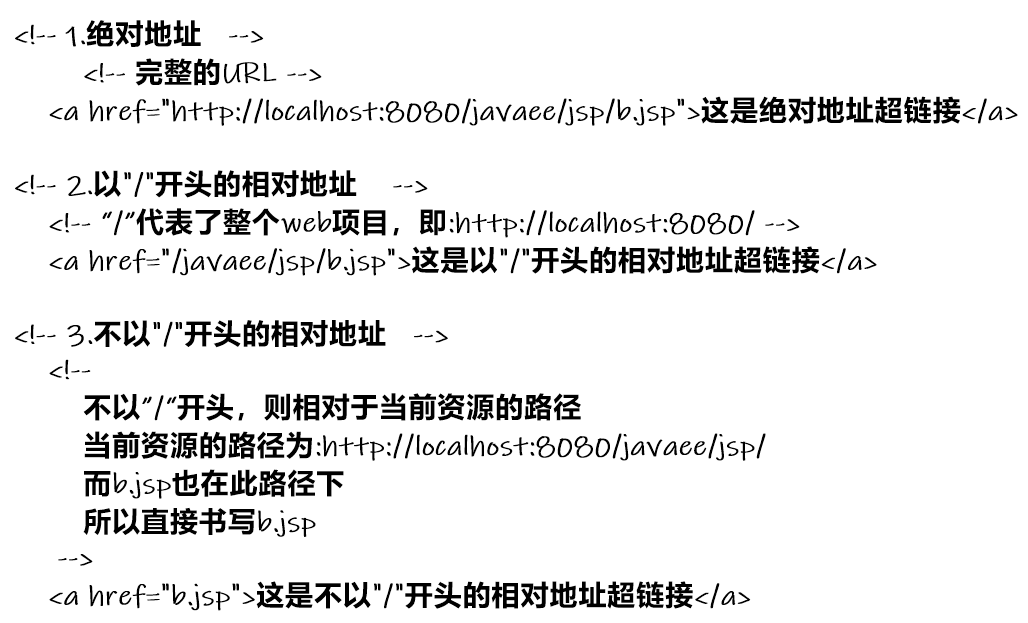
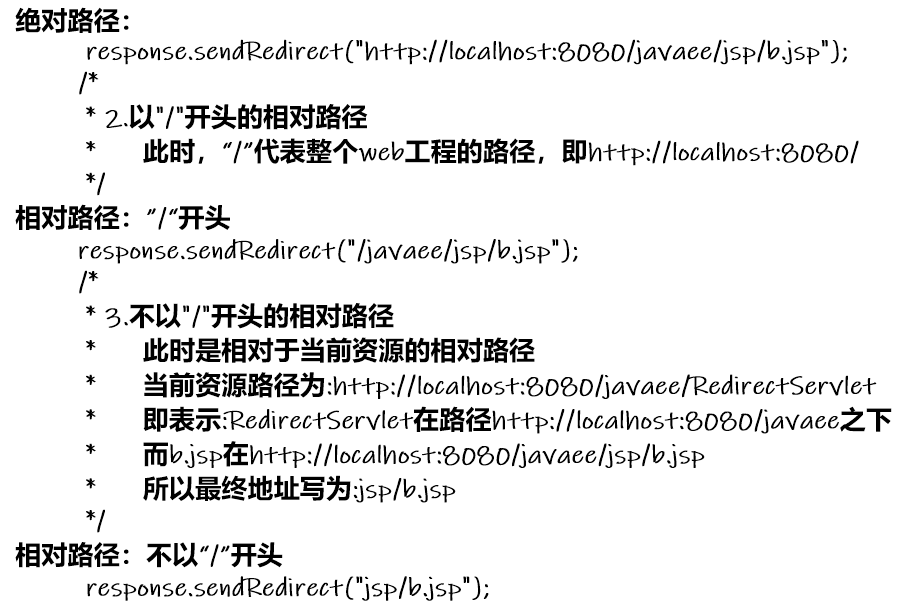
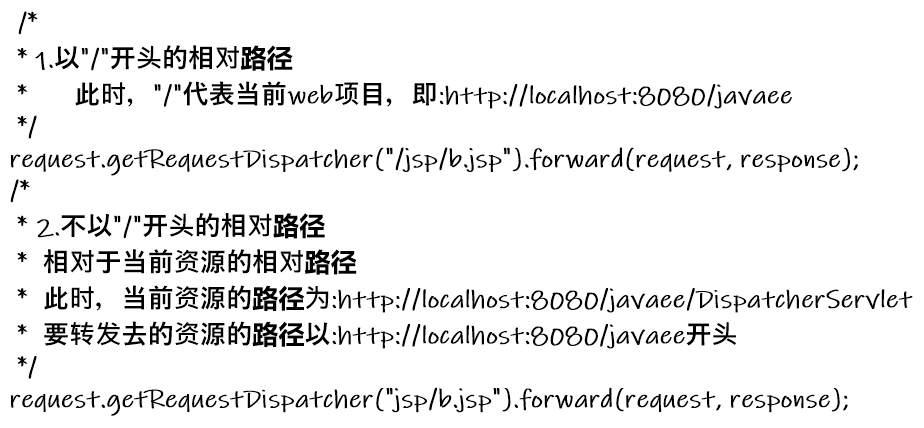


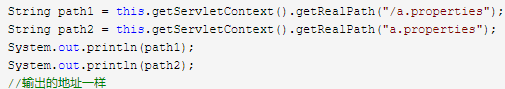
客户端路径：（路径分为绝对与相对， 相对又分为相对主机与相对当前请求）

1. 表单
   1. 三种书写路径的方式：
      1. 绝对路径：
      2. 以“/“开头的相对路径
      3. 不以“/“开头的相对路径
      4. 
      5. 
2. 超链接
   1. 三种书写路径的方式：
      1. 绝对路径
      2. 以“/“开头的相对路径
      3. 不以“/“开头的相对路径
      4. 
      5. 分析：
         1. 绝对路径（以协议开头）
         2. 相对路径
            1. 相对主机的路径，以“/“开头：相对主机，可以简单理解为ip地址
            2. 相对与请求路径，不以“/“开头：相对于当前请求，浏览器的请求的路径
3. Js（location.href）
4. Ajax（url）
5. Css与js的引入
6. 重定向
   1. 三种书写路径的方式：这里一般只用相对路径
      1. 绝对路径
      2. 以“/“开头的相对路径
      3. 不以“/“开头的相对路径
      4. 

服务端路径：指在服务器上运行的请求，比如请求转发，请求包含

1. 服务端路径有俩种：
   1. 相对于当前应用的路径
   2. 相对于当前请求的路径
2. 请求转发：
   1. 俩种书写路径方式
      1. 以"/"开头的相对路径
      2. 不以"/"开头的相对路径
      3. 注意：服务器端路径不能时绝对路径，只能时相对路径
      4. 
3. 请求包含：
   1. 和上面的请求转发相同

资源获取路径：

1. 主要分三种：
   1. ServletContext
      1. Web阶段，Tomcat提供的一种获取资源的方式
      2. 获取资源的路径时相对系统的绝对路径，在win中带盘符的可以用来上传或下载的具体路径
      3. 基本语法：servletContext.getRealPath("路径");
      4. 注意：参数中的路径必须时相对路径，建议以”/“开头，无论是否使用“/”都是相对当前应用路径
      5. 获取方式：
         1. 使用request获取： request.getSession().getServletContext();
         2. 在Servlet中获取：this.getServletContext()；
         3. 用FilterConfig对象获取（在Filter中使用）：config.getServletContext();
      6. 
   2. Class
      1. 用法： clazz.getResourceAsStream("路径")；
   3. ClassLoader
      1. 上俩者时JavaAPI提供的获取流的方式
      2. 用法：classLoader.getResourceAsStream("路径")；
      3. 参数可以以“/“开头，无论带不带，都表示相对于当前类的路径
      4. 注意：不能通过加载绝对路径获取资源，只能通过相对地址来加载资源

<url-pattern>路径

1. 虚拟路径，只有相对路径，相对于当前应用路径