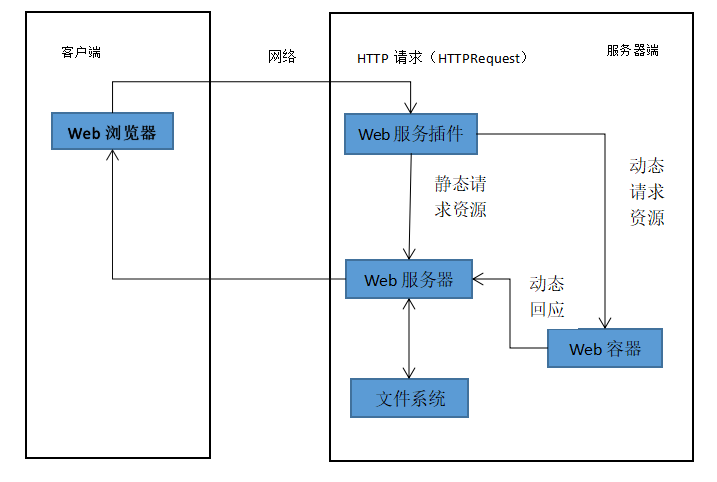
# Web容器简介

## 容器的作用

要想运行一个Java Web程序，则必须要有相应的Web容器（Web container）的支持。因为所有的动态页面的程序代码都要在Web容器中运行，最后将处理的结果交付给用户使用。



客户端通过Web浏览器发送一个基于HTTP协议的请求到服务器上后，服务器端使用Web服务插件（Web Server Plugin）接收客户端的请求，并对接收的用户请求进行判断，判断是动态请求还是静态请求。如果是静态请求，则直接通过Web服务器（Web Server）从文件系统中取出对应的文件，并通过HTTP协议返回到客户端进行显示。如果是动态请求，则将内容提交到Web容器中，在Web容器中由程序动态的生成显示的结果，最后通过Web服务器利用HTTP协议返回到客户端。

## 静态请求与动态请求

Web服务插件可以用于区分请求是静态请求还是动态请求，从文件的后缀名上看，html、htm 之类的后缀往往都属于静态请求，jsp、asp之类的后缀往往是动态请求。但这只是从表名上看，最好的解释是：静态请求的所有代码都是固定好的，而动态请求的所有代码都是拼凑而成的。

# Tomcat简介

## 认识Tomcat

Tomcat是Apache软件基金会的一个项目，可以从Apache官网上下载因为Tomcat技术先进、性能稳定、而且免费，因而深受Java爱好者的喜爱，并且得到了部分软件开发商的认可，现在已经成为了目前比较流行的Web应用服务器（Web容器）。

## 下载与安装Tomcat

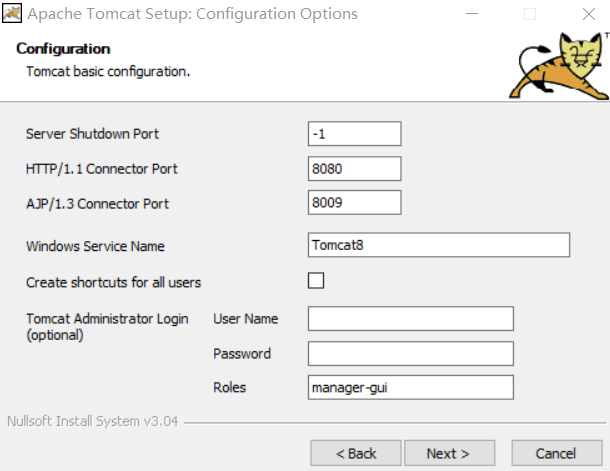
1. **下载**

用户可以直接登录Apache的网站上下载Tomcat。下载网址如下：

<http://tomcat.apache.org/>

1. **安装**

安装Tomcat时必须有JDK的支持，所以需要在本机上先配置JDK的安装环境。假设用户已安装JDK。



在安装过程中，需要选择安装路径，配置JRE路径，设置端口和密码等

## Tomcat文件目录结构



1. bin文件夹：存放了所有的可执行命令，例如启动和关闭服务器的命令就在此文件夹

中。

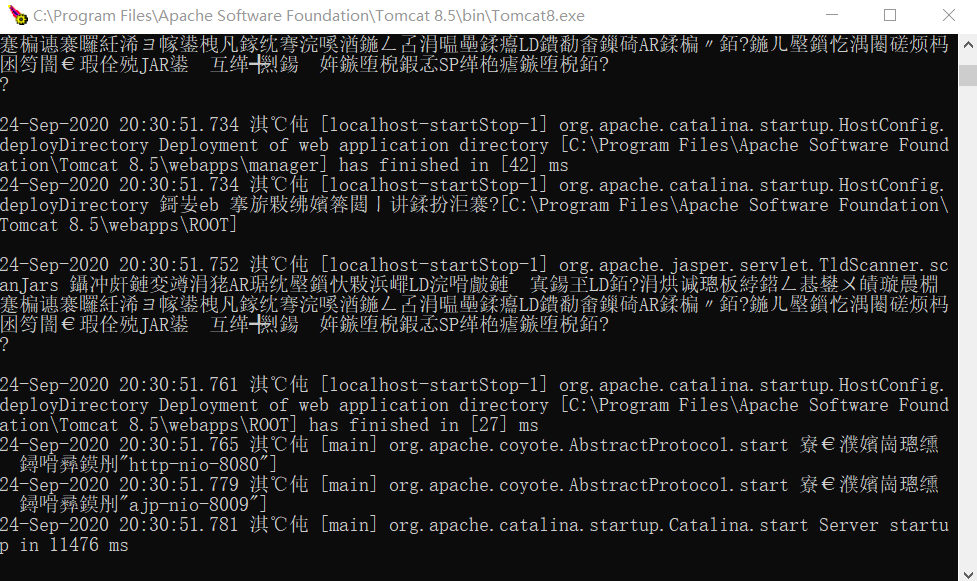
1. conf文件夹：存放了服务器的一些配置文件，如其中的server.xml文件可以修改端

口号。

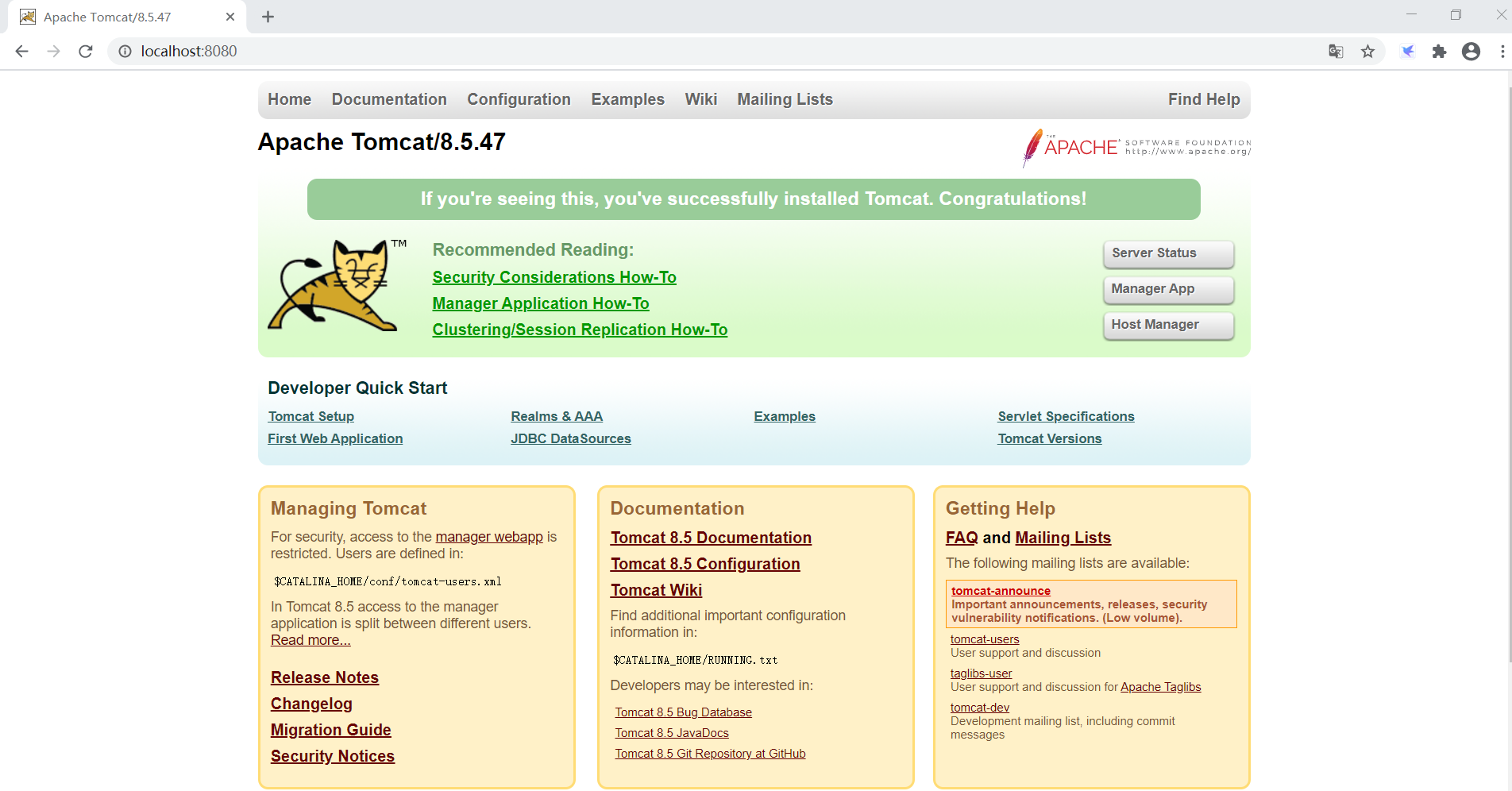
1. lib文件夹：存放了Tomcat服务器所需要的各个库文件。
2. logs文件夹：保存了服务器的系统日志。
3. webapps文件夹：Web应用程序存放的目录，Web项目保存到此目录中即可发布。
4. work文件夹：临时文件夹，生成所有的临时文件，如（\*.java、\*.class）。

## 测试Tomcat

Tomcat服务器安装完成后，即可通过Tomcat中的bin目录下的tomcat8.exe命令启动Tomcat服务器，启动界面如下：



注意启动后不可关闭，打开浏览器，访问<http://localhost:8080>或<http://127.0.0.1:8080>，即可看到如下图所示的界面：

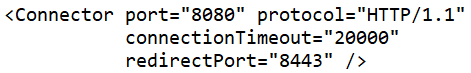


表示Tomcat安装成功。

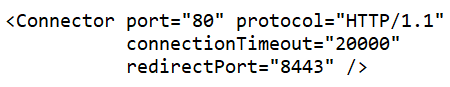
# Tomcat常用配置

## 修改端口号

在Tomcat安装时，默认的端口号8080，如果要想修改端口号，则可以打开Tomcat安装目录中的conf/server.xml文件，找到以下内容：



将port属性的值修改即可。例如，将端口号修改为80端口：

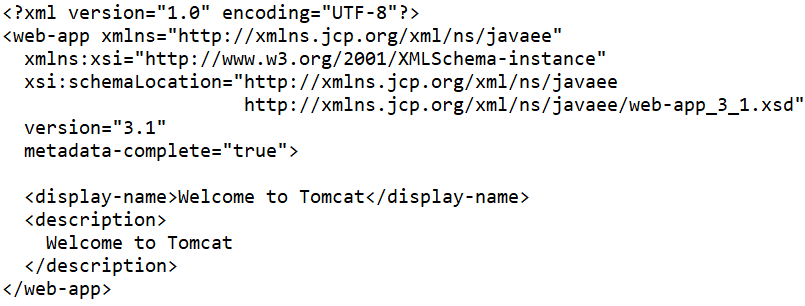


这样，以后就可以直接输入<http://localhost>即可访问，不用再输入端口号。需要特别注意的是，修改配置文件后，必须重启服务器才能生效，因为服务器每次启动时都会加载server.xml文件中的内容。

## 配置虚拟目录

每一个虚拟目录都保存了一个完整的Web项目，配置虚拟目录对于项目的开发及运行维护有很大的帮助。下面演示了如何配置虚拟目录：

1. 在D盘建立一个myweb，并在此文件中建立一个WEB-INF的子文件夹，同时在WEB-INF文件夹中建立web.xml文件，此文件的格式如下：



·web.xml文件是整个Web的核心配置文件，也称为Web的部署描述符。

·以上内容如果手动敲会很麻烦，还容易出错，所以，可以从Tomcat安装目录中

找到，路径为webapps/ROOT/WEB-INF/web.xml。

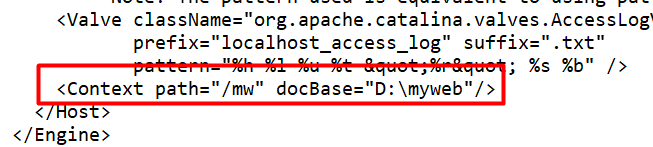
1. 配置完项目目录后，打开Tomcat安装目录conf/server.xml文件，对其进行修改，添加以下内容：

**<Context path=”/mw” docBase=” D:\myweb”/>**

<Context>是一个固定的标签，表示配置虚拟目录，其中两个参数的意义如下：

·path：表示浏览器上的访问虚拟路径名称，前面必须加上“/”

·docBase：表示此虚拟目录名称所代表的真实路径地址

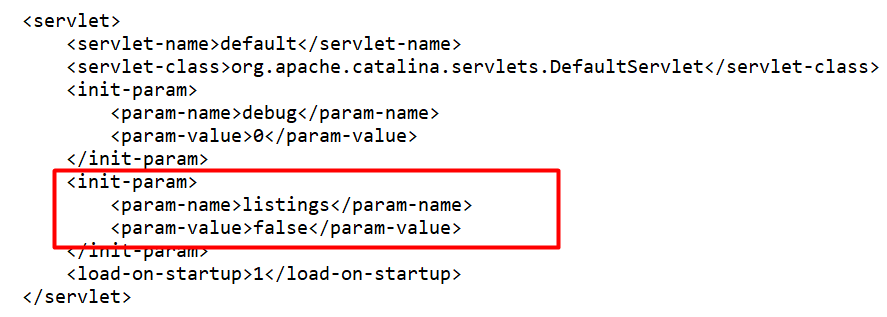


一个Tomcat服务器可以配置多个虚拟目录，但是每一个虚拟目录的path名称不能重复，否则服务器将无法启动。

1. 配置完成后，重新启动服务器使配置生效，在浏览器中输入<http://localhost/mw>，（mw就是之前配置好的虚拟路径的名称），如果看到的是如下页面：

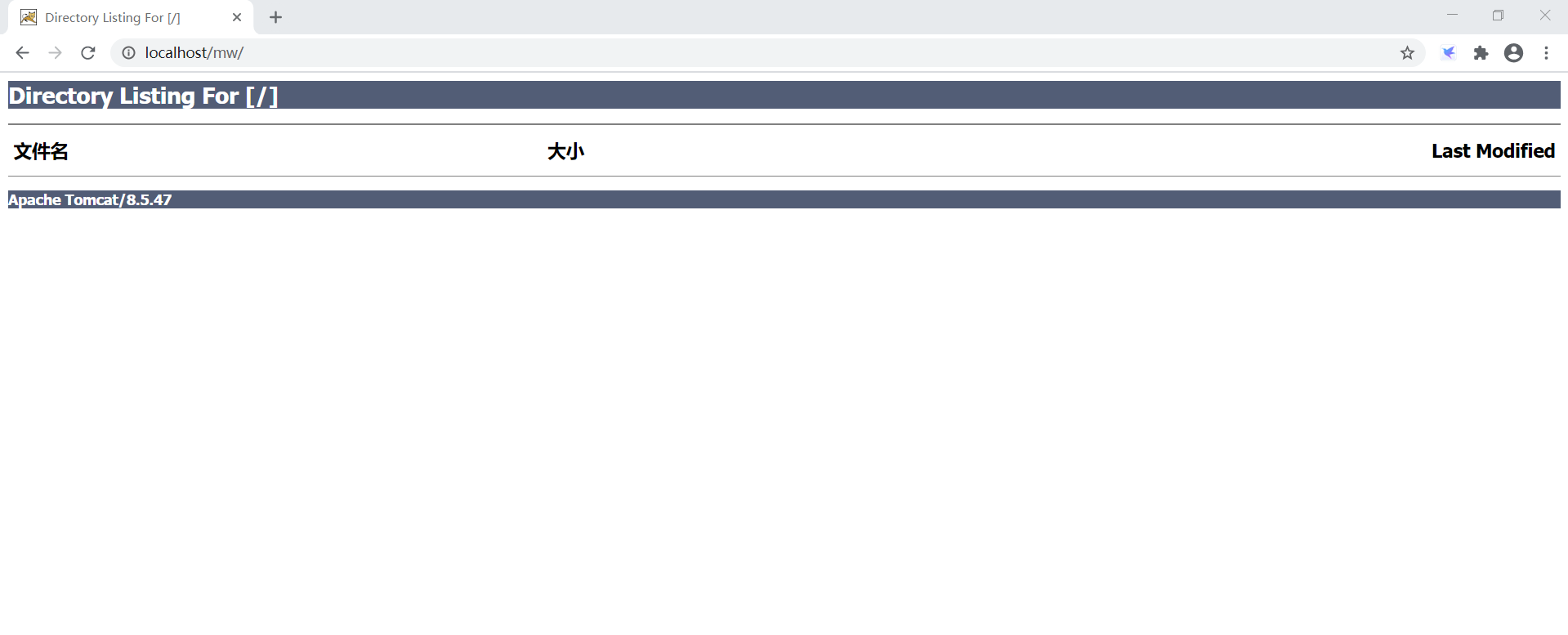


则在conf/web.xml中找到如下内容：



将listings对应的值修改为true，表示显示文件列表页，因为在Tomcat服务器上，默认情况下访问Web项目根目录是不会出现文件列表页的。

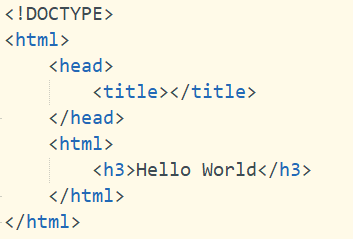
1. 重启服务器，访问<http://localhost/mw>，即可看到如下界面：



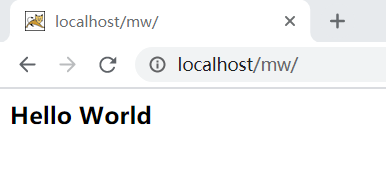
## 配置首页

Tomcat服务器配置完虚拟目录后，可以配置一个Web项目的首页。

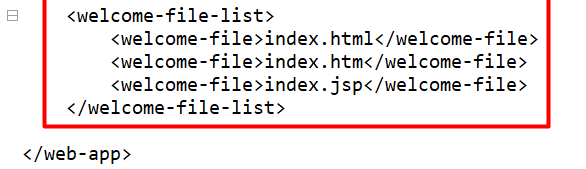
1. 承接上一章节，在配置好的虚拟目录中建立一个index.html文件，文件内容随意，我的如下：



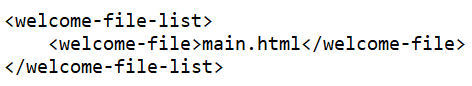
1. 此时再次输入<http://localhost/mw，服务器就会自动将index.html>页面的内容显示给用户，如下：



之所以会有这样的情况主要是在Tomcat的conf/web.xml中默认配置了首页，如下：



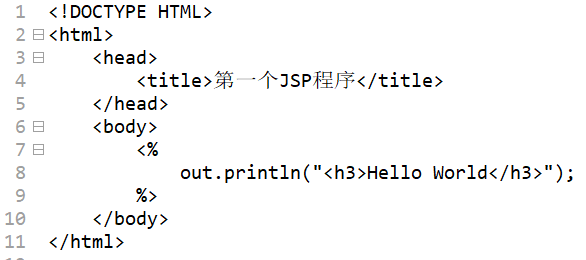
如果想要更换某个项目中首页（因为tomcat中的首页配置是全局的），则可以直接在项目文件夹的WEB-INF/web.xml文件添加或修改以上代码即可。例如，将main.html作为首页：



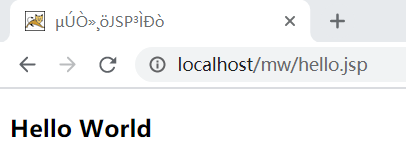
# 编写第一个JSP文件

## 第一个JSP程序

服务器配置完成后，编写第一个JSP程序，输出“Hello World”。JSP文件的后缀名是\*.jsp。在项目目录建立hello.jsp，内容如下：



启动服务器，通过浏览器输入地址<http://localhost/mw/hello.jsp>访问，程序的运行结果如下：



1. 从以上程序中可以发现，该JSP程序主要还是由HTML代码组成的，只是在HTML代码中加入了Java代码。所以，所谓的JSP程序代码开发就是指在HTML中嵌入大量的Java代码而已。
2. 如果在HTML代码中加入的Java代码出现了语法错误，如某条Java语句少了一条分号，则页面会500的HTTP状态码。

## JSP文件执行流程

JSP程序在第一个次执行的速度是很慢的，而之后的执行速度就变得非常快，这是由于JSP在第一次执行时会首先将\*.jsp文件翻译成\*.java文件，然后再将\*.java编译成\*.class文件后才执行的。

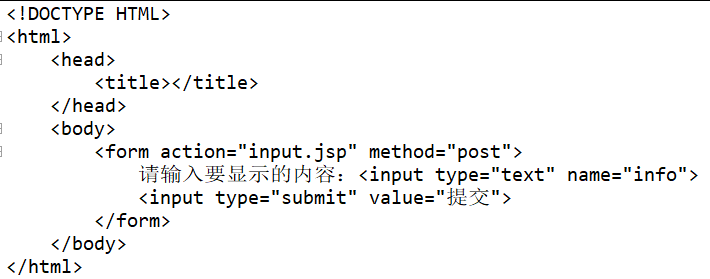
当每次修改\*.jsp文件后，所有的\*.java文件也会重新生成，每次修改对于程序来将都相当于是一个新的程序重新执行一样，所以第一次的执行速度依然后很慢。

在Tomcat的安装目录中，可以看到有一个work的文件夹，所有生成的\*.java及\*.class文件都会保存在此文件夹中。如果想要清理掉这些临时文件，可以删除work目录中的全部内容，这样在每次运行时就会重新生成新的文件。

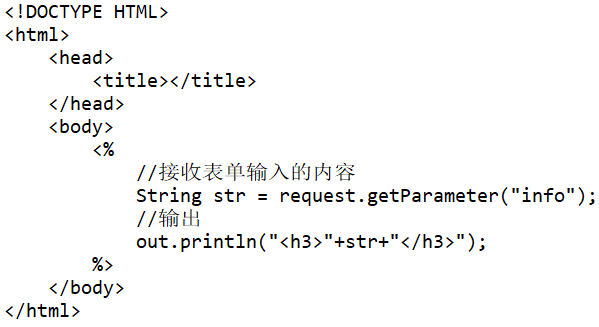
# 交互性

在动态Web中最大的特点就是交互性，所谓的交互性就是指，在服务器端可以接收前页面输入的内容并进行显示，而要输入内容就必须通过表单的支持。示例如下：

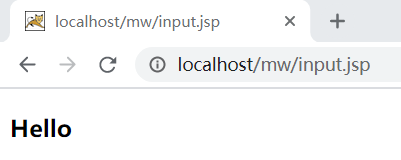
1. 编写input.html文件：



1. 接收输入表单参数并显示，input.jsp：



1. 访问input.html，输入内容，点击显示后，运行结果如下：



从以上简单的交互程序可以看出，在JSP中，服务端的操作要想取得客户端的输入信息，则需要使用request.getParameter(“info”)操作，其中，request就是一个JSP提供的内置对象，而getParameter( )方法接收的参数就是表单中对应的文本框的名称，所以此示例中编写的就是info。

# HTTP状态码

在Web开发中，经常会碰见HTTP状态码，通过HTTP状态码能够快速定位出错误的类型。常用的HTTP状态码如下

|  |  |  |
| --- | --- | --- |
| **状态码** | **表示的含义** | |
| 2xx | 请求成功 | |
| 3xx | 重定向 | |
| 4xx | 客户端出现错误 | |
| 404 | 服务器找不到指定的资源，文档不存在 |
| 5xx | 服务器端出现错误 | |
| 500 | 服务器内部错误 |