



# 软件开发环境

Software Development Environment

主讲教师 刘凡  
[fanliu@hhu.edu.cn](mailto:fanliu@hhu.edu.cn)



# 第三章

## JSP简介



- ▲ 什么是JSP
- ▲ 安装配置JSP运行环境
- ▲ JSP页面
- ▲ JSP运行原理
- ▲ JSP与Servlet的关系
- ▲ 小结



# 什么是JSP

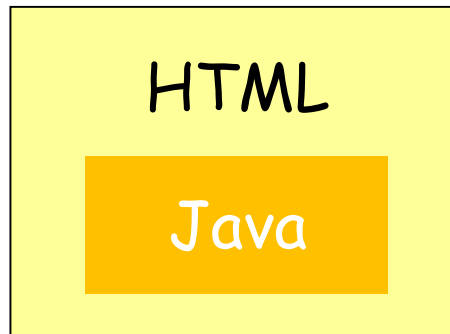
👉 **JSP** is an alternate way of creating servlets.

**JSP** is written as ordinary **HTML**, with a little **Java** mixed in.

👉 **JSP** files must have the extension **.jsp**.

**JSP** is translated into a **Java** servlet, which is then compiled.

**Servlets** are run in the usual way.





# 什么是JSP

---

<HTML>

<BODY>

Hello! The time is now **<%= new java.util.Date() %>**

</BODY>

</HTML>

## Notes:

The **<%= ... %>** tag is used, because we are computing a *value* and inserting it into the HTML

The fully qualified name (**java.util.Date**) is used, instead of the short name (**Date**), because we haven't yet talked about how to do **import** declarations



## 安装配置JSP运行环境



一个服务器上可以有多个基于JSP的Web应用程序，以满足各种用户的需求。这些Web应用程序必须有一个**软件**来统一管理和运行，这样的软件被称作**JSP引擎**或**JSP容器**，而安装JSP引擎的计算机被称作一个支持JSP的**Web服务器**。



Tomcat是一个免费的开源JSP引擎，将安装了Tomcat的计算机称作一个Tomcat服务器。

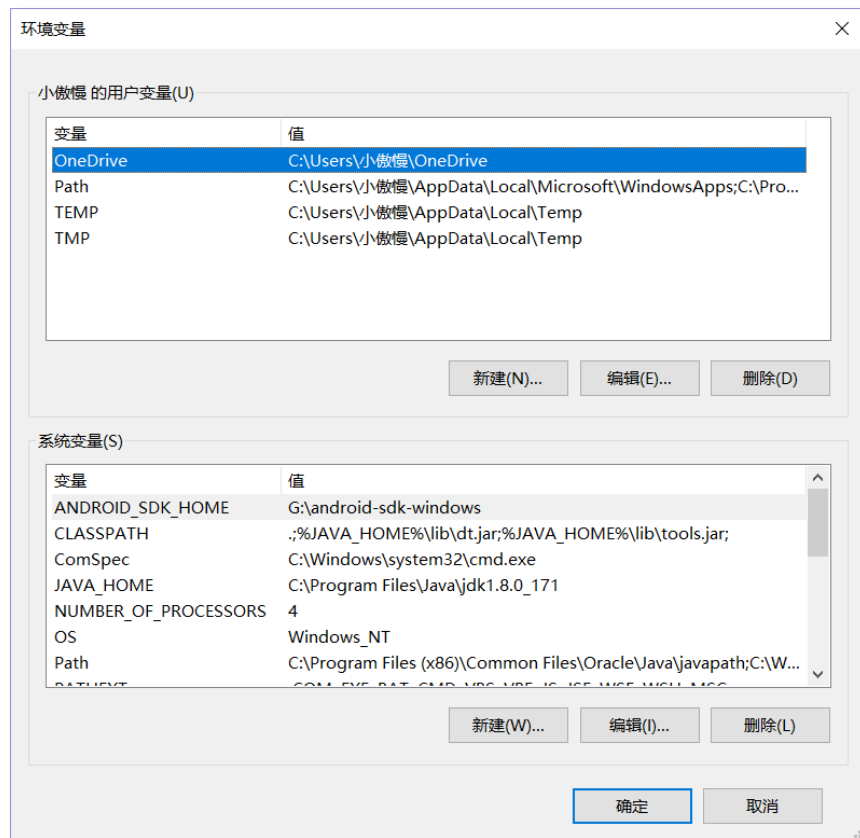


# 安装配置JSP运行环境

## 1、安装和配置JDK

👉 下载Java 标准平台(J2SE)提供的Java开发工具JDK1.7,例如 **jdk-7-windows-i586**。择安装的目录是：**D:\JDK1.7**

👉 配置环境变量





# 安装配置JSP运行环境

## 1、安装和配置JDK

### 配置环境变量

变量名 **JAVA\_HOME**， 变量值：**D:\jdk1.7**

变量名 **PATH**， 变量值：**D:\jdk1.7\bin**

变量名 **CLASSPATH**， 变量值：**D:\jdk1.7\lib**



图 1.1 设置 Java\_home



图 1.2 编辑 Path





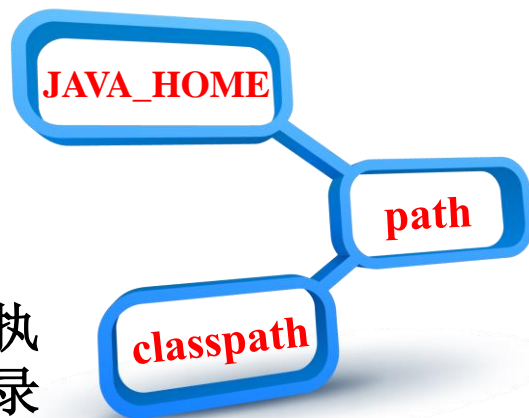
# 安装配置JSP运行环境

## 1、安装和配置JDK


### 配置环境变量的意义

使其它相关环境变量的配置更加方便简洁

告诉java虚拟机所有可执行的.class文件所在目录



帮助系统操作java，先将源码编译为.class文件，然后再调用java命令让程序去执行。因此，配置path环境变量可以帮助系统找到并调用java.exe或者是javac.exe程序，两者均在bin目录下。

 eclipse一般不用设置环境变量。从上述来看，所谓环境变量就是一个路径，让程序在编译时能够通过这一变量找到java存在的路径，所以一般要配置环境变量。而eclipse自带JDK，编译时调用JDK是从它自带的JDK里调用，路径都是固定好的，所以就不存在配置环境变量的问题了。



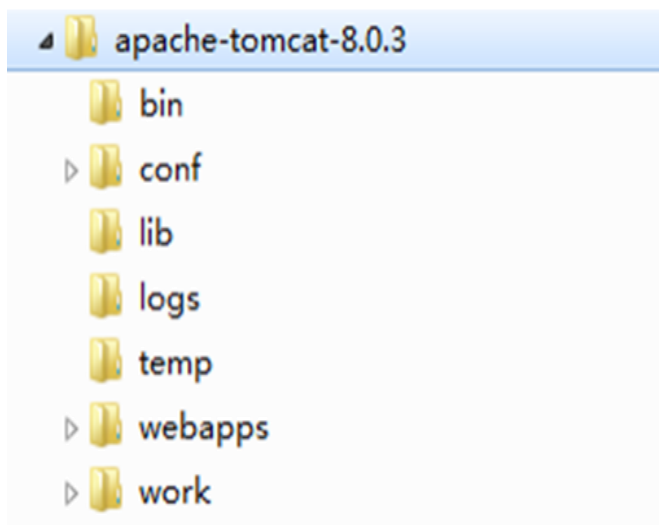
# 安装配置JSP运行环境

## 2、安装和启动Tomcat服务器

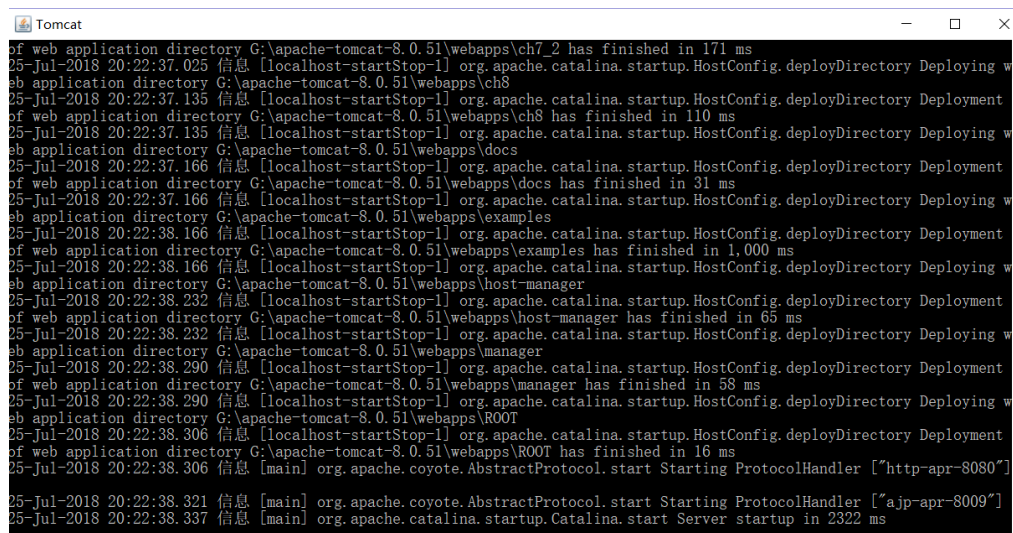
安装版与解压版Tomcat服务器的区别在于启动方式的不同！



如果下载的是apache-tomcat-8.0.30.zip文件，那么只需将该zip文件解压到磁盘某个分区中即可。执行Tomcat安装根目录中bin文件夹中的startup.bat来启动Tomcat服务器。



Tomcat服务器的目录结构



启动Tomcat服务器

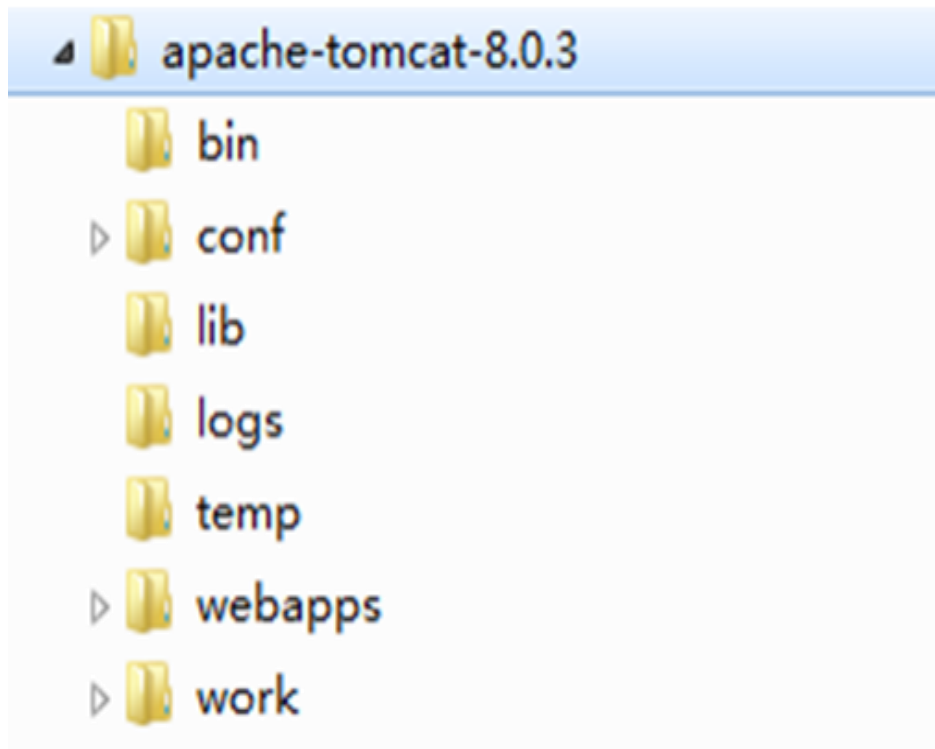


# 安装配置JSP运行环境

## 2、安装和启动Tomcat服务器



如果下载的是apache-tomcat-8.0.30.exe,双击下载的apache-tomcat-8.0.30.exe文件,将出现“**安装向导**”界面,然后根据安装提示进行安装。



Tomcat服务器的目录结构



# 安装配置JSP运行环境

## 2、安装和启动Tomcat服务器-安装版与解压版区别

- 安装版的tomcat文件双击后选择路径和jdk配合安装，启动的时候在services.msc服务中启动；解压版的tomcat文件直接解压后在bin目录中有startup.bat和startup.sh启动（windows下用startup.bat，linux下运行startup.sh）。
- 在程序运行过程中它们分别运行获取的路径是有差别的。这也是导致许多项目在本地可以运行，但部署到其它计算机上时会报错的原因之一。
- 在bin目录下启动起来，如果关闭运行窗口的话，tomcat会自动停止运行，而在服务中不会。一般服务器上都是以服务的方式运行tomcat，所以在读取项目外的配置文件的时候一定要注意路径的问题。



# 安装配置JSP运行环境

## 2、安装和启动Tomcat服务器



### Tomcat目录结构

<b>/bin</b>	存放启动、关闭和监控Tomcat的命令文件。
<b>/conf</b>	存放Tomcat服务器的各种配置文件，其中最重要的是server.xml文件
<b>/server/lib</b>	存放Tomcat服务器所需的各种JAR文件。
<b>/server/webapps</b>	存放Tomcat自带的两个Web应用：admin应用和manager应用。
<b>/shared/lib</b>	存放所有Web应用都可以访问的JAR文件。
<b>/common/lib</b>	存放Tomcat服务器以及所有Web应用都可以访问的JAR文件。
<b>/webapps</b>	当发布Web应用时，默认情况下把Web应用文件存放在此目录下。
<b>/work</b>	Tomcat把由JSP生成的Servlet放于此目录下。
<b>/logs</b>	存放Tomcat服务器的日志文件。



# 安装配置JSP运行环境

## 3、测试Tomcat服务器

如果在浏览器的地址栏中输入<http://localhost:8080>或<http://127.0.0.1:8080>，会出现如图所示的Tomcat服务器的测试页面。



测试Tomcat服务器



# 安装配置JSP运行环境

## 4、配置端口



**8080**是Tomcat服务器默认占用的端口，可以通过修改Tomcat服务器安装目录中**conf**文件下的主配置文件**server.xml**来更改端口号。用记事本打开server.xml文件，找到出现：

**<Connector port =“8080”**

**maxTheads=“150” minSpareThreads=“25” maxSpareThreads=“75”**

**enableLookups=“true” redirectPort=“8443” acceptCount=“100”**

**debug=“0” connectionTimeout=“2000”**

**disableUploadTimeout=“true” URLEncoding=“GBK”/>**





# 安装配置JSP运行环境

## 4、配置端口

**port:**定义TCP/IP端口号，默认是8080

**maxThreads:**设定处理客户请求的线程的最大数目，这个值也决定了服务器可以同时响应请求的最大数目，默认值是200

**enableLookups:**如果为true表示支持域名解析，可以把IP地址解析为主机名

**redirectPort:**指定转发端口。如果当前端口只支持non-SSL请求，在需要安全通信的场合，将把客户请求转发到基于SSL的redirectPort端口

**acceptCount:**设定在监听端口队列中的最大客户请求数，如果队列已满，客户请求将被拒绝

**connectionTimeout:**定义建立客户连接超时的时间，以毫秒为单位，如果设置为-1，表示不限制建立客户连接的时间





# 安装配置JSP运行环境

## 5、设置环境变量



**TOMCAT\_HOME:** Tomcat的安装目录，如：**D:\apache-tomcat-8.0.30**



**CLASSPATH:****%TOMCAT\_HOME%\lib**



**PATH:****%PATH%;%TOMCAT\_HOME%;%TOMCAT\_HOME%\bin**



# JSP页面

## 1、JSP页面简介

一个JSP页面中可以有普通的HTML标记、JSP规定的JSP标记以及通过标记符号“<%”，“%>”加入的Java程序片。

一个JSP页面按文本文件保存，扩展名是.jsp

**注：**文件名必须符合标识符规定；

文件名区分大小写；



# JSP页面

## 1、JSP页面简介-基本结构

**<!--JSP标记-->**

**<%@ page contentType="text/html;charset=gb2312"%>**

**<!--HTML标记-->**

**<html>**

**<head><title>第一个JSP程序</title></head>**

**<body>**

**<!--Java程序片-->**

**<%**

**out.println("Hello World!");**

**%>**

**<!--HTML标记-->**

**</body>**

**</html>**



# JSP页面

## 1、JSP页面简介-基本结构





## 2、设置web服务目录

只有将编写好的JSP页面文件保存到Tomcat服务器的某个Web服务目录中，远程用户才能通过浏览器访问该Tomcat服务器上的JSP页面。

人们常说的一个网站，实际上就是一个Web服务目录。



# JSP页面

## 2、设置web服务目录

**I 根目录：** %TOMCAT\_HOME%\webapps\Root

**http://192.168.1.100:8080/example1\_1.jsp**

**http://127.0.0.1:8080/example1\_1.jsp**

**必须省略Web根目录名！**

**II Webapps下的Web服务目录：**

Tomcat服务器安装目录webapps下的任何一个子目录都可以作为一个Web服务目录。如在webapps下新建子目录ch1，那么ch1就成为一个Web服务目录

**http://127.0.0.1:8080/ch1/example1\_1.jsp**



# JSP页面

## 2、设置web服务目录

### III 新建Web服务目录:

用记事本打开**conf**文件夹中的主配置文件**server.xml**,  
在 **</Host>**的前面加入

```
<Context path="/apple" docBase="D:\MyBook\zhang" debug="0"  
    reloadable="true" />
```

```
<Context path="/cloud" docBase="C:\wang" debug="0"  
    reloadable="true" />
```

保存并重新启动 **tomcat**服务器!

[http://127.0.0.1:8080/apple/example1\\_1.jsp](http://127.0.0.1:8080/apple/example1_1.jsp)

或[http://127.0.0.1:8080/cloud/example1\\_1.jsp](http://127.0.0.1:8080/cloud/example1_1.jsp)



# JSP页面

## 2、设置web服务目录

### IV 相对目录:

Web服务目录下的目录称为该Web服务目录下的相对目录。

比如在Web服务目录D:\MyBook\zhang下再建立一个子目录image，将example1\_1.jsp文件保存到image中。那么可以在浏览器的地址栏中输入

[http://127.0.0.1:8080/apple/image/example1\\_1.jsp](http://127.0.0.1:8080/apple/image/example1_1.jsp)





## 3、Web应用目录结构

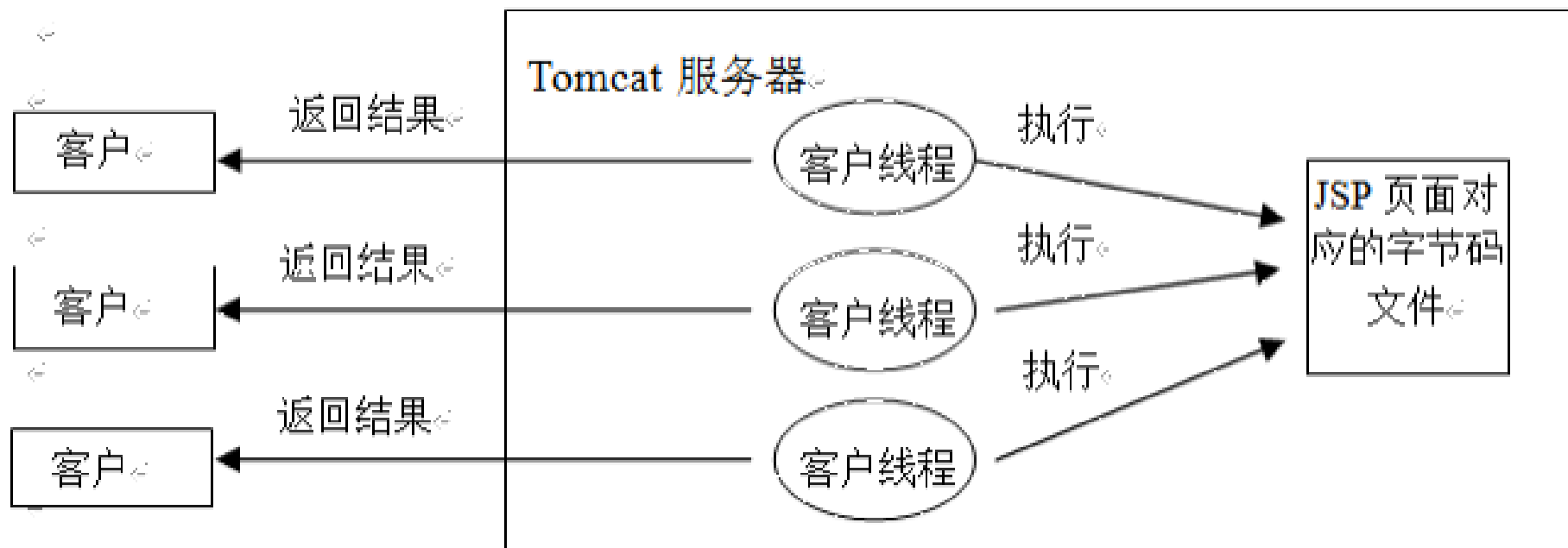
Web应用具有固定的目录结构，假定开发一个userLoginapp的Web应用。首先应该在webapps目录下创建userLoginapp的目录结构。

目录	描述
<b>/userLoginapp</b>	<b>Web</b> 应用的根目录，所有的 <b>JSP</b> 和 <b>HTML</b> 文件都存放在此目录下。
<b>/userLoginapp/WEB-INF</b>	存放 <b>Web</b> 应用的发布描述文件 <b>web.xml</b> 。
<b>/userLoginapp/WEB-INF/classes</b>	存放各种 <b>class</b> 文件， <b>Servlet</b> 类文件也存放于此目录下。
<b>/userLoginapp/WEB-INF/lib</b>	存放 <b>Web</b> 应用所需的各种 <b>JAR</b> 文件，如今后要学习的 <b>JDBC</b> 驱动。



# JSP运行原理

当服务器上的一个JSP页面被第一次请求执行时，服务器上的JSP引擎首先将JSP页面文件转译成一个java文件，并编译这个java文件生成字节码文件，然后执行字节码文件响应客户的请求。



JSP的运行原理



# JSP运行原理

## 字节码文件的主要工作：

- ❏ 把JSP页面中的HTML标记符号（页面的静态部分）交给客户的浏览器负责显示。
- ❏ 负责处理JSP标记，并将有关的处理结果发送到客户的浏览器。
- ❏ 执行“<%”和“%>”之间的Java程序片（JSP页面中的动态部分），并把执行结果交给客户的浏览器显示。



# JSP与Servlet的关系

- ✎ **Java Servlet**就是编写在服务器端创建对象的Java类，习惯上称之为Servlet类，Servlet类的对象习惯上称之为一个Servlet。
- ✎ **JSP技术**就是以Java Servlet为基础，提供了Java Servlet的几乎所有好处。但是JSP技术不是Java Servlet技术的全部，它只是Java Servlet技术的一个成功应用。
- ✎ 对于某些Web应用，就可能需要JSP+Javabeen+Servlet来完成，即需要服务器再创建一些Servlet对象，配合JSP页面来完成整个Web应用程序的工作。



## 小结

- 👉 **JSP引擎是支持JSP程序的Web容器，负责运行JSP，并将有关结果发送到客户端。目前流行的JSP引擎之一是Tomcat。**
- 👉 **安装Tomcat服务器，首先要安装JDK，并需要设置JAVA\_HOME环境变量。**
- 👉 **JSP页面必须保存在Web服务目录中。Tomcat服务器的webapps下的目录都可以作为Web服务目录。如果想让webapps以外的其他的目录作为Web服务目录，必须要修改Tomcat服务器下conf文件夹中的server.xml文件，并重新启动Tomcat服务器。**



## 小结

- 👉 当服务器上的一个**JSP**页面被第一次请求执行时，服务器上的**JSP**引擎首先将**JSP**页面文件转译成一个**Java**文件，再将这个**Java**文件编译生成字节码文件，然后通过执行字节码文件响应客户的请求。
- 👉 当多个客户请求一个**JSP**页面时，**Tomcat**服务器为每个客户启动一个线程，该线程负责执行常驻内存的字节码文件来响应相应客户的请求。这些线程由**Tomcat**服务器来管理，将**CPU**的使用权在各个线程之间快速切换，以保证每个线程都有机会执行字节码文件。