

教育是用生命影响生命的过程

Broadview®  
www.broadview.com.cn

# Java Web 开发

## 就该这样学

我用了**8年时间**教授Java Web这门课，尝试着让初中毕业生、被传统教育淘汰的人，或是退休在家的老人掌握编程技术，大约**15000名**各种基础和理解能力的学生轻松地掌握了这项技术，令人欣慰的是所有的学生从始至终都将这个学习过程当成一场游戏，并将编程这个工作发展成自己的兴趣。在这本书，我将阐述我与众不同的做法，希望学习者能够在一开始便建立起新的学习思想，这样我们才能一起玩代码。



王洋 编著

 电子工业出版社  
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY  
<http://www.phei.com.cn>

金牌讲师为您开启编程的快乐之旅

# Java Web开发 就该这样学

---

王洋 编著

---

电子工业出版社

Publishing House of Electronics Industry

北京·BEIJING

## 内 容 简 介

本书基于建构主义教育思想,通过大量循序渐进的案例,让学生在体验中掌握 Java Web 相关知识,同时获得编程能力、排错能力和学习能力,本书多次使用陷阱式教学法,帮助学生深刻理解所学知识,掌握实现 Web 编程的不同技术特点。

本书详细介绍了 Java Web 程序设计的前端技术、开发和部署,以及一些衍生技术变化。在内容上,本着使用不同技术尽可能实现相同功能的原则,让读者能够充分体会认识每个技术的优缺点。

本书的内容和组织形式立足于高校教学教材的要求,适用于从职业院校到重点本科院校的教师教学和学生学习,可以作为 Java Web 程序设计的入门教材,或者面向就业的实习实训教材,同时可作为计算机技术的培训教材,读者完全可以通过本书自学 Java Web 技术。

未经许可,不得以任何方式复制或抄袭本书之部分或全部内容。  
版权所有,侵权必究。

### 图书在版编目(CIP)数据

Java Web 开发就该这样学 / 王洋编著. —北京: 电子工业出版社, 2013.6  
ISBN 978-7-121-20453-1

I. ①J… II. ①王… III. ①JAVA 语言—程序设计 IV. ①TP312

中国版本图书馆 CIP 数据核字(2013)第 103759 号

策划编辑: 孙学瑛

责任编辑: 徐津平

印 刷: 三河市鑫金马印装有限公司

装 订: 三河市鑫金马印装有限公司

出版发行: 电子工业出版社

北京市海淀区万寿路 173 信箱 邮编 100036

开 本: 787×1092 1/16 印张: 15.5 字数: 365 千字

印 次: 2013 年 6 月第 1 次印刷

印 数: 4000 册 定价: 49.00 元

凡所购买电子工业出版社图书有缺损问题,请向购买书店调换。若书店售缺,请与本社发行部联系,联系及邮购电话:(010) 88254888。

质量投诉请发邮件至 [zltz@phei.com.cn](mailto:zltz@phei.com.cn), 盗版侵权举报请发邮件至 [dbqq@phei.com.cn](mailto:dbqq@phei.com.cn)。

服务热线:(010) 88258888。

# 前 言

---

一直以来人们都认为教师和书籍是知识的载体，教学的过程就是将这些知识传递给学生，于是书上写满了正确的知识，学生看书就可以迅速掌握知识，理论上这是高效率的系统，但事实上只有极少数人能够适应这样的系统，因为这些知识也是有人经过一个过程得到的，忽略了发现知识的过程，而直接将结果传递给学生，似乎高效率，但是学生却常常无所适从，因为学习是发现知识的过程，而不是记住知识的过程。

好在建构主义教育思想指明了更加适合学生的教学过程，在这一思想下，书和教师从正确知识的传递者，转变成探索知识的引领者，带领着学生去体验、去感受、去发现属于学生自己的知识，正如这本书，读者会发现 60% 的内容是不正确的，这些错误是学习过程避免不了的，优秀的学生不同之处就是，通过自己的努力在到达学习目标的道路上，不断地调整，将错误的理解剔除掉，问题是大多数学生无法完成这个过程，要么陷入错误的包围中，最终放弃了探索，要么通过死记硬背来自己达到学习目的，结果学生能够通过考核，却没有运用知识的能力。学习离正确的轨道越来越远，甚至很多人迷失了学习的真正目标，将记住知识作为唯一的目标，很多教育者苦苦探索的正确教育途径，在教育理论研究中早已经被发现，那就是建构主义教育。

建构主义教育思想从来不认为掌握知识是学习的最终目的，我认为学习的目标是认知、能力和精神。认知和知识是不同的，知识停留在人的头脑中，而认知是能够被熟练使用的知识；能力在不同的领域是不同的，由于 Java Web 技术是前端页面技术和后端编程技术的结合，其中前端技术结合了 HTML、CSS 和 JavaScript，这些几乎完全不同语法特点的编程，通常是 Java Web 程序员的薄弱环节，所以综合运用能力就成为 Java Web 前端技术学习的关键。

后端编程技术经过多钟不同思路的发展，产生了更加适合编程的 Servlet 技术和 JSP 技术，并且在两个主流的分支基础上又演变出了 JSP+JavaBean 以及 JSP+Tag 的应用手段，为了改善用户体验，还出现了 AJAX 技术，所以在本书的范畴内，一个优秀的 Java Web 程序员，既需要有前端编程能力，又需要掌握后端编程的特点，清晰地理解每一种技术适用的范围和优缺点；如果学习的目的仅仅是为了掌握一项技术，那么人终将被新的技术手段所替代，任何学习过程都是生命价值的提升，一个程序员需要有严谨的态度、专注的品质、探索的精神和创新的意识。这些学习目标不是一节课或一个章节的任务，需要通过整个教学过程来建构。

一直以来学生的学习动机都是教育理论界热衷讨论的话题，我认为学生学习的动机有三个方面，一是为了获得喜悦，二是为了消除恐惧，三是自我效能。好的成绩可以获得家长、老师的表扬，可以有更好的名次甚至奖学金。而差的成绩会被批评、留级，甚至拿不到毕业证。我们发现普遍的教学手段是为了推动学习动机的前两个方面，这造成了两个可能的结果，有些学生对于奖励或是惩罚麻木了，一旦丧失了学习动机，自然好的成绩无从谈起，在另外的学生身上，这些手段一直能够起到作用，我们会得到所谓的好学生，问题是这些动机是外界推动的，而非内生的，这些习惯于此的好学生或许一生都在意别人的评价。如果教学过程能够激发学生的自我效能，让学生的学习是基于自己强烈的爱好和成功的喜悦，我们就一定能够培养出来优秀的学生，而他们也必将一生受益。

问题是为什么建构主义教育思想如此的好，却很少在教学实践中应用，这是因为建构主义和现有的教学形式相比仍有些弱点：第一，建构主义在教育的效率理论上比较低，现在我们能够在短时间内将大量的正确知识传递给学生，学生只需要理解记忆就好了，而建构主义教育要呈现知识探索的过程，这样会消耗更多的时间和精力。第二，建构主义教育的效果不可控，学生是通过体验自己发现整合知识，那么不同的学生或许得到的结论不同，深度不同。第三，考核困难，我们不能再用知识点来考核学生，因为教学过程中就没有传递经典的知识点。第四，实施建构主义教育对于教师的要求比较高，教学过程的设计建立在对学生的深入理解的基础上，教师不仅要准备教学知识了。

为了实现上述效果，老师将扮演不同以往的角色，教师不再是知识的载体，教师将陪伴着学生一同探索，带领着学生犯错误，引导着学生进行思考整合。为了克服建构主义教育思想的弱点，在写这本书时，我基于对学生和技术的理解，剔除了大量知识点的讲解，在反复的教学实践中，已经能够获得和传统教学相同的教学效率。另外我大量总结和研究了学生的学习过程，建立了学生在学习 Java 过程中的学习曲线，依照学习曲线来评估和考核学生的学习效果。

本书总结了作者多年在这条道路上的探索，力求提供基于建构主义教育思想的 Java Web 教学材料，帮助学生轻松地掌握作为 Java Web 程序员所需要的知识和能力，通过对比运用相似技术，帮助学生将最适合的技术运用到项目中。书中的内容并不是简单的案例堆砌，每个部分的任务都包含了对相关知识的整合，都基于学生的学习曲线特点。

我在 8 年教学探索后才动手写这本书，因为我一直相信“教育是用生命影响生命的过程”，我无法在一本书中实现和我亲自上课同样的影响过程，课堂上一遍遍的重复代码所传递的严谨态度，无法在书中呈现，加上我对技术、对学生理解，以及对于建构主义教育思想理解的局限，让我清楚地知道，我并没有完成一部让我心满意足的作品，书中不可避免的有很多不足，恳请读者批评指正。

这本书的内容是我数以万计的学生的成果，甚至有很多案例是我的学生在学习的过程中发明的，这段从 8 年前开始的探索并不是一蹴而就的，我诚挚地感谢我所教过的学生，是他们的忍受、包容和努力帮助我完成了这本书。我要感谢我的家人，我儿子的出生和成长，让我开始接触和研究教育理论，给我之前漫无目的地探索指明了方向，为了让这本书通俗易懂，我那学文科的

爱人像一名真正的学生一样，通过这本书来学习 Java 技术，在她的努力下，这本书具备了更强的覆盖范围，确保读者即便是没有任何专业基础，也能够通过这本书掌握 Java 技术。同时也要感谢电子工业出版社的老师为本书的出版所付出的辛勤工作。

光盘使用：我一直希望这本书的定价尽可能低，希望有更多的人能够没有负担的学习 Java 技术，本意不想提供光盘，但是权衡再三，由于本书中的代码是伴随着讲解逐步展开的，很多代码没有整体呈现（否则会增加太多页数），所以不得不通过光盘提供书中的代码，这些代码被放在 `codes` 目录中，按照书中的章节组织，需要强调的是，请不要直接编译运行，或是复制我提供的代码，光盘中的代码是我的，只有你亲手输入到电脑里的内容才属于你。

王 洋

于 2013 年 4 月 14 日

# CONTENTS

# 目 录

第 1 章 认识 Tomcat	1
1.1 什么是 Web Server	1
1.2 选择 Tomcat 来学习	2
1.3 安装 Tomcat	2
1.4 MyTomcat	6
1.5 MyIE	10
1.6 再谈 Tomcat	12
第 2 章 学习 HTML	14
2.1 认识 HTML	14
2.2 做百度的首页	15
2.3 搜狐邮箱的用户登录	24
2.4 京东的购物车	28
2.5 用表格定位搜狐邮箱的用户登录界面	32
2.6 使用 CSS 实现搜狐邮箱的用户登录	41
2.6.1 绝对定位	41
2.6.2 div	43
2.6.3 级联样式	44
2.7 在网页上显示时间	49
2.7.1 为什么要学习 JavaScript	49
2.7.2 获取时间	50
2.7.3 定义函数	54
2.7.4 js 文件	55
2.7.5 显示到其他地方	57
2.7.6 能动的的时间	58

2.7.7	漂浮的时间显示	60
2.8	再看搜狐邮箱的用户登录	63
2.9	京东商城的新用户注册	67
2.9.1	String 对象操作	73
2.9.2	正则表达式	74
2.9.3	密码框验证	78
2.9.4	邮箱地址验证	84
2.10	搜狐首页的菜单条	85
2.11	QQ 空间的设置	89
<b>第 3 章</b>	<b>Servlet</b>	<b>94</b>
3.1	Servlet 怎么运行	94
3.1.1	编写第一个 Servlet	96
3.1.2	部署	100
3.2	用户登录	104
3.3	重要的 XML	111
3.3.1	XML	112
3.3.2	DTD	113
3.3.3	Schema	117
3.3.4	CSS 和 XSL	122
3.3.5	DOM	126
3.3.6	SAX	131
3.3.7	XML 总结	132
3.4	购物网站的商品展示	133
3.4.1	数据库设计	133
3.4.2	展示页面程序	134
3.4.3	查询评论数量	141
3.4.4	分离数据库连接	141
3.4.5	分页显示	144
3.4.6	在每个页面上都显示用户名	149
3.5	用户注册	151
3.5.1	生成验证码图片	152
3.5.2	绘制干扰线	154
3.5.3	更新验证码	156
3.5.4	注册处理程序	157
3.5.5	使用 AJAX 验证用户名是否冲突	159
3.5.6	用 AJAX 实现分页显示	162



第 4 章 JSP .....	177
4.1 用户登录 .....	178
4.1.1 设置中文编码 .....	180
4.1.2 编写脚本 .....	180
4.1.3 连接数据库 .....	181
4.1.4 跳转 .....	182
4.2 购物网站的商品展示 .....	185
4.3 将用户登录结合到商品展示页面中 .....	190
4.3.1 使用 Cookie .....	192
4.3.2 将两个网页合并 .....	195
4.4 购物车 .....	196
4.4.1 实现加减按钮和删除商品的功能 .....	201
第 5 章 使用 JavaBean .....	205
5.1 使用 JavaBean 实现用户验证 .....	205
5.1.1 定义 JavaBean .....	205
5.1.2 运用 JavaBean .....	208
5.1.3 JavaBean 的作用域 .....	209
5.1.4 在 JavaBean 中使用内置对象 .....	210
5.2 使用 JavaBean 来实现商品展示 .....	212
5.2.1 规划和设计 JavaBean .....	212
5.2.2 改造 JSP .....	214
5.2.3 将数据库和页面彻底分离 .....	216
5.3 实现购物车逻辑 .....	221
第 6 章 使用自定义标记 TAG .....	226
6.1 使用 JSP、JavaBean 和 TAG 实现商品显示 .....	229

## 1.1 什么是 Web Server

今天来讨论 Web 对大多数人来说都不是多么陌生的概念，我们在电脑中打开浏览器，在地址栏中输入 `www.sohu.com`，就能够看到搜狐公司提供给我们的网页，来思考一下在你的操作背后，计算机本身以及 Internet 为我们在做些什么事情。

首先是浏览器，经过 Java 部分编程的学习，你不应该对计算机里面的任何程序有所敬畏，那些程序都是和你一样的程序员编写出来的，理论上只要你愿意，花些时间你也可以编写出相似的程序！

那么，如果由你来编写一个浏览器，你会做些什么？我们能够想象，用户输入的 `www.sohu.com` 是一个网址，它对应了一个 IP 地址，通过网址得到一个 IP 地址，并不是一件多么难的事情，互联网上的 DNS 服务就可以提供这样的转换。作为一个程序员，得到一个 IP 地址后，会创建 Socket 对象和对方连接上，目的是为了得到一个特定的网页，我们常常能够在浏览器的地址栏中看到这样的网址：`http://www.360buy.com/product/258313.html`，意思是我要看京东网站上的 `258313.html` 这个文件的内容。这是由于一个网站上的网页文件太多了，所以还会有路径结构。

`www.sohu.com` 这样的网址请求是个例外，你并没有指定要看哪个网页，通常网站会提供一个默认的网页，如果没有指定，就将这个默认的网页传送给你。无论是哪种形式，我们将浏览器中输入的地址称为 URL。

我们不看这样的例外情况，来想一下，如果要看 `258313.html` 这个网页，是不是要通过 Socket 将这个请求传送到京东的网站服务器去呢。

京东的网站服务器一定是一台计算机，但是只有计算机是不够的，计算机上需要运行一个程序，我们能够理解，这个程序将是服务器端的 Socket 应用，里面有 `ServerSocket`，这个程序将监听在一个端口，大家约定了提供 Web 服务的端口是 80，当然也可以改变这个约定，那么在浏览器访问的时候，就要额外指定新的端口号，如果不这样做，就意味着浏览器访问的是 80 端口。

服务器的程序将接收到浏览器发出的请求，然后它用 IO 流到自己所管理的目录中找到这个 `258313.html` 文件，将这个文件的内容读到内存中，然后通过网络发送给你的浏览器，浏览器

得到了这个文件的内容，将按照 html 的规则显示出来。这就是一次最简单的 Web 访问的过程，如图 1-1 所示。

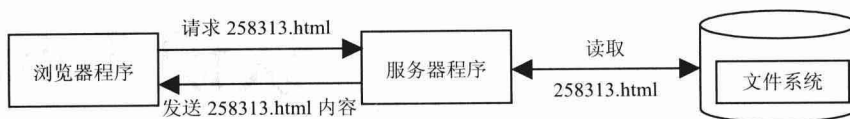


图 1-1

作为一个计算机的使用者，浏览器是一个非常常用的程序，有很多公司会把这样的程序做成产品，所以现在我们已不需要来编写这样的程序了。

那么服务器程序在互联网世界中，是不是也是非常常见的程序呢，其实编写服务器程序更加有利可图，所以现在在市场上有很多服务器端的软件产品，这些产品被统称为 Web Server。

## 1.2 选择 Tomcat 来学习

比较著名的有这样几种，在微软的服务器端操作系统中，会附带提供名字叫做 Microsoft Internet Information Server（简称 IIS）的产品，由于微软公司的成功，这款产品被广为使用。但是它有两个弱点：一是，它通常只能运行在 Windows 系列产品中，虽然在个人计算机中 Windows 的普及率非常的高，但是作为服务器端的计算机，使用 Windows 系统的情况相对少得多；二是它不能直接支持 Java。

Tomcat 却没有这样的问题，Tomcat 能够运行在任何操作系统中，也是 Java 官方推荐的 Web Server，当然它不支持微软所倡导的 ASP 或是 ASP.NET。Tomcat 还有一个非常重要的优点，这个软件是免费的，这并不意味着 Tomcat 是简陋的，事实上有很多著名的网站是由 Tomcat 支持而建立起来的，基于这些优点 Tomcat 成为学习 Java Web 的首选 Web Server。

此外还有 BEA 公司的产品 WebLogic，IBM 公司的产品 WebSphere，这些都是大名鼎鼎的 Web Server，它们往往提供了一些 Tomcat 所不具备的功能。问题是它们太昂贵了，当然这些额外的功能并不是我们现在这个阶段所涉及的。

## 1.3 安装 Tomcat

Tomcat 是一款开源的免费软件，这意味着 Tomcat 的版本升级速度是非常快的，问题是每次 Tomcat 的升级，都会带来配置和使用细节的变化。所以你新下载的 Tomcat 和这本书中的一些细节，有可能不是那么吻合，但是这个现象并不打紧，因为基本的思想和方法并没有太大变化。

截止目前，Tomcat 的最新版本是 Tomcat 7，虽然对于初学 Tomcat 的人来说，这个版本的升级离我们还有些遥远，但是我确实想不出来拒绝一个最新版本的理由，下载 Tomcat 的官方网址

是 tomcat.apache.org，你可以找到在国内的下载地址，当然即便是在官网上下载也是非常快的，我们也发现这里出现了一个名字叫做 Apache，我不能称 Apache 是一个公司的名字，我们通常将其叫做 Apache 组织，Tomcat 是这个组织提供的一款产品，随着学习的深入，你未来可能还会遇到这个组织的其他产品。我下载的版本是 Tomcat 7.0.25，文件分为 32 位版本和 64 位版本，你需要根据自己的计算机来选择。

下载下来的文件是.zip 的压缩包，Tomcat 是一个纯绿色的软件，不需要安装，只要将这个压缩包解压就可以了。解压后的目录如图 1-2 所示。

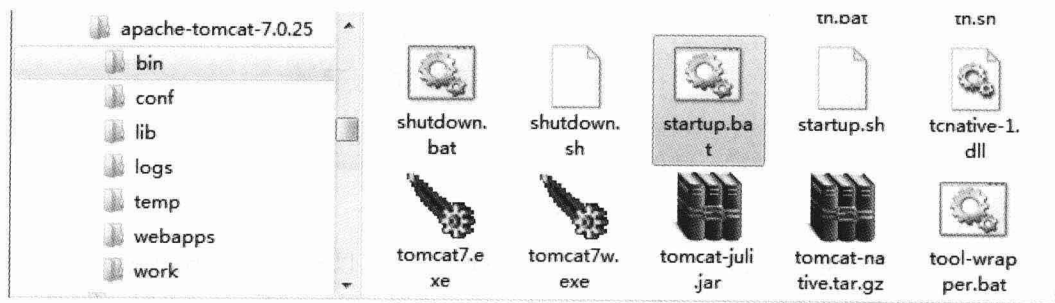


图 1-2

选择放在 Tomcat 目录下的 bin 目录，并选择其中的 startup.bat 文件，.bat 文件是微软系统环境下的批处理文件，通常人们将一系列的命令放到这个文件中，这样就能通过运行这个批处理文件，同时启动其中的命令了。这个文件用于启动 Tomcat 程序，像我们熟悉的那样，双击运行这个程序，你会发现有一个黑色的控制台窗体一闪而过，我抱歉的告诉你，这代表运行是失败的。

Tomcat 和 JDK 的联系紧密程度超乎想象，如果没有 JDK 的支持 Tomcat 根本就无法运行，可是 Tomcat 自身并不带 JDK，它需要找到你计算机中的 JDK，由于我们没有安装，也就是说没有一个安装程序能够帮助 Tomcat 找到 JDK，所以我们要按照 Tomcat 的要求指定 JDK 的位置，Tomcat 需要一个叫做 JAVA\_HOME 的环境变量。

设置环境变量的过程在上一本书《Java 就该这样学》的开始进行过类似的讲解，用鼠标右键单击“我的电脑”图标，选择“属性”命令，然后在“高级”选项卡中找到“环境变量”按钮，“系统属性”对话框如图 1-3 所示。

单击“环境变量”按钮，会弹出一个设置环境变量的窗体，如图 1-4 所示，窗体分成上下两个部分，上面那个部分设置的是影响当前用户的环境变量，而下面设置的是影响这台计算机上所有用户的环境变量，可根据你的情况来选择，如果这台计算机是你的，选择那个都无所谓。

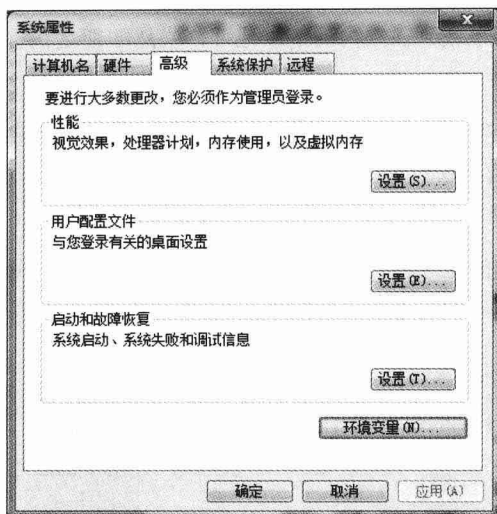


图 1-3

单击“新建”按钮，输入环境变量名为 JAVA\_HOME，你可以通过资源管理器找到你正在使用的 JDK，然后将路径复制到变量值输入框中。

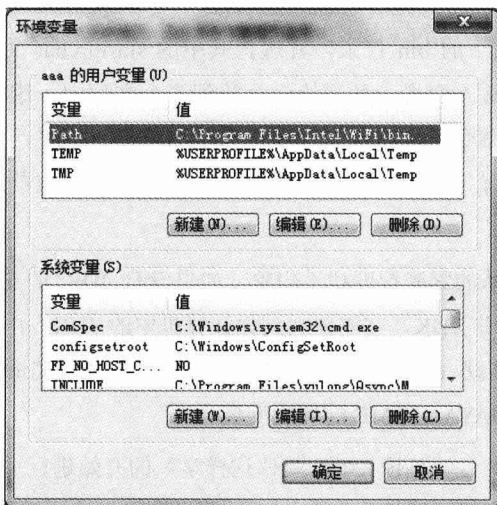


图 1-4

单击“确定”按钮后，再回到 Tomcat 目录中，到 bin 目录下找到 startup.bat 文件，双击运行它，经过一番等待后，你会看见在一个黑色的控制台窗体里面出现很多输出，如图 1-5 所示。

见到这句 Server startup in 2480 ms，就说明 Tomcat 成功运行了，当然 ms 数针对每台计算机是不同的，千万不要关闭这个黑色窗体，关闭了 Tomcat 就不再运行了。为什么？还记得我们所编写的服务器端 Socket 程序吗？这个 Tomcat 就是那个程序，现在程序在 8080 端口监听，为什么不是 80 端口呢？咱们这不是在做实验吗？默认的 Tomcat 实验端口是 8080。

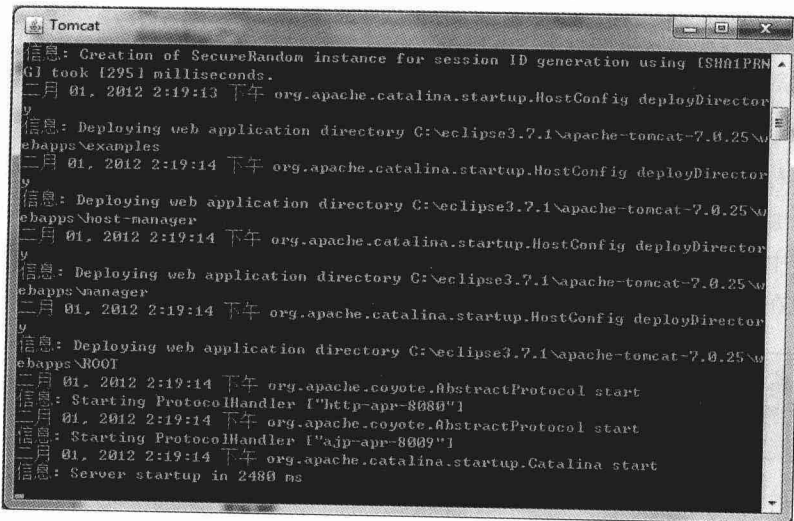


图 1-5

我们要验证一下 Tomcat 运行了，打开 IE 浏览器，输入 URL: <http://127.0.0.1:8080>，这个 127.0.0.1 是本机的回传地址，通过:8080 来指定 80 以外的端口号，至于“http://”是什么意思回头再解释。

如果你能够在浏览器中看到如图 1-6 所示的画面，说明 Tomcat 是工作的，当然不同版本的 Tomcat 提供的页面是不同的，不过我想这只小猫将一直存在。

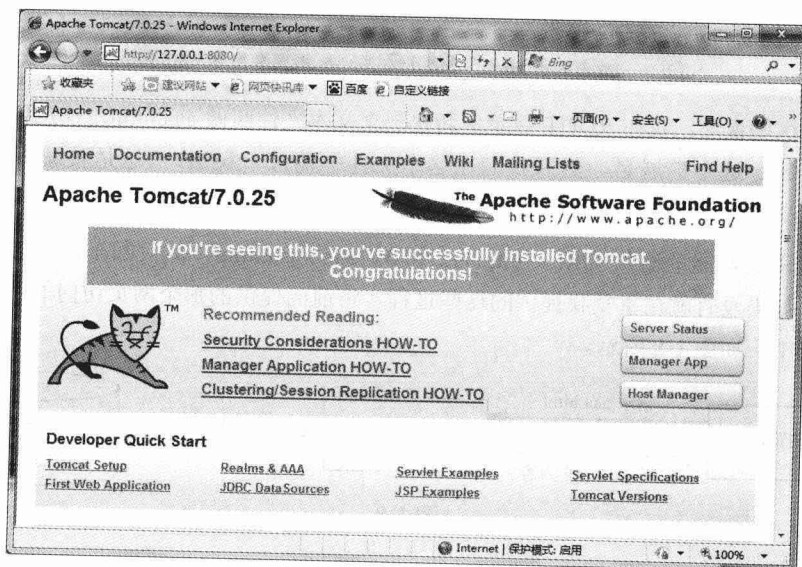


图 1-6

现在访问的自然就是 Tomcat 所提供的默认网页，我们是希望能够提供我们所编写的网页，还记得前面我描述的 Web 访问的工作过程吗，如果我们请求的网页是 `aaa.html`，那么 Tomcat 得到

这个请求后，会到自己所管理的目录中寻找这个文件，注意我所使用的语言，是到自己所管理的目录中，而不是在计算机的硬盘上，当然 Tomcat 所管理的目录也在硬盘上，但是它不能肆无忌惮地访问整个硬盘，Tomcat 所管理的目录就是我们运行的 Tomcat 目录中的 webapps 目录。

我们能够看到在这个 webapps 中有一个子目录的名字是大写的 ROOT 目录，这是默认的根目录，我们能够看到这个目录中已经有了一些文件，这些文件就是上面那个有小猫的网络文件。我们还看到其中有一个目录叫做 WEB-INF，这是作为 Tomcat 要求的，后面我们会频繁地接触到这个目录，现在还不需要做什么。如图 1-7 所示。

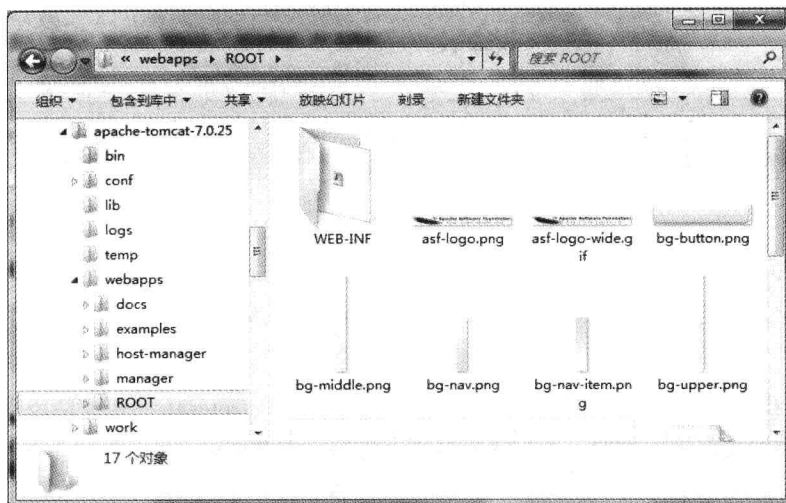


图 1-7

为了验证这些说法，我在 ROOT 目录中新建一个文本文件叫做 aaa.html，注意要避免计算机自动给你提供.txt 这个扩展名。然后我们在这个文件中输入一句话：“这是我编写的第一个网页”，输入后保存。

我们再到浏览器那里输入 URL：<http://127.0.0.1:8080/aaa.html>，然后回车，你的浏览器上有这句话了吗？如果没有就在重复我提供的这些过程，将前面提供的那个网页访问的示意图套到这个应用的例子中，如图 1-8 所示。

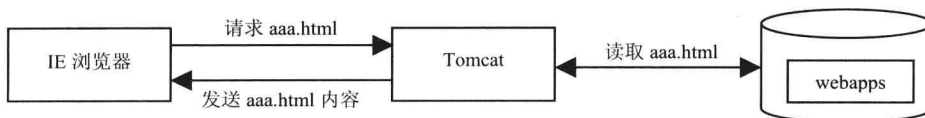


图 1-8

## 1.4 MyTomcat

有人不相信我所描述的这个过程，如果你有这个念头，我觉得你会是一个有前途的程序员，

你不应该相信任何人或文档上面对原理的描述，除非你自己亲身验证过，这么说不仅仅是因为在计算机行业里没有任何权威，有的时候同样看到听到的话，两个人的理解会有很大的不同。

你们现在完全有能力用程序来验证我所描述的 Web 访问流程，你可以关闭掉 Tomcat，自己做一个假的 Tomcat，然后让真的 IE 浏览器进行同样的访问，看看会不会有 aaa.html 这样的字符串发送给 Tomcat。

整理一下思路，真正的 Tomcat 会做什么事情呢？Tomcat 会监听 8080 端口，当用户启动浏览器，输入网址访问到你的计算机和定义好的 8080 端口时，我们的程序会获得一个服务器端的 Socket，然后获取这个 Socket 的输入流，这样我们就可以通过这个输入流接收用户浏览器发送过来的信息，目前我们好奇的是浏览器会发生什么？下面的工作我们先不考虑，现在来看看是否能够接收到请求。

我们来编写一个叫做 MyTomcat 的程序。

```

/*
 *
 * 伪装的 Tomcat，来验证 Web 访问流程
 *
 */
import java.net.*;
import java.io.*;

public class MyTomcat {
    public static void main(String args[]){
        try{
            ServerSocket ss = new ServerSocket(8080);

            System.out.println("Server startup in 1428ms");
            Socket s = ss.accept();

            InputStream is = s.getInputStream();
            InputStreamReader isr = new InputStreamReader(is);
            BufferedReader br = new BufferedReader(isr);

            System.out.println(br.readLine());
        }catch(Exception e){}
    }
}

```

上述程序对于经过 Java 学习的学生应该相当简单，运行的时候千万不要忘记，要关闭真的 Tomcat，否则将发生 8080 端口被占用的异常。运行这个程序后，控制台上将显示 Server startup in 1428ms 这句话，这纯粹是为了让假的 Tomcat 尽可能的像真的。然后到浏览器那里单击“刷新”按钮，回到控制台，你将看见这样一句话：GET /aaa.html HTTP/1.1。

里面有一个部分是/aaa.html，这就是我所说的请求，字符串，还有别的信息，前面有一个单词 GET，你要记住这个单词，后面的学习会用到。最后是 HTTP/1.1，我在测试 Tomcat 有没有工



作的时候用到过这个 HTTP，当时我将这个单词的解释留了下来，别急，我现在还不解释。

多出来的这些信息让我有了一个想法，或许会有更多的信息存在，我将输入流读取的语句多复制了一些，我们再来试试。

```
/*
 *
 * 伪装的 Tomcat，来验证 Web 访问流程
 *
 */
import java.net.*;
import java.io.*;

public class MyTomcat {
    public static void main(String args[]){
        try{
            ServerSocket ss = new ServerSocket(8080);

            System.out.println("Server startup in 1428ms");
            Socket s = ss.accept();

            InputStream is = s.getInputStream();
            InputStreamReader isr = new InputStreamReader(is);
            BufferedReader br = new BufferedReader(isr);

            System.out.println(br.readLine());
            System.out.println(br.readLine());
            System.out.println(br.readLine());
            System.out.println(br.readLine());
            System.out.println(br.readLine());
            System.out.println(br.readLine());
            System.out.println(br.readLine());
            System.out.println(br.readLine());
            System.out.println(br.readLine());
            System.out.println(br.readLine());
            System.out.println(br.readLine());
            System.out.println(br.readLine());
            System.out.println(br.readLine());
            System.out.println(br.readLine());
            System.out.println(br.readLine());
        }catch (Exception e){}
    }
}
```

不出所料，刷新浏览器的时候，出现了大量的信息被打印出来，如下所示。

```
Server startup in 1428ms
GET /aaa.html HTTP/1.1
Accept: */*
Accept-Language: zh-CN
User-Agent: Mozilla/4.0 (compatible; MSIE 8.0; Windows NT 6.1;
Trident/4.0; SLCC2; .NET CLR 2.0.50727; .NET CLR 3.5.30729; .NET CLR
```