

高等学校应用型本科创新人才培养计划指定教材
高等学校计算机类专业“十三五”课改规划教材



Java Web

■ 程序设计及实践

青岛英谷教育科技有限公司 编著



西安电子科技大学出版社
<http://www.xduph.com>

高等学校应用型本科教材
高等学校计算机类专业“十三五”课改规划教材

Java Web 程序设计及实践

青岛农业大学
编著
青岛英谷教育科技有限公司

西安电子科技大学出版社

内 容 简 介

本书系统地讲解了使用 Servlet、JSP、JSTL、AJAX 和 Web 服务等技术实现动态商业网站的方法,全面解析了 B/S 架构开发的基本技能。全书共分为 10 章,分别介绍了 Servlet 基础、Servlet 会话跟踪、JSP 基础、JSP 指令和动作、JSP 内置对象、EL 和 JSTL、监听和过滤、AJAX 技术、Web 服务概念和开发技术以及 SOAP、WSDL 和 UDDI 等概念。通过对本书的学习,读者可以掌握 Servlet 的生命周期及会话应用;熟练使用 JSP 脚本代码动态显示页面;通过 EL 表达式和 JSTL 标准库标签来简化 JSP 页面代码;掌握如何使用过滤和监听技术增强网站的完整性;使用 AJAX 技术增强页面交互性;使用 SOAP 开发 Web 服务。

本书重点突出、偏重应用,结合理论篇的实例和实践篇的案例讲解、剖析,读者可以迅速理解并掌握相关知识,全面提高动手能力。

本书适用面广,可作为本科计算机科学与技术、软件工程、网络工程、计算机软件、计算机信息管理、电子商务和经济管理等专业的程序设计课程的教材。

图书在版编目(CIP)数据

Java Web 程序设计及实践/青岛英谷教育科技有限公司编著.

—西安:西安电子科技大学出版社,2016.1

高等学校计算机类专业“十三五”课改规划教材

ISBN 978-7-5606-3976-5

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

中国版本图书馆 CIP 数据核字(2016)第 002827 号

策 划 毛红兵

责任编辑 买永莲

出版发行 西安电子科技大学出版社(西安市太白南路 2 号)

电 话 (029)88242885 88201467 邮 编 710071

网 址 www.xduph.com 电子邮箱 xdupfb001@163.com

经 销 新华书店

印刷单位 陕西华沐印刷科技有限责任公司

版 次 2016 年 1 月第 1 版 2016 年 1 月第 1 次印刷

开 本 787 毫米×1092 毫米 1/16 印 张 27.5

字 数 654 千字

印 数 1~3000 册

定 价 70.00 元

ISBN 978-7-5606-3976-5/TP

XDUP 4268001-1

如有印装问题可调换

高等学校计算机类专业 “十三五”课改规划教材编委会

主编 陈龙猛

编委	王 燕	王成端	薛庆文	孔繁之
	李 丽	张 伟	李树金	高仲合
	吴自库	张 磊	吴海峰	郭长友
	王海峰	刘 斌	禹继国	王玉锋

◆◆◆ 前 言 ◆◆◆

本科教育是我国高等教育的基础，而应用型本科教育是高等教育由精英教育向大众化教育转变的必然产物，是社会经济发展的要求，也是今后我国高等教育规模扩张的重点。应用型创新人才培养的重点在于训练学生将所学理论知识应用于解决实际问题，这主要依靠课程的优化设计以及教学内容和方法的更新。

另外，随着我国计算机技术的迅猛发展，社会对具备计算机基本能力的人才需求急剧增加，“全面贴近企业需求，无缝打造专业实用人才”是目前高校计算机专业教育的革新方向。为了适应高等教育体制改革的新形势，积极探索适应 21 世纪人才培养的教学模式，我们组织编写了高等院校计算机类专业系列课改教材。

该系列教材面向高校计算机类专业应用型本科人才的培养，强调产学研结合，经过了充分的调研和论证，并参照多所高校一线专家的意见，具有系统性、实用性等特点，旨在帮助读者系统掌握软件开发知识，同时着重培养其综合应用能力和解决问题的能力。

该系列教材具有如下几个特色。

1. 以培养应用型人才为目标

本系列教材以培养应用型软件人才为目标，在原有体制教育的基础上对课程进行了改革，强化“应用型”技术的学习，使读者在经过系统、完整的学习后能够掌握如下技能：

- ◇ 掌握软件开发所需的理论和技术体系以及软件开发过程规范体系；
- ◇ 熟练地进行设计和编码工作，并具备良好的自学能力；
- ◇ 具备一定的项目经验，包括代码调试、文档编写、软件测试等内容；
- ◇ 达到软件企业的用人标准，做到学校学习与企业的无缝对接。

2. 以新颖的教材架构来引导学习

本系列教材采用的架构打破了传统的以知识为标准编写教材的方法，采用理论篇与实践篇相结合的组织模式，引导读者在学习理论知识的同时，加强实践动手能力的训练。

- ◇ 理论篇：学习内容的选取遵循“二八原则”，即，重点内容由企业中常用的 20% 的技术组成。每个章节设有本章目标，明确本章学习重点和难点，章节内容结合示例代码，引导读者循序渐进地理解和掌握这些知识和技能，培养学生的逻辑思维能力，掌握软件开发的必备知识和技巧。
- ◇ 实践篇：集多点于一线，任务驱动，以完整的具体案例贯穿始终，力求使学生在动手实践的过程中，加深对课程内容的理解，培养学生独立分析和解决问题的能力，并配备相关知识的拓展讲解和拓展练习，拓宽学生的知识面。

另外，本系列教材借鉴了软件开发中的“低耦合，高内聚”的设计理念，组织结构上遵循软件开发中的 MVC 理念，即在保证最小教学集的前提下可以根据自身的实际情况对

整个课程体系进行横向或纵向裁剪。

3. 提供全面的教辅产品来辅助教学实施

为充分体现“实境耦合”的教学模式，方便教学实施，该系列教材配备可配套使用的项目实训教材和全套教辅产品。

- ✧ 实训教材：集多线于一面，以辅助教材的形式，提供适应当前课程(及先行课程)的综合项目，遵循软件开发过程，进行讲解、分析、设计、指导，注重工作过程的系统性，培养读者解决实际问题的能力，是实施“实境”教学的关键环节。
- ✧ 立体配套：为适应教学模式和教学方法的改革，本系列教材提供完备的教辅产品，主要包括教学指导、实验指导、电子课件、习题集、实践案例等内容，并配以相应的网络教学资源。教学实施方面，本系列教材提供全方位的解决方案(课程体系解决方案、实训解决方案、教师培训解决方案和就业指导解决方案等)，以适应软件开发教学过程的特殊性。

本书由青岛农业大学、青岛英谷教育科技有限公司编写，参加编写工作的有陈龙猛、王燕、宁维巍、宋国强、何莉娟、杨敬熹、田波、侯方超、刘江林、方惠、莫太民、邵作伟、王千等。本书在编写期间得到了各合作院校专家及一线教师的大力支持与协作，在此衷心感谢每一位老师与同事为本书出版所付出的努力。

由于编者水平有限，书中难免有不足之处，欢迎大家批评指正！读者在阅读过程中发现问题，可以通过邮箱(yinggu@121ugrow.com)发给我们，以帮助我们进一步完善。

本书编委会
2015年9月

❖❖❖ 目 录 ❖❖❖

理 论 篇

第 1 章 Servlet 概述.....3	2.5.2 ServletContext 的生命周期.....43
1.1 动态网站技术概述.....4	2.5.3 ServletContext 示例.....43
1.1.1 动态网站技术.....4	2.5.4 初始化参数和 ServletConfig.....45
1.1.2 B/S 架构.....5	本章小结.....47
1.2 Servlet 基本知识.....5	本章练习.....47
1.3 第一个 Servlet.....7	
1.4 Servlet 的生命周期.....9	第 3 章 JSP 基础.....49
1.5 Servlet 数据处理.....11	3.1 JSP 概述.....50
1.5.1 读取表单数据.....11	3.1.1 JSP 的特点.....50
1.5.2 处理 HTTP 请求报头.....17	3.1.2 JSP 与 Servlet 的比较.....50
1.5.3 设置 HTTP 响应报头.....20	3.1.3 第一个 JSP 程序.....51
1.6 重定向和请求转发.....22	3.1.4 JSP 执行原理.....52
1.6.1 重定向.....22	3.2 JSP 基本结构.....53
1.6.2 请求转发.....23	3.2.1 JSP 指令.....53
本章小结.....26	3.2.2 JSP 声明.....54
本章练习.....27	3.2.3 JSP 表达式.....54
	3.2.4 JSP 脚本.....55
	3.2.5 JSP 动作标签.....56
	3.2.6 JSP 注释.....57
	本章小结.....58
	本章练习.....59
第 2 章 Servlet 会话跟踪.....29	
2.1 会话跟踪基本知识.....30	第 4 章 JSP 指令和动作.....61
2.2 Cookie.....30	4.1 JSP 指令.....62
2.2.1 Cookie 的创建及使用.....31	4.1.1 page 指令.....62
2.2.2 Cookie 示例.....32	4.1.2 include 指令.....66
2.3 Session.....35	4.1.3 taglib 指令.....68
2.3.1 Session 的创建.....35	4.2 JavaBean.....68
2.3.2 Session 的使用.....36	4.2.1 JavaBean 简介.....68
2.3.3 Session 的生命周期.....37	4.2.2 JavaBean 的应用.....69
2.3.4 Session 的演示.....38	
2.4 URL 重写.....41	
2.5 ServletContext 接口.....42	
2.5.1 ServletContext 的方法.....42	

4.3 JSP 标准动作	69	第 7 章 监听和过滤	117
4.3.1 <jsp:useBean>	70	7.1 监听器	118
4.3.2 <jsp:setProperty>	71	7.1.1 监听器概述	118
4.3.3 <jsp:getProperty>	71	7.1.2 上下文监听	118
4.3.4 <jsp:include>	73	7.1.3 会话监听	122
4.3.5 <jsp:forward>	74	7.1.4 请求监听	124
4.3.6 <jsp:param>	74	7.2 过滤器	127
本章小结	75	7.2.1 Filter 简介	127
本章练习	75	7.2.2 实现 Filter	129
第 5 章 JSP 内置对象	77	7.2.3 过滤器链	132
5.1 内置对象概述	78	本章小结	133
5.2 常用内置对象	78	本章练习	133
5.2.1 out	78	第 8 章 AJAX 基础	137
5.2.2 request	79	8.1 AJAX 简介	138
5.2.3 response	82	8.2 AJAX 工作原理	138
5.2.4 session	83	8.3 XMLHttpRequest 对象	139
5.2.5 application	84	8.3.1 XMLHttpRequest 对象简介	139
5.3 其他内置对象	84	8.3.2 XMLHttpRequest 的方法和属性	140
5.3.1 page	84	8.3.3 XMLHttpRequest 对象的运行周期	141
5.3.2 pageContext	85	8.4 AJAX 示例	142
5.3.3 config	87	8.4.1 时钟	142
5.3.4 exception	88	8.4.2 动态更新下拉列表	144
本章小结	89	8.4.3 工具提示	151
本章练习	89	本章小结	157
第 6 章 EL 和 JSTL	91	本章练习	157
6.1 EL	92	第 9 章 Web Service 概述	159
6.1.1 EL 基础语法	92	9.1 Web Service 简介	160
6.1.2 EL 使用	93	9.1.1 引言	160
6.1.3 EL 隐含对象	94	9.1.2 Web Service 的特点	161
6.1.4 EL 运算符	96	9.1.3 Web Service 的组成	162
6.2 JSTL	98	9.1.4 Web Service 的优势与局限	163
6.2.1 JSTL 简介	98	9.2 Web Service 体系结构	165
6.2.2 核心标签库	100	9.2.1 Web Service 理论模型	165
6.2.3 I18N 标签库	107	9.2.2 Web Service 协议	167
6.2.4 EL 函数库	112	9.2.3 Web Service 通信模型	169
本章小结	114	9.2.4 实现 Web Service	170
本章练习	115	本章小结	173

本章练习	174	10.2.2 WSDL 文档结构	187
第 10 章 SOAP、WSDL 和 UDDI	175	10.2.3 WSDL 绑定	192
10.1 SOAP	176	10.3 UDDI	193
10.1.1 SOAP 介绍	176	10.3.1 UDDI 注册中心	194
10.1.2 SOAP 消息结构	177	10.3.2 UDDI 数据结构	194
10.1.3 SOAP 消息交换模型	182	10.3.3 UDDI API	196
10.1.4 SOAP 应用模式	184	10.3.4 WSDL 映射到 UDDI	198
10.2 WSDL	186	本章小结	204
10.2.1 WSDL 概述	186	本章练习	205

实 践 篇

实践 1 Servlet 基础	209	实践 4.2	289
实践指导	209	实践 4.3	303
实践 1.1	209	知识拓展	309
实践 1.2	214	拓展练习	311
知识拓展	219	实践 5 JSP 内置对象	312
拓展练习	225	实践指导	312
实践 2 Servlet 会话跟踪	226	实践 5.1	312
实践指导	226	实践 5.2	316
实践 2.1	227	实践 5.3	318
实践 2.2	238	实践 5.4	323
知识拓展	246	知识拓展	330
拓展练习	251	拓展练习	335
实践 3 JSP 基础	253	实践 6 EL 和 JSTL	336
实践指导	253	实践指导	336
实践 3.1	253	实践 6.1	336
实践 3.2	261	实践 6.2	337
实践 3.3	268	实践 6.3	348
知识拓展	276	实践 6.4	354
拓展练习	281	实践 6.5	360
实践 4 JSP 指令和动作	282	知识拓展	361
实践指导	282	拓展练习	365
实践 4.1	282	实践 7 监听和过滤	366

实践指导	366	实践 8.3.....	402
实践 7.1.....	366	实践 8.4.....	406
实践 7.2.....	369	知识拓展	410
实践 7.3.....	373	拓展练习	417
实践 7.4.....	382		
知识拓展	391	实践 9 SOAP	418
拓展练习	392	实践指导	418
		实践.....	418
实践 8 AJAX 基础	393	知识拓展	424
实践指导	393	拓展练习	427
实践 8.1.....	393		
实践 8.2.....	397	附录 常用的 Servlet API.....	428

理论篇





第1章 Servlet 概述



本章目标

- 了解动态网站开发技术
- 理解 Servlet 运行原理及生命周期
- 掌握 Servlet 的编写及部署
- 掌握 Servlet 对表单数据的处理
- 掌握 Servlet 对 HTTP 请求报头的处理



1.1 动态网站技术概述

在“Web 编程基础”课程中已经学习了使用 HTML、CSS、JavaScript 等相关的静态网站技术，静态网页文件的扩展名为“.htm”或“.html”，这些页面不能直接与服务器进行数据交互。随着站点内容和功能需求的不断复杂化，单一的静态网站技术往往不能满足应用的要求。例如，静态网站无法将数据传输到服务器上进行处理并存储到数据库中，此时就需要动态网站技术。

1.1.1 动态网站技术

动态网站并不是指具有动画功能的网站，而是指基于数据库架构的网站，一般由大量的动态网页(如 JSP)、后台处理程序(如 Servlet)和用于存储内容的数据库组成。所谓的动态网页，本质上与网页上的各种动画、滚动字幕等视觉上的“动态效果”无关。动态网页可以是纯文字内容，也可以包含各种动画的内容，这些只是网页具体内容的表现形式；无论网页是否具有动态效果，采用动态网站技术生成的网页都称为动态网页。

动态网站一般具有以下几个特点：

- ✧ 交互性：网页会根据用户的要求和选择做出动态改变和响应。例如，用户在网页中填写表单信息并提交，服务器可以对数据进行处理并保存到数据库中，然后跳转到相应页面。因此，动态网站可以实现用户注册、信息发布、订单管理等功能。
- ✧ 自动更新：无需手动更新 HTML 文档，便会自动生成新的页面，大大节省工作量。例如，在论坛中发布信息时，后台服务器可以产生新的网页。
- ✧ 随机性：在不同的时间，不同的用户访问同一网页时可能产生不同的页面。



注意

动态网站一般采用动静结合的原则：网站中内容频繁更新的可采用动态网页技术；网站中内容不需要更新的则可采用静态网页进行显示。通常一个网站既包含动态网页也包含静态网页。

动态网站技术在早期有 CGI(Common Gateway Interface, 公用网关接口)技术。CGI 提供了一种机制，可以实现客户和服务器之间真正双向交互。这种技术为在线用户支持和电子商务等新思想的实现铺平了道路，同时，因 CGI 技术可以使用不同的语言编写适合的 CGI 程序，如 Visual Basic、Delphi 或 C/C++等，并且功能强大，故被早期的很多网站采用。但由于其编程困难、效率低下、修改复杂，所以逐渐被新技术所取代。目前被广泛应用的动态网站技术主要有以下三种：

- ✧ PHP(Hypertext Preprocessor)：超文本预处理器，其语法大量借鉴了 C、Java、Perl 等语言，只需要很少的编程知识就能使用 PHP 建立一个真正交互的 Web 站点。由于 PHP 开放源代码，并且是免费的，所以非常流行，是当今 Internet 上最为火热的脚本语言之一。



- ✧ **ASP(Active Server Pages)**: 一种类似 HTML、Script 与 CGI 结合体的技术, 它没有提供自己专门的编程语言, 允许用户使用许多已有的脚本语言编写 ASP 应用程序。但 ASP 技术局限于微软的操作系统平台之上, 主要工作环境为微软的 IIS 应用程序结构, 而且 ASP 技术很难实现在跨平台 Web 服务器上工作, 因此一般只适合一些中小型站点。但目前由 ASP 升级演变而来的 ASP.NET 可支持大型网站的开发。
- ✧ **JSP(Java Server Pages)**: 基于 Java Servlet 以及整个 Java 体系的 Web 开发技术。JSP 是由 Sun 公司于 1999 年 6 月推出的新技术, 它与 ASP 有一定的相似之处, 但 JSP 能在大部分的服务器上运行, 而且其应用程序易于维护和管理, 安全性方面也被认为是这三种基本动态网站技术中最好的。

1.1.2 B/S 架构

在动态网站技术中, 一般使用浏览器作为客户端, 客户在浏览器中发出请求, Web 服务器得到请求后查找资源, 然后向客户返回一个结果, 这就是 B/S(Browser/Server)架构, 如图 1-1 所示。

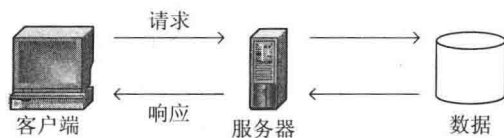


图 1-1 B/S 模型

在 B/S 架构中, 用户的请求与 Web 服务器的响应需要通过 Internet 从一台计算机发送到另一台计算机, 不同计算机之间是使用 HTTP(HyperText Transfer Protocol)协议进行通信的。HTTP 是超文本传输协议, 包含命令和传输信息, 不仅用于 Web 访问, 也可以用于其他因特网/内联网应用系统之间的通信, 从而实现各种资源信息的超媒体访问集成。



注意

C/S 架构: 即 Client/Server(客户机/服务器)结构, 这种架构将任务合理分配到 Client 端和 Server 端。用户的程序主要是在客户端, 服务器端程序主要提供数据管理、数据共享、数据及系统维护和并发控制等, 客户端程序只需完成用户的具体业务。

1.2 Servlet 基本知识

Java EE 是基于分布式和多层结构的企业级应用开发规范和标准, Servlet 是 Java EE 架构中的关键组成。目前, 在企业应用开发中不仅会使用传统的 Java EE 组件(例如 JDBC、Servlet、EJB 等), 还会使用一些轻量级的框架结构(例如 Struts、Hibernate 和 Spring), 以提高企业的开发效率。在 Java 企业级开发应用中会使用到的技术如图 1-2 所示。

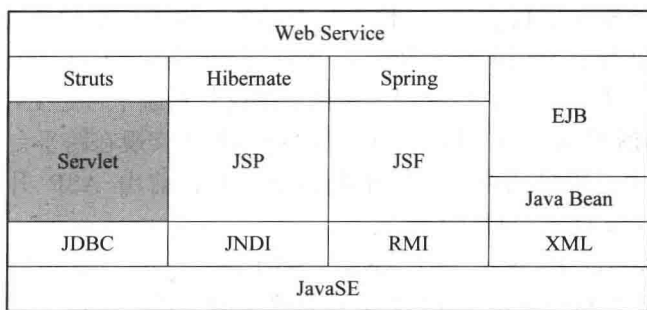


图 1-2 Java EE 技术组成

Servlet 技术是 Sun 公司提供的一种实现动态网页的解决方案，它是基于 Java 编程语言的 Web 服务器端编程技术，主要用于在 Web 服务器端获得客户端的访问请求信息并动态生成对客户端的响应信息。此外，Servlet 技术也是 JSP 技术的基础。

Servlet 是 Web 服务器端的 Java 应用程序，它支持用户交互式地浏览和修改数据，生成动态的 Web 页面。比如，当浏览器发送一个请求到服务器时，服务器会把请求送往一个特定的 Servlet，这样 Servlet 就能处理请求并构造一个合适的响应(通常以 HTML 网页形式)返回给客户，如图 1-3 所示。

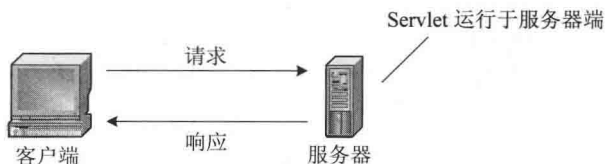


图 1-3 Servlet 的作用

Servlet 与普通 Java 程序相比，只是输入信息的来源和输出结果的目标不一样，例如，对于 Java 程序而言，用户一般通过 GUI 窗口输入信息，并在 GUI 窗口上获得输出结果，而对于 Servlet 程序而言，用户一般通过浏览器输入并获取响应结果。通常普通 Java 程序所能完成的大多数任务，Servlet 程序都可以完成。Servlet 程序具有以下特点：

- ✧ 高效。在传统的 CGI 中，如果有 N 个并发的对同一 CGI 程序的请求，则该 CGI 程序的代码在内存中重复装载 N 次；而对于 Servlet，处理请求的是 N 个线程，只需要一份 Servlet 类代码。在性能优化方面，Servlet 也比 CGI 有着更多的选择，比如缓冲以前的计算结果，保持数据库连接的活动等。
- ✧ 方便。Servlet 提供了大量的实用工具例程，例如自动解析和解码 HTML 表单数据、读取和设置 HTTP 头、处理 Cookie、跟踪会话状态等。
- ✧ 功能强大。在 Servlet 中，许多使用传统 CGI 程序很难完成的任务都可以轻松地完成。例如，Servlet 能够直接和 Web 服务器交互，而普通的 CGI 程序不能。Servlet 还能够在各个程序之间共享数据，轻松实现数据库连接池之类的功能。
- ✧ 良好的可移植性。Servlet 是用 Java 语言编写的，所以具备 Java 的可移植性特点。此外，Servlet API 具有完善的标准，支持 Servlet 规范的容器都可以运



行 Servlet 程序，例如 Tomcat、Resin 等。

1.3 第一个 Servlet

编写 Servlet 需要遵循以下规范：

- ✧ 创建 Servlet 时，需要继承 `HttpServlet` 类。同时需要导入 Servlet API 的两个包：`javax.servlet` 和 `javax.servlet.http`。`javax.servlet` 包提供了控制 Servlet 生命周期所必需的 Servlet 接口，是编写 Servlet 时必须实现的；`javax.servlet.http` 包提供了从 Servlet 接口派生出的专门用于处理 HTTP 请求的抽象类和一般的工具类。
- ✧ 根据数据的发送方式，覆盖 `doGet()`、`doPost()` 方法之一或全部。`doGet()` 和 `doPost()` 方法都有两个参数，分别为 `HttpServletRequest` 和 `HttpServletResponse` 类型。这两个参数分别用于表示客户端的请求和服务端端的响应。通过 `HttpServletRequest`，可以从客户端中获得发送过来的信息；通过 `HttpServletResponse`，可以让服务端对客户端做出响应，最常用的就是向客户端发送信息。关于这两个参数，将在后续内容中详细讲解。



注意

如果在浏览器中直接输入地址来访问 Servlet 资源，属于使用 Get 方式访问。

【示例 1.1】 使用 Servlet 输出 “Hello World” 页面。

创建一个类文件 `HelloServlet.java`，编写代码如下：

```
//创建一个Servlet类，继承HttpServlet
public class HelloServlet extends HttpServlet
{
    // 重写doGet()
    public void doGet(HttpServletRequest request,
        HttpServletResponse response)
        throws ServletException, IOException {
        //设置响应到客户端的文本类型为HTML
        response.setContentType("text/html");
        //获取输出流
        PrintWriter out = response.getWriter();
        out.println("Hello World");
    }
}
```

上述代码会向客户端浏览器中打印 “Hello World” 信息。通过 `response` 对象的 `getWriter()` 方法可以获取向客户端输出信息的输出流：

```
PrintWriter out = response.getWriter();
```

调用输出流的 `println()` 方法可以在客户端浏览器中打印消息。例如：