

CSS



张亚飞

飞思科技产品研发中心

编著
监制

建立正确的认知、完整系统的知识体系

- 究竟什么是CSS，让本书来给你答案
- 完全范例讲解，内容通俗易懂
- 范例完全通过W3C推荐验证程序验证，真正使你的网页完全符合行业标准
- 完全涵盖CSS与HTML，全面地学习网页语言
- 直击CSS技术的核心，用开发理论的阐述，辅以实际案例的开发，学习效果最佳
- 循序渐进地指导读者将CSS融入到网站建设中，掌握如何优化网站总体布局



开发王



CSS

Learn CSS from the ground up, from the basics of selectors and styling to advanced topics like animations and transitions. This book is a comprehensive guide to CSS, covering everything you need to know to create beautiful, responsive websites.



开始吧

CSS

张亚飞 编著
飞思科技产品研发中心 监制



电子工业出版社

Publishing House of Electronics Industry

北京·BEIJING



内容简介

本书源于作者对CSS功能的认识和多年的应用实践，详细介绍CSS语言与HTML结合实现网页的显示和布局，具有严格的逻辑性。本书首先帮助用户建立首要的、基础的正确认知，然后以深入浅出的实例，说明如何利用CSS来辅助HTML网页实现显示和布局这两大功能。本书教你轻松掌握通过CSS盒子模型实现复杂的网页布局，并通过一个完整的餐厅网站案例，循序渐进地指导读者将CSS融入到网站建设中，掌握如何优化网站总体布局。

本书直击CSS技术的核心，理论与实务并进，通过对开发理论的阐述，辅以实际案例的开发，学习效果最佳！

本书每一章绝非泛泛而谈理论，而是围绕实际案例代码来组织的，让读者马上可以上手，并且加深理解。“实用”是技术类书籍好坏的唯一衡量标准，而本书更是将这个词语演绎到了极致。

网页设计师可以通过本书学习到更为高级的网页设计应用技巧，程序开发者也可以通过本书学习到如何利用CSS制作出界面更华丽的应用程序。

本书相关源码请到<http://www.fecit.com.cn>的“下载专区”进行下载。

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

CSS开发王 / 张亚飞编著. —北京: 电子工业出版社, 2008.6
(网站开发专家)

ISBN 978-7-121-06414-2

I. C… II. 张… III. 主页制作—软件工具, CSS IV. TP393.092

中国版本图书馆CIP数据核字 (2008) 第053092号

责任编辑: 王树伟

印 刷: 北京天宇星印刷厂

装 订: 三河市皇庄路通装订厂

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: 860×1092 1/16 印张: 24.75 字数: 673.2千字 彩插: 8

印 次: 2008年6月第1次印刷

印 数: 5 000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zlt@phei.com.cn。盗版侵权举报请发邮件至dbqq@phei.com.cn。

服务热线: (010) 88258888。



自从 20 世纪 70 年代因特网诞生以来，以 www 技术为支撑的万维网近十多年来迅猛发展，渗透到社会生活的各个方面。从旧金山郊外的硅谷，到新竹日夜通明的半导体工厂，它正在改变着我们工作和生活的方式。

全世界有 3/4 的人还从未接触过因特网，这也预示着我们这些正在从事与因特网相关工作的人员，以及即将从事此类工作的设计者、开发者都将有一个美好的明天。

必须掌握的万维网常识

用户首先必须掌握万维网的基本常识，这是万维网的基本知识结构，也是“开发王”这套书所基于的基本理论基石和贯穿于始终的纲要。并且，这也不单单是一个知识结构的问题，其实任何的技术知识都是随着技术本身的发展慢慢积累起来的，是技术不断变化、不断延伸的结果，所以知识结构往往也是一个怎样正确认识技术发展史的问题。下面我们就来简要介绍一下万维网的技术简史和万维网的基本知识结构，这也是阅读“开发王”这套书的基本前提。

你不得不了解它

它是进行万维网应用程序开发的认识论基石，也是实践的指南针，设想存在一个不遵守该指南的万维网应用程序是不可能的。

www 也就是我们经常提到的万维网，它是 Internet 上那些支持 www 和超文本传输协议 (HTTP) 的客户机与服务器的集合，通过它可以存取世界各地的超文本文件，内容包括文字、图形、声音、动画、资料库，以及各式各样的软件等。

HTML 是万维网所基于的标准的格式化语言，它自从被发明以来就构成了整个万维网设计和开发的核心，这也是我们每个从业人员都必须熟知于心、了然于胸的，它是我们从业于万维网的基石。

鉴于 HTML 扮演的重要角色和万维网不断发

展的客观要求，我们需要不断改进和细化它的功能——将结构、内容和样式分离已是大势所趋，这也是 CSS(层叠样式表)出现的原因。

在样式表没有出现之前，HTML 用于呈现的功能实际肩负着两种责任：一种是语义的呈现，一种是表现形式的呈现。例如，b 元素(英文 bold 的缩写)既要表现加粗的语义，实际上还要用做呈现的标识。

样式表的出现简化了 HTML 语言并且减轻了

HTML 对表现形式的责任，它使设计者和用户都可以控制文档的表现形式，包括字体信息、对齐方式、颜色等。

除此之外别无其他

每个开发者都必须将知识建立在两个最基本的基石之上，除此之外别无其他。每一本图书和教材都应该帮你做到，这也正是你从业于万维网的最坚实的技术基础。

HTML 和 CSS 一道构成了万维网应用的客户端基石，它们又被称为万维网应用程序的展示层。

万维网在不断前进，人们对于新功能的渴望也在不断增强，而这些新功能理所应当能够针对不同用户的需求做出不同的响应，这也就是动态网页技术层出不穷的原因。

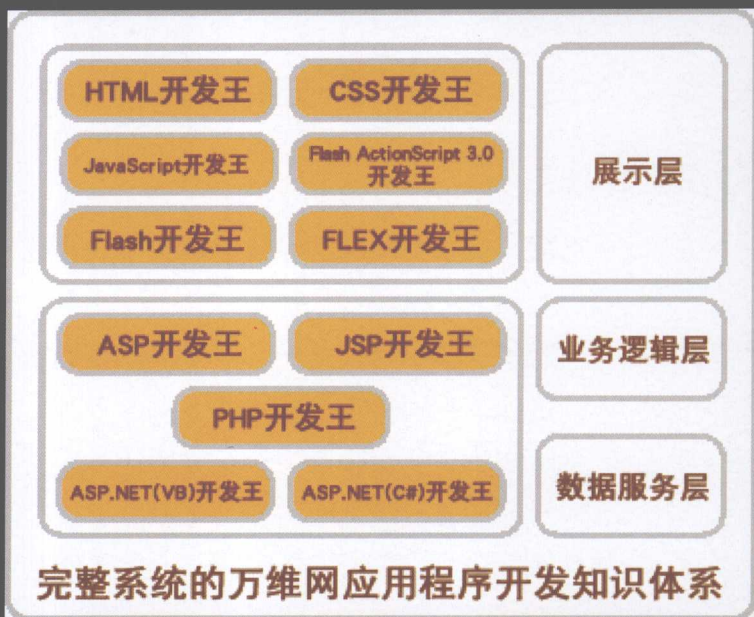
在这个目标的驱使下，微软创造了 ASP 技术，进而又创造了 ASP.NET；SUN 创造了 Servlet，进而又创造了 JSP；Adobe 创造了 ColdFusion；而万维网的技术社区会聚着不计其数的开发者，他们中的杰出者创造了 PHP 并使之成为流行的动态网页开发技术。

这些动态网页技术可以针对用户的不同请求做出不同的响应。“请求 / 响应”正是万维网应用程序的基本模式（不管是静态网页还是动态网页）。不同的动态网页开发技术有不同的对应于“请求 / 响应”的语言特点，但它们也

都有共同的实现特点，那就是针对请求生成不同的 HTML 网页，并把它传递到用户的客户端软件中。正如你经常用到的那样，这个客户端软件被称为“浏览器”。

由于动态网页总是位于服务器上，而它生成的 HTML 网页又总是被浏览器浏览，所以这个开发经常被描述成 B/S，也就是 Browser（浏览器）和 Server（服务器）的缩写。

所有的基于 www 技术的动态网页技术都会生成 HTML 网页（网页中也许会包含样式表，也有可能仅仅是裸数据而不包含 HTML 标签），这也是为何你会看到不同的动态网页技术却可以创



建同样呈现结果的根本原因。

同样，在客户端浏览器中你也看不到动态网页的源代码，因为这些源代码在服务器上就已经被解释、执行，最后被转化成 HTML 格式的代码并被传递到客户端。这些网页之所以被称为动态网页，也正是因为它们能够根据不同的请求生成不同的 HTML 网页，这个作为结果的 HTML 网页是“动态”产生的。

这些动态网页技术和支持它们的底层技术一道也被称为中间件，它们构成了网络应用的中间层。由于这个

夸夸其谈并不可取

所传达给用户的知识要么必须有坚实的事实基础，要么是由严谨的逻辑推导而出，而这样的逻辑也应被所有读者轻松地理解。

因此，这个层又被称为业务逻辑层。

中间层包含了用于分析用户不同请求，并对不同请求做出不同响应的业务逻辑代码，



位于业务逻辑层之下的是数据服务层，用户请求所要获取的数据从这里被“筛选”出来，并经过业务逻辑层被发送到客户端——它们被包含在 HTML 文件中。

出于人们对数据海量存储和便于查询访问的渴望，推动着整个数据服务产业的发展，关系数据库 (Relational Database) 凭借其自身的特点从它被发明那天起就成为整个数据服务产业的核心。Oracle 也正是凭借其著名的关系数据库产品 Oracle 9i 成为软件产业中仅次于微软的软件巨擘。IBM DB2、微软 SQL Server 和开源的 MySQL 也都是关系数据库产品中的佼佼者。

而微软 Office 产品组中的 Access 以其高智能、轻量化和简单易用性的特点在办公自动化软件 (OA) 领域内广泛应用，它也是很多开发人员学习数据库开发的极佳入门工具。

数据服务层不仅仅是数据库，而且还包含了其他可以用做数据的资源，例如 XML。XML 已经成为因特网间标准的数据交换格式，它也正在成为数据存储的标准格式。

展示层、业务逻辑层、数据服务层构成了当前万维网应用程序的基础架构，它们也被统称为“三层架构”。

实际上你不得不了解它，它是进行万维网应用程序开发的认识论基石，也是实践的指南针。设想存在一个不遵守该指南的万维网应用程序是不可能的，同样，设想任何不了解该指南而进行的万维网应用程序开发也是近似荒谬的。

在“开发王”系列图书中，我们针对不同的

万维网技术，对三层架构做了精确的、同一的阐述，这是你在全球其他任何地方都无法完整系统地获得的。不管你愿不愿意承认，你永远都无法摆脱它。

在万维网软件开发日益区分为泾渭分明的三层架构后，市场和技术并未停止它们前进的步伐，它们依旧在不断地向前延伸。在技术厂商的推动下，在万维网的技术社区的遥相呼应下，三层架构中的每个层次都为了某些特殊的应用不断向其邻近的层次延伸，首当其冲的便是展示层向逻辑层的渗透。

展示层利用客户端脚本和浏览器插件技术开始分担业务逻辑层的部分运算，而且，它们还有一个响亮的名字，被称为“富客户端”。

JavaScript、Flash 和 Java Applet 目前是展示层分担业务逻辑层运算的主要软件技术。JavaScript 因为 AJAX 解决方案和 DHTML 解决方案的结合而“重获”新生，而 Flash 凭借其分布广泛的 Flash Player 插件及高效的开发环境而正日益成为下一代万维网客户端运算的核心。与此形成鲜明对比的是 Java Applet 技术，因为其开发的难度极高，以及其客户端运行环境自身的原因，从其诞生那天起就从未引人注目过，今天，它依然被搁置在某个角落。

正如你在“开发王”系列图书中所看到的那样，我们已经将 JavaScript、Flash 技术完全融入到整个万维网开发体系中，现在你理应将它们视为一个整体，而不是再孤立地去看它们中的任何一个。



完整系统的万维网知识体系 能为你带来什么

是的，这个问题是每一个开发者、雇主、任课教师和学员所关心的，而且，很显然，每个用户理应关心。实际上，每一个人都应该问自己这样一个问题。

用户还会问：你为何为该套书起名为“开发王”？它（的内容）真的如描述的那么好吗？

这真的很直截了当，而且我本人也喜欢直截了当。当然，每一个开发者也都有权利这样提问，而我也正想在这里为大家分享这个问题的答案。



这本书为何被冠以“王”

每本图书都可能会被冠以“王”这样的词汇，以表明这本书是多么伟大，但是并非所有图书都能名副其实。

作为这套书的作者，我的眼中只有“王”——这就是我的回答，并且，毫不犹豫。

当然，智者从不相信信口而来，每一个读者理应都是智者，而唯一值得相信的理应是事实。

这套书从最初的准备，到今天的出版，前后经过长达6年的时间，前后经过了数以千计次大小修改。你现在看到的这个序言我也写了二三年时间，序言的目标就是做到“增一字嫌冗余，减一字则无从下手”。

写作的过程是漫长而痛苦的（偶尔也会有快

乐），出版的历程也是如此。如果作者想出版一本书，就必须首先给出版机构一个策划书——也就是这本书的基本情况和设想。

每一个作者都会想该如何打动出版机构出版它，这是毫无疑问的。

你得苦思冥想……

但实际上，这很简单，你必须告诉他们这本书是最棒的，我想，除此之外别无他法。

如果没有证据，我想你也根本不会相信，更何况那些“眼疾手快”的编辑呢？

当然，那是因为我给他们提供了充足可信的证据。如果你有某个事实作为证据，我想你也可以这样做，让我们来看一个简单的现实吧：

根据 JavaBeans 规范的要求，一个 JavaBeans 必须支持序列化或者外部化，这在 JavaBeans 规范 5.5 节的第一行就明确地定义了这个要求。

证据确凿，低级错误正伺机阻碍您企业的发展。

如果我没有记错，那上面白纸黑字写着下面的一行英文：

```
All beans must support either Serialization or Externalization.
```

这意味着只有实现序列化或者外部化，或者继承于某个实现了序列化或者外部化的基类，才能将一个 Java 类视做 JavaBeans。但是遗憾的是，

当你在查阅有关 Java 网络开发（包括 JSP 开发、JSF、Servlet、Struts 等）的图书和资料时，又有多少图书和资料教你遵守规范的要求呢？绝大多数的 Java 网络开发图书和资料在没有实现序列化或者外部化时就告诉我们年轻的开发者：这是一个 JavaBeans。很明显，这是跟我们年轻的开发者开了一个天大的玩笑！

当然，仅仅是玩笑尚不足对我们年轻的开发者构成致命的打击，关键是按照这样的阐述开发出的应用程序将会给雇主造成致命的、无法挽回的损失。

由此我也想到，曾经有开发人员在读完一些计算机图书，再经历自己多年的开发实践后有如下感慨：国内很多技术图书领域都是这样，好多计算机图书只要最初的那一本错了，后面的就全是 Copy，甚至都不去关心事物的本源是什么。

但，这仅仅不过是悲剧的开端……

如果究其产生的根源，那么基础认知上存在根源性错误是导致这种情况发生的直接原因。

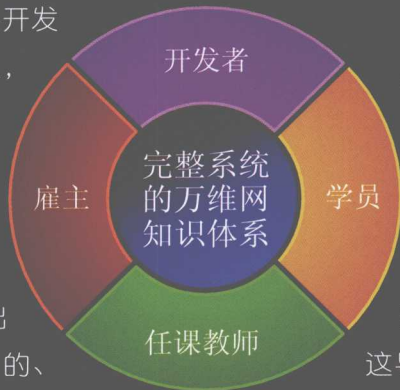


严格遵守规范和标准，警惕低级错误可能造成的悲剧

JavaBeans 问题的发生是整个 Java 网络开发领域内的悲剧，实际上也是整个网络开发领域内的悲剧。连这样基础的东西都漠视，视标准和规范的要求于不顾，更不要说了解使用 JavaBeans 的主要目的是什么了，那么我们可爱的、年轻的开发者如何才能得到正确的网络开发知识呢？

或者，每一个开发者都应自问：我如何才能

下面要介绍的这个认识和实践的错误显得更加复杂，可能对于一个初入门的作者和译者而言显得有些难了，因为要想理解 JavaBeans 的真正作用对他们来说还是有些难度的。



归于 JSP 规范并未规定 `<jsp:useBean>` 动作标签必须使用 JavaBeans，所以大多数的 Java 应用程序服务器（JSP 实现）在转化阶段并未对其进行鉴别（实际上，就我所知的 Java 应用程序服务器中，没有一款对其进行鉴别），这导致不计其数的开发者在使用 `<jsp:useBean>` 动作标签时将一个简单的 Java

Class 当做 JavaBeans 来使用，而不是有甄别地去应用这个动作标签。

仅仅是 `<jsp:useBean>` 动作标签中有 useBean 这个英文单词，于是就造成了这样巨大的错误。

你可以看到，在《JSP 开发王》这本书中，我甚至不建议用户使用 `<jsp:useBean>` 动作标签。当然，如果非要使用它，那么就必须遵守《JSP 开发王》提出的几点约束。

为雇主开发出负责任的应用程序来呢？

开发者一定要破除“雇主付钱请我来编写程序”的错误观念，这实际上是忽略了后面的这句：“目的是我的程序能使雇主有更大的回报”。

你可以将一个范例代码写错，就像写一个错别字，可能在所难免，但你绝不能犯如此的低级错误，这是影响你整个开发体系的大事。



每一个人都知道，悲剧显然不是由某个特定的错误引起的，而是由某个根深蒂固的错误思想导致的，如果这种思想不除去，挡在开发者和雇

主之间的魔障亦不会消除。它也会逐渐抽干雇主多年努力所换来的投入……当雇主独处一室苦苦思索时，你是不会体会到的。



错误还在蔓延？

如果仅仅是这样一个小事例，尚不足使这套书被称为“王”，更多的基础性错误还继续在用户的软件开发知识体系中间蔓延，它们正伺机阻碍你企业的发展。

忘记了这样做的根本是什么，好像这根本就不应被考虑。好在，现在该是结束的时候了……

如果说 JavaBeans 仅仅是个偶然性的错误的话，那么在基本验证和网络应用程序安全这样最基础的问题上，你同样几乎找不到一本像样的教材和资料能够完整系统地正确阐述它，更不要说其他的安全措施了。

AJAX 解决方案的出现仿佛使 JavaScript “重新”回到人们关注的视线中，但如果没有 DHTML 解决方案作为铺垫，AJAX 又能做什么呢？答案很简单：它什么也做不了。

这个错误在 ASP、ASP.NET、JSP、PHP 开发领域内存在的时间之长，几乎可以与这些语言被发明那天起同步。导致这个错误发生的根源与导致 JavaBeans 错误发生的根源惊人地相同，都是根本不顾规范的要求——只要最开始的一本书错了，那么后面的全是 Copy。

AJAX 和 Flash 现在是富客户端的主流，但是，如果你不能完整系统地正确认识三层架构的基本体系，如果不了解 C/S 和 B/S 的区别，那么设想能真正理解 AJAX 和 Flash 用于开发富客户端应用程序只能是空想。

错误也是积累的，在三层架构万维网应用程序的展示层，如何正确认识 HTML、CSS、JavaScript 和 Flash 已经成了当务之急，富客户端应用程序已经成为当前因特网应用程序开发中极受重视的领域，因为它能给消费者带来最佳的客户体验，而这也正是雇主所需要的。

虽然 Flash 已是一个成熟的体系，但是，目前 Flash 开发对很多开发人员来说还是一个较新的领域。两年前，很多开发者还在喋喋不休地争论 AS2 和 AS1 的优、缺点，争论往往意味着还不能正确地认识它们的真正实质。所以，两年前，在拙著中就完整系统地介绍了这方面的知识，帮助用户完成了这一基本认知。现在，它已经被完全融入到这个万维网知识体系中，过去的大批量错误认知已经被更正，我们还建立了完整的基于时间轴的 Flash 应用程序开发的理论体系，所以现在看起来，这个领域内的灾难并没有 AJAX 领域那么深重。

HTML 这个最基础并且最简单的语言可能用户从未认真审视过——可能只是因为它简单了，一个承担了内容结构和内容呈现责任的语言如何在新的体系面前被正确理解呢？CSS 究竟使 HTML 发生了何种变化呢？大多数的教材都是教授开发者和设计者怎样使用 CSS 去呈现网页，却

这些错误的发生不但使得开发者深受其害，重要的是，它有可能使得雇主的网络应用程序因此而遭受重大挫折，并失去客户的信任。

在这里，我提到了很多不应该发生却又事实上发生了的，也是现在正在发生的悲剧，而更为



重要的是，对规范、标准、实现、实践的完备性认识还有很多、很长的路要走。如果没有完整系统的知识体系，开发者还不知道要在“黑夜”里摸索多久。

想一想，除了这里提到的错误，还有哪些错误被隐藏起来了，你还未发现呢？这也是为何我在书中对易出错的部分做了突出显示的原因。

也许不远处真的就有冰山出没……

当我看到有无数的教材和资料在长篇累牍、多达数十页的论述抽象类和接口的区别时，我真的感到悲哀，是在为整个使用面向对象进行软件开发的开发者们悲哀。当我两年前用一句话来区分抽象类和接口时，可能并没有人真正地去注意过它，虽然，在拙著中还用了文字加粗……

就像 6 年前，当在拙著中第一次提出 Flash Player 虚拟机的概念时，我想也并没有多少人真正注意过这个问题；但是在 2 年前，我在拙著中第一次明确阐述了在 ActionScript 2.0 和 ActionScript 1.0 中存在的基本错误认知，现在，两年过去了，我想再也没有人会去犯同样愚蠢的错误了。

有时候，夜真的很黑、很深沉，它会淹没你，并让你忘记你自己……

想来应该有不少任课教师应该可以给他们的学员讲授正确的 Flash 开发了。

但是，如果我说很多开发者连 Flash 基本的变量作用范围都不能搞清楚，你是不是很诧异呢？

是啊，两年前我也经常听一些人说他们会 Flash 开发，可是我却发现他们连最基本的变量的作用范围都一知半解，如果连这个都不能系统掌握，能开发出一个完美的程序来吗？又如何为雇主创造价值呢？

想来，我的前一本 Flash 作品已经有不少人看过了，但是，同样，大多数 Flash 开发人员可能并没有得到这个知识。

是啊，因为并没有特别声张这件事，就是想看一看用户究竟有没有真正去学习 Flash 开发，去认真获取那些最真实的东西。

但是，这终究还是造成了遗憾——当拙著第一次为整个 ActionScript 语言带来了完整系统的变量作用范围的介绍时，可能很少人去注意它。

现在，新版的《Flash 开发王》我想不会再有人错过了。

开发王系列图书 有两个最基本的基石

当然，即使有点点滴滴的、零星的低级错误，也不能说本书不能称为“王”。开发王必须有两个最基本的基石，除此之外别无其他，这两个基石是：

- 帮助用户建立正确的认知
- 帮助用户建立完整系统的知识体系



正确的认知就是一切的根本

正确的认知就是一切的根本，这无须进行任何说明，而关键是什么是正确的认知。

认知有两个层面的意思，一个是认字，一个是知道。

每个人都是从认字开始的，这毫无疑问，认字就是了解事实是什么，知道就是怎样做才能正确地认识到这个事实。

我们经常听到一个成语“授之以鱼，不如授之以渔”。是的，不单单是应该告诉用户一个事实，还要告诉他们怎样去正确地认识到这个事实。

事实往往是某种规定，就像你学习汉语拼音那样，a 这个英文字母就是发音为“啊”。当然，对于应用程序开发来说，这个规定性或由某个标准定义，或由某个规范定义，或由某个程序的业务逻辑所要求，而你的“认字”也就是了解

这种规定性究竟规定了什么。这个“认字”很枯燥，你只有死记硬背，完全遵照，就像完全遵照 JavaBeans 规范那样。

现在关键点是：这个死记硬背的事实用户如何才能认识到它呢？这真的很难办。为此我们提出了 3 个最简单、最基本的问题可以帮助你一举解决这个难题：

- 这是什么？
- 为什么是这样？
- 该怎样做？

你在“开发王”每一本书中，甚至每一章中都可以看到或体会到这样的 3 个最简单、最基本的问题。



完整系统的知识体系就是顺理成章的一切

正确的认知往往是一点一滴汇聚起来的，如果这些点点滴滴的认知被相互肢解开来，那么最终形成的则是一片混乱。这就需要有一个知识体系将这些点点滴滴的认知相互关联起来形成完整系统的知识体系，这也是日渐庞大的万维网的必然要求。

正如你看到的那样，我们将万维网应用程序的知识体系建立在三层架构的基础之上，在这个最基本的认知和知识体系下不断将知识延伸。

我不想在这里对什么是“完整系统的知识体系”进行过多的介绍，因为每个人都理解它的字

面意思，也理解它对每个开发者都是梦寐以求的，而关键是“完整系统的知识体系”的内容是什么。除了应该牢记这篇序言前面所介绍的万维网的技术史和万维网的基本知识结构，我也不想在前言中再多费笔墨，因为知识体系往往融入在各个知识点的相互结合中，不是一两页纸所能阐述清楚的。

如果能用几页纸阐述清楚那就好了，不过话又说过来，一个用几页纸就能阐述清楚的“完整系统的知识体系”能可信吗？

“完整系统的知识体系”包含了某个领域内的

完整内容，不可能使用几页纸就能阐述清楚，而且领域越广，所需要的笔墨就越多。例如，针对 JSP，你可能需要用一本书来介绍，但对整个万

维网应用程序来说，你可能需要 10 本书甚至更多，我想每一个用户都不难理解。

谁将从这套书中受益

开发者、雇主、主讲教师、学员都将从“开发王”系列图书中受益



开发者是理所应当的最大受益者

毫无疑问，“开发王”这套书是最终面向开发者的，不但初学者，而且资深的开发者也应视它为必读之作。在正确的认知和完整、系统的知识体系面前，初学者和资深的开发者处于同一起跑线上。

员工应该完整系统地掌握万维网应用程序开发的知识。只有这样才能更好地服务于企业和雇主。

当然，如果你有错误的认知，并且编出了错误的应用程序，那么就要小心了，应该尽快通读本书，趁雇主还未发现之前将漏洞补上。

“初学者”能够在开发伊始就将技术建立在正确的认知的基础上，并且循序渐进，获得完整系统的知识体系，这不但可以加速学习进程，而且也是正确开发的保证。

“资深的开发者”虽然在实践过程中获得了很多好的实战经验，但未必能够运用自如，这就是过去混乱不堪的知识体系和错误的认知共同作用的结果。如果不砍掉错误和混乱，必然使自己的开发认识踌躇不前，最终走火入魔也未必不可能。



雇主是最终的受益人

雇主首先要关心的是如何在有效的投入情况下获得较高的产出，也就是如何实现经济产出。

正确的软件开发来源于正确的认知和完整系统的知识体系，这是毫无疑问的。

正如你前面看到的，正确的认知是编写正确程序的关键，试想一个没有实现序列化的 JavaBeans 将会带来什么样的危害？假设是一个银行结算程序，资金（一个 JavaBeans）汇出去了，但是却因为中间运算出了故障而死机，这个时候

自动保护装置本应该能够恢复 JavaBeans 的状态（也就是你的资金），但却因为拙劣的错误认知进而编写出拙劣的程序，这一程序从而又导致这一恢复过程无法完成。对



一个企业而言，这是致命的一击——连自己的基础设施都无法保证，企业信誉便荡然无存。

实现经济产出的另一个要关心的事情是如何实现团队协作，这个时候正确的认知和完整系统的知识体系显得尤为重要。

现在，抛开那些点点滴滴的零星错误（虽然它们也很重要），让我们眼光看得更长更远，即使是组织行为学的经典理论也认为仅仅是团队精神



学员应该拥有这套书

虽然学员大多数选择教材都还是为了应付考试，但是不管是用于考试还是用于以后从业于因特网，“开发王”系列图书是建立在正确的认知和完整系统的知识体系之上的，这符合最一般的逻辑要求，有利于记忆和总结，它会加快你的学习进程，提高效率，节省时间。

虽然教材并非由学员自己来选择，但任课老

和团队意识并不足以成功实现团队协作，共同的技术知识和技术能力才是团队合作得以有效进行的基础。

共同的技术知识和技术能力不但能够很容易使成员清楚要达到的目标，而且相互间也清楚自己应承担的责任和应该实现的功能。因此，一个完整、系统的知识体系对于团队合作而言便显得极为重要。仅仅靠热情，员工们能够合作得像一个团队吗？恐怕员工们自己也说服不了自己，更何况位于组织核心的人力资源主管和信息技术主管呢？

人力资源主管和信息技术主管作为雇主的代表应该将此书作为实现团队合作的技术知识和技术能力评估的“客观指标”。

人力资源主管和信息技术主管作为雇主的代表，同时也是这一知识体系的受益者。这一知识体系首先使你可以认识到一个合格的开发者所应具备的基本知识技能。

除此之外，这套图书包含了大量的测试题，可以帮助你从客观的技术层面上选择那些适用于本企业的开发人员，以及用于员工考评和员工的再培训。并且这些测试题可以提高你的工作效率，从而为你和你的雇主节省成本。

师一般都会非常尊重学员的建议，你应告诉他你看到的真实的东西，以及你想到的东西，如果你这样做了，就理应得到赞许。

如果你发现了教材中的一些错误，就应该及时提醒任课教师，并且也应该和同学切磋这个认识是否正确，这个过程也是培养团队协作精神的过程。

“开发王”系列图书不但适合作为教材，也适合作为自学用书，在3个最基本问题的指引下，学员也可以迅速有效地掌握万维网应用程序开发的知识。并且，书中大量的范例可以用于实际应用程序开发和设计中，每章后面的习题可以用于巩固所学的知识。



任课教师应该使用这套书作为教材以赢得学员的信任

主讲网络开发的任课教师应该成为正确认知和完整系统的万维网知识体系的传播者，正确的认知和完整系统的万维网知识体系是相对于错误认知和混乱的知识而言的，在教学上，混乱的知识往往与肢解式的授课方式紧密连在一起。

肢解式教学往往是要么盲目地对某个语法进行割裂式的使用方法讲解，要么沉迷于对某个具体范例的长篇累牍的分析之中。这种教学虽然容易让学员沉醉于暂时的范例成果而“沾沾自喜”，但也往往让他们迷失在这种“沾沾自喜”当中，从而忘记软件开发的基本事实。

是的，我这学期正准备这样按照完整系统的知识体系进行教学。



教授，我想我们学生应该完整系统地学习网络开发的知识。



这样做的教学目标看似明确，其实贻害无穷。

“庖丁解牛，不见全牛”是流传几千年的老故事，讲述了庖丁最初解牛时“所见无非全牛者”，而其“所好者道也，进乎技矣”，这个“道”就是指要完整系统地把握一件事物，然后再勤加练习，到“三年之后，未尝见全牛也”。

任课教师应该将学员从这种“沾沾自喜”的悬崖边缘拉回来，授以庖丁之“道”，勉励其自学以成庖丁之“技”，并最终集万维网应用程序开发之大成。

任课教师要做到这一点，一个重要的方式就是将过去肢解式的授课永远忘掉，那些肢解式的教材也理应被扔到垃圾堆中。

“开发王”这套书能够帮助你完成你的教学目标，这套书的一个编写特点就是其完整系统的知识体系，它能够帮助学员掌握“道”之本质。

为了使教学能够更顺利地展开，并便于学生课后自学，我们在每章的前面对本章内容做了总结，而且，每章的后面提供了习题和小结。



以坚实恒定的基石应对变化

对开发者、雇主、主讲教师、学员来说，市场的不断发展变化是不可避免的，市场的变化进而催生技术的不断发展变化，这些变化对每一个卷入 IT 产业的组织和人员都提出了挑战。

企业如何才能迅速响应市场变化带来的挑战呢？开发者如何才能迅速应对企业的新需求呢？在面对新技术时，如何将教学工作与新技术迅速衔接起来呢？

为了保持竞争优势并更迅速地响应市场变化，人员的技术知识和企业的业务必须保持流程的一致性，以实现更大的灵活性和扩展性，而这一切必须建立在一个“坚实恒定”的基石之上才能有

可靠的技术保证。

在“开发王”这套技术丛书中，每一个参与者都将看到这块“坚实恒定”的基石，这个基石被称为“完整系统的万维网知识体系”，在这个基石之上实现的灵活性和扩展性必然是可持久、可度量的，且可以应对迅速的变化。

完整系统的万维网知识体系针对变化提供了更多机会，无论变化多么复杂、多么迅速，“开发王”都能够帮助简化它们，从而轻松获取竞争优势。

从此，变化不再令人苦恼，相反它为你带来一系列机遇：灵活性、业务一致性，当然，还有竞争优势，不管是对个人，还是对组织……

作者的话

从开始准备写作到本套书的出版，前后经历了近 6 年的时间，也经历了多个软件版本变迁，以及万维网的沉沉浮浮，在这紧张的几十个月后，本套书终于得以问世。我希望能通过本书给万维网应用程序开发者和设计者带来无限广阔的、基本的、正确认知和完整系统的网络开发知识。

本书在写作过程中参考了“大量”的参考资料，包括数十份标准和规范，也包括作者多年创作和开发体验的结晶，所有范例也都经过了严格的测试，从而确保了内容准确翔实。

文章中提出的任何观点和主张都必须经得起事实的考验。

虽然如此，错误和不足也在所难免，恳请读者不吝赐教和指正，我们一定会全力改进，在以后的工作中加强和提高。

本书在创作和出版的过程中得到了电子工业出版社易飞思公司的大力支持，感谢他们在图书的设计及内容审定方面所给予的指导，同时也感谢为本书的出版而努力工作的出版社其他工作人员。

怎样阅读这套书

这套书的知识结构（包括范例），前后衔接的逻辑非常紧密，所以，这里特别提示用户不要试图跳跃章节阅读，这将恐怕使你欲速而不达。

保持与作者沟通

有时与作者沟通是十分必要的，用户可以从作者处获取知识的更新，或者勘误（如果书中有话），同时也可以让作者了解到你的想法。作者目前有以下联系方式：

E-mail（首选的推荐方式）

zhang-yafei@hotmail.com

BLOG 留言（简体版和繁体版）

<http://cn.zhang-yafei.com>

<http://tw.zhang-yafei.com>

发送邮件的注意事项

如果你发送邮件，请在标题处以以下格式书写，以利于作者分类处理。

假定你对本书第 2 章中的内容有疑问，只需在标题处如下书写（请不要在标题处添加其他内容）：

CSS2008_2_Chapter_02

假定你对本书第 12 章中内容有疑问：

CSS2008_2_Chapter_12

假定你对本书附录 B 中的内容有疑问：

CSS2008_2_Chapter_B

我们喜欢听到你的声音！

编 著 者

概览

第 1 篇 CSS 基础入门篇	1
第 1 章 建立首要的、基础的正确认知	3
第 2 章 CSS 语言和 HTML 语言结合基础	15
第 3 章 CSS 语言基础——CSS 的基本语法	27
第 4 章 CSS 语言基础——了解属性值	43
第 5 章 CSS 语言基础——CSS 的选择符语法	55
第 2 篇 CSS 视觉显示篇	67
第 6 章 CSS 盒子模型	69
第 7 章 颜色和背景	91
第 8 章 字体和文本	99
第 9 章 伪元素、伪类选择符和动态生成内容	121
第 10 章 为列表定义样式	143
第 11 章 溢出和裁剪	155
第 12 章 HTML 文档设置和 CSS 的层叠规则	165
第 3 篇 CSS 视觉布局篇	175
第 13 章 CSS 中盒子的定位	177
第 14 章 CSS 中布局盒子的计算	201
第 15 章 CSS 基本的布局范例	213
第 16 章 使用 CSS 表格布局	233
第 17 章 将样式表应用于不同介质	255
第 18 章 利用 CSS 分页显示和打印	261