

网站设计与开发

周 耿 编著

復旦大學 出版社

前言

FOREWORD

互联网近 10 年来获得了迅猛发展。20 年前很少有人能够接触到网站,而现在各种各样的网站在人们的日常生活中起着越来越重要的作用。网站每天都有新的变化,网站的开发与设计更是如此。早期的网站非常简单,几乎就是几个简单的网页。而现在的电子商务网站具有规模大、复杂度高等特点,大中型网站的开发普遍引入了类似工业化生产的方法,靠个人单打独斗的时代已经结束了,现在的网站都需要一个团队来开发、维护。网站设计各个环节的分工也越来越明确,从策划、需求分析到美工、设计和编码都可能是由不同的人完成的,这样,可以让每个人集中精力投入到自己的分工中去,从而提高工作效率。但是,在实际工作过程中,这样的分工也未必就能提高效率。比如编代码的不懂美工,做美工的不懂代码。当美工设计好界面后,做代码的往往不小心破坏了界面,而做界面的不懂代码不知如何修改。这类问题的出现除了团队内部沟通不够以外,还因为团队成员基础知识不扎实。虽然每个工序各司其职,但是如果某个环节的人对下个环节的操作一点都不了解的话,这种分工只是形式上的分工,并不能真正提高效率。这也是不少国内中小公司宁愿把自己的网站交给一个人来开发,而不愿意交给一个团队来开发的原因。不管从事网站设计的哪个环节,都需要了解网站设计的其他各个环节。这就对当前的网站设计人员提出了更高的要求。

网站设计(Web Design)和网站开发(Web Development)的含义是不一样的。前者主要指界面的设计,而后者主要指功能的开发。对于一个优秀的网站来说,两者是密不可分的。缺少了界面,功能即使再强大也吸引不到人气;而空有华丽的外表没有实质功能的网站也只能是昙花一现。本书将网站设计与网站开发一些最精华的东西奉献给大家。

目 录

CONTENTS

第 1 章 网站设计的基本概念	001
1.1 Web 与网络	001
1.2 Web 设计的核心问题	004
1.3 Web 开发的技术	005
第 2 章 HTML 快速入门	008
2.1 HTML 的基本结构	008
2.2 说明性元素	010
2.2.1 HTML 的注释 <code><!-- --></code>	010
2.2.2 说明性元素 meta	010
2.2.3 调用外部资源 link	011
2.3 文本元素	012
2.3.1 超链接元素 a	012
2.3.2 水平线 hr	014
2.3.3 标题 h1~h6	014
2.3.4 段落 p	014
2.3.5 手动换行 br	015
2.4 表格与列表	015
2.4.1 table	015
2.4.2 列表 ul 和 ol	016
2.5 多媒体元素	017
2.5.1 图像 img	017
2.5.2 多媒体对象 object	018
2.6 界面,样式性元素	019
2.6.1 框架 frameset	019
2.6.2 块元素 div	021
2.6.3 行内元素 span	021
2.7 非标准元素	021



2.7.1	applet, embed	021
2.7.2	marquee	023
2.8	HTML 的表单元素	023
2.8.1	form 元素	023
2.8.2	input	024
2.8.3	select	027
2.8.4	textarea	028
2.8.5	label 元素的妙用	028
2.9	HTML 编码需要注意的问题	029
2.9.1	建议不使用的元素	029
2.9.2	建议不使用的属性	030
2.9.3	其他要注意的问题	031
2.10	XML 与 XHTML 简介	031
2.10.1	XML 基本概念	031
2.10.2	XML 命名空间	032
2.10.3	XML 解析	033
2.10.4	XHTML	034
第 3 章	CSS 快速入门	036
3.1	CSS 的定义	036
3.2	CSS 的选择符与选择符的层叠	041
3.2.1	一般选择符	041
3.2.2	高级选择符	043
3.2.3	样式表,规则的优先级——层叠机制.....	044
3.2.4	层叠,继承与特殊性.....	045
3.3	CSS 常用样式的定义	046
3.3.1	定义字体样式	046
3.3.2	定义文本样式	047
3.3.3	定义链接样式	048
3.3.4	定义背景样式	048
3.3.5	定义鼠标光标样式	049
3.4	CSS 排版	050
3.4.1	CSS 盒模型	050
3.4.2	CSS 定位	054
3.4.3	CSS 按钮实现	057
第 4 章	Web 页面的布局	060
4.1	主流浏览器	060
4.1.1	Internet Explorer	060



4.1.2	Mozilla Firefox	061
4.1.3	Opera	061
4.1.4	Netscape	061
4.1.5	Safari	061
4.1.6	Konqueror	061
4.1.7	其他二次开发浏览器	062
4.2	布局的基本要素	062
4.2.1	页头横幅	062
4.2.2	导航	063
4.2.3	辅助要素	064
4.2.4	页脚	064
4.2.5	内容与背景	065
4.3	布局的色彩	065
4.3.1	三原色	065
4.3.2	色彩三属性	066
4.3.3	色彩的搭配	066
4.4	常见布局的结构	067
4.5	CSS 实现布局	068
4.5.1	表现和结构相分离	068
4.5.2	CSS 布局的常用元素	070
4.5.3	CSS 布局示例	072
4.6	其他方式布局	075
4.6.1	表格实现布局	076
4.6.2	图像实现布局	078
4.6.3	框架实现布局	080
4.6.4	Flash 布局	081
4.6.5	CSS 布局与表格布局的对比	082
4.7	Web 标准	084
4.7.1	什么是 Web 标准	084
4.7.2	结构的标准	084
4.7.3	行为的标准	085
4.7.4	表现的标准	085
4.7.5	Web 标准的校验	085
第 5 章	Web 页面的行为	087
5.1	文档对象模型(DOM)	087
5.1.1	HTML 的 DOM 对象	088
5.1.2	对象的属性	088
5.1.3	对象的方法	089



5.1.4	对象的事件	092
5.2	JavaScript	094
5.2.1	JavaScript 概要	094
5.2.2	JavaScript 与 HTML	095
5.2.3	JavaScript 的语法	096
5.3	DOM 与 JavaScript 应用举例	098
5.3.1	HTML 文档的 DOM 的 Tree 结构	098
5.3.2	菜单折叠	101
5.4	AJAX 技术简介	102
第 6 章	服务器端动态网页环境	105
6.1	Apache 服务器的安装步骤	105
6.2	PHP 环境的安装与配置	107
6.3	MySQL 环境的安装与配置	108
6.4	phpMyAdmin 的安装配置	111
第 7 章	PHP 语言快速入门	115
7.1	php.ini 的一些基本配置	115
7.2	PHP 的基本语法	116
7.2.1	变量	116
7.2.2	数组	117
7.2.3	操作符	119
7.2.4	语句与引用	119
7.3	字符串的处理	120
7.3.1	字符串的格式化	120
7.3.2	字符串的分割与连接	120
7.3.3	字符串查找	121
7.4	正则表达式及其应用	121
7.4.1	字符匹配	121
7.4.2	重复	122
7.4.3	字符串定位	123
7.4.4	分支	123
7.4.5	特殊字符	123
7.4.6	PHP 正则表达式的函数	123
7.5	代码重用	124
7.5.1	外部文件直接调入	124
7.5.2	自定义函数	125
7.5.3	类和对象	125
7.5.4	Web 中代码重用方式的比较	131



7.6	PHP 异常处理	133
7.7	PHP 文件操作	134
7.8	PHP 与 MySQL 数据库连接与操作	137
7.9	错误调试技巧	138
7.9.1	3 类常见错误	138
7.9.2	调试技巧	140
第 8 章	MySQL 数据库快速入门	142
8.1	MySQL 数据库的基本操作	142
8.1.1	创建数据库和数据库中的表	142
8.1.2	对数据库中表的操作	147
8.2	MySQL 数据库的权限与数据库的安全	150
8.2.1	创建新用户	150
8.2.2	设置用户权限	151
8.3	MySQL 数据库的优化设置	154
8.4	MySQL 数据库的事务与存储过程	155
8.4.1	事务处理	155
8.4.2	存储过程	156
8.5	MySQL 数据库的备份与恢复	158
8.5.1	备份数据库	158
8.5.2	备份表	160
8.5.3	恢复	160
8.6	数据库设计	161
第 9 章	服务器端动态网页综合应用	162
9.1	PHP 中日期和时间的相关问题	162
9.2	网页间变量的传递方法	163
9.2.1	GET 方法传递变量	163
9.2.2	POST 方法传递变量	164
9.2.3	COOKIE 方法传递变量	164
9.2.4	SESSION 方法传递变量	166
9.2.5	利用文件和数据库传递(存储)变量	167
9.3	计数器	167
9.4	留言本	168
9.5	用户登录与身份验证	170
9.6	动态图像处理	172
9.6.1	单一图像输出的步骤	172
9.6.2	在页面中嵌入自动生成图像	174
9.6.3	图片缩放	174



9.7	文件上传	175
9.8	内容分页显示	177
9.9	目录操作	179
9.10	动态网页的安全	180
9.10.1	数据安全与保密	180
9.10.2	SQL 注入漏洞	181
9.11	PHP 模板技术	182
第 10 章	网站的优化	185
10.1	面向性能的优化	185
10.1.1	操作系统的优化	185
10.1.2	Apache 服务器性能的优化	185
10.1.3	PHP 程序性能的优化	186
10.1.4	MySQL 数据库性能的优化	187
10.1.5	Web 压力测试——ab 工具的使用	190
10.2	面向用户的优化	192
10.2.1	尽量少让用户思考	192
10.2.2	用户使用 Web 的方式	193
10.2.3	视觉设计优化	193
10.2.4	精简内容	193
10.2.5	导航的优化	194
10.2.6	可用性测试	194
10.3	面向搜索引擎的优化 (SEO)	194
10.3.1	链接的优化	194
10.3.2	面向搜索关键字的匹配	196
10.3.3	日志分析	196
10.3.4	静态地址格式——Apache 中 rewrite 模块	198
10.3.5	SEO 其他要注意的问题	199



第 1 章

网站设计的基本概念

1.1 Web 与网络

Web 是 World Wide Web(WWW, 俗称万维网)的简称,它是因特网上的一种信息服务系统,并逐渐成为集成其他信息服务的门户。作为一种新兴的信息获取的渠道,Web 与传统的媒介(广播、杂志、报纸、电视等)相比,具有两个最大的特点。首先,Web 系统是非线性的,信息在网页上可以以跳跃的方式呈现,人们所做的事情仅仅是点击一下链接而已。其次,Web 系统是可交互的,除了能给用户的信息外,还能接受用户的请求,与用户进行互动交流。一个 Web 系统一般由 4 个部分构成: Web 服务器、Web 浏览器、HTTP 协议以及网页(HTML),如图 1-1 所示。

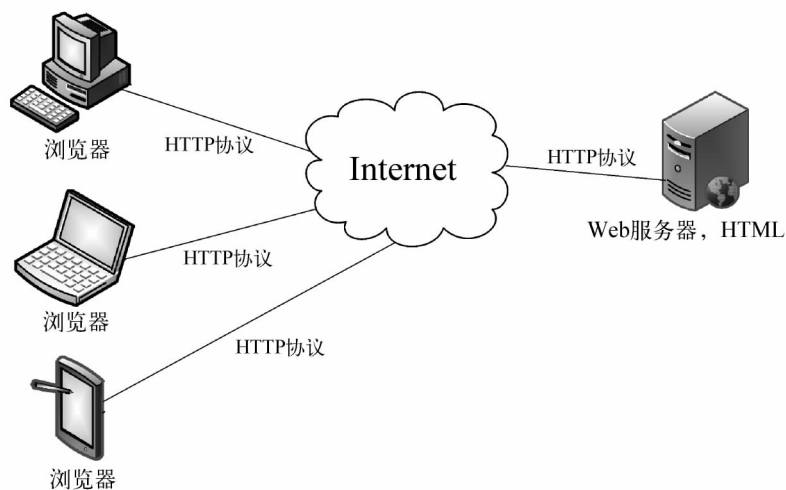


图 1-1 Web 系统的组成

在计算机的世界中,任何信息的传递必须由收发双方通过特定的协议才能发生。而在 Web 系统中,承担这个任务的协议就是超文本传输协议 HTTP (HyperText Transfer Protocol)。它负责浏览器和服务器的信息交换,是工作在 TCP/IP 协议模型应用层上的协议。图 1-1 中最右边的是 Web 服务器,它执行的功能通常叫做 WWW 服务。超文本标记语言



HTML (HyperText Markup Language)及其演化的其他文件(俗称网页)规定了信息在网络上传放的格式。各种网页都存放在 Web 服务器中。WWW 服务器使用 HTTP 协议接受外界的请求并进行处理,将处理好的信息通过 HTTP 协议发送出去。WWW 服务的默认端口是 80。

在任何客户端,客户要查看 Web 服务器上的信息必须具备浏览器。浏览器使用 HTTP 协议与服务器交互。接受数据后将其翻译成 Web 超文本,然后以 Web 页面的形式显示在用户屏幕上。要注意的是,上面所说的服务器与浏览器指的是逻辑上,即软件上的概念。目前最常用的 Web 服务器有开源机构的 Apache 和微软公司的 IIS 等;而浏览器主要有微软公司的 IE 和 Opera,以及 Firefox 等等。

如果一个用户要访问某个网站,就需要在客户端浏览器中输入 URL(Uniform Resource Locator)或点击页面的超链接。这时浏览器向 DNS(域名解析服务器)询问 URL 中域名的 IP 地址,随后 DNS 将返回被询问的 IP 地址。知道了服务器的 IP 地址后,浏览器通过 3 次握手方法与 Web 服务器的 80 端口建立一个 TCP 连接。建立连接的同时,浏览器通过 HTTP 协议向 Web 服务器发送请求消息(Request Message)。一般请求信息包括请求方法、URL、协议版本以及相关的 MIME(Multipurpose Internet Mail Extensions)样式。Web 服务器通过 HTTP 协议向浏览器返回响应消息(Response Message),而响应信息一般包括包含消息的协议版本、一个成功或失败码以及相关的 MIME 样式消息(包含服务器的信息、资源实体的信息和可能的资源内容)。接着浏览器解释接收来的资源实体信息——Web 页面(即网页)或者多媒体资源。一个典型的 Web 页面由许多页面对象组成,如 HTML 文本、各种客户端脚本,而多媒体资源包括 GIF/JPEG 图像文件、音视频文件、Java Applet、ActiveX 控件等,它们都用 URL 来标志。

从以上过程我们可以看出,HTTP 对如何在客户机上显示 Web 页面不起作用,它仅仅定义在客户机和服务器之间如何传送数据。上面所讲述的是一般正常情况。在实际应用中,可能出现一些其他情况,比如当 DNS 瘫痪的时候,浏览器到服务器的连接无法建立,那么服务器就无法访问。不过如果客户直接在浏览器地址栏输入服务器的 IP 地址,那么就可以绕过 DNS,直接执行下面的步骤。如果浏览器请求的信息服务器无法响应,或者无法处理,那么服务器就会给浏览器发送一个失败的信息,告诉浏览器出错的细节。浏览器上仅仅显示一些出错的信息。需要注意的是,Web 服务器与浏览器之间的连接仅仅维持一段很短的时间,一旦文档或者多媒体资源传输完毕,TCP 连接就关闭,浏览器并不与服务器始终保持连接,即使马上就访问同一服务器的其他网页,也要重复上述过程。

从上面的描述我们可以看出 Web 系统依靠浏览器与服务器的交互来完成一系列功能。这种网络交互的方式叫做浏览器/服务器(Browser/Server)模式,简称 B/S 模式。在系统的性能方面,B/S 的优势是其灵活性。用户只需使用浏览器,不需安装额外的软件,操作方便;应用程序的维护只要在服务器端进行,便于应用系统的维护和更新。任何时间、任何地点、任何系统,只要可以使用浏览器上网,就可以使用 B/S 系统的终端。不过,采用 B/S 结构,客户端只能完成浏览、查询、数据输入等简单功能,绝大部分工作由服务器承担,这使得服务器的负担很重。由于客户端使用浏览器,使得 Web 功能受到浏览器的约束,只能执行一些相对简单的功能。

除了 B/S 模式外,还有一种常用的网络交互模式,客户端/服务器(Client/Server)模式,



简称 C/S 模式。在部署 C/S 构架时,要专门为客户端定制相应功能,比如规定客户端如何与服务器连接,客户端如何获取数据等。不像在部署 B/S 构架时,浏览器是通用的,只需将精力集中在服务器的功能实现上。由于定制客户端,C/S 模式的客户端就有可能采用较高效的方式运行,可以采用更有效率的通讯协议,或者将一部分运算、处理量放在客户端,降低服务器的工作量等等。我们可以观察到用 C/S 模式实现的即时通讯系统(如 QQ, MSN 等)与用 B/S 模式实现的网上聊天室的速度和执行的效率就有明显的不同。C/S 模式的缺点是客户端的软件升级、更新和维护比较复杂。如服务器端要求客户端软件有一点点改动,那对每一个客户端的软件都要维护(升级),其维护工作量很大。针对上述缺陷,目前 C/S 结构的应用一般都配备自动在线升级的功能,省去了很多人工的工作。C/S 模式最简单的是 2 层 C/S 模式,即客户机直接连接数据库服务器。在 2 层 C/S 模式的基础上,在客户机与数据库服务器中间插入多个层次的其他应用服务器就构成了 3 层甚至 N 层 C/S 结构。3 层结构模式是伴随着中间件技术的成熟而兴起的,核心概念是利用中间件将应用分为表示层、业务逻辑层和数据存储层 3 个不同的处理层次。这 3 个层次是从逻辑上来划分的,具体的物理分法可以有多种组合。中间件作为构造 3 层结构应用系统的基础平台,提供了以下主要功能:负责客户机与服务器、服务器与服务器间的连接和通信;实现应用服务器与数据库服务器的高效连接;提供一个 3 层结构应用的开发、运行、部署和管理的平台。

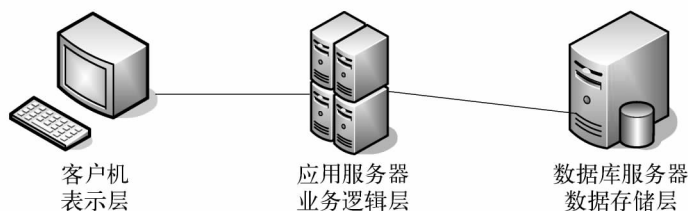


图 1-2 三层 C/S 结构

C/S 模式逐渐融合于 B/S 模式中。目前出现了一种浏览器+插件/服务器的方式,这种方式实际上是对传统意义上 B/S 模式的修正,将 C/S 模式定制的优点融入其中。许多浏览器没法实现的功能可以采取用户安装相应的插件的方式实现,而插件一般可以定时升级。在互联网上最常见的插件就是 Flash,一般浏览器是不能查看 Flash 动画的,只有安装了相应的插件才能观看。而对于大型电子商务网站,一般采用 4 层结构的 B/S 模式。处于第一层的是浏览器,处于第二层的是 Web 服务器,负责 Web 的外在表现。而应用服务器由一台或多台服务器组成,该层具有良好的可扩充性,服务器的数目可以随着应用的需要增加而增加。处于第四层的是数据层,由数据库系统组成。

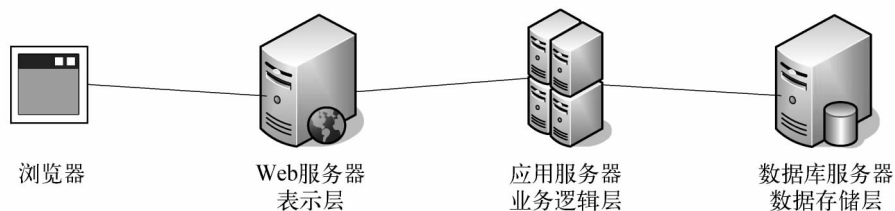


图 1-3 四层 B/S 结构



1.2 Web 设计的核心问题

Web 设计理念的发展大概经历了 3 个阶段。第一个阶段是存在阶段,设计者们认为他们的 Web 站点将会吸引访问者只是因为它们存在。这在互联网发展的初期是非常常见的,而且在当时情况下确实是这样,因为可访问的站点实在太少了。这些站点一般界面非常难看,并且难以使用,甚至没有任何可用的信息。在我国,这类网站大都集中在政府部门和垄断企业的站点上。第二个阶段是美观阶段,设计者们为了展示自己的水平,添加了大量漂亮的图片、花哨的动画与脚本。这些东西虽然给网站一个非常漂亮的外观,但是这种漂亮的外观大大超出了实际的需要,很多网站空有其骨架,没有什么内容。而且这类网站由于大量使用不必要的图片和加入过多的冗余代码,使得网页变得臃肿无比,大大降低了网站的访问速度。花哨的外观设计会诱惑用户访问一个站点,但这些无法留住用户,如果站点空洞无物,用户最终还是会选择离开。

访问者的态度最终使得设计者意识到,人们通常在访问一个站点时,并未将它当作艺术作品来欣赏,而是为了获得有关产品的信息,或者有一些技术问题需要咨询,或是为了阅读有关开发者的期刊。如果这些信息不能快速地呈现在访问者面前,他很快会对这个网站失去兴趣。于是,Web 设计的理念进入了第三个阶段,针对内容的优化。它的特点是以内容为中心,人性化,简捷,快速。在这个阶段,设计者意识到,Web 设计最核心的 3 个问题是内容、功能和形式。

当前的 Web 更加专注于内容。站点与其内容的边界经常非常模糊。甚至从某种意义上讲,内容就是站点。内容可以是文本、图片、音频、视频或多种的混合体。文本是网页的核心和灵魂,平均而言,网站提供的信息至少有 90% 是通过文本展示的。文本在互联网上通常有两种形式,普通文本和图形文本。普通文本的特点是占有空间小,便于编辑。但是普通文本有时候也会出现这样不理想的情况:用户仅仅是增加字体的尺寸、缩小窗口或者扩大窗口,就会毁掉一段优美的排版样式,而且普通文本遇到浏览器不支持的字体时会出现乱码。而图形文本一般就可以避免这种情况,能够在一定程度上控制字母间距、字体大小选择等情况的发生。但是图形文本带来了更大的问题:图片比较大时下载速度变慢,更新困难,可访问性受限制,搜索引擎会忽略等。所以,目前主要还是以普通文本为主。图形文本一般应用在 Logo、标题、导航等需要特殊文字效果的地方。动画、视频等多媒体资源近年来发展非常迅速,在线视频的网站甚至以多媒体为主要内容。但是对于目前的互联网状况和现实的需求,网站内容还是以文本为主,多媒体内容在网页中起到画龙点睛的作用。

除了内容以外,设计者最关心的两个问题是 Web 的功能与 Web 的形式。有高质量的内容,但用户没法找到它,这些内容也毫无价值。网站需要良好的功能来让用户迅速找到他想要的资源,所以,对于 Web 而言,功能就体现在提高用户的访问效率上。Web 最常见的功能是导航与搜索以及如何提高访问的速度。Web 需要良好的导航结构,即词能达意的链接名称,合乎逻辑的按钮组合以及语义清晰的页面标题。用户能够根据设定的导航结构快速找到他所要的内容,用户从进入首页到找到他需要的内容点击数最好不要超过 3 次。搜索技术在 Web 上的应用使得用户从茫茫信息海洋中解放出来,有了搜索,人们不需要顺着



导航依次点击下去,而可以在搜索栏直接输入关键字从而直接定位到自己需要的资源。注意导航和搜索只是达到目标的手段,用户并不关注导航或者搜索本身,他们希望得到的是内容和结果。

Web 的形式(外观)也是非常重要的,因为外观会显著影响用户最初的价值判断。用户离开网站的时候会有一个综合的印象——站点是否成功的感觉。而外观能够加强成功的感觉,尤其是在最初的几次访问中。但继续使用后,用户会减少对外观的关注。Web 的形式主要包括颜色、图像、布局。颜色的搭配在用户形成价值判断的过程中起着非常重要的作用,对于颜色如何搭配本应该是美术方面的问题,幸好我们有许多专门的色彩设计软件,使得我们不必纠缠于无穷的色彩调试。图像在互联网上以 GIF、JPEG 与 PNG 3 种格式为主,这是由于反映同样的信息,这三种格式的文件所占用的存储空间能够大大小于其他的图像格式。占用的存储空间小意味着下载速度快,就会受到人们的宠爱,这就是互联网规律。Web 布局其实是指综合颜色与图像,以及将内容以什么样的方式展示出来(类似文本的排版)。站点的布局必须表明其所提供的服务,可以使用一个很短的描述性的标题指明站点的主题是什么,而且需要加入徽标(Logo)。必须使用易读的字体与合适的行距,以结构良好的方式表现出来。一般而言电子商务网站必须设置站点地图、关于我们和帮助(FAQ)栏目。Web 实际设计中可能遇到鱼与熊掌不可兼得的情况,即形式、功能与内容出现一定程度的矛盾,这时我们应该遵从形式服从功能,功能服从内容的原则。

1.3 Web 开发的技术

Web 系统在服务器端上是由网页组成的,Web 的开发主要是指网页的开发,完全“静止不动”的网页是指除了网页在服务器和浏览器之间传送(参见图 1-4),浏览器对网页进行 HTML 解析外,服务器和浏览器没有其他功能。这种网页在互联网发展初期非常流行,但是在现阶段已经完全不能满足需要。网页首先存放在服务器中,经过浏览器请求调用再传给浏览器。因此,如何处理网页以及在哪儿处理,成了 Web 开发首先关注的问题。

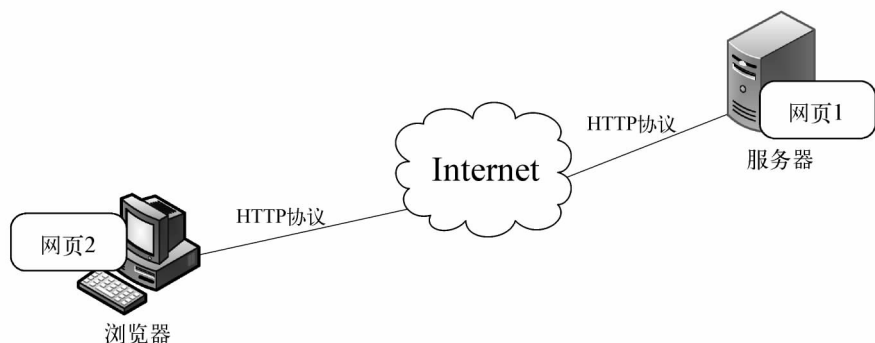


图 1-4 网页处理示意图

如图 1-4,有两个地方可以处理网页,一个是客户端,另外一个服务器端。如果服务器端对网页不进行任何处理,仅仅起到存放网页的作用,那么客户端请求后,服务器端发送到客



户端的网页与服务器端存放的网页是完全相同的。即图 1-4 中网页 2 是网页 1 的副本。那么这时候网页“动”起来的任务就全部塞给浏览器了。因此网页中必须包含让其“动”起来的指令,不然浏览器仅仅是将 HTML 解释出来。这些具体的指令,可以在网页中,也可以由一个外部链接指向一个专门的指令文件。不管是什么情况,这些指令都必须下载到浏览器中,浏览器再根据这些具体的指令作出反应,让网页动起来。这些指令是在浏览器中执行的,因此我们可以称其为浏览器端动态网页技术。这些技术主要有 CSS, JavaScript 以及一些插件技术如 Flash 等。这类技术的特点是指令(源代码)能被任何浏览器端的使用者查看,这对辛辛苦苦做出动态效果的开发人员可不是一个好消息——动态效果能够很容易被模仿。CSS 是一种指定文档如何呈现给用户的语言, JavaScript 能让网页的各个元素进行互动, Flash 则是前 Macromedia^① 公司推出的一种浏览器中实现平面动画的技术,而 VRML 则是未来在浏览区实现 3D 动画的语言。

如果服务器端对网页进行处理呢?这种服务器端可以根据服务器上存储的网页(母体)的指令,根据服务器上的数据库动态“生产”出一个临时的网页,将这个临时“生产”出的网页而不是母体发送给浏览器。这个“生产”的过程就是服务器端处理的过程,这种技术也称作服务器端动态网页技术。有了这种技术,开发人员就可以完全不用担心自己编写的服务器端处理指令能被浏览器端查看到了,因为浏览器端能看到的仅仅是处理后(生产出)的网页。这种技术还有一个更大的好处,可以用一个母体(服务器上的网页),根据浏览器不同的状况“生产”不同的网页,将“生产”出来的不同的网页发送给不同的浏览器,让不同的浏览器访问同一个页面的时候得到的结果是不一样的。BBS、BLOG、甚至网上银行系统非常需要这种技术支持。这类技术主要有 PHP、ASP、JSP、Cold Fusion、PERL、CGI 等。CGI 是早期人们使用的服务器和外部程序的接口,但由于太复杂非常不灵活,目前较少使用。PERL 能使得 CGI 难以驾驭的缺点得以改善,但是对于初学者而言仍然有一定的难度。ASP 和 JSP 分别是微软公司和 SUN 公司出品的,是存在版权的,而 PHP 则是开源的。它们共同的特点是较容易学习,而且目前都是免费的,是目前使用最多的 3 大服务器端动态网页技术。PHP 的优势在于它的开源性质,这决定了它永远是免费的,而 ASP 和 JSP 的免费只是暂时的商业策略,未来不一定是免费的。最后, Cold Fusion 是前 Macromedia 公司推出的一个产品,它的特点是初学者容易上手,能够快速开发出需要的程序,并且能非常方便地和 Macromedia 公司的其他产品结合起来,对多媒体支持较好,不过价格不菲成为它应用较少的主要原因。

服务器端处理网页通常都要用到数据库技术来存储一些持久信息,从而形成有交互能力的动态网页。有些设计师认为,数据库驱动(使用数据库)是区别服务器端动态网页与其他类型网页的标志。网络上常用的数据库有: MySQL、SQLite、SQLServer、Oracle。此外,以文本方式存储的 XML 经常被当作数据存储库使用,也起到了数据库的作用。不过需要注意的是, XML 是一个能够存放数据的标准,而不是一个具体的数据库产品。而且它相对于其他数据库产品,存储的效率比较低,但是兼容性较好。

^① 2005 年 12 月 3 日 Macromedia 公司被 Adobe 公司收购。



复习与思考题

1. 什么是 B/S 模式？什么是 C/S 模式？它们各有什么优缺点？试举出现实中 B/S 和 C/S 模式的几个例子。
2. Web 的发展经历了哪几个阶段？
3. 客户端动态网页和服务器端动态网页有什么区别？

上机实习

登录 <http://www.worldbest.com/shopping.htm>, 浏览国外销售行业获奖的网站, 重点关注 Web 的内容、外观与国内大部分网站相比有何不同。



第 2 章

HTML 快速入门

HTML(超文本标记语言)是用来创建网页最基本的语言,是 SGML(标准通用标记语言)的一个子集。目前的版本是 HTML4.01,它是在 W3C 的支持下开发的。按照 W3C 的说法,HTML4.01 是 HTML 的终点,而取代它的是 XHTML1.0。XHTML 实际上是用 XML(可扩展标记语言)重新定义了 HTML 的各个元素,虽然大部分元素都是相同的,不过它对元素的书写规则要求更加严格,使得它对未来的标准有更好的兼容性。关于 HTML,XML 以及 XHTML 的标准、规范都可以在 W3C 的网站([Http://www.w3.org](http://www.w3.org))上下载到。它定义了形成网页上各种元素的语法和放置方法。根据 W3C 的定义,所有网页要素都是由元素构成的,这些元素指示浏览器如何显示网页,而元素名称本身是不会显示的。本章结合 XHTML1.0 的标准介绍 HTML 中各常用的元素。

2.1 HTML 的基本结构

用 Dreamweaver 8 创建一个新的网页,操作过程是文件,新建,新建文档(见图 2-1)。

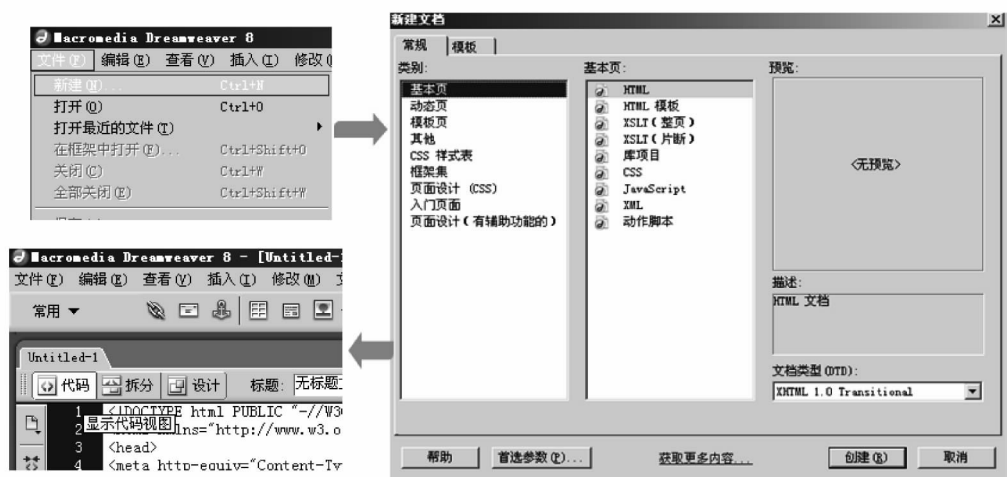


图 2-1 新建网页操作

HTML 文件的后缀是 .htm 或者是 .html,当前使用前者较多,因为它少了一个字符,更加

简洁。使用 Dreamweaver 新建文件后最基本的 HTML 代码呈现在我们面前：

```
1. <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
   "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
2. <html xmlns="http://www.w3.org/1999/xhtml">
3. <head>
4. <meta http-equiv="Content-Type" content="text/html; charset=gb2312"/>
5. <title>无标题文档</title>
6. </head>
7. <body>
8. </body>
9. </html>
```

第 1 行是对 HTML 的版本和格式说明,该行引用了 W3C 的 XHTML1.0 过渡(transitional)格式,DTD(具体格式的规则)参照给定的 W3C 地址: <http://www.w3c.org/TR/xhtml1/DTD/xhtml1-transitional.dtd>。

第 2 行是 HTML 的开头,其中 html 是元素,而 xmlns 是该元素的一个属性,表示使用 XHTML 1.0 的名字空间。这虽然和第一行表达的意思有些重复,不过 W3C 既然规定这么写,我们就照办。html 是该文档的第一个元素,HTML 所有的元素书写方法和它一样,都是用尖括号“<>”将元素名称括入其中。而 xmlns 是 HTML 元素的属性, <http://www.w3.org/1999/xhtml> 则是该属性的值。XHTML 规定所有元素的属性值书写时必须用引号“”括起来,避免和其他产生混淆。

第 3 行的 <head> 和第 6 行的 </head> 是匹配的。HTML 中的大部分元素都有清楚的开头和结束相匹配。元素开头和结束之间是元素的内容,它可以嵌套别的元素。在 HTML 中,head 这个元素中可以设置 HTML 的一些头部信息。

第 4 行是 meta 元素,它嵌套在 title 元素中。它怎么没有另外一半?其实这是 HTML 中另外一类元素,即空元素。空元素一般没有单独的结束标记,而是在元素的结尾加上斜杠“/”。空元素“空”的含义是指不能够嵌套任何内容。但是空元素可以拥有自己的属性,比如 meta 中就含有 http-equiv 和 content 两个属性。而“Content-Type”(内容类型)和“text/html; charset=gb2312”(文本 HTML 格式和字符集是简体中文)则分别是两个属性的值。meta 除了可以设置内容类型外,还可以设置一系列其他内容,我们在下面会详细介绍。

第 5 行是一个有内容的 title 元素,我们不难发现这个元素的内容是“无标题文档”。这个内容将反映在浏览器窗口的左上角。

第 7 行和第 8 行又是一对闭合的元素,它的内容才是在浏览器中显示出来的东西。

第 9 行是第 2 行 html 元素的另一半。

最后需要说明的是,为了符合 Web 标准,我们对于所有的元素的书写应该尽量使用小写。

