



项目 1 Web 前端开发基础 / 1

任务 1.1 初识 Web	2	子任务 1.2.2 HTML 开发历程	6
子任务 1.1.1 网页与网站	2	子任务 1.2.3 HTML 5 设计理念	8
子任务 1.1.2 网页相关名词	3	任务 1.3 熟悉网站开发流程和工具	11
子任务 1.1.3 Web 标准	4	子任务 1.3.1 网站开发流程	11
任务 1.2 认识浏览器和 HTML	5	子任务 1.3.2 Web 前端开发工具	13
子任务 1.2.1 浏览器介绍	5	子任务 1.3.3 编写第一个 HTML 5 页面	15



项目 2 Web 语言基础 / 17

任务 2.1 了解 HTML 基础语法	18	子任务 2.4.2 创建图片超链接	43
子任务 2.1.1 HTML 文档结构	18	子任务 2.4.3 创建电子邮件超链接	43
子任务 2.1.2 HTML 标签	18	子任务 2.4.4 创建下载超链接	44
子任务 2.1.3 标签的属性	28	子任务 2.4.5 创建锚点超链接	45
任务 2.2 学习文本控制标签	31	子任务 2.4.6 创建热点区域	46
子任务 2.2.1 标题段落标签	31	任务 2.5 学习表格标签	48
子任务 2.2.2 文本格式化标签	33	子任务 2.5.1 创建表格	48
子任务 2.2.3 特殊字符标签	34	子任务 2.5.2 设置属性	49
任务 2.3 学习图像标签	36	子任务 2.5.3 表格结构化	54
子任务 2.3.1 常用图像格式	36	任务 2.6 学习 <meta> 标签和 <link> 标签	56
子任务 2.3.2 嵌入图像	36	子任务 2.6.1 <meta> 标签	56
子任务 2.3.3 路径的表示方法	41	子任务 2.6.2 <link> 标签	57
任务 2.4 学习超链接标签	41		
子任务 2.4.1 创建文字超链接	42		



项目 3 Web 页面元素和属性 / 59

任务 3.1 学习结构元素	60	子任务 3.3.3 details 元素和 summary 元素	75
子任务 3.1.1 header 元素	60	子任务 3.3.4 menu 元素	77
子任务 3.1.2 article 元素	61	子任务 3.3.5 command 元素	77
子任务 3.1.3 aside 元素	63	任务 3.4 学习文本层次语义元素	78
子任务 3.1.4 footer 元素	65	子任务 3.4.1 cite 元素	78
子任务 3.1.5 figure 和 figcaption 元素	66	子任务 3.4.2 mark 元素	78
任务 3.2 学习页面元素	67	子任务 3.4.3 time 元素	79
子任务 3.2.1 section 元素	68	任务 3.5 学习全局属性	80
子任务 3.2.2 nav 元素	71	子任务 3.5.1 hidden 属性	80
子任务 3.2.3 address 元素	73	子任务 3.5.2 spellcheck 属性	81
任务 3.3 学习交互元素	74	子任务 3.5.3 contenteditable 属性	82
子任务 3.3.1 progress 元素	74	子任务 3.5.4 contextmenu 属性	83
子任务 3.3.2 meter 元素	75	任务 3.6 设计个人博客主页	83



项目 4 构建 Web 表单 / 87

任务 4.1 了解表单的结构	88	任务 4.3 学习表单属性	96
任务 4.2 学习表单控件	89	任务 4.4 了解 HTML 5 对表单的改进	97
子任务 4.2.1 input 控件	89	子任务 4.4.1 新增的 input 输入类型	97
子任务 4.2.2 textarea 控件	93	子任务 4.4.2 新增的 input 属性	103
子任务 4.2.3 select 控件	94		



项目 5 构建 Web 音频与视频 / 113

任务 5.1 认识编解码器	114	子任务 5.3.2 播放视频	122
子任务 5.1.1 音频编解码器	114	任务 5.4 学习音频与视频的属性、方法、事件	124
子任务 5.1.2 视频编解码器	115	子任务 5.4.1 音频和视频的相关属性	124
任务 5.2 在 HTML 5 中播放音频	116	子任务 5.4.2 音频和视频的相关方法	126
子任务 5.2.1 认识 audio 元素	116	子任务 5.4.3 音频和视频的相关事件	126
子任务 5.2.2 播放音频	118	任务 5.5 在网页中插入多媒体	128
任务 5.3 在 HTML 5 中播放视频	120		
子任务 5.3.1 认识 video 元素	120		



项目 6 CSS3 语言基础 / 131

任务 6.1 认识 CSS	132	子任务 6.2.2 供应商特定前缀	144
子任务 6.1.1 CSS 概述	132	任务 6.3 理解 CSS 过渡	146
子任务 6.1.2 CSS 编码规范	135	子任务 6.3.1 CSS 过渡概述	146
子任务 6.1.3 CSS 的命名规则	136	子任务 6.3.2 transition 语法	147
任务 6.2 应用 CSS3	137		
子任务 6.2.1 CSS3 核心属性	137		



项目 7 使用 CSS 设计样式 / 153

任务 7.1 了解 CSS 引入方式	154	任务 7.6 学习列表项符号	184
子任务 7.1.1 外部样式表	154	子任务 7.6.1 定义列表项符号	184
子任务 7.1.2 内部样式表	155	子任务 7.6.2 去除列表项符号	186
子任务 7.1.3 行内样式	155	子任务 7.6.3 列表项图片	188
任务 7.2 学习 CSS 选择器	156	任务 7.7 学习表格样式	189
子任务 7.2.1 元素的 id 和 class 属性	156	子任务 7.7.1 表格标题位置	189
子任务 7.2.2 选择器应用	157	子任务 7.7.2 表格边框的合并	190
任务 7.3 学习字体样式	165	子任务 7.7.3 表格边框间距	192
子任务 7.3.1 字体类型 font-family	166	任务 7.8 学习图片样式	193
子任务 7.3.2 字体大小 font-size	166	子任务 7.8.1 图片大小	193
子任务 7.3.3 字体粗细 font-weight	167	子任务 7.8.2 图片边框	194
子任务 7.3.4 字体风格 font-style	168	子任务 7.8.3 图片对齐	194
子任务 7.3.5 字体颜色 color	169	子任务 7.8.4 文字环绕	199
子任务 7.3.6 CSS 注释	169	任务 7.9 学习背景样式	201
任务 7.4 学习文本样式	171	子任务 7.9.1 背景颜色	201
子任务 7.4.1 文本样式简介	171	子任务 7.9.2 背景图片样式	203
子任务 7.4.2 文本修饰	173	子任务 7.9.3 背景图片重复	204
子任务 7.4.3 大小写	174	子任务 7.9.4 背景位置图片	206
子任务 7.4.4 行高	176	子任务 7.9.5 背景图片固定	208
子任务 7.4.5 间距	177	任务 7.10 学习超链接样式	209
任务 7.5 学习边框样式	178	子任务 7.10.1 超链接伪类	209
子任务 7.5.1 整体样式	179	子任务 7.10.2 深入了解 :hover	211
子任务 7.5.2 局部样式	181	子任务 7.10.3 鼠标样式	213



项目 8 CSS 高级应用及设计 / 217

任务 8.1 学习 CSS 盒子模型	218	任务 8.4 浮动与清除浮动	234
子任务 8.1.1 认识 CSS 盒子模型	218	子任务 8.4.1 浮动	234
子任务 8.1.2 宽和高	220	子任务 8.4.2 清除浮动	237
子任务 8.1.3 边框	223	任务 8.5 定位布局	238
任务 8.2 学习内边距与外边距	224	子任务 8.5.1 初识定位布局	238
子任务 8.2.1 内边距	224	子任务 8.5.2 固定定位	239
子任务 8.2.2 外边距	226	子任务 8.5.3 相对定位	241
任务 8.3 学习文档流	230	子任务 8.5.4 绝对定位	243
子任务 8.3.1 正常文档流	230	子任务 8.5.5 静态定位	245
子任务 8.3.2 脱离文档流	232		
参考文献			246

项目 1

Web 前端开发基础

项目目标 >

- ① 理解 Web 的相关概念。
- ② 掌握常用浏览器及其使用的内核。
- ③ 掌握网站开发基本流程。
- ④ 熟悉 HTML 开发历程及 HTML 5 设计理念。
- ⑤ 熟悉前端开发常用软件。

知识导图 >



网站是一种新型的信息传播渠道，人们通过它来发布自己想要公开的资讯以及提供相关的网络服务。Web 前端开发是创建 Web 页面或 App 等前端界面并呈现给用户的过程，通过 HTML、CSS、JavaScript 以及衍生出来的各种技术、框架、解决方案，来实现互联网产品的用户界面交互。

从 2010 年开始，HTML 5+CSS3 就一直是互联网技术中最受关注的两个话题。2010 年 MIX10 大会上微软的工程师在介绍 IE9 时，从前端技术的角度将互联网的发展分成 3 个阶段：第一阶段是以 Web 1.0 为主的网络阶段，前端主流技术为 HTML+CSS；第二个阶段是 Web 2.0 的 Ajax 应用阶段，热门技术是 JavaScript/DOM/ 异步数据请求；第三阶段是即将迎来的 HTML 5+CSS3 阶段，这两者相辅相成，将使互联网进入一个崭新的时代。

任务 1.1 初识 Web

Web，本意是蜘蛛网和网的意思，中文译为“网页”。通俗而言，人们在网上网的时候打开浏览器，输入一个网址，展现的就是一个网页。对于普通用户来说，Web 仅仅是一种环境——互联网的使用环境、氛围、内容等。而对于网站设计和制作者来说，它是一系列技术的复合总称——网站前台布局、后台程序、美工、数据库开发等。

子任务 1.1.1 网页与网站

网页是一个包含 HTML 标签的能够被浏览器解释执行的纯文本文件。网页中包含文字、图像、表格、超链接、音频、视频等元素，其中，文字、图像、超链接是组成网页最基本的 3 个元素。例如，在浏览器地址栏中输入网易的网址 <https://www.163.com>，回车即可打开其官方网站，如图 1-1 所示。



图 1-1 网易首页

为了快速了解网页是如何形成的,可以查看网页的源代码。在打开的页面中右击,在弹出的快捷菜单中选择“查看源代码”选项即可查看源代码,可以看出它是一个纯文本,如图 1-2 所示。而人们看到的网页效果则是由这些纯文本代码被浏览器渲染后的效果。

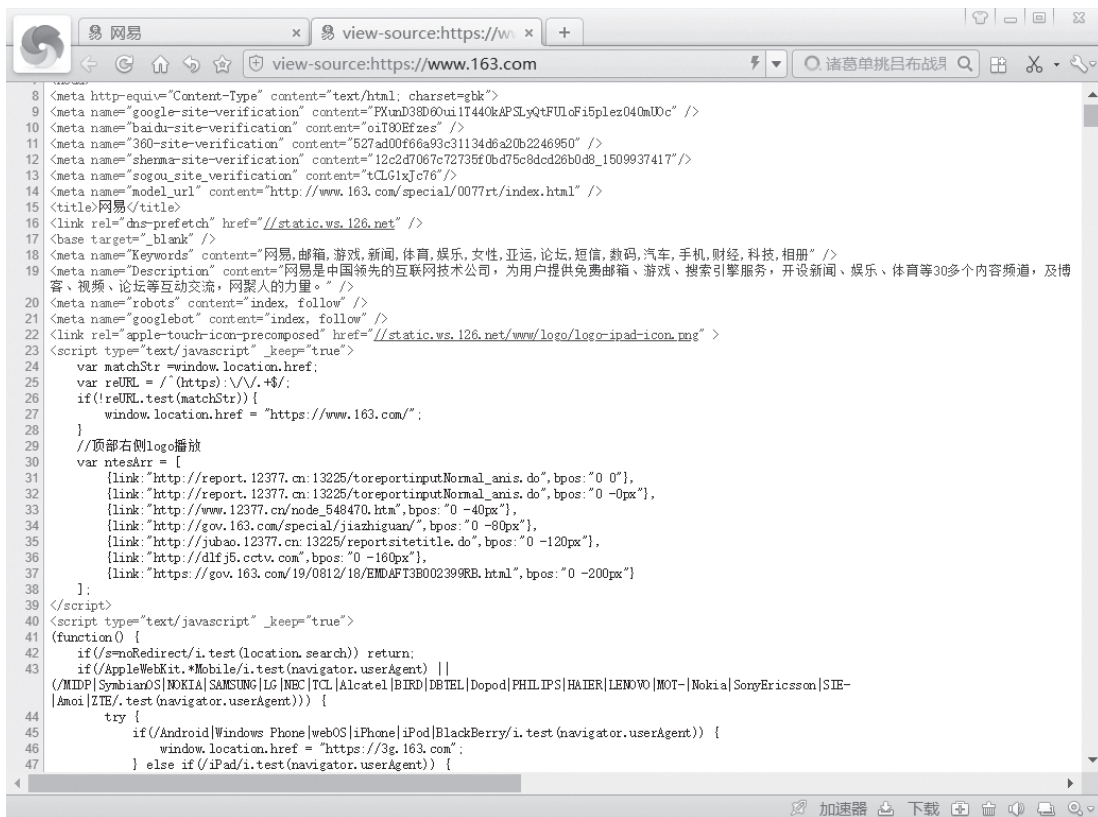


图 1-2 网易首页源代码

子任务 1.1.2 网页相关名词

对于从事网页制作的人来说,与互联网相关的一些专业术语是必须要了解的,例如,Internet、WWW、URL、DNS、HTTP 等。

1. Internet

Internet 即通常所说的互联网,是由一些使用公用语言互相通信的计算机连接而成的网络,简单来说,互联网就是将世界范围内不同国家、不同地区的众多计算机连接起来形成的网络。互联网实现了全球信息的共享,形成了一个能够共同参与、相互交流的互动平台,通过互联网,远在千里之外的朋友可以相互发送邮件、共同完成一项工作或共同娱乐。

2. WWW

万维网(World Wide Web, WWW)不是网络,它是 Internet 提供的一种网页浏览服务,人们上网时通过浏览器阅读网页信息就是在使用 WWW 服务,WWW 是 Internet 提供的最主要的服务,其他网络功能如网上聊天、网上购物等,都是基于 WWW 服务的。

3. URL

统一资源定位符(Uniform Resource Locator, URL)其实就是 Web 地址,俗称“网址”,在万维网上所有文件(HTML、CSS、图片、音乐、视频等)都有唯一的 URL,

提示

网站是通过超链接建立起来的多个网页的集合,网页与网页之间通过超链接互相访问。在一个网站中,打开一个网站显示给人们的第一个页面即是网站主页。

只要知道资源的 URL，就能进行访问。URL 可以是“本地磁盘”，也可以是局域网上的某一台计算机，更多的是 Internet 上的资源，如 <http://www.sougou.com>（域名）、<http://168.103.123.465>（IP）、<http://zhidao.baidu.com/question/2072537.html> 都是 URL，也就是网址。

4.DNS

在 Internet 上，域名与 IP 地址之间是一一对应的，域名（如网易的域名 www.163.com）虽然便于记忆，但计算机不认识域名，只认识 IP 地址，将好记的域名转换成 IP 地址的过程称为域名解析，域名解析系统（Domain Name System，DNS）就是进行域名解析的系统。在浏览器中输入域名，然后由域名解析服务器 DNS 将域名解析成 IP 地址，找到对应的主机，最后寻找到对应的资源。

5.HTTP

超文本传输协议（HyperText Transfer Protocol，HTTP）是一种详细规定了浏览器和万维网服务器之间如何互相通信的规则。HTTP 是非常可靠的协议，它具有强大的自检能力，所有用户请求的文件到达客户端时，一定是准确无误的。

6.W3C

万维网联盟（World Wide Web Consortium，W3C）是国际上最著名的标准化组织。W3C 最重要的工作是发展 Web 规范，自 1994 年成立以来，已经发布了 200 多项影响深远的 Web 技术标准及实施指南，如超文本标记语言（HTML）、可扩展标记语言（XML）等。这些规范有效地促进了 Web 技术的相互兼容，对互联网的发展和应用起到了基础性和根本性的支撑作用。

子任务 1.1.3 Web 标准

由于不同的浏览器对同一个网页文件解析出来的效果可能不一致，为了让用户能够看到正常显示的网页，Web 开发者常常需要为多版本的开发而艰苦工作，当新的硬件（如移动电话）和软件（如微浏览器）出现时，这种情况会变得更加严重。为了使 Web 更好的发展，在开发新的应用程序时，浏览器开发商和站点开发商共同遵守标准就显得尤为重要，为此 W3C 与其他标准化组织共同制定了一系列的 Web 标准。

Web 标准并不是某一个标准，而是一系列标准的集合，主要包括结构（Structure）、表现（Presentation）和行为（Behavior）三个方面。

1. 结构标准

结构用于对网页元素进行整理和分类，主要包括 HTML、XML 和 XHTML。它们的具体区别如下。

（1）超文本标记语言（Hyper Text Markup Language，HTML）被用来结构化网页信息，如标题、段落和列表等，也可用来在一定程度上描述文档的外观和语义。其在 1982 年由蒂姆·伯纳斯·李创建，互联网工程任务组（Internet Engineering Task Force，IETF）用简化的标准通用标记语言（Standard Generalized Markup Language，SGML）语法对其进行规范化，现已成为国际标准，由万维网联盟（W3C）维护。

（2）可扩展标记语言（Extensible Markup Language，XML）最初的目的是为了弥补 HTML 语言的不足，它具有强大的扩展性，可用于数据的转换和描述。

(3) 可扩展超文本标记语言 (Extensible HyperText Markup Language, XHTML) 遵循 W3C 于 2000 年 1 月 26 日推荐的 XML 1.0。XML 虽然数据转换能力强大, 完全可以替代 HTML, 但面对成千上万已有的站点, 直接采用 XML 还为时过早。因此, 人们在 HTML 4.0 的基础上, 用 XML 的规则对其进行扩展, 得到了 XHTML。简单来说, 建立 XHTML 的目的就是实现 HTML 向 XML 的过渡。

2. 表现标准

表现用于设置网页元素的版式、颜色、大小等外观样式, 主要是指层叠样式表 (Cascading Style Sheets, CSS)。W3C 创建 CSS 标准的目的是以 CSS 为基础进行网页布局, 控制网页的表现。CSS 布局与 XHTML 结构语言相结合能帮助设计师分离外观与结构, 使站点的访问及维护更加容易。

3. 行为标准

行为是指网页模型的定义及交互的编写, 主要包括 DOM 和 ECMAScript 两个部分。

(1) 文档对象模型 (Document Object Model, DOM) 是一种中立于平台和语言的接口, 它允许程序和脚本动态地访问和更新文档的内容、结构和样式。

(2) ECMAScript (European Computer Manufacturers Association Script) 是 Ecma 国际 (Ecma International) 以 JavaScript 为基础制定的标准脚本语言。JavaScript 是一种基于对象和事件的驱动, 并具有相对安全性的客户端脚本语言, 广泛用于 Web 开发, 常用来给 HTML 网页添加动态功能, 如响应用户的各种操作。

任务 1.2 认识浏览器和 HTML

子任务 1.2.1 浏览器介绍

制作好的网页文件必须要使用浏览器打开才能看到网页所呈现的效果, 也就是说, 浏览器是网页运行的平台。目前常用的浏览器有 IE 浏览器 (Internet Explorer, IE)、火狐浏览器 (Mozilla Firefox, Fx)、谷歌浏览器 (Google Chrome)、Opera 浏览器 (Opera) 等, 如图 1-3 所示。基于某些因素, 这些浏览器不能完全采用统一的 Web 标准, 或者说不同的浏览器在渲染网页时的机制是不一样的, 因此不同的浏览器打开相同的网页可能会得到不一样的展现效果, 图 1-4 为不同的浏览器中的显示方式。因此在开发网站时需要 Web 开发人员对不同的浏览器进行适配。



图 1-3 常见浏览器图标



图 1-4 列表元素在不同的浏览器中的显示方式

1. 常见浏览器内核

浏览器内核是用来渲染网页内容的, 将网页的代码转换为最终呈现在访客面前的页面。



笔记



(1) Trident 内核。Trident 内核代表产品是 Internet Explorer，说起 Trident，很多人都会感到陌生，但提起 IE（Internet Explorer）则无人不知、无人不晓，由于其被集成在全世界使用率最高的操作系统 Windows 中，得到了极高的市场占有率，所以又经常称其为 IE 核心。其浏览器代表主要为 Microsoft Internet Explorer 系列。

(2) Gecko 内核。Gecko 是开源的浏览器内核，目前最主流的 Gecko 内核浏览器是 Mozilla Firefox，所以也常常称之为火狐内核。Gecko 是跨平台的，能在 Microsoft Windows、Linux 和 Mac OS X 等主要操作系统上运行。

(3) Webkit 内核。WebKit 内核是最有潜力而且已经有相当成绩的新兴内核，性能非常出色，而且对 W3C 标准的支持很完善，其浏览器代表主要有 Chrome、Safari、360、搜狗等。

(4) Presto 内核。Presto 是一个由 Opera Software 开发的浏览器内核，供 Opera 7.0 及以上版本使用。Presto 取代了旧版的 Opera 4~Opera 6 版本所使用的 Elektra 排版引擎，其加入了动态功能，其浏览器代表有 Opera、Opera Next 等。

2. 常见浏览器

(1) IE 浏览器。IE 浏览器的全称是 Internet Explorer，由微软公司推出，直接绑定在 Windows 操作系统中，无须下载安装。IE 有 6.0、7.0、8.0、9.0、10.0、11.0、edge 等版本。但是由于各种原因，一些用户仍然在使用低版本的浏览器如 IE6、IE7 等，所以在制作网页时，低版本一般也是需要兼容的。

(2) 火狐浏览器。Mozilla Firefox，中文通常译为“火狐”，是一个开源网页浏览器，使用 Gecko 引擎（非 IE 内核），可以在多种操作系统如 Windows、Mac 和 linux 上运行。Firebug 是火狐浏览器下的一款开发插件，属于火狐强力推荐的插件之一，它集 HTML 查看和编辑、JavaScript 控制台、网络状况监视器于一体，是开发 HTML、CSS、JavaScript 等的得力助手。

(3) 谷歌浏览器。Google Chrome，又称为谷歌浏览器，是由谷歌（Google）公司开发的开放原码网页浏览器。该浏览器是基于其他开放原始码软件所撰写的，包括 WebKit 和 Mozilla，目标是提升稳定性、速度 and 安全性，并创造出简单有效的使用界面。

说明

IE、火狐、谷歌是目前互联网上的三大浏览器，其他常用的浏览器还有苹果的 Safari、欧朋的 Opera 等。在实际开发中对于一般网站，只要兼容 IE、火狐和谷歌浏览器，就能满足绝大多数用户的需求了。

子任务 1.2.2 HTML 开发历程

超文本标记语言是万维网的统一语言，主要用来对网页中的文本、图片、声音等内容进行描述。HTML 提供了许多标签，如段落标签、标题标签、超链接标签、图片标签等，网页中需要定义什么内容，就用相应的标签进行描述，通过它所提供的标签，人类已经创建了各种各样令人惊奇的超链接文档网络，如 Amazon、eBay、Wikipedia、个人博客等。HTML 5 是这门通用语言的最新版。超文本标记语言的发展过程如表 1-1 所示。

表 1-1 超文本标记语言的发展过程

版本	发布日期	说明
超文本标记语言（第一版）	1993 年 6 月	作为互联网工程工作小组（IETF）工作草案发布（并非标准）
HTML 2.0	1995 年 11 月	作为 RFC1866 发布，于 2000 年 6 月 RFC2854 发布后被宣布已经过时

续表



版本	发布日期	说明
HTML 3.2	1996 年 1 月 14 日	W3C 推荐标准
HTML 4.0	1997 年 12 月 18 日	W3C 推荐标准
HTML 4.01	1999 年 12 月 24 日	W3C 推荐标准，微小改进
ISO HTML	2000 年 5 月 15 日	基于严格的 HTML 4.01 语法，是国际标准化组织和国际电工委员会的标准
XHTML 1.0	2000 年 1 月 26 日	W3C 推荐标准（修订后于 2002 年 8 月 1 日重新发布）
XHTML 1.1	2001 年 5 月 31 日	较 XHTML 1.0 有微小改进
HTML 5 草案	2008 年 1 月	前身名为 Web Applications 1.0，2004 年被 WHATWG 提出，于 2007 年被 W3C 接纳，并成立了新的 HTML 工作团队。
HTML 5	2014 年 10 月 28 日	W3C 推荐标准

1. 从 IETF 到 W3C——HTML 4 的诞生过程

实际上，根本不存在 HTML 1。最早的 HTML 官方规范是由因特网工程任务组（Internet Engineering Task Force，IETF）发布的 HTML 2.0。这一规范中的许多特性都是在已有实现的基础上归纳总结出来的。例如，1994 年居于市场领导地位的 Mosaic 浏览器提供了 标签，开发人员可以通过该标签在自己的文档中嵌入图像。后来， 标签就出现在了 HTML 2.0 中。

继 IETF 之后，W3C 成为 HTML 后续标准的制定者。20 世纪 90 年代中期以后，W3C 对 HTML 进行了几次升级，直至 1999 年发布的 HTML 4.01。

2.XHTML 1——符合 XML 标准的 HTML

HTML 4.01 之后的修订版为 XHTML 1.0。其中，X 表示的是“eXtensible（可扩展）”。XHTML 1.0 规范的内容与 HTML 4.01 完全相同。没有添加任何新元素或新属性。这两个规范唯一的差别就是对 HTML 语法作出了不同的规定。HTML 为开发人员提供了很大的自由度，他们可以按照自己的意愿去编写元素和属性，但 XHTML 要求开发人员遵从 XML 规则。XML 是 W3C 大多数技术规范的基础，也是一种更为严格的标记语言。

更加严格的语法规则并没有什么坏处，反而可以促使开发人员按照统一的样式来编写标签。此前的标签和属性可以是大写、小写，或者任意大小写字母的组合，而 XHTML 1.0 文档则要求所有标签和属性都必须为小写。

XHTML 1.0 发布的时候恰逢浏览器普遍开始支持 CSS。开发人员意识到了网页标准的出现，特别是在 Web 标准项目（The Web Standards Project）的倡导下，XHTML 规定的这种更为严格的语法被看成是编写标记的“最佳实践”。

在此之后，W3C 发布了 XHTML 1.1，较 XHTML 1.0 有微小改进。如果说 XHTML 1.0 只不过是 XML 重新表示的 HTML，那么 XHTML 1.1 才是真正且纯粹的 XML。也就是说，不能将 text/html 的 MIME 类型提供给 XHTML 1.1 文档。但是如果开发人员以 XML 的 MIME 类型来发布文档，那么当时世界上最流行的 Web 浏览器——Internet Explorer 就无法呈现该文档。



3.XHTML 2——不被接受

W3C 在接管 HTML 的时候，HTML 已经发展到了第 4 版（Version 4）。然后他们又着手开发 XHTML 2，其目的是将 Web 建立在 XML 之上。

虽然 XHTML 2 的名字听起来与 XHTML 1 非常类似，但它们的差别却非常之大。与 XHTML 1 不同，XHTML 2 与已有的网页内容都不兼容，甚至与以前版本的 HTML 也不兼容。XHTML 2 的目的就是成为一门纯粹的语言，也就是不与以前的规范建立任何关系。但这却成为影响 HTML 发展的重要因素。

4. 分裂——WHATWG TF

W3C 意图放弃 HTML，而力图发展 XML 技术。于是，一个以推动网络 HTML 5 标准为目的的组织成立了，即网页超文本应用技术工作组（Web Hypertext Application Technology Working Group, WHATWG）。WHATWG 邮件列表公布于 2004 年 6 月 4 日，在 Opera-Mozilla 宣布加入后的第二天便否决了由万维网联盟成员在 W3C 工作室的 Web 标准。

5. 从 Web Apps 1.0 到 HTML 5

从一开始，WHATWG 的工作方式就与 W3C 截然不同。W3C 采取基于表决的方式：提出议题、讨论议题，然后投票表决。而在 WHATWG，同样会提出和讨论议题，但哪些特性可以写入规范的最终决定权在编辑手中。这个编辑就是伊恩·希克森。

表面上看，W3C 的流程更民主也更公平。但实际上，由于内部争论等各种因素，经常会导致流程停滞不前。而在 WHATWG，任何人都可以自由地发表意见相对自由，但负责拍板的只有编辑一个人，因此工作效率明显提高很多。其实编辑也并非拥有绝对的权力：一个仅由受邀人员组成的指导委员会可以质疑编辑的做法。

最初，WHATWG 的大部分工作被分为两个规范：Web Form 2.0 和 Web Apps 1.0。这两个规范都是在 HTML 的基础上扩展而来的。后来，这两个规范又被合并到一起，简单地称作 HTML 5。

子任务 1.2.3 HTML 5 设计理念

HTML 5 是一个里程碑式的规范，它为下一代 Web 发展指明了方向，下面就来探析 HTML 5 语言设计理念。

1. 避免不必要的复杂性

例如，使用 HTML 4.01 规范时，如果定义 DOCTYPE，其声明格式如下。

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/html4/strict.dtd">
```

很少有人能够记住这行代码，本来它要告诉浏览器的是“这个文档是 HTML 4.01 的文档”。而在 HTML 5 中，省掉了不必要的复杂性，DOCTYPE 声明简化为如下格式。

```
<!DOCTYPE html>
```

再如，使用 HTML 4.01 时，如果要指定文档的字符编码为 utf-8，需要在头部信息中进行如下定义。


```
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
```

而在 HTML 5 中，只要输入如下代码即可。

```
<meta charset="UTF-8">
```

2. 支持已有内容

支持已有内容这一点非常重要，因为很多人都认为 HTML 5 规范很新，它应该代表着未来发展的方向，应该将 Web 推向一个新的发展阶段。虽然人们都会考虑让 Web 的未来发展更好，但工作组则必须考虑过去，WHATWG 这个工作组中有很多人代表的是浏览器厂商，他们肯定必须考虑支持已有的内容。只要想构建一个浏览器，就必须记住一个原则：必须支持已有内容。

如，在 HTML 中编写包含一个 `img` 元素和 `p` 元素的代码段，可以使用 4 种不同的语法，将其中一段代码交给浏览器，浏览器都会生成相同的 DOM 树，没有任何问题，这 4 种写法唯一不同的就是语法。

```
<body>
  <div class="first">
    
    <p> 欢迎选购 </p>
  </div>
  <div class="second">
    
    <p> 欢迎选购
  </div>
  <div class="THIRD">
    <IMG SRC="images/yx.jpg" ALT="mi">
    <p> 欢迎选购 </p>
  </div>
  <div class="fourth">
    <img src=images/yx.jpg alt=mi>
    <p> 欢迎选购 </p>
  </div>
</body>
```

3. 解决现实问题

HTML 5 的另一个设计理念是解决现实问题。在这之前的格式和规范要解决的基本上都是理论问题，而非现实问题。这个设计理念是要从理论上承认人们普遍存在的问题，消除敏感问题，理念真正用于解决今天人们所面临的现实问题、令人头痛的问题。

例如，假设 HTML 或 XHTML 1 已有一块内容，这块内容包含一个标题、一个段落，如果想使它们全部都可以生成链接，必须使用以下 2 个链接元素。



笔记



```
<h2><a href=""> 标题文本 </a></h2>
<p><a href=""> 段落文本 </a></p>
```

在 HTML 5 中，只要简单地将所有内容都写在一个链接元素中就可以了。

```
<a href="#">
<h2> 标题文本 </h2>
<p> 段落文本 </p>
</a>
```

链接包含的都是块元素，但现在可以用一个元素包含它们。碰到类似的情况，就给几个块元素加上相同的链接，因为这样做解决了一个现实的问题：不必重新写代码来支持这种写法。

4. 求真务实

对于 HTML 来说，求真务实的含义是，在解决那些令人头痛的问题之前，先看看人们为应对这些问题都想出了什么办法。HTML 5 中新的语义元素就是遵循求真务实理念的反映。新增的元素不算多，谈不上无限扩展，但意义非同一般。这些新元素涉及头部（header）、脚部（footer）、分区（section）、文章（article）等，即使不使用 HTML 5 大家也不会陌生，因为它们都是经常使用的类名或 id 名，如 class="header"（head），id=footer（foot）。

（1）未使用 HTML 5 时规划网页结构的核心代码如下。

```
<body>
  <div id="header"> 头部 </div>
  <div id="navigation"> 导航 </div>
  <div id="main"> 内容区 </div>
  <div id="sidebar"> 侧边栏 </div>
  <div id="footer"> 脚部 </div>
</body>
```

（2）使用 HTML 5 新增语义元素后的核心代码如下。

```
<body>
  <header> 头部 </header>
  <nav> 导航 </nav>
  <div id="main">...</div>
  <aside> 侧边栏 </aside>
  <footer> 脚部 </footer>
</body>
```

直观上看，这些新增元素仅仅取代了原来的 div，其实不然，这些元素在一个页面中可以使用多次，详细内容将在后面进行介绍。

5. 平稳退化

HTML 遵循平稳退化这条原理的例子就是使用 type 属性增强表单。现在开始就可以使用这些新值，如果浏览器不支持就将它解析成“text”，而这正是一个浏览器实现平稳退化的好例子。HTML 5 中新增的 type 属性列举如下。

```
<input type="number" >
<input type="search" >
<input type="range" >
<input type="email">
<input type="tel" >
<input type="date" >
<input type="url">
```

6. 最终用户优先

这个设计理念本质上是一种解决冲突的机制。换句话说，当面临一个要解决的问题时，如果 W3C 和 WHATWG 给出不同的解决方案，一旦遇到冲突，最终用户优先，然后是作者，其次是实现者，再次是标准制定者，最后才是理论的完满。

理论上的完满，大致是指尽可能创建出最完美的格式，标准制定都指的是 WHATWG、W3C 等，实现者是指浏览器厂商，作者是开发人员，用户是第一位的。

根据最终用户优先的原理，开发人员在链条中的位置高于实现者，假如开发人员发现了规范中的某些地方有问题，又不支持实现这个特性，那么就等于将相应的特性给否定了，规范中就得删除，因为开发人员的声音具有最高的权重，本质上开发人员就有了更大的发言权。

任务 1.3 熟悉网站开发流程和工具

子任务 1.3.1 网站开发流程

网站的创建需要经历前期准备（需求分析）、中期制作和后期测试发布 3 个大阶段。前期准备包括了解网站的业务背景、明确网站的设计风格、确定网站的内容等；中期制作主要包括创建站点、制作页面、制作样式；后期测试和发布包括检查页面效果是否美观、链接是否完好、是否与浏览器兼容，以及发布网站等，如图 1-5 所示。

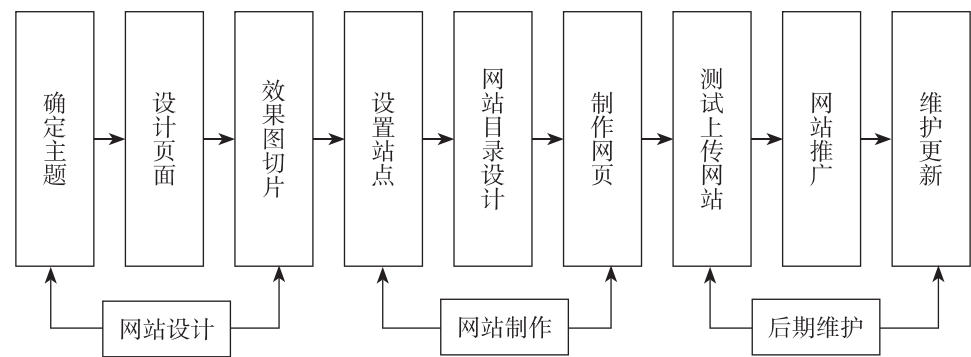


图 1-5 网站开发流程





1. 确定主题

确定网站主题可以分为分析策划和资料收集两个部分。制作网站前，首先需要对网站进行整体分析，分析网站的功能与建站的目的，确定用户群和网站内容，即确定网站的主题。在确定主题后，即可绘制网站架构图，并搜集建站所需的相关资料和素材。图 1-6 为某企业的网站架构。另外可要求客户提供与公司相关的文字和图片等资料，如公司介绍、产品图片等。

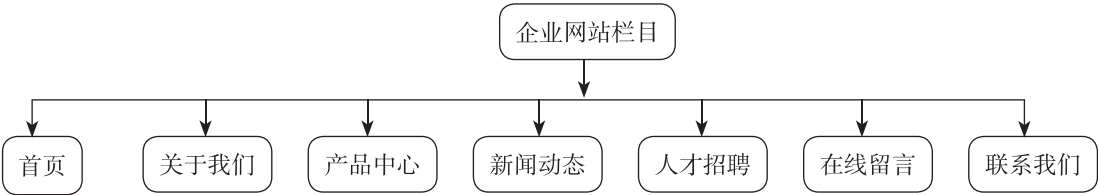


图 1-6 某企业的网站架构

2. 设计页面

网站设计师与客户沟通，并了解客户的基本要求后，制定网站建设方案并使用 Photoshop 等图像处理软件进行页面效果图设计。页面效果图主要包括首页效果图和各栏目效果图。将效果图设计好后交给客户查看，客户查看后提出修改意见，设计人员根据客户意见进行修改，并最终确定网站页面效果图。

3. 效果图切片

当效果图得到客户认可后，设计师可使用 Photoshop 等图像处理软件中的切片工具将效果图切割保存为 JPG、GIF 或 PNG 等格式的小图片，将它们作为网页制作的图像素材。

4. 网站目录设计

从这一步开始进入到网页制作阶段，首先在本地磁盘创建网站根目录及子文件目录，将制作网页需要的素材资源分类放置在各文件夹中，如图 1-7 所示。

F:\web >			
名称	修改日期	类型	大小
css	2019/8/1 19:50	文件夹	
images	2019/8/1 19:50	文件夹	
js	2019/8/1 19:50	文件夹	
index.html	2019/8/1 19:51	360 Chrome HT...	0 KB

图 1-7 网站目录

5. 制作网页

网页的制作可以分为结构层制作、样式层制作和行为层制作三部分。结构层制作就是使用 HTML 代码搭建网页的主体结构，如文字、图片、超链接等。在结构层制作好后，就可以使用层叠样式表（CSS）及动态脚本语言（JavaScript）来制作网页的样式层及行为层，完成网页布局 and 外观设置。

6. 测试上传网站

在将网站上传到服务器前，需要对其进行测试及优化，测试包括兼容性测试、链接测试等；优化是尽可能减小网页文件的体积及日后发生错误的概率。完成测试和优化后，就可以利用 FTP 工具将网站发布到所申请的空间服务器上。网站上传成功后，可通过浏览

器进行实际测试。

7. 网站推广

网站上传后, 需要进行推广和宣传, 以提高网站的访问量及知名度。推广网站的方法有很多, 如群发邮件、注册到搜索引擎、借助同类网站留言、加入友情链接、传统媒体宣传等。

8. 维护更新

制作好网站后, 还需要日常对网站进行维护和更新。站点只有不断地补充新内容, 才能吸引浏览者, 延长使用寿命。

子任务 1.3.2 Web 前端开发工具

网页制作过程中, 为了开发方便, 通常会选择一些较便捷的工具, 如 Notepad++、Sublime、WebStorm、DreamWeaver 等。

1. Notepad++

Notepad++ 是 Windows 操作系统下的一套文本编辑器 (软件版权许可证: GPL), 有完整的中文化接口及支持多国语言编写的功能 (UTF8 技术), 其支持 C、C++、Java、C#、XML、HTML、PHP、JavaScript 等语言。

Notepad++ 功能比 Windows 中的记事本 (Notepad) 强大, 除了可以用来制作一般的纯文字说明文件, 也十分适合编写计算机程序代码。Notepad++ 不仅有语法高亮度显示, 也有语法折叠功能, 并且支持宏以及扩充基本功能的外挂模组, 图 1-8 为 Notepad++ 工作界面。

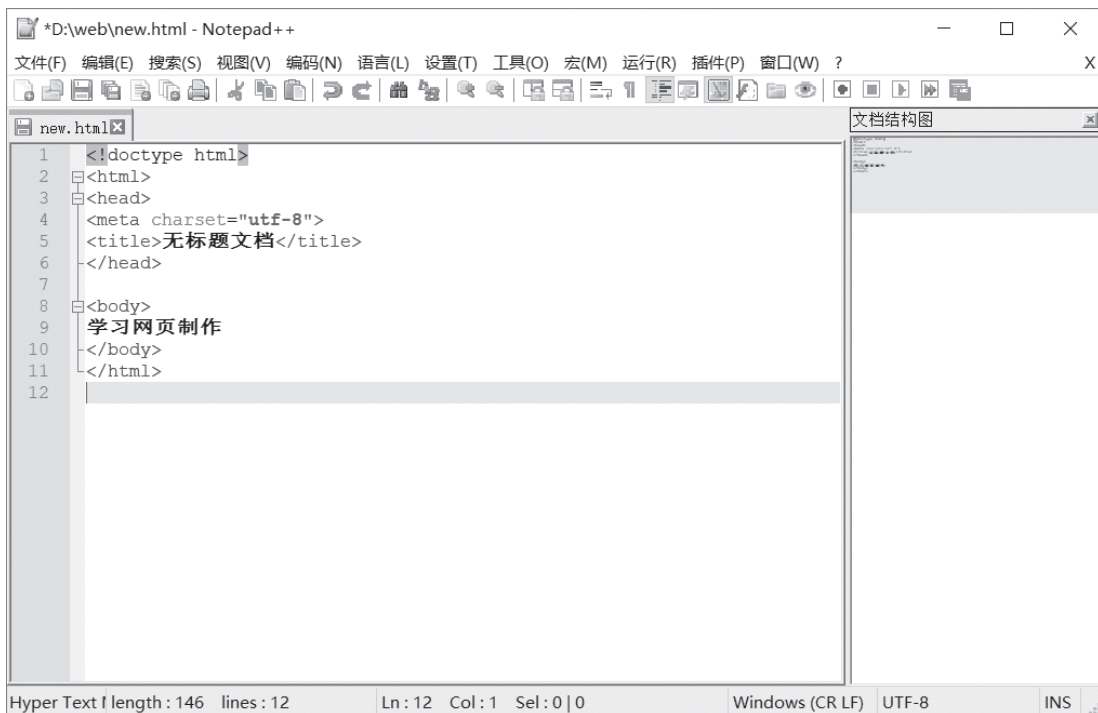


图 1-8 Notepad++ 工作界面

2. Sublime Text

Sublime Text 是一个代码编辑器, 具有漂亮的用户界面和强大的功能, 如代码缩略图、



笔记



Python 插件、代码段等，还可自定义绑定按键、菜单和工具栏。Sublime Text 的主要功能包括：拼写检查、书签、完整的 Python API、Goto 功能、即时项目切换、多选择、多窗口等。Sublime Text 是一个跨平台的编辑器，同时支持 Windows、Linux、Mac OS X 等操作系统，Sublime Text 工作界面如图 1-9 所示。

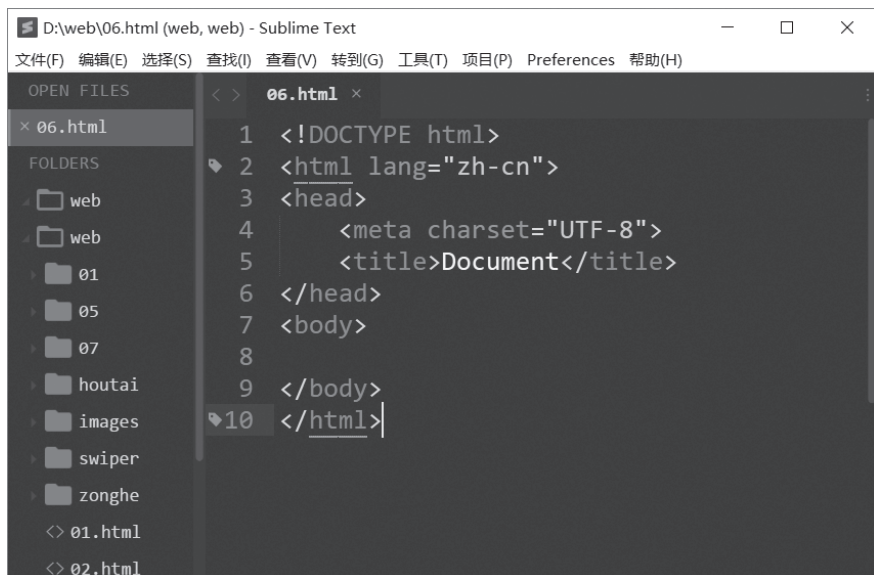


图 1-9 Sublime Text 工作界面

3.WebStorm

WebStorm 是 JetBrains 公司旗下的一款 JavaScript 开发工具。被中国广大 JavaScript 开发者誉为“Web 前端开发神器”“最强大的 HTML 5 编辑器”“最智能的 JavaScript IDE”等。其与 IntelliJ IDEA 同源，继承了 IntelliJ IDEA 强大的 JS 部分的功能。其特色是具有专业的 JavaScript 开发工具。WebStorm 工作界面如图 1-10 所示。

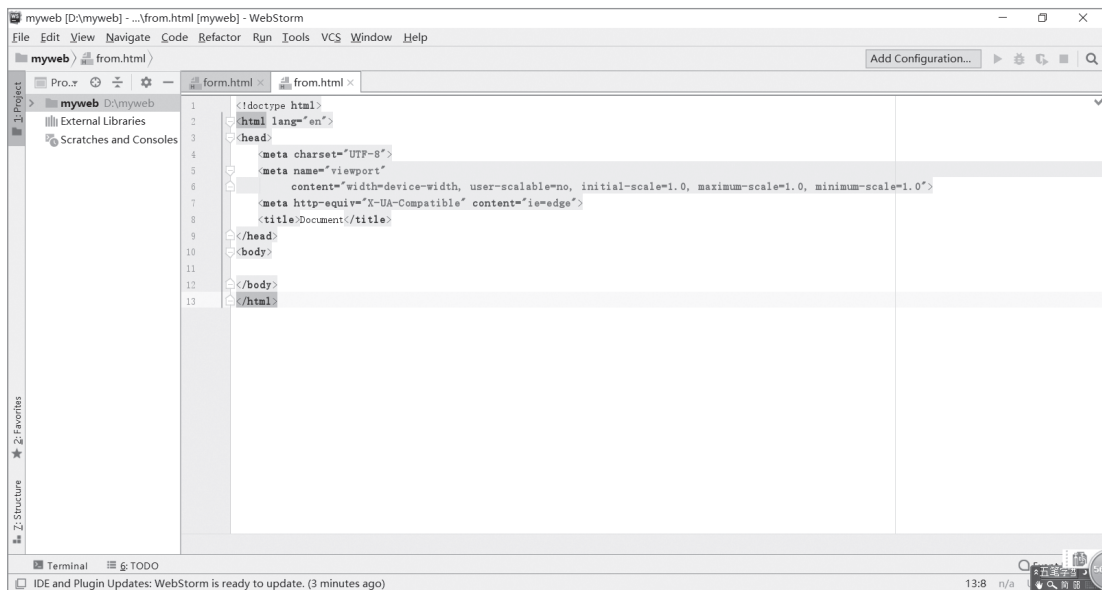


图 1-10 WebStorm 工作界面

4.Dreamweaver

Adobe Dreamweaver，简称为“DW”，中文名称为“梦想编织者”，是美国 MACROMEDIA

公司开发的集网页制作和管理网站于一身的所见即所得网页编辑器，DW 是第一套为专业网页设计师特别开发的视觉化网页开发工具，利用它可以轻而易举地制作出跨越平台和浏览器限制的充满动感的网页。其特色是具有最易掌握的可视化网页编辑工具。Dreamweaver 工作界面如图 1-11 所示。

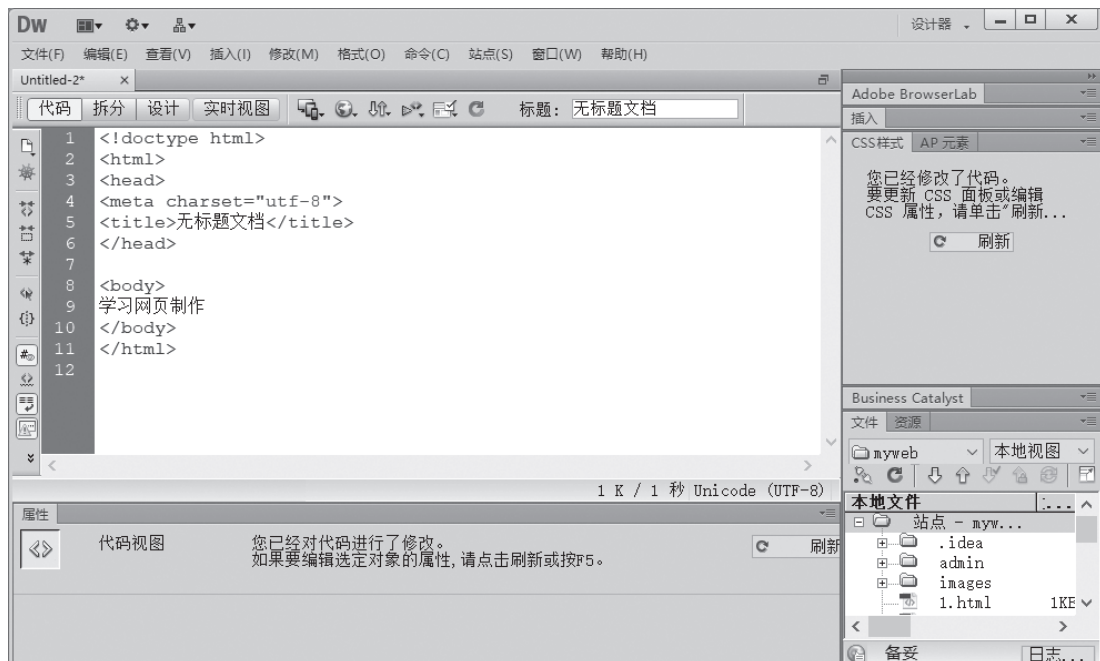


图 1-11 Dreamweaver 工作界面

子任务 1.3.3 编写第一个 HTML 5 页面

HTML 5 中不仅增加了很多的页面标记，而且与 HTML 4 相比，整体页面的结构也发生了根本的变化。下面将使用 HTML 5 来编写一个简单的页面，代码如下。

```
<!DOCTYPE html>
<html lang="zh-cn">
<head>
  <meta charset="UTF-8">
  <title>我的第一个网页 </title>
</head>
<body>
  HTML 是什么？ <br>
  HTML 指的是超文本标记语言 (Hyper Text Markup Language)<br>
  HTML 不是一种编程语言，而是一种标记语言 (markup language)<br>
  标记语言是一套标记标签 (markup tag)<br>
  HTML 使用标记标签来描述网页 <br>
</body>
</html>
```

笔记

该页面在浏览器中的显示效果即 HTML 页面效果如图 1-12 所示。



图 1-12 HTML 页面效果