

Web
前端开发
系列丛书

CSS入门与实践

胡俊卿 编著



清华大学出版社

CSS入门与实践

胡俊卿 编著

本书介绍CSS的入门知识，并详细讲解CSS控制网页布局、表现、动画、特效、交互等知识，并附带大量实例，帮助读者快速掌握CSS的入门知识。本书可作为高等院校计算机专业及相关专业的教材，也可作为从事Web前端开发的工程师的参考书。

1973年，随着个人电脑的普及，人们开始使用图形用户界面(GUI)来操作计算机。随着互联网的普及，人们开始使用浏览器来浏览网页。浏览器为网页提供了丰富的内容，而网页就是这些内容的载体。随着浏览器及网络带宽的不断提升，以及个人计算机和网络设备的不断提升，浏览器已经不再单纯用来呈现博客、新闻等内容，它已经成为了人们办公、社交、娱乐甚至游戏应用出现在浏览器中的应用平台。越来越多的办公、社交、娱乐甚至游戏应用出现在浏览器中。

无论Web应用有多强大，我们都需要使用CSS控制网页的布局及内容的表现。读者将通过本书学习CSS的入门知识，并掌握通过CSS控制网页布局、表现、动画、特效、交互等方法。

本书内容包含浏览器和开发工具的介绍，以及CSS的入门知识。通过大量实例引导读者了解和学习CSS的入门知识，并掌握通过CSS控制网页布局、表现、动画、特效、交互等方法。

本书是面向初学者的入门书籍，适合在学习本书之前，对Web前端开发有一定的了解。在准备好一定的基础知识后，读者可以通过本书快速掌握CSS的入门知识。

由于定位及篇幅的原因，本书仍然会涉及一些Web前端开发的相关知识。读者在学习本书的过程中，可以结合其他相关书籍，进一步加深对Web前端开发的理解。

本书由清华大学出版社出版，定价39.00元。本书的ISBN号为978-7-302-51181-1。本书的出版日期为2018年12月。本书的开本为185mm×260mm。本书的印张为12.5。本书的字数为300千字。本书的ISBN号为978-7-302-51181-1。



内 容 简 介

虽然近几年前端相关的技术不断推新,但 CSS 作为网页开发的三大基础技术之一,其地位不会被轻易取代。本书内容包含浏览器和开发工具介绍、CSS 基础知识和应用方法,并通过贯穿全书的应用案例引导读者了解和学习 CSS 的相关知识,最终利用这些知识分析和解决问题。本书旨在帮助前端初学者快速入门和上手 CSS 开发,并掌握 CSS 的基础知识和应用方法,最终能够分析和使用 CSS 实现网页布局和样式控制。由于读者定位和篇幅限制的原因,本书的重点是 CSS 基础知识的讲解和实际应用,同时会涉及一些流行的技术和应用方式,以及部分 HTML 和 JavaScript 的相关知识。

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。
版权所有,侵权必究。侵权举报电话:010-62782989 13701121933

图书在版编目(CIP)数据

CSS 入门与实践/胡俊卿编著. —北京:清华大学出版社,2018
(Web 前端开发系列丛书)
ISBN 978-7-302-51004-8

I. ①C… II. ①胡… III. ①网页制作工具 IV. ①TP393.092.2

中国版本图书馆 CIP 数据核字(2018)第 191831 号

责任编辑:郭 赛
封面设计:傅瑞学
责任校对:徐俊伟
责任印制:宋 林

出版发行:清华大学出版社

网 址: <http://www.tup.com.cn>, <http://www.wqbook.com>

地 址:北京清华大学学研大厦 A 座 邮 编:100084

社 总 机:010-62770175 邮 购:010-62786544

投稿与读者服务:010-62776969, c-service@tup.tsinghua.edu.cn

质量反馈:010-62772015, zhiliang@tup.tsinghua.edu.cn

课件下载: <http://www.tup.com.cn>, 010-62795954

印 装 者:清华大学印刷厂

经 销:全国新华书店

开 本:185mm×260mm 印 张:10.5 字 数:251 千字

版 次:2018 年 12 月第 1 版 印 次:2018 年 12 月第 1 次印刷

定 价:39.00 元

产品编号:077652-01

清华大学出版社

010-62770175

需要完整PDF请访问: www.ertongbook.com

前 言

图形用户界面(Graphical User Interface, GUI)是指使用图形化的方式呈现计算机等设备的操作界面。GUI通过屏幕等设备上不同颜色的点(像素)为用户提供文本、图像等内容的呈现及鼠标单击、键盘输入和触摸屏点击等操作的反馈。

1973年,施乐公司首次在 Alto 计算机中搭载 GUI。随后,苹果公司和微软公司相继推出搭载 GUI 的个人计算机和操作系统。目前,GUI 已经成为用户与个人计算机、智能手机等电子设备最主要的沟通和交互方式。

对大部分用户来说,浏览器是最常用的 GUI 程序之一,也是用户进行搜索、阅读、分享和娱乐的平台。浏览器为用户提供了文本、图像和音/视频等丰富的内容,而网页就是这些内容的载体。随着浏览器及前端技术的发展,以及个人计算机和移动设备性能的大幅提升,浏览器已经不再单纯用来呈现博客、新闻等内容,它已经成为最广泛、最受欢迎的应用平台,越来越多的办公、社交、娱乐甚至游戏应用出现在浏览器中。

无论 Web 应用有多强大、多复杂,它们都需要使用 CSS 控制网页的布局及内容的呈现。读者将通过本书学习 CSS 的基础知识,并掌握通过 CSS 控制网页布局及内容呈现的方法。

本书内容包含浏览器和开发工具的介绍,以及 CSS 的基础知识和应用方法。通过应用案例引导读者了解和学习 CSS 的相关知识,并利用这些知识分析和解决问题。读者在阅读本书后,基本能够掌握通过 CSS 控制网页布局及内容呈现的方法,了解常用的 CSS 技巧,并具备分析和解决 CSS 相关问题的能力。

本书是面向前端初学者的自学教材,以及面向初级前端工程师的 CSS 参考资料。读者在学习本书时,只需要掌握基本的计算机操作和文件管理知识,并对 HTML 有基本的了解。在准备好一台计算机,甚至平板电脑和手机之后,就可以开始学习和练习了。

由于定位及篇幅限制,本书将内容重点放在 CSS 基础知识的讲解和实际应用上,但仍然会涉及一些流行的技术和应用方法,以及部分 HTML 和 JavaScript 的相关知识。读者在学习和实践的过程中,也可以通过互联网或相关书籍学习前端开发及 Web 应用开发的相关知识和技术。

3.4.1 @charset	15
3.4.2 @import	15
3.4.3 @media	16
3.5 注释	17
3.6 错误处理	17
3.6.1 无效的值	17
3.6.2 使用浏览器解决样式问题	18

胡俊卿

2018年5月

目 录

第 1 章 了解 CSS	1
1.1 CSS 的作用	1
1.2 工作机制	1
1.3 网页浏览器	2
1.4 兼容性问题	2
第 2 章 开发工具	4
2.1 浏览器和开发者工具	4
2.1.1 Google Chrome 浏览器	4
2.1.2 开发者工具	5
2.2 代码编辑工具	7
2.2.1 Visual Studio Code	7
2.2.2 在线编辑和预览工具	8
2.3 静态文件服务器	8
2.4 代码存储和版本控制——Git	9
2.5 参考文档和资料	9
第 3 章 开始编写 CSS	11
3.1 CSS 规则	11
3.2 选择器	11
3.3 属性声明	13
3.3.1 属性名称	13
3.3.2 属性前缀	13
3.3.3 属性值	14
3.4 @规则	15
3.4.1 @charset	15
3.4.2 @import	15
3.4.3 @media	16
3.5 注释	17
3.6 错误处理	17
3.6.1 无效的值	17
3.6.2 使用浏览器解决样式问题	18

第 4 章 网页中的样式	21
4.1 将 CSS 应用到网页中	21
4.1.1 内联样式	21
4.1.2 内部样式	21
4.1.3 外部样式	22
4.2 使用 JS 控制样式	22
4.3 相对路径和绝对路径	23
4.4 网页中存在的样式	24
4.4.1 浏览器默认样式	24
4.4.2 网页开发者定义的样式	24
4.4.3 用户自定义样式	25
4.5 样式的层叠	25
4.5.1 根据来源确定优先级	25
4.5.2 根据规则顺序确定优先级	26
4.5.3 !important 标记	26
4.6 样式的继承	27
第 5 章 选择器	28
5.1 基本选择器	28
5.2 属性选择器	29
5.3 伪类选择器	29
5.3.1 a 元素专属的几种状态	30
5.3.2 :active 状态	30
5.3.3 :hover 状态	30
5.3.4 :focus 状态	30
5.3.5 :enabled 状态与 :disabled 状态	30
5.3.6 :checked 状态	31
5.3.7 :root 状态	31
5.3.8 子元素位置	31
5.3.9 子元素类型	32
5.3.10 :not(selector)	32
5.3.11 :fullscreen	32
5.4 伪元素选择器	32
5.4.1 ::after	33
5.4.2 ::before	33
5.4.3 ::first-letter	33
5.4.4 ::first-line	33
5.4.5 ::selection	33

5.5	关系选择器	34
5.6	选择器组合	34
5.6.1	叠加	34
5.6.2	组合	35
5.7	选择器优先级	35
第 6 章 CSS 属性值		37
6.1	整数	37
6.2	数值	37
6.3	百分数	37
6.4	尺寸值	37
6.5	角度值	38
6.6	时间	38
6.7	字符串	39
6.8	关键词	39
6.9	颜色值	39
6.10	函数	41
第 7 章 字体和文本		42
7.1	本章实例	42
7.2	字体相关属性	44
7.2.1	font-family 属性	44
7.2.2	font-size 属性	45
7.2.3	font-style 属性	46
7.2.4	font-weight 属性	46
7.2.5	line-height 属性	47
7.2.6	font 属性	47
7.2.7	@font-face 声明	47
7.3	文本相关属性	49
7.3.1	direction 属性	49
7.3.2	letter-spacing 属性	49
7.3.3	word-spacing 属性	49
7.3.4	white-space 属性	50
7.3.5	word-break 属性	52
7.3.6	text-align 属性	52
7.3.7	text-justify 属性	55
7.3.8	text-indent 属性	55
7.3.9	text-transform 属性	55

7.3.10	text-overflow 属性	56
7.3.11	word-wrap 属性	57
7.3.12	user-select 属性	57
7.4	装饰性样式	58
7.4.1	color 属性	58
7.4.2	text-decoration 属性	58
7.4.3	text-shadow 属性	59
第 8 章	盒模型	60
8.1	本章实例——音乐网站主页	60
8.1.1	页面框架	61
8.1.2	网格布局	64
8.2	元素的呈现方式	66
8.3	盒子的组成	68
8.4	宽度和高度	70
8.5	内边距	71
8.6	边框	71
8.7	外边距	75
8.8	尺寸计算	78
8.9	内容溢出	78
第 9 章	弹性盒模型	81
9.1	本章实例——弹性多列布局	81
9.2	相关概念	82
9.3	弹性容器相关属性	83
9.4	弹性子元素相关属性	88
第 10 章	装饰性样式	92
10.1	边框背景图	92
10.2	轮廓	95
10.3	圆角	96
10.4	背景	99
10.5	渐变	104
10.6	盒阴影	107
10.7	透明度	108
第 11 章	定位	110
11.1	相关属性	110
11.1.1	position 属性	110

11.1.2	top、bottom、left、right 属性	112
11.1.3	z-index 属性	113
11.2	应用场景	114
11.2.1	Tooltip 组件	114
11.2.2	Dropdown 组件	116
11.2.3	Dialog 组件	118
第 12 章	浮动	120
12.1	浮动的特征	120
12.2	清除浮动	122
12.3	应用场景	124
第 13 章	变换	127
13.1	添加变换效果	127
13.2	支持的变换类型	129
13.3	设置变换原点	130
第 14 章	过渡	131
14.1	添加过渡效果	131
14.2	相关属性	132
第 15 章	动画	134
15.1	动画实例	134
15.2	定义动画	136
15.3	使用和控制动画	137
第 16 章	其他属性	140
16.1	visibility 属性	140
16.2	resize 属性	141
16.3	cursor 属性	142
16.4	content 属性	143
16.5	filter 属性	144
16.6	vertical-align 属性	146
第 17 章	内置元素的样式	149
17.1	按钮	149
17.2	列表	150
17.3	表格	153
17.4	表单	156

第 1 章 了解 CSS

层叠样式表(Cascading Style Sheets, CSS)是一种描述性语言,主要用来描述网页(主要是 HTML)的内容样式和页面布局。

早期的网页内容比较少,布局和视觉效果相对简单,HTML 中曾经包含一些可以用来定义内容样式的标签。但随着用户的要求越来越复杂,这些标签已经无法实现更加丰富的视觉效果。万维网联盟(World Wide Web Consortium, W3C)于 1996 年发布了 CSS 标准的第一个版本,让用户能够更方便地描述网页的内容样式和页面布局。

本章介绍 CSS 的作用和工作机制,以及主流浏览器和兼容性问题。

1.1 CSS 的作用

HTML 通过嵌套的标签、标签属性和标签内的文本描述网页的结构以及文本、超链接、图像等内容,CSS 则用来描述这些内容的呈现样式以及呈现的位置和尺寸等信息。

(1) 描述内容样式。包括文本内容的字体、字号和颜色,元素的尺寸等信息。

(2) 描述页面布局。将内容以列表、网格或表格的形式呈现出来,或者对元素进行排列、调整元素的位置等。

(3) 丰富视觉效果。为元素添加边框、阴影和动画等效果。

1.2 工作机制

下面通过一幅图了解 CSS 是如何影响网页布局及内容呈现的,如图 1-1 所示。

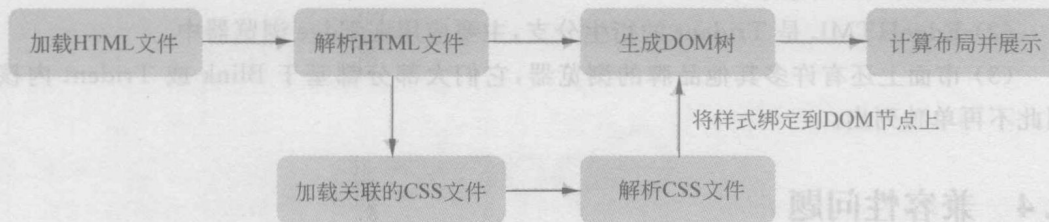


图 1-1 CSS 的工作机制

(1) 当用户在浏览器地址栏中输入 URL^① 并按 Enter 键或单击超链接后,浏览器会加载 URL 指定的 HTML 文件。

① URL: 统一资源定位符(Uniform Resource Locator),通常称为网址。

(2) 在 HTML 文件加载完成后,浏览器会解析该文件并生成 DOM^①树,然后继续加载该文件关联的 CSS、JS^②和图像等文件。

(3) 浏览器在加载完 CSS 文件后,会解析其中的样式规则,并将其绑定到关联的 DOM 节点上。

(4) 浏览器根据 DOM 树计算页面布局并将内容展示出来。

1.3 网页浏览器

网页浏览器(Web Browser)简称浏览器,是一种用来显示网页的应用程序。浏览器负责从网络上获取和展示文本、图像、超链接和音/视频等内容,处理用户操作(鼠标单击、键盘敲击和触摸屏操作等)并给予相应的反馈。

现代浏览器主要包括排版引擎和 JS 引擎。排版引擎负责解析网页结构和样式,并将网页的布局和内容渲染成图像输出到屏幕等显示设备中;JS 引擎则负责解析和执行 JS 代码。

目前主流的浏览器(正式发布的稳定版本)及其排版引擎和 JS 引擎的数据(截至 2018 年 3 月)见表 1-1。

表 1-1 主流浏览器及其排版引擎和 JS 引擎

	Google Chrome	Mozilla Firefox	Internet Explorer	Microsoft Edge	Safari
版本	65	59	11	41	11
渲染引擎	Blink	Gecko	Trident	EdgeHTML	WebKit
JS 引擎	V8	SpiderMonkey	Chakra	Chakra	JavaScriptCore

(1) Blink 是 WebKit 的衍生分支。由于技术架构、发展方向和更新节奏与 WebKit 不同,Chrome 团队基于 WebKit 开发了自己的独立分支——Blink^③,并应用在 Chrome 28 浏览器及其以后的版本中。

(2) EdgeHTML 是 Trident 的衍生分支,主要应用在 Edge 浏览器中。

(3) 市面上还有许多其他品牌的浏览器,它们大部分都基于 Blink 或 Trident 内核,因此不再单独列出。

1.4 兼容性问题

虽然现代浏览器的大部分功能都是基于 Web 标准实现的,但由于浏览器品牌的多样性以及用户设备上安装的浏览器的版本有所不同,导致不同用户的浏览器在功能上有一

① DOM: 文档对象模型(Document Object Model)。浏览器将 HTML 文件解析成由节点(Node)组成的树结构(DOM 树),根据该结构渲染网页,并提供可通过 JavaScript 访问和控制 DOM 树的接口。

② JS: JavaScript 的简称,一种广泛应用在浏览器中的编程语言。

③ Blink: 访问网址为 <https://blog.chromium.org/2013/04/blink-rendering-engine-for-chromium.html>。

些差异,这就是网页所面临的兼容性问题。一些有创意的“打新标签”使用者,兼容性问题是让前端工程师最头疼的问题之一。本书总结了一些应对方法,供读者参考。

- (1) 使用现代浏览器,并及时更新至最新版本,这样既能够享受最新的功能,也能够保证良好的网页浏览体验及安全。
 - (2) 只针对现代浏览器最新的两个版本编写网页和 Web 应用(如 Chrome 64/65 或 IE 10/11 浏览器等),督促用户及时更新浏览器。
- 虽然上述方法有些理想化,但仍然建议读者照做。在实际操作中,读者可以参考以下方案处理兼容性问题:

- (1) 针对现代浏览器最新的两个版本编写网页和 Web 应用。
 - (2) 保证大部分功能可以运行在现代浏览器最近一年内发布的不同版本中。
 - (3) 对于部分浏览器无法支持的功能,采用兼容的方式处理(如将无法播放的动画替换为静态图像),保证网页的其他内容和布局能够正常呈现,主要功能仍然可用。
 - (4) 提醒使用老旧版本浏览器的用户使用现代浏览器,并更新至最新的版本。
- 本书的内容将面向现代浏览器最新的两个版本为读者提供标准化且被浏览器广泛支持的 CSS 知识。对于尚未标准化但被浏览器广泛支持,或可能存在兼容性问题,本书将会特别标记出来,以供读者参考。

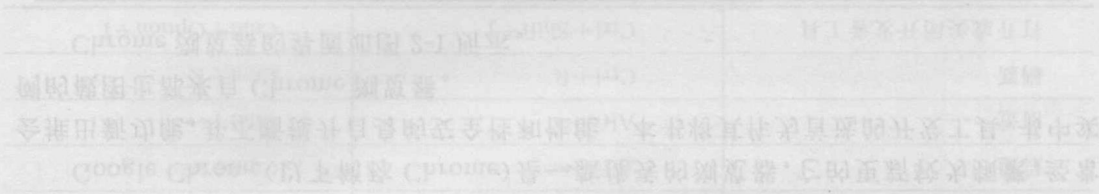


图 1-3-1 图 1-3-2

图 1-3-1 图 1-3-2

第2章 开发工具

在开始编写 CSS 代码之前,读者将从本章了解 CSS 的运行环境——浏览器,和浏览器中的开发者工具,以及用来编写代码的文本编辑器和其他辅助开发的工具与文档资料等内容。

2.1 浏览器和开发者工具

第1章介绍了几种主流的浏览器以及不同浏览器之间的兼容性问题。在编写 CSS 代码时,读者需要随时通过浏览器查看网页的效果,并通过浏览器中的开发者工具发现问题中的问题。

2.1.1 Google Chrome 浏览器

Google Chrome(以下简称 Chrome)是一款优秀的浏览器,它的更新较为频繁,经常会推出新功能,并不断提升自身的安全性和性能。本书将其作为首选的开发工具,书中实例的截图也都来自 Chrome 浏览器。

Chrome 浏览器的界面如图 2-1 所示。

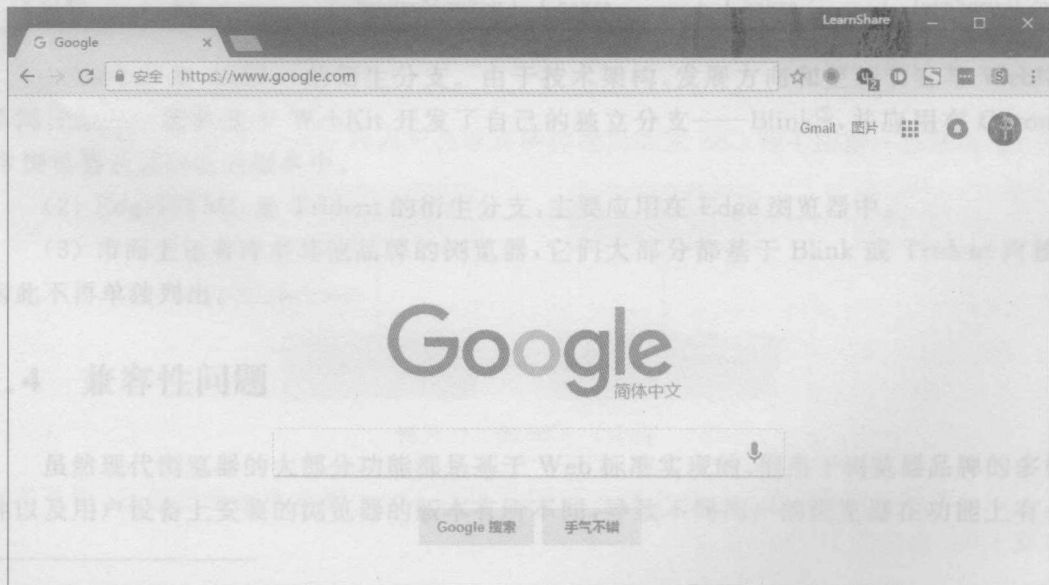


图 2-1 Chrome 浏览器

Chrome 浏览器的界面包括以下几部分。

(1) 标题栏。左侧为已打开的标签,标签右侧是“打开新标签”按钮。

(2) 工具栏。左侧分别为“返回”“前进”和“刷新”按钮,中间为地址栏和搜索框,右侧为“菜单”按钮和已安装的扩展程序。

(3) 工具栏下方是浏览器的内容展示区域,用于呈现网页。

和其他桌面软件一样,Chrome 浏览器也支持快捷键操作。使用快捷键可以有效提高浏览器的操作效率。Chrome 浏览器中常用的快捷键及功能见表 2-1。

表 2-1 Chrome 浏览器中常用的快捷键

功 能	快 捷 键	
	Windows/Linux	MacOS
打开标签	Ctrl+T	Cmd+T
重新打开刚刚关闭的标签	Ctrl+Shift+T	Cmd+Shift+T
关闭当前标签	Ctrl+W	Cmd+W
关闭所有标签	Ctrl+Shift+W	Cmd+Shift+W
将光标定位到地址栏	Ctrl+L	Cmd+L
将当前页面添加为书签	Ctrl+D	Cmd+D
在当前标签打开搜索功能	Ctrl+F	Cmd+F
后退	Alt+←	Cmd+←
前进	Alt+→	Cmd+→
刷新	Ctrl+R	Cmd+R
打开或关闭开发者工具	Ctrl+Shift+J	Cmd+Option+I

更多快捷键可以参考 Chrome 浏览器提供的帮助文档^①。

2.1.2 开发者工具

开发者工具是 Chrome 浏览器内置的开发和调试工具,可以用来对网页进行调试和分析。开发者工具的主要界面和功能如图 2-2 所示。

(1) 第一行为工具栏,从左到右分别为元素选择、设备模拟、功能切换标签、错误数量、菜单和关闭。

(2) 功能切换标签用于切换开发者工具的主要功能,其中,Elements(元素标签)是编写 CSS 时最常用的功能。

(3) 工具栏下方是功能区域,用于呈现当前选中标签对应的功能。

元素标签分为以下四个功能区。

(1) HTML 结构:用于呈现网页的 HTML 结构。

(2) CSS 规则:用于呈现与当前选中元素匹配的样式,可以在这里添加或修改 CSS

① 帮助文档:访问网址为 <https://support.google.com/chrome/answer/157179>。

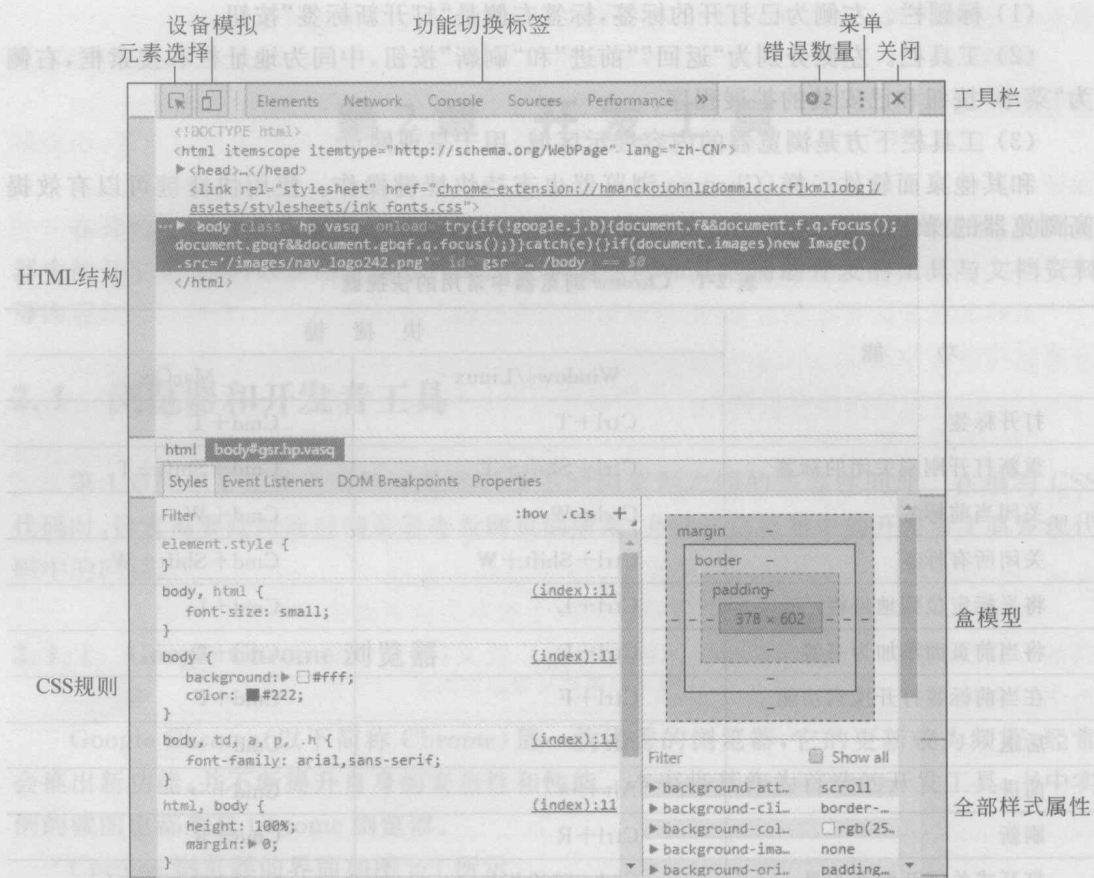


图 2-2 Chrome 浏览器中的开发者工具

属性。

(3) 盒模型：用于呈现当前选中元素的盒模型及相关数值。

(4) 全部样式属性：用于展示当前选中元素关联的所有 CSS 属性。

读者可以通过阅读谷歌开发者网站提供的文档^①了解更多关于开发者工具的知识。

提醒：Chrome 浏览器的下载地址为 <https://www.google.com/chrome/>。

如果由于网络原因无法下载 Chrome 浏览器时，则可以下载 Vivaldi 浏览器^②作为替代。Vivaldi 浏览器的主要功能及开发者工具都和 Chrome 浏览器类似，同样适用于本书内容的学习。

注意：请读者务必从官方网站下载软件，并确保下载的文件未经第三方修改，以保证系统和数据的安全。

^① 文档：访问网址为 <https://developers.google.com/web/tools/chrome-devtools/>。

^② Vivaldi 浏览器：访问网址为 <https://vivaldi.com/>。

2.2 代码编辑工具

除了浏览器及开发者工具以外,读者还需要一款编写 HTML、CSS 和 JS 代码的文本编辑器。一款合适的代码编辑工具的主要功能应该包括纯文本编辑、多种编程语言支持、代码高亮、智能提示等。

选择代码编辑工具时需要注意以下两点。

(1) Microsoft Word 等富文本编辑工具并不合适,它所保存的文件为私有的二进制数据,并非纯文本的文件。

(2) 记事本虽然是纯文本编辑工具,但其功能过于简单,也不建议用它编写程序源代码。

2.2.1 Visual Studio Code

Visual Studio Code(以下简称 Code)是由微软公司推出的一款新一代开源代码编辑工具。Code 是基于 Web 技术(Electron^①)实现的,它提供了多种语言支持、代码高亮、智能提示、编译和调试等开发者常用的功能,并提供了丰富的配置选项及众多第三方插件和主题。

Code 是一款深受开发者喜爱和支持的代码编辑工具,所以本书将其作为首选的代码编辑工具。Code 的工作界面如图 2-3 所示。

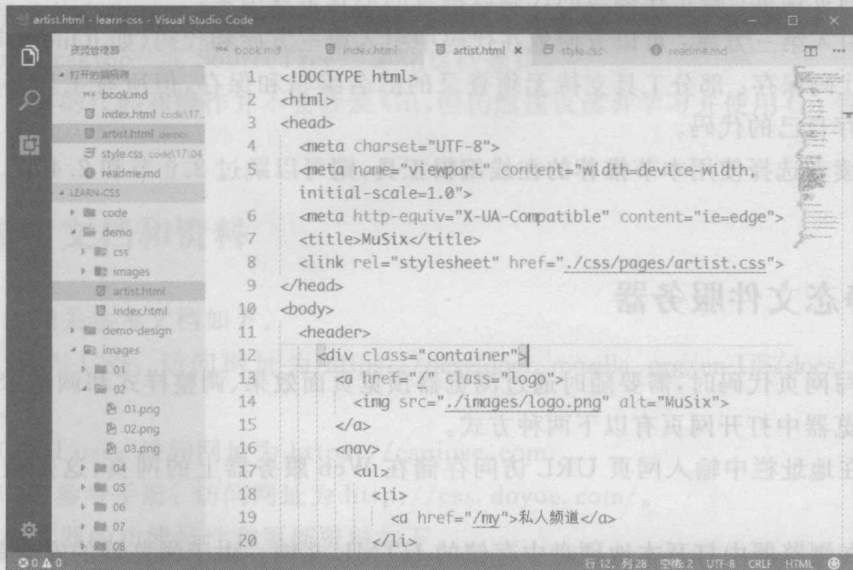


图 2-3 Visual Studio Code 的工作界面

① Electron: 访问网址为 <https://electronjs.org/>。

读者可以通过阅读 Code 网站提供的文档^①了解它的更多功能、配置和扩展方法。

当然,读者也可以根据自己的喜好选择 Sublime Text、Atom 等其他流行的代码编辑工具或 IDE(Integrated Development Environment, 集成开发环境)。但不建议使用 Adobe Dreamweaver,因为它相对封闭,缺少第三方库的支持,功能迭代速度较慢,也很少有前端工程师在工作中使用它。

2.2.2 在线编辑和预览工具

如果读者不方便在计算机上搭建开发环境和保存代码,则可以选择功能丰富的在线编辑和预览工具,可以借助这些工具快速开始网页的编写和预览,也可以长期保存自己的代码。

下面列举几种比较流行的在线编辑和预览工具。

(1) CodePen: 访问网址为 <https://codepen.io/>。

(2) JS Bin: 访问网址为 <https://jsbin.com/>。

(3) JSFiddle: 访问网址为 <https://jsfiddle.net/>。

(4) Plunker: 访问网址为 <https://plnkr.co/>。

(5) RunJS: 访问网址为 <http://runjs.cn/>。

上述在线工具能够提供的功能如下。

(1) 在线代码编辑: 通常支持 HTML、CSS 和 JS 等前端相关的代码。

(2) 网页预览: 将正在编写的代码以网页的形式呈现出来。

(3) 引入第三方库: 可以方便地在代码中引入第三方库和代码(如 jQuery 等)。

(4) 代码保存: 部分工具支持无须登录的匿名编辑和保存,但所有工具都支持注册账户并保存自己的代码。

如果读者选择使用本节推荐的在线编辑工具,则可以跳过 2.3 节和 2.4 节,直接阅读 2.5 节。

2.3 静态文件服务器

在编写网页代码时,需要随时通过浏览器预览页面效果、调整样式和调试 JS 代码。

在浏览器中打开网页有以下两种方式。

(1) 在地址栏中输入网页 URL 访问存储在 Web 服务器上的网页,这是最常用的访问方式。

(2) 在浏览器中打开本地硬盘中存储的 HTML 文档。由于浏览器的安全策略,部分功能可能无法正常使用。

本书建议读者始终通过 Web 服务器预览和调试正在编写的网页。

不过在网页对外发布之前,读者并不需要考虑如何注册域名、购买服务器及安装配置

^① 文档: 访问网址为 <https://code.visualstudio.com/docs>。