



国家精品教材
应用型本科计算机类专业系列教材
应用型高校计算机学科建设专家委员会组织编写

Web前端开发技术

第2版

王媛媛 张海艳 章 慧 主编



南京大学出版社

南京大学出版社
版权所有

内容简介

本教材以培养具备扎实 Web 前端开发技能、良好职业素养和创新能力的应用型技术人才为目标,系统讲解 HTML、CSS、JavaScript、Vue.js 等前端核心技术,涵盖网页结构、样式设计、交互编程、响应式布局及前端框架应用等内容。教材贯穿产教融合理念,融入课程思政元素,注重培养学生的职业道德、创新意识与责任感,实现专业能力与综合素质的协同提升。

全书内容分为三个递进层次:基础构建层、交互增强层和工程应用层,以高校教学信息平台项目为主线,实现理论与实践、教学与产业的深度融合。本教材可作为高等院校计算机类及相关专业的 Web 前端开发课程教学,也可作为前端开发初学者和自学者的实用参考书。全书结构清晰、由浅入深,每章均设置实例演练、本章小结与拓展训练,注重理论与实践结合,帮助学生在动手操作中掌握知识、提升技能。通过从静态页面构建到动态交互实现,再到完整前端管理系统设计的全过程训练,学生能够系统掌握现代 Web 前端开发的技术体系,具备独立完成企业级项目开发的能力,为后续专业发展和职业成长奠定坚实基础。

图书在版编目(CIP)数据

Web 前端开发技术 / 王媛媛, 张海艳, 章慧主编.
2 版. — 南京: 南京大学出版社, 2025. 11. — (应用型本科计算机类专业系列教材). — ISBN 978-7-305-29797-7

I. TP393.092.2

中国国家版本馆 CIP 数据核字第 2025BY0269 号

出版发行 南京大学出版社

社 址 南京市汉口路 22 号 邮 编 210093

丛 书 名 应用型本科计算机类专业系列教材

书 名 Web 前端开发技术

Web QIANDUAN KAIFA JISHU

主 编 王媛媛 张海艳 章 慧

责任编辑 杨蒙蒙 编辑热线 025-83592655

照 排 南京开卷文化传媒有限公司

印 刷 南京百花彩色印刷广告制作有限责任公司

开 本 787 mm×1092 mm 1/16 印张 16.75 字数 420 千

版 次 2025 年 11 月第 2 版 2025 年 11 月第 1 次印刷

ISBN 978-7-305-29797-7

定 价 49.80 元

网 址: <http://www.njupco.com>

官方微博: <http://weibo.com/njupco>

官方微信号: njupress

销售咨询热线: (025)83594756

* 版权所有, 侵权必究

* 凡购买南大版图书, 如有印装质量问题, 请与所购
图书销售部门联系调换

南京大学出版社
版权所有

前 言

随着互联网技术的飞速发展,Web 前端开发已成为信息技术领域的重要组成部分,广泛应用于电子商务、教育、政务、金融、社交平台等各行各业。前端技术不断演进,从最初的静态网页到如今的动态交互、组件化开发与智能化辅助设计,对开发人员的技术能力、工程思维和综合素质提出了更高要求。为适应新时代信息技术人才培养的需求,我们结合当前高校教学实际与产业技术发展趋势,对第 1 版进行了修订,编写了《Web 前端开发技术(第 2 版)》这本兼具基础性、实用性与前瞻性的教材,旨在帮助学生系统掌握 Web 前端开发的核心知识与实践技能,提升解决复杂工程问题的能力。

《Web 前端开发技术(第 2 版)》教材是由淮阴工学院与江苏卓易信息科技股份有限公司合作编撰的课程教材,本书作者系高校一线教师和企业资深工程师,具有多年项目开发经验和授课经验。本教材由王媛媛、张海艳、章慧担任主编,并负责全书的整体策划与统稿;由陈晓兵、李翔、詹奇、刘杰、高燕、徐义晗、王红华等担任副主编,编写成员有经玮乐、高古学、尤洪峰、朱永青、张文巧等。本教材由谢乾担任顾问并主审。

本教材以培养具备扎实专业能力、良好职业素养和创新精神的应用型人才为目标,内容涵盖 HTML、CSS、JavaScript 基础技术,深入讲解网页结构构建、样式控制、交互逻辑实现等关键环节,并系统介绍 CSS+DIV 布局、盒子模型、浮动与定位等核心布局技术。在此基础上,教材进一步引入 Vue.js 前端框架、Element Plus 组件库、Axios 通信、Vue 路由与状态管理等现代前端开发核心技术,强化工程化开发理念。通过综合项目设计,学生可完成从页面布局到管理系统前端开发的全流程实践,提升项目实战能力。同时,教材注重产教融合,融入课程思政元素,强调职业道德、社会责任与创新意识的培养,体现技术前沿性,助力学生适应智能化开发新环境。

本教材共三部,分 8 章,第一部分(1~4 章)聚焦前端基础,系统讲解 HTML 文档结构、文本与多媒体标签、列表与表格,以及 CSS 样式设计与选择器应用,深入解析盒子模型、浮动

与定位等布局技术,夯实静态页面开发能力;第二部分(5~6章)强化动态交互,涵盖 JavaScript 编程基础与事件处理,引入 Vue.js、Element Plus、Axios 与 Vue Router,完成工程环境搭建与组件化开发;第三部分(7~8章)面向综合应用,通过组件整合、ECharts 数据可视化及 Trae 等 AI 辅助开发技术,实现从页面开发到系统集成的全流程实践,全面提升真实项目开发能力。全书结构清晰、循序渐进,理论讲解与实例演练相结合,每章均配有小结与拓展训练,便于教师教学与学生自学。书中案例贴近实际应用场景,注重动手能力培养,既适合高校计算机类及相关专业的本科与高职学生使用,也可作为前端开发初学者和从业者的参考用书。

本教材的编写得到了淮阴工学院计算机与软件工程学院领导和老师们给予的大力支持和帮助,以及江苏省计算机学会应用型高校计算机学科建设专家委员会同行的关心,在此一并致谢。同时,感谢南京大学出版社为本书的出版提供的帮助、指导意见和参考资料,以及感谢对本书出版给予大力支持与帮助的各位同仁、院校和相关企业单位。

我们衷心希望这本教材能够成为学习 Web 前端开发技术的得力助手,为教师的教学工作提供有益参考。也期待广大师生在使用过程中提出宝贵意见和建议,以便我们不断完善教材内容,更好地服务于新时代高素质软件技术人才的培养。反馈意见请告知编者(zhfwyy@hyit.edu.cn)。

编 者

2025 年 11 月

南京大学出版社
版权所有

目 录

第一章 Web 前端开发技术概述

.....	001
1.1 Web 前端开发技术基础	002
1.2 前端开发框架	004
1.3 思政融入	006
1.4 本章小结	006
1.5 拓展训练	007

第二章 HTML 基础

2.1 HTML 文件编写方法	008
2.2 HTML 基本语法	010
2.3 HTML 基本文档结构	012
2.4 HTML 文档类型	014
2.5 常用文本格式化标签	015
2.6 图像与多媒体	018
2.7 列表	021
2.8 HTML 表格标签	025
2.9 实例演练	029
2.10 思政融入	031
2.11 本章小结	031
2.12 拓展训练	032

第三章 CSS 基础

3.1 CSS 概述	036
3.2 CSS 的构成	036
3.3 CSS 样式表分类	037
3.4 CSS 基本选择器	039
3.5 CSS 派生选择器	044
3.6 CSS 单位	047
3.7 CSS 文字与文本属性	048
3.8 CSS 列表属性	053
3.9 CSS 表格属性	055
3.10 实例演练	057
3.11 思政融入	059
3.12 本章小结	059
3.13 拓展训练	060

第四章 CSS+DIV 布局

4.1 盒子模型	063
4.2 标准文档流	070
4.3 position 定位方式	071
4.4 float 浮动方式	076
4.5 CSS+DIV 布局理念	091

4.6 两列布局	092	6.8 本章小结	183
4.7 三列布局	099	6.9 拓展训练	184
4.8 实例演练	103		
4.9 思政融入	108	第七章 Web 前端页面综合设计	
4.10 本章小结	108		186
4.11 拓展训练	109	7.1 项目需求描述	187
第五章 JavaScript 编程	111	7.2 快捷链接和顶部导航组件	187
5.1 JavaScript 简介	112	7.3 轮播图组件	193
5.2 JavaScript 基础	114	7.4 内容区域组件	197
5.3 JavaScript 事件分析	130	7.5 公告组件	200
5.4 实例演练	132	7.6 底部组件	205
5.5 思政融入	136	7.7 实例演练	208
5.6 本章小结	137	7.8 思政融入	210
5.7 拓展训练	137	7.9 本章小结	211
		7.10 拓展训练	212
第六章 Web 前端开发环境配置		第八章 管理系统前端页面综合设计	
.....	139		214
6.1 配置项目开发环境	140	8.1 项目组件创建	215
6.2 项目创建	142	8.2 路由配置	247
6.3 Vue.js 基础	147	8.3 ECharts 可视化开发	249
6.4 Vue.js 进阶	161	8.4 结合 AI 插件的前端开发	257
6.5 Element Plus 组件	179	8.5 思政融入	260
6.6 实例演练	180	8.6 本章小节	260
6.7 思政融入	183	8.7 拓展训练	261

第一章

Web 前端开发技术概述

Web 前端开发技术是指用于构建网站或 Web 应用程序中直接与用户交互的部分的技术集合,这些技术主要涉及网页的设计、布局、内容组织以及用户界面的开发等多个方面,为用户提供直观、易用的使用体验。随着互联网技术的快速发展,现代 Web 前端已从最初的静态页面演变为高度交互、跨平台、响应式的复杂应用系统。



教学目标

【知识目标】

- 掌握 HTML、CSS、JavaScript 三大核心技术的基本功能及其协同工作机制,理解前端开发的技术构成与核心地位。
- 了解主流前端框架(如 Vue.js)和 UI 组件库(如 Element Plus)的应用场景,熟悉现代前端技术生态与发展脉络。

【能力目标】

- 能够描述前端开发流程,识别关键技术角色,并具备根据项目需求进行初步技术选型的能力。
- 能结合实际案例分析前端技术在页面结构、样式表现与交互实现中的综合应用,为后续深入学习奠定实践基础。

【素养目标】

- 养成规范编码、语义化结构、可维护性强的工程习惯,具备持续学习与技术追踪的自主能力。
- 增强团队协作意识与用户中心理念,理解前端开发在多角色项目中的责任与价值。

【思政元素】

- 科技报国与创新自信:通过国产技术发展案例,增强学生对我国前端技术自主创新的认同感与使命感。
- 工匠精神与职业素养:强调代码质量与工程规范,培育精益求精、严谨负责的职业态度。

1.1 Web 前端开发技术基础

Web 前端开发的核心技术主要包括用于描述网页结构和内容的 HTML(Hyper Text Markup Language,超文本标记语言)、描述网页样式和布局的 CSS(Cascading Style Sheet,层叠样式表)以及实现网页交互功能和动态效果的 JavaScript 脚本语言。

1.1.1 HTML 简介

HTML 是一种用于创建网页的标准标记语言。HTML 构建一套标记符号和语法规则,将所要显示的文字、图像和声音等要素按照一定的标准要求排放,形成一定的标题、段落和列表等单元,通过浏览器的解析,将网页中所要呈现的内容、排版展现给用户。

HTML 的特点主要体现在以下几个方面:

(1) 简易性

HTML 的语法相对简单,简洁的标签结构使得 HTML 易于学习和使用。

(2) 平台无关性

得益于 HTML 的标准化和广泛支持,HTML 编写的网页可以在不同的操作系统和浏览器中运行。

(3) 可扩展性

HTML 具有良好的扩展性。HTML 采取子类元素的方式,为系统扩展带来保证,这使得 HTML 能够满足不断变化的 Web 开发需求。

(4) 交互性

虽然 HTML 本身主要用于描述网页结构,但结合 JavaScript 和 CSS 等技术,可以实现丰富的交互效果。

(5) 超链接功能

HTML 的“超文本”特性允许在网页中嵌入链接(Hyperlink),这些链接可以指向其他网页、图像、文件或电子邮件地址等。

(6) 多媒体支持

HTML 支持嵌入多种媒体类型,如图像、音频和视频。这些特点使得 HTML 成为 Web 前端开发的基础语言之一,为开发者提供了强大的工具来创建和展示网页内容。

1.1.2 CSS 简介

CSS 是一种格式化网页的标准方式,用于控制网页样式,并允许样式信息与网页内容分离的一种技术。开发者采用 CSS 设计网页,可以精确地控制网页元素的样式,例如页面布局、字体、颜色、背景等效果,从而设计出美观、易用的 Web 界面。

CSS 的特点主要体现在以下几个方面:

(1) 层叠性

CSS 允许对同一个元素多次设置样式,并使用最后一次设置的属性值。这种层叠的方式可以灵活地加入自己的设计。

(2) 选择器

CSS 选择器用于选择需要应用样式的 HTML 元素。通过选择器,开发者可以精确地定

位到网页中的特定元素,并对其应用样式。

(3) 属性和属性值

CSS 样式由属性和属性值组成。属性定义了要改变的样式方面(如颜色、字体大小等),而属性值则指定了这些方面的具体设置。

CSS 实现了简洁并语义化的网页布局,使得网站维护更加简便。网页的结构与样式是相互分离的,人们完全可以改变一个基于 CSS 设计的网站的视觉效果而不去改动网站的内容,开发者可以更好地控制网页的外观和布局。CSS 是 Web 开发中不可或缺的一部分,使得开发者能够创建具有丰富视觉效果和用户体验的网页。

1.1.3 JavaScript 简介

JavaScript 是面向 Web 的编程语言,是目前前端开发的核心技术之一。JavaScript 使得网页不再是静态的展示内容,而是根据用户行为和外部条件做出动态改变及响应。

JavaScript 的特点主要体现在以下几个方面:

(1) 客户端脚本语言

JavaScript 是一种在客户端(通常是浏览器)执行的脚本语言,用于增强网页的交互性和动态性,可以动态地改变网页的内容、样式和结构,从而提供丰富的用户体验。

(2) 弱类型语言

JavaScript 是一种弱类型语言,变量的类型在运行时可以动态改变,不需要显式声明变量的类型。

(3) 面向对象

JavaScript 支持面向对象编程(Object-Oriented Programming, OOP),具有类、对象、继承等常见的面向对象特性,同时支持函数式编程和事件驱动编程等多种编程范式,使得开发者可以根据实际需求选择最适合的编程方式。

(4) 事件驱动

JavaScript 通过处理用户交互事件(如点击、鼠标移动等)来触发特定的操作,实现与用户的互动,提供动态和交互式的用户体验。

(5) 跨平台性

JavaScript 可以在多个平台上运行,包括各种操作系统和设备,如浏览器、服务器(通过 Node.js 等技术)、移动应用等。

(6) 动态性

JavaScript 允许在运行时修改和更新代码、添加新的功能或改变网页的外观和行为,从而提供更加灵活和智能的用户体验。

(7) 可扩展性

通过使用 JavaScript 库和框架,开发人员可以扩展 JavaScript 的功能和能力,简化开发过程,使得开发者能够更加高效地构建复杂的 Web 应用程序。

JavaScript 可以动态地改变网页的内容、样式和结构,实现页面元素的动态显示和隐藏,以及响应用户的交互行为。

1.2 前端开发框架

在前端开发框架出现之前,开发人员通常需要手动编写大量的 JavaScript 代码构建各种页面元素和交互功能。但因缺乏组织,代码臃肿混乱,同时耗时费力,容易出错,因此催生了前端开发框架的诞生。前端框架提供了一系列预定义的组件和 UI 元素,将这些组件组合在一起可以快速构建出复杂的 Web 界面。前端开发框架减少了手动编码的工作量,提高了开发效率,使开发者能够更高效地创建用户界面。目前主流前端开发框架有 Vue.js、React 和 Angular.js 等,如图 1-1 所示。

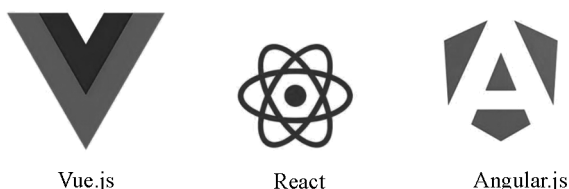


图 1-1 主流 Web 前端开发框架

Vue.js 简称 Vue, 基于标准 HTML、CSS 和 JavaScript 构建, 是目前使用最广泛的 JavaScript 框架之一, 是一款友好的、多用途且高性能的构建用户界面的渐进式框架。Vue 允许将网页分割成可复用的组件, 组件化开发使得开发者能够将复杂的用户界面拆分成多个小的、独立的、可复用的部分。

React 是一个功能强大、灵活高效用于构建用户界面的 JavaScript 库, 由 Meta 开发。React 框架采用声明式设计、组件化思想以及虚拟 DOM 技术, 为开发者提供了简洁易用的 API 和丰富的生态系统。

Angular 是谷歌开发的一个开源的、功能全面的、强大的基于 TypeScript 的前端框架, Angular 适用于构建大型、复杂的 Web 应用, 特别是交互式 Web 应用程序、电子商务网站、数据驱动的仪表盘和渐进式 Web 应用程序(PWA)。

1.2.1 Vue.js 框架简介

Vue.js 凭借其简洁性、高效性、组件化开发、双向数据绑定、丰富的生态系统、良好的文档和社区支持等特性, 成为许多开发者的首选前端框架。

Vue.js 的特点主要体现在以下几个方面:

(1) 轻量级框架

Vue.js 核心库关注视图层, 是一个构建数据的视图集合, 大小只有几十 KB。Vue.js 通过简洁的 API 提供高效的数据绑定和灵活的组件系统。

(2) 良好的文档和社区支持

Vue.js 的官方文档提供了大量的示例和教程, 能帮助开发者快速上手。

(3) 数据驱动视图

Vue.js 使用双向数据绑定和单向数据流确保数据和视图之间的同步。当数据变化时, 视图会自动更新。

(4) 组件化

Vue.js 的组件化可以把页面拆分成多个组件,组件是资源独立的,在系统内部可复用,组件和组件之间可以嵌套。组件化使数据的更改更为简单,不需要进行逻辑代码的修改,只需要操作数据就能完成相关操作。

(5) 生态系统丰富

Vue.js 拥有丰富的生态系统,提供了许多第三方库和工具,包括 Vue Router(用于构建单页面应用)、Vuex(用于状态管理)、Vue CLI(用于快速搭建项目)等。

(6) 过渡和动画

Vue.js 内置了过渡和动画的支持,使开发者可以轻松地实现元素的淡入淡出、滑动等动画效果,开发者可以方便地控制动画的进入、离开和列表的过渡。

(7) TypeScript 支持

Vue.js 提供了对 TypeScript 的支持,使开发者可以使用静态类型检查来提高代码的质量和可维护性。

(8) 虚拟 DOM

Vue.js 使用虚拟 DOM 提高渲染性能。虚拟 DOM(Virtual DOM)是一种编程概念,使用 JavaScript 对象来表示真实 DOM(Document Object Model,文档对象模型)的结构和状态。

1.2.2 Element UI

Element UI 是一个为开发者、设计师和产品经理准备的基于 Vue 2.0 的桌面端组件库。它提供了一套丰富的 UI 组件,用于快速构建网站界面,提高开发效率。Element UI 遵循 Material Design 和 Ant Design 的设计规范,为用户提供了统一、美观的交互体验。Vue.js 与 Element UI 协同使用是一种高效的前端开发模式,特别适用于构建 Web 应用,可以显著提高开发效率和用户体验。

Vue.js 与 Element UI 的关系如下:

(1) 相互关联

Vue.js 和 Element UI 是相互关联的。Element UI 基于 Vue.js 开发,因此它依赖于 Vue.js。在使用 Element UI 之前,需要先引入 Vue.js。

(2) 配合使用

Vue.js 和 Element UI 可以一起使用,使开发者能够更高效地构建用户界面。Vue.js 提供了构建用户界面的工具和机制,如组件、响应式数据绑定等;而 Element UI 提供了丰富的 UI 组件和样式,简化了开发者的 UI 开发过程。

(3) 灵活性

虽然 Element UI 是一个基于 Vue.js 的 UI 组件库,但 Vue.js 并不限于使用 Element UI。开发者也可以选择其他 UI 组件库或自行开发组件来满足项目的需求。

随着 Vue 3.0 的发布,Element UI 团队也推出了基于 Vue 3.0 的版本,即 Element Plus。因此,在选择使用 Element UI 时,需要根据项目的 Vue 版本进行决策。

1.3 思政融入

1. 科技报国与创新自信:增强技术自主的使命担当

当前,我国在前端技术领域取得了长足进步,从百度、阿里、腾讯等企业主导的开源项目(如 ECharts、Ant Design、TDesign),到 Vue.js(拥有庞大中文社区)等具有全球影响力的框架生态,再到 Element UI、NutUI 等国产优秀组件库在国际社区的广泛使用,中国开发者和企业在全球 Web 前端技术生态中的创新活力与贡献度不断提升,充分展现了自主技术发展的强劲势头。

2. 工匠精神与职业素养:追求卓越的技术品格

前端开发直接面向用户,页面的每一像素、每一段动画、每一次交互都影响用户体验。这要求开发者具备严谨细致、精益求精的工匠精神。需要加强 HTML 语义化标签的规范使用、CSS 命名的可读性、JavaScript 代码的可维护性,养成良好的编码习惯和工程意识。同时,尊重标准、注重细节、追求完美,把每一行代码都视为作品来打磨。

1.4 本章小结

学完本章后读者对 Web 前端开发技术能有一个总体的认识,了解主流 Web 前端开发技术及前端开发框架。

本章知识点如图 1-2 所示。

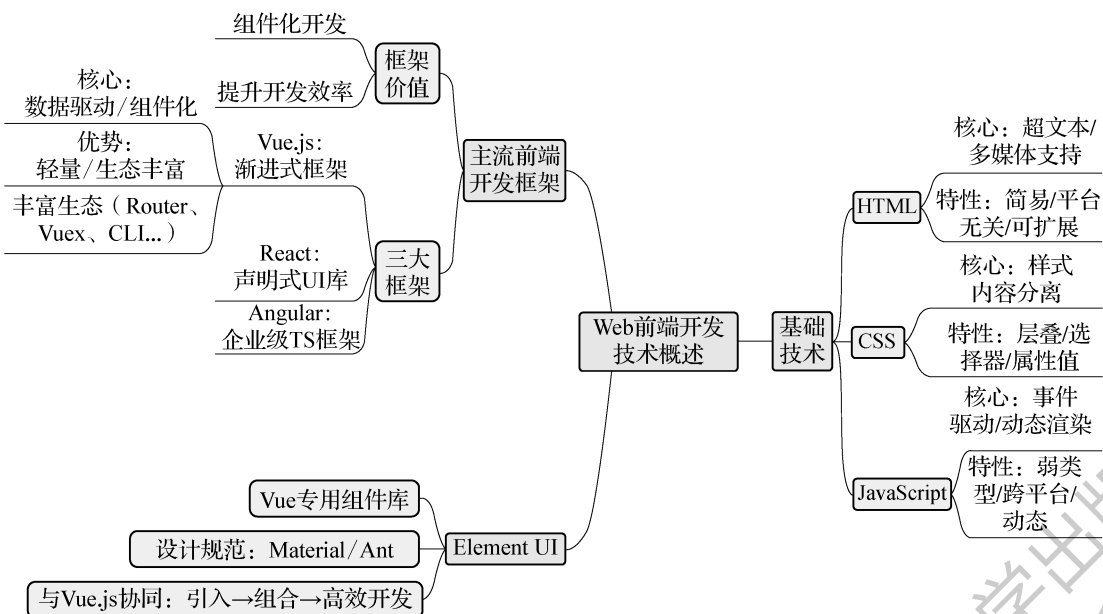


图 1-2 Web 前端概述思维导图

1.5 拓展训练

- (1) HTML 的全称是什么()
- A. Hyper Text Markup Language B. High Tech Modern Language
C. Home Tool Markup Language D. Hyperlinks and Text Markup Language
- (2) CSS 的全称是什么()
- A. Creative Style Sheets B. Computer Style Sheets
C. Cascading Style Sheets D. Colorful Style Sheets
- (3) JavaScript 是一种什么类型的语言()
- A. 编译型语言 B. 解释型语言 C. 标记语言 D. 样式语言
- (4) Vue.js 是一个什么类型的框架()
- A. 后端框架 B. 前端框架 C. 全栈框架 D. 数据库框架
- (5) Vue.js 的核心特性是什么()
- A. 双向数据绑定 B. 单向数据流 C. 静态模板 D. 服务器渲染
- (6) Element UI 是基于哪个前端框架的组件库()
- A. React B. Angular C. Vue.js D. jQuery
- (7) Element UI 主要用于什么目的()
- A. 后端 API 开发 B. 数据库设计
C. 快速构建用户界面 D. 服务器配置



【微信扫码】
本章参考答案 & 相关资源

第二章

HTML 基础

HTML(Hyper Text Markup Language)是 Web 前端开发的基石,是构建网页内容结构的核心技术。本章系统介绍 HTML 的基本语法、文档结构、常用标签及其语义化应用,涵盖文本格式化、图像多媒体嵌入、列表与表格等基础内容的编写方法。本章内容注重实践性与规范性,通过实例演练,帮助读者掌握标准 HTML 文档的编写规范,理解标签的语义化意义,为后续 CSS 样式设计和 JavaScript 交互开发奠定坚实的内容结构基础。



教学目标

【知识目标】

- 掌握 HTML 的基本语法结构、文档类型声明与标准文档结构,理解常用标签的语义及使用场景。
- 熟悉文本、图像、多媒体、列表与表格等核心标签的语法及其属性配置方法。

【能力目标】

- 能够独立编写结构清晰、语义合理的静态网页,具备组织网页内容的基本能力。
- 能根据设计需求灵活运用 HTML 标签实现图文混排、数据展示等常见页面布局。

【素养目标】

- 养成语义化编码习惯,注重代码可读性与可维护性,树立规范开发意识。
- 增强信息表达的责任感,理解网页内容组织对用户体验与信息传播的重要影响。

【思政元素】

- 规范意识与工匠精神:通过强调 HTML 标签的语义化使用和结构规范,培养学生严谨细致、追求卓越的职业态度。
- 信息责任与传播伦理:引导学生认识到网页内容结构直接影响信息传达的准确性与可及性,增强作为技术开发者的社会责任感。

2.1 HTML 文件编写方法

网页(Web Page)是信息在互联网上的基本呈现单元,是用户通过浏览器访问的内容页

面。一个完整的网站(Website)由多个相互关联的网页组成,可以说,网站是网页的集合。

使用 HTML(Hyper Text Markup Language)编写的超文本文档称为 HTML 文档。每一个 HTML 文件本质上就是一个网页文件。简单来说,一个.html文件就是一个网页。

仅使用 HTML 开发的网页主要负责内容的结构化展示,不具备复杂的交互功能(如新闻发布系统的自动更新、在线支付、用户登录等),这类网页被称为静态网页。静态网页文件的扩展名通常为.html或.htm,其中.html更为通用和推荐。

HTML 是一种纯文本标记语言,具有良好的跨平台特性,对操作系统和运行环境无特殊要求。任何文本编辑器均可用来编写 HTML 代码。

目前,常用的前端开发工具包括:

Notepad++:轻量级文本编辑器,支持语法高亮,适合初学者;

HBuilderX:国产集成开发环境,专为 Web 和移动端开发设计,具备智能提示、快速预览等功能,适合初学者;

Visual Studio Code:微软推出的免费开源代码编辑器,插件丰富、功能强大,广泛应用于企业开发。

如图 2-1 所示,这些工具均可高效支持 HTML 文件的编写与调试,开发者可根据习惯和需求选择合适的开发环境。



图 2-1 HTML 常用开发工具

Windows 操作系统自带的记事本是一款基础的文本编辑工具,虽功能简单,但可用于编写包括 HTML 在内的任何纯文本格式的代码。下面介绍如何使用记事本创建一个网页:

1. 打开记事本,输入文本(如图 2-2 所示);

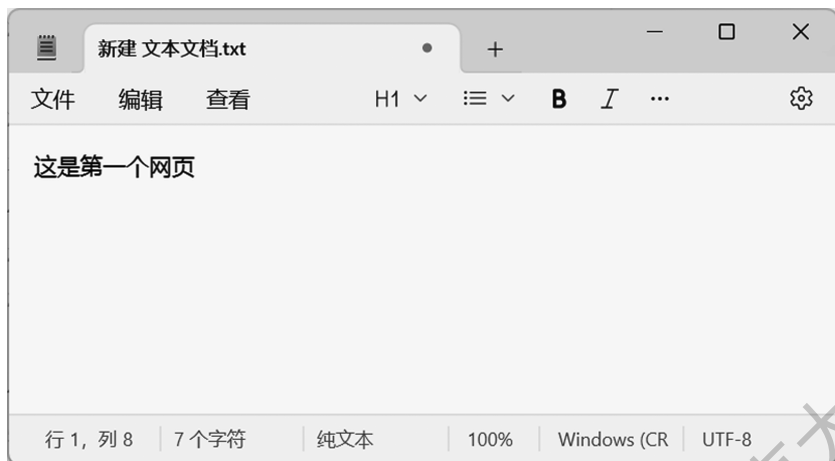


图 2-2 使用记事本制作网页

2. 点击“文件”菜单中的“另存为”，在保存对话框中设置文件名，并将“保存类型”选择为“所有文件”，文件扩展名设置为.html或.htm(例如:index.html)；
3. 保存后，双击该文件即可在默认浏览器中打开并查看网页效果(如图 2-3 所示)；

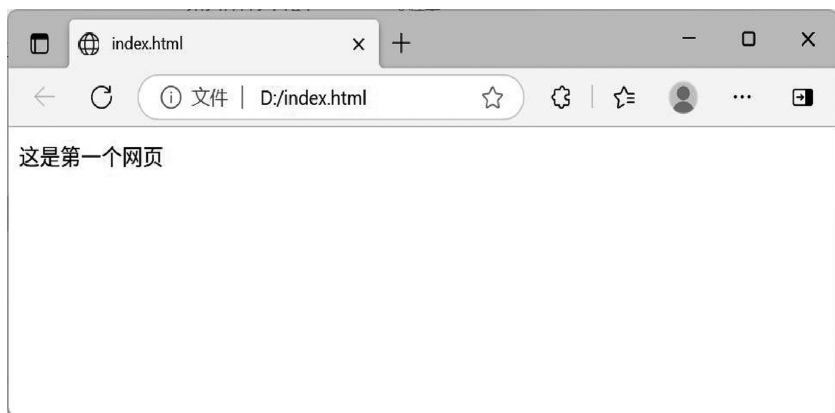


图 2-3 网页在浏览器中运行效果

4. 如需修改，右键点击 HTML 文件，选择“打开方式”->“记事本”，编辑完成后保存，刷新浏览器页面即可看到更新内容，无需重新打开浏览器。

2.2 HTML 基本语法

HTML 是一种由浏览器解析的标记语言，其核心结构由一系列具有特定含义的标签(Tag)构成。标签用于定义网页内容的结构与语义，如标题、段落、图片、链接等。通过合理使用标签，开发者可以将纯文本组织为结构清晰、语义明确的网页内容。

HTML 标签通常成对出现，包括开始标签(如<p>)和结束标签(如</p>)，中间包含具体内容。部分标签为自闭合标签(如
)，无需单独闭合。

【例 2-1】初识 HTML 标签。

使用记事本创建一个 HTML 文件，输入一段纯文本内容并保存，在浏览器中打开，显示效果如图 2-4 所示。

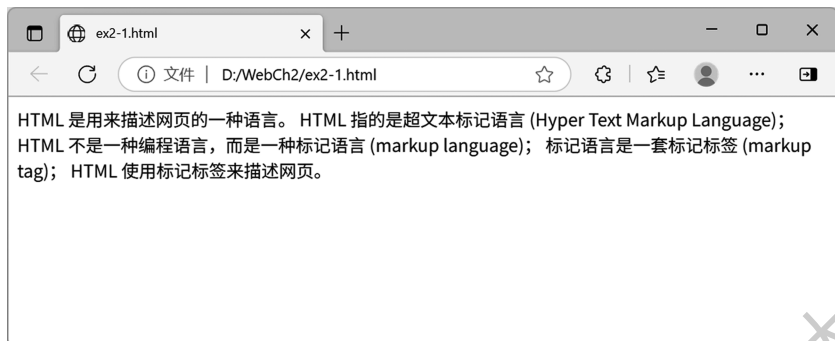


图 2-4 浏览器中的显示效果

观察发现,所有文字在浏览器中“扎堆”显示,无标题、无段落分隔,排版混乱,可读性差。接下来,为文本添加 HTML 标签使其结构化显示。

```
<html>
<head></head>
<title></title>
<body>
  <h2 style="color: red;"> HTML 是用来描述网页的一种语言。</h2>
  <p> HTML 指的是超文本标记语言 (Hyper Text Markup Language) ;</p>
  <p> HTML 不是一种编程语言,而是一种标记语言 (markup language) ;</p>
  <p> 标记语言是一套标记标签 (markup tag) ;</p>
  <p> HTML 使用标记标签来描述网页。</p>
</body>
</html>
```

保存后刷新浏览器,页面显示效果如图2-5所示:标题以红色醒目显示,段落之间有了清晰的间距,整体布局更加清晰有序。

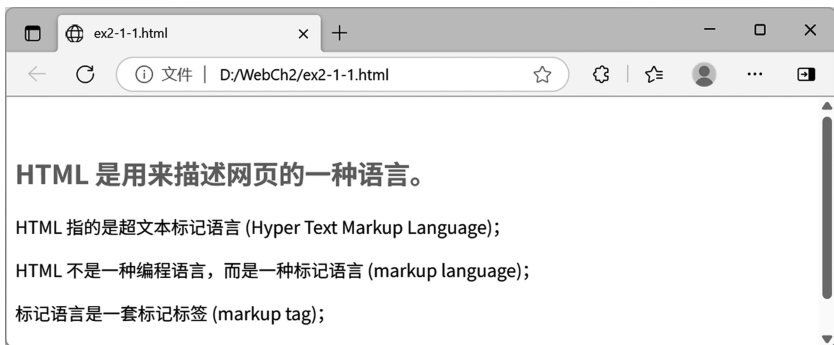


图 2-5 添加标签后页面显示效果

代码解析:

- `<h2>...</h2>`:二级标题标签,赋予文本“标题”语义。
- `style="color: red;"`:内联样式属性,用于设置字体颜色为红色。
- `<p>...</p>`:段落标签,赋予文本“段落”语义,浏览器自动为其添加上下边距。

(1) 标签

HTML 是一种通过标签来定义网页结构的标记语言,不区分大小写。标签用于描述内容的语义与功能,每个标签及其包含的内容共同构成网页中的一个元素。通过合理组织标签结构,可实现网页内容的层次化与语义化表达。

- 标签必须使用尖括号。
如`<p></p>`,`<h1></h1>`,`<br/>`。
- 标签分为双标签和单标签两种类型。双标签由“开始标签”和“结束标签”两部分构成,必须成对使用,单标签只有开始标签。
如`<p></p>`标签,其中`<p>`是段落的开始标志,`</p>`是段落的结束标志。
- 标签可以相互嵌套,但不能交叉。

如 `<html> <body> </body> </html>` (正确)

`<html> <body> </html> </body>` (错误)

(2) 标签的属性、属性值

标签的附加信息称为属性,为属性赋的值称为属性值。

- 属性的声明必须位于开始标签里。

如 `<a href="http://www.baidu.com">百度一下</a>`。

- 一个元素可以有多个属性,属性之间由空格分开。

如 ``。

- 多个属性之间不区分先后顺序。

(3) 注释语句

HTML 语言中,采用 `<!--注释信息-->` 添加注释。

【例 2-2】 给网页添加注释,代码如下所示,页面效果如图 2-6 所示。

```
1. <!DOCTYPE html>
2. <html>
3.   <body>
4.     <!--这是一段注释,注释不会在浏览器中显示。-->
5.     <p>注释不显示</p>
6.   </body>
7. </html>
```

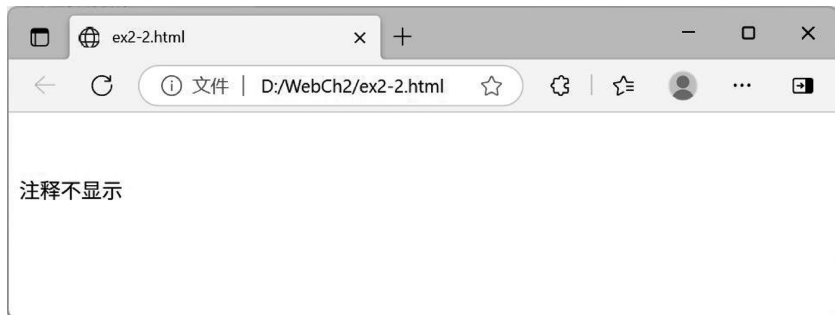


图 2-6 添加注释应用

2.3 HTML 基本文档结构

为确保网页在不同浏览器中正确解析与显示,编写 HTML 时应遵循 Web 标准,从构建规范的基本文档结构开始。

HTML 文档采用树形结构(也称文档对象模型,DOM 树),以 `<html>` 标签为根节点,整体上分为两个主要部分:

`<head>`: 头部区域,用于定义页面的元信息(如标题、字符编码、引用样式表等),不直接显示在页面内容中;在此对标签对之间可以使用 `<title> </title>`、`<meta />`、`<script> </script>`、`<style> </style>`、`<link />` 等标签,这些标签用于描述 HTML 文档相关信息。

`<body>`: 主体区域,包含网页的实际内容,如文本、图像、链接等,是用户可见的部分。

这种结构化的组织方式不仅有助于浏览器正确解析页面,也为后续的样式设计与交互开发奠定了基础。

HTML 文档基本结构如下:

```
1. <html>
2.     <head>
3.         <title></title>
4.     </head>
5.     <body>
6.     </body>
7. </html>
```

【例 2-3】 HTML 标题和主体的简单应用,页面显示效果如图 2-7 所示。

```
1. <html>
2.     <head>
3.         <title> HTML 结构示例</title>
4.     </head>
5.     <body>
6.         浏览器中显示的内容
7.     </body>
8. </html>
```

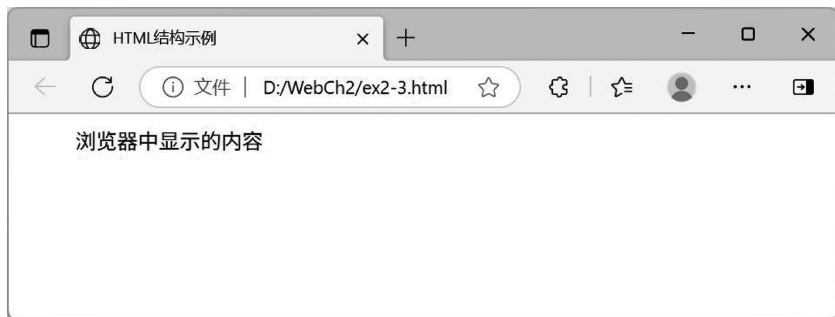


图 2-7 HTML 标题和主体的简单应用

代码中第 3 行是文档标题,显示在浏览窗口的标题栏上;第 5—7 行是主体标签包含的代码,也是网页要显示的主要信息。

- <meta />

meta 是元信息标签,用来描述一个 HTML 网页文档的属性,这些内容不会显示在网页上。比如编码方式、文档的摘要、文档的关键词、文档的刷新时间等,其中网页的摘要、关键词是为了使搜索引擎能对网页内容的主题进行识别和分类。

元信息标签在网页的 <head> </head> 标签对之间,下面有一段代码:

```
1. <html>
2.     <head>
3.         <meta charset="utf-8">
4.         <meta name="keywords" content="HTML, CSS, Javascript"/>
```

```
5.          <meta name="description"content ="HTML 从语义的角度描述页面结构,  
           CSS 从审美的角度美化页面, Javascript 从交互的角度提升用户体验"/>  
6.          <meta http-equiv="Refresh"content ="10"url = http:// info.cern.ch>  
7.          <title>设置 meta 相关功能</title>  
8.      </head>  
9.      <body>  
10.     </body>  
11. </html>
```

代码中第 3 行 charset 属性用来设定网页字符集,采用国际编码 utf-8,需要注意:当源文件保存时编码与源文件声明 <meta charset="编码方式"> 不一致,就会出现乱码问题。

代码中第 4 行 name 属性用来告诉搜索引擎站点的主要内容,格式为 <meta name="keywords"content="关键词"/>。

代码中第 5 行 name 属性用来设定站点在搜索引擎上的描述,格式为 <meta name="description"content="网页的描述信息"/>。

代码中第 6 行 http-equiv 属性用来设定网页自动跳转,当前网页等待 10 秒后,自动跳转到指定网页 http:// info.cern.ch,如果没有指定网页,则经过设定的时间后自动刷新页面内容。

- <title> </title>

<title> </title> 标签的作用是设置网页的“标题”,浏览器窗口标题栏部分显示的文本信息就是网页的标题。

代码中第 7 行用来标签网页的标题。

- <body> </body>

<body> </body> 定义 HTML 文档的主体部分,是双标签。在 <body> </body> 之间放置要显示的内容(文本、图像、表格等)和其他用于控制内容显示方式的标签,如 <p> </p>、<h1> </h1>、
、<hr>等,这些内容将会在浏览器的窗口内显示出来。

2.4 HTML 文档类型

在 Web 世界中,存在多种文档格式,浏览器需要通过文档类型声明来识别并正确解析页面内容。HTML 有多个版本(如 HTML 4.01、XHTML、HTML5),只有明确指定所使用的 HTML 版本,浏览器才能按照对应标准渲染页面,确保显示效果的一致性与兼容性。

<!DOCTYPE> 并非 HTML 标签,而是一条文档类型声明(Document Type Declaration),用于告知浏览器当前文档遵循的 HTML 规范版本。它必须位于 HTML 文件的最顶部,紧接在首行之前,且不区分大小写。

现代网页普遍采用 HTML5,其文档声明为: <!DOCTYPE>,该声明简洁明了,被所有主流浏览器(如 Chrome、Firefox、Edge、Safari 等)广泛支持,是当前标准的 HTML 文档开头方式。如图 2-8 所示,主流浏览器均能正确识别并处理标准的文档类型声明,保障网页的跨平台兼容性。



图 2-8 常用浏览器

```
1. <!DOCTYPE html>
2. <html>
3.   <head><title></title></head>
4.   <body></body>
5. </html>
```

代码中第 1 行表示当前文档是符合 HTML5 标准的一个文档, <!DOCTYPE> 声明必须是 HTML 文档的第一行, 位于 <html> 标签之前, 便于浏览器确定渲染模式。
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"

"http://www.w3.org/TR/html4/strict.dtd">

代码中第 1 行表示当前文档是符合 HTML 4.01 标准的一个文档, HTML 4.01 规定了三种 DTD 类型: 严格型 (Strict)、过渡型 (Transitional) 以及框架型 (Frameset)。

2.5 常用文本格式化标签

文本是网页中最基本、最核心的内容形式, 无论是新闻报道、博客文章还是教学资料, 大多以文本为主体。为了使文本结构清晰、层次分明, HTML 提供了一系列文本格式化标签, 用于定义标题、段落、强调内容等, 提升页面的可读性与语义化程度。

本节将介绍常用的文本控制标签, 帮助开发者合理组织网页内容, 实现结构与表现的初步分离。

2.5.1 标题标签

HTML 提供了六级标题标签: <h1> 到 <h6>, 用于定义不同层级的标题。

<h1> 表示最高级、最重要的标题 (如页面主标题), 字体最大;

<h2> 至 <h5> 用于次级标题, 体现内容的层次结构;

<h6> 为最低级别标题, 字体最小。

标题标签的对齐属性 align。它用来说明标题的对齐方式。align 的属性值包括 Left (左对齐)、Center (居中对齐)、Right (右对齐), 用户可以根据需求进行相应设置。

【例 2-4】 输入以下代码, 页面显示效果如图 2-9 所示。

```
1. <!DOCTYPE html>
2. <html>
3.   <head>
4.     <meta charset="utf-8">
```

```
5.         <title>标题标签</title>
6.     </head>
7.     <body>
8.         <h1 align="center">我是一级标题!居中对齐</h1>
9.         <h2>我是二级标题!默认居左对齐</h2>
10.        <h3 align="right">我是三级标题!居右对齐</h3>
11.        <h4>我是四级标题!</h4>
12.        <h5>我是五级标题!</h5>
13.        <h6>我是六级标题!</h6>
14.    </body>
15. </html>
```



图 2-9 标题标签

代码中第 8 行 h1 为一级标题,其后是 h2,再其后是 h3,以此类推。搜索引擎使用标题为网页的结构和内容编制索引,同时用户可以通过标题标签来快速浏览网页。

2.5.2 段落标签

段落标签 <p> 用于定义网页中的一个段落,是组织正文内容的基本单元。浏览器在渲染时会自动在段落前后添加一定的空白(外边距),使文本内容层次清晰、易于阅读。

与标题标签类似,<p> 标签也支持 align 属性,用于设置段落文本的水平对齐方式。

【例 2-5】 输入以下代码,页面显示效果如图 2-10 所示。

```
1. <!DOCTYPE html>
2. <html>
3.     <head>
4.         <meta charset="utf-8">
5.         <title>段落标签</title>
6.     </head>
7.     <body>
```

8. <h1 align="center">万维网联盟</h1>
9. <p>万维网联盟,又称 W3C 理事会。1994 年 10 月在麻省理工学院计算机科学实验室成立。建立者是万维网的发明者蒂姆·伯纳斯·李。</p>
10. <p>W3C 最重要的工作是发展 Web 规范(称为推荐,Recommendations),这些规范描述了 Web 的通信协议(比如 HTML 和 XHTML)和其他的构建模块。</p>
11. <p>每项 W3C 推荐的发展是通过由会员和受邀专家组成的工作组来完成的。工作组的经费来自公司和其他组织,并会创建一个工作草案,最后是一份提议推荐。一般来说,为了获得正式的批准,推荐都会被提交给 W3C 会员和主任。</p>
12. </body>
13. </html>

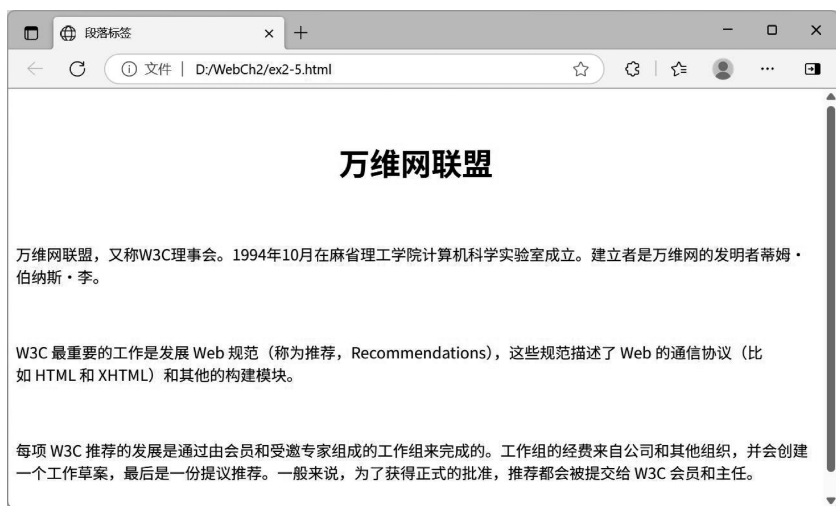


图 2-10 段落样式

代码中第 9—11 行分别插入段落标签,段落与段落之间会默认自动插入一个空行(段间距)。

2.5.3 换行标签

段落之间默认存在较大间距(由浏览器自动添加的外边距决定),若需在段落内部或相邻内容之间进行紧凑换行(如诗歌、地址、多行提示等),可使用换行标签
。

 是一个自闭合标签(空元素),不包含内容,仅表示换行。可通过连续使用多个
实现多行间距控制。

2.5.4 预格式标签

使用预格式化标记<pre></pre>格式化文本,可以在浏览器中显示的内容保留原始的排版格式,该标记中的文本内容将按原格式显示,保留所有空格、换行和定位符等。

【例 2-6】 输入以下代码,页面显示效果如图 2-11 所示。

1. <!DOCTYPE html>
2. <html>

```
3.      <head>
4.          <meta charset="utf-8">
5.          <title>预格式标签</title>
6.      </head>
7.      <body>
8.          <p> Pre 标签很适合显示计算机代码:</p>
9.          <pre>
10.              for i = 1 to 10
11.                  print i
12.              next i
13.          </pre>
14.      </body>
15. </html>
```

第 9—13 行对文字段落进行预格式化



图 2-11 预格式化

2.6 图像与多媒体

图像和多媒体是现代网页不可或缺的重要元素。合理使用图片、音频和视频,不仅能丰富页面内容、提升视觉表现力,还能增强用户体验,使网页更具吸引力和传播力。在 Web 前端开发中,掌握图像与多媒体的插入方法及其属性设置,是构建完整、交互性强的网页的基础技能。本节将介绍如何在 HTML 中嵌入图片与多媒体元素,并结合实际案例说明其应用场景与最佳实践。

2.6.1 图片的基本格式

网页中常用的图片格式主要有三种:gif 格式、jpg 格式和 png 格式。

(1) GIF 图像格式

GIF 扩展名是.gif。它在压缩过程中,图像的像素资料不会被丢失,丢失的是图像的色彩。GIF 格式最多只能储存 256 色,所以通常用来显示简单图形及字体。有一些数码相机会有—种名为 Text Mode 的拍摄模式,就可以储存成 GIF 格式。

GIF 可以包含透明区域和多帧动画。因此 GIF 通常适用于卡通、图形、logo、带有透明区域的图形和动画等。

(2) JPEG 图像格式

JPEG 扩展名是.JPG,其全称为 Joint Photographic Experts Group。它利用—种失真式

的图像压缩方式将图像压缩在很小的储存空间中,其压缩比率通常在 10 : 1~40 : 1 之间。这样可以使图像占用较小的空间,所以很适合应用在网页中。JPEG 格式的图像主要压缩的是高频信息,对色彩的信息保留较好,因此也普遍应用于需要连续色调的图像中,如广告 banner 图、照片、产品图片等。

(3) PNG 图像格式

PNG(Portable Network Graphics)的原名称为“可移植性网络图像”。PNG 能够提供长度比 GIF 小 30% 的无损压缩图像文件。它同时提供 24 位和 48 位真彩色图像支持以及其他诸多技术性支持。

2.6.2 图像标签

图像标签: , 其属性可以控制图像的路径、尺寸和替换文字等各种功能。其中 src 属性为必需属性,用于指定图像的路径,支持相对路径和绝对路径,基本语法: 。

width 和 height 属性用于定义图像的宽度和高度,单位可为像素或百分比,基本语法: 。

alt 属性用于设置替代文字,当图像无法加载或无法显示时,将在图像位置显示该提示信息,有助于提升网页的可访问性与用户体验,基本语法: 。

【例 2-7】 图像标签示例代码如下,页面显示效果如图 2-12 所示。

```
1. <!DOCTYPE html>
2. <html>
3.     <head>
4.         <meta charset="utf-8">
5.         <title>图像标签</title>
6.     </head>
7.     <body>
8.         <h2> HTML </h2>
9.         
10.    </body>
11. </html>
```

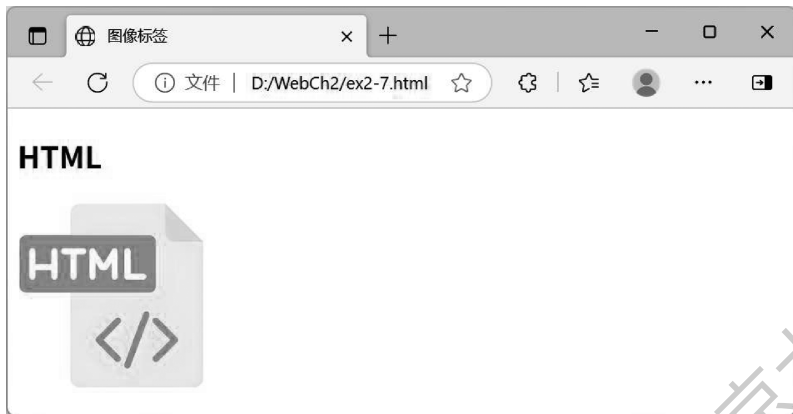


图 2-12 图像标签

2.6.3 插入多媒体元素

`<embed>` `</embed>` 是用来嵌入外部多媒体内容的另一个重要标签,可用于插入音频、视频或 Flash 动画等资源,<embed> 标记语法: `<embed src="" autostart="" loop="" hidden="">` `</embed />`,常用属性取值及其说明如下:

- `src` 属性用于指定音频、视频等多媒体文件及其路径,可以是相对路径或绝对路径。
- `autostart` 属性用于控制多媒体内容是否自动播放,取值为“true”表示文件在下载完成后自动播放,取值为“false”表示文件在下载完成后不自动播放。一般默认为自动播放。
- `loop` 属性用于控制多媒体内容是否循环播放,取值可以为正整数、true 和 false。设为 true 表示无限循环播放音频或视频文件;设为 false 表示不循环播放;若设置为正整数,则文件循环播放次数为该正整数数值。
- `hidden` 属性用于设置播放界面的显示和隐藏,取值为“true”代表隐藏面板,取值为“no”表示显示面板,默认值为“no”。

【例 2-8】 在页面嵌入 mp4 文件,代码如下,页面显示效果如图 2-13 所示。

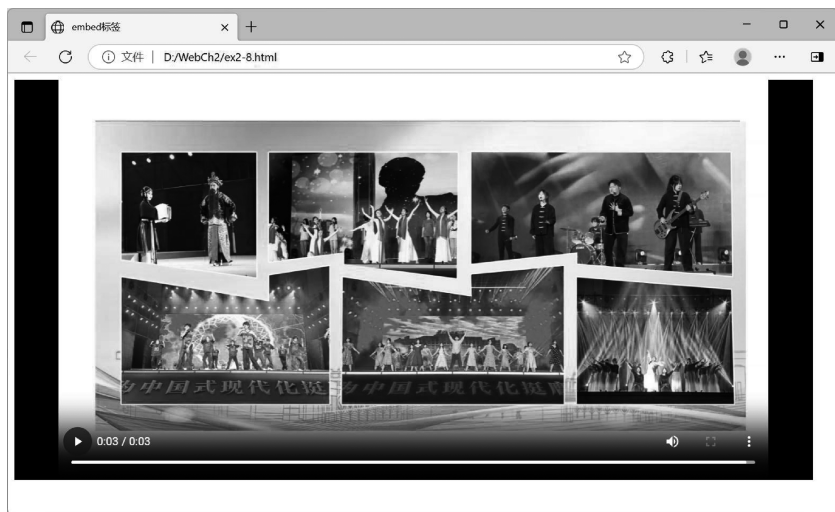


图 2-13 页面中嵌入 mp4 文件

```
1. <!DOCTYPE html>
2. <html>
3.     <head>
4.         <meta charset="utf-8"/>
5.         <title> embed 标签</title>
6.     </head>
7.     <body>
8.         <embed src="./demo.mp4"width="1000"height="500"></embed>
9.     </body>
10. </html>
```

第 8 行代码使用 `<embed>` 标记嵌入 flash 文件,并设置了播放界面的宽度和高度。

2.7 列表

列表是网页中组织相关内容的重要工具。HTML 提供了三种列表类型：无序列表、有序列表和定义列表，用于呈现具有结构化特征的信息内容。

2.7.1 无序列表标签

无序列表用于表示一组无固定顺序的并列项目，也称为项目符号列表。列表中的各项内容属于同一层级，无先后之分，常用于组织新闻列表、文章目录、导航菜单等结构化内容。

无序列表使用 `<ul>` 标签定义，列表项使用 `<li>` 标签表示，一个 `<ul>` 中可包含一个或多个 `<li>` 项。

【例 2-9】 在 body 标签中插入无序列表结构，代码如下，页面显示效果如图 2-14 所示。

```
1. <!DOCTYPE html>
2. <html>
3.   <head>
4.     <meta charset="utf-8">
5.     <title>无序列表</title>
6.   </head>
7.   <body>
8.     <h4>HTML 支持有序、无序和定义列表</h4>
9.     <ul>
10.      <li>无序列表是一个项目的列表。</li>
11.      <li>此列表项使用粗体圆点(典型的小黑圆圈)进行标记。</li>
12.      <li>列表项内部可以使用段落、换行符、图片、链接以及其他列表等。</li>
13.    </ul>
14.  </body>
15. </html>
```



图 2-14 无序列表

代码中第 9—13 行定义了无序列表，默认列表符号是实心圆形，通过设置 `type` 属性，还可以实现空心圆形、实心正方形的显示风格。

2.7.2 有序列表标签

有序列表使用 `<ol>` 标签定义,列表项使用 `<li>` 标签表示,其基本结构与无序列表类似。`<ol>` 用于组织具有明确顺序或优先级的内容,如文章排行榜、考试步骤、操作流程等,浏览器会默认为其列表项添加数字编号。

【例 2-10】 在 `body` 标签中插入有序列表结构,代码如下,页面显示效果如图 2-15 所示。

```
1. <!DOCTYPE html>
2. <html>
3.     <head>
4.         <meta charset="utf-8">
5.         <title>有序列表</title>
6.     </head>
7.     <body>
8.         <h2>HTML 支持有序、无序和定义列表</h2>
9.         <ol>
10.            <li>有序列表也是一列项目。</li>
11.            <li>列表项目使用数字进行标记。</li>
12.            <li>列表项内部可以使用段落、换行符、图片、链接及其他列表等。</li>
13.        </ol>
14.    </body>
15. </html>
```



图 2-15 有序列表

代码中第 9—13 行定义了有序列表,默认列表项前面编号为数字列表,通过设置 `type` 属性,可以实现大小写字母编号、大小写罗马字母编号。

2.7.3 定义列表标签

定义列表使用 `<dl>` 标签创建,用于组织术语与其定义或描述之间的配对关系。它包含两个子标签:

- `<dt>`: 定义列表中的术语或关键词;
- `<dd>`: 对 `<dt>` 的描述或解释内容。

与无序列表和有序列表不同,定义列表本身不带项目符号或编号,结构更灵活,常用于词汇表、参数说明等场景。

【例 2-11】 在 body 标签中插入定义列表结构,代码如下,页面显示效果如图 2-16 所示。

```
1. <!DOCTYPE html>
2. <html>
3.   <head>
4.     <meta charset="utf-8">
5.     <title>定义列表</title>
6.   </head>
7.   <body>
8.     <h4> Web 标准中结构、表现和行为: </h4>
9.     <dl>
10.      <dt> HTML </dt>
11.      <dd>从语义的角度,描述页面结构。</dd>
12.      <dt> CSS </dt>
13.      <dd>从审美的角度,美化页面。</dd>
14.      <dt> JavaScript </dt>
15.      <dd>从交互的角度,提升用户体验。</dd>
16.    </dl>
17.  </body>
18. </html>
```



图 2-16 定义列表

代码中第 9—16 行定义了定义列表,第 10、12、14 行定义了列表项的标题,第 11、13、15 行定义了列表项的描述。

2.7.4 嵌套列表

在网页设计中,常需要实现二级或多级菜单结构,通过列表的嵌套可以轻松实现多层次内容的组织。嵌套列表是指在一个列表项 `<li>` 内部再嵌套另一个完整的列表(`<ul>` 或 `<ol>`),从而形成层次分明的结构。

需要注意的是:子列表必须作为父级 `<li>` 元素的内容,不能独立于 `<li>` 之外,否则将破

坏列表的语义结构与布局。

【例 2-12】 在 body 标签中插入列表的混合嵌套结构,代码如下,页面显示效果如图 2-17 所示。

```
1. <!DOCTYPE html>
2. <html>
3.     <head>
4.         <meta charset="utf-8">
5.         <title>嵌套列表</title>
6.     </head>
7.     <body>
8.         <ol>
9.             <li>Web 前端开发工具
10.                <ul type="disc">
11.                    <li>Notepad++</li>
12.                    <li>HBuilder</li>
13.                    <li>Visual Studio Code</li>
14.                    <li>WebStorm</li>
15.                </ul>
16.            <li>浏览器
17.                <ul>
18.                    <li>Microsoft Edge</li>
19.                    <li>Google Chrome</li>
20.                    <li>Mozilla Firefox</li>
21.                    <li type="square">Safari</li>
22.                    <li>QQ 浏览器</li>
23.                </ul>
24.            </li>
25.        </ol>
26.    </body>
27. </html>
```

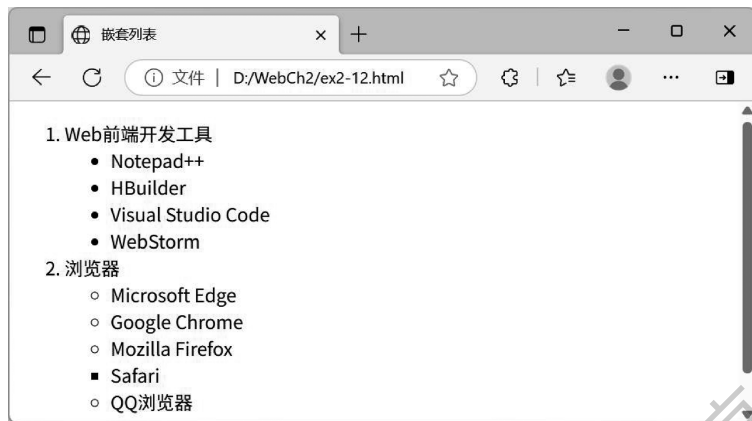


图 2-17 嵌套列表

代码中第 8—25 行定义了有序列表;第 10—15 行在有序列表中嵌套了一个列表符号为实心圆形的无序列表;第 17—23 行又在有序列表中嵌套了一个无序列表,其中第 21 行设置了列表项的 type 属性为“square”,所以此项之前为实心正方形。

2.8 HTML 表格标签

表格在网页中被广泛应用于数据的展示与组织,如课程表、成绩表、价格清单等。它不仅能清晰呈现结构化信息,还具有良好的对齐与排版能力,使内容呈现更加直观、有序。

2.8.1 表格标签

HTML 使用 <table> 标签定义一个表格。表格由行和单元格组成,基本结构如下:

- <tr>:定义表格中的一行(table row);
- <td>:定义标准单元格(table data),用于存放具体数据内容;
- <th>:定义表头单元格(table header),通常用于首行或首列,内容默认加粗并居中显示;每个 <tr> 可包含多个 <td> 或 <th>,构成一行中的多个单元格。
- <td> 和 <th> 是表格的最基本结构单元,单元格内可嵌套文本、图像、列表、链接等多种 HTML 元素,具有良好的内容包容性。

表格标签常用属性如表 2-1 所示。

表 2-1 表格标签的属性

标 签	说 明
border	表格边框的宽度,单位为像素,默认值为 0
cellspacing	表格的单元格与单元格之间的间距,单位为像素
cellpadding	表格的单元格边框与内容之间的间距,单位为像素
width	表格的宽度,单位为像素,默认值为能容纳表格内容的最小宽度
height	表格的高度,单位为像素,默认值为能容纳表格内容的最小高度
bgcolor	整个表格的背景颜色
align	表格相对于页面的水平对齐方式,默认值为 left,即左对齐

【例 2-13】 设计一个四行四列的表格,代码如下所示,页面显示效果如图 2-18 所示。

```
1.  <!DOCTYPE html>
2.  <html>
3.    <head>
4.      <meta charset="utf-8">
5.      <title>设计表格</title>
6.    </head>
7.    <body>
8.      <table border="1"cellspacing="0"width="300" height="200"align="center">
9.        <tr bgcolor="silver"><th>课程编号</th><th>课程名称</th><th>学时</th><th>学分</th><th>开课学期</th></tr>
10.       <tr align="center"><td>1356310001 </td><td>数据结构</td><td>64 </td><td>4 </td><td>秋季</td></tr>
```

```
11.         <tr align="center"><td> 1356310002 </td><td>机器学习</td><td>
            48 </td><td> 3 </td><td>秋季</td></tr>
12.         <tr align="center"><td> 1356310003 </td><td>计算机网络</td><td>
            > 64 </td><td> 4 </td><td>春季</td></tr>
13.         <tr align="center"><td> 1356310004 </td><td>深度学习及应用</td>
            <td> 48 </td><td> 3 </td><td>春季</td></tr>
14.     </table>
15. </body>
16. </html>
```



图 2-18 表格的应用

代码中第 8—14 行定义了一个表格,第 8 行定义了表格的边框、单元格间距、宽度和高度等属性,第 9 行采用<th>标签定义了表头,第 10—13 行采用<td>标签定义了表内容。单元格标签<th>、<td>具有共同的属性,如表 2-2 所示。

表 2-2 <td>、<th>共同的属性

标 签	说 明
align	单元格水平对齐属性,取值为 left、center 和 right,其中 left 为默认值
valign	单元格垂直对齐属性,取值为 middle、top 和 bottom,其中 middle 为默认值
width	单元格宽度属性,若不设置单元格宽度,则默认为能容纳该单元格内最宽字符
height	单元格高度属性,若不设置单元格高度,则默认为能容纳该单元格内最高字符
bordercolor	单元格边框的颜色
bgcolor	单元格背景色

2.8.2 单元格的跨行设置

在网页表格中,当某单元格内容需要纵向跨越多行时,可使用 rowspan 属性实现跨行合并。该属性应用于<td> 或 <th>标签,用于指定当前单元格所占的行数。设置后,被合并的下方单元格应省略,以确保表格结构正确。

【例 2-14】 设计跨行合并的表格,代码如下所示,页面显示效果如图 2-19 所示。

```
1. <!DOCTYPE html>
```

```
2. <html>
3.   <head>
4.     <meta charset="utf-8">
5.     <title>表格跨行合并</title>
6.   </head>
7.   <body>
8.     <table border="1" cellspacing="0" width="300" height="200" align
      ="center">
9.       <tr bgcolor="silver">
10.        <th>课程编号</th><th>课程名称</th>
      <th>学时</th><th>学分</th><th>开课学期</th>
11.      </tr>
12.      <tr align="center">
13.        <td>1356310001</td><td>数据结构</td><td>64</td><td>
      4</td><td rowspan="2">秋季</td>
14.      </tr>
15.      <tr align="center">
16.        <td>1356310002</td><td>机器学习</td><td>48</td><td>
      3</td>
17.      </tr>
18.      <tr align="center">
19.        <td>1356310001</td><td>计算机网络</td><td>64</td>
      <td>4</td><td rowspan="2">春季</td>
20.      </tr>
21.      <tr align="center">
22.        <td>1356310002</td><td>深度学习及应用</td><td>48</td>
      <td>4</td>
23.      </tr>
24.    </table>
25.  </body>
26. </html>
```

对单元格执行跨行操作



课程编号	课程名称	学时	学分	开课学期
1356310001	数据结构	64	4	秋季
1356310002	机器学习	48	3	
1356310001	计算机网络	64	4	春季
1356310002	深度学习及应用	48	3	

图 2-19 表格跨行合并

2.8.3 单元格的跨列设置

在网页表格中,当某单元格内容需要横向跨越多列时,可使用 `colspan` 属性实现跨列合并。该属性应用于 `<td>` 或 `<th>` 标签,用于指定当前单元格所占的列数。设置后,同一行中后续对应的单元格应省略,以保持表格结构正确。

【例 2-15】 设计跨列合并的表格,代码如下所示,页面显示效果如图 2-20 所示。

```
1.  <!DOCTYPE html>
2.  <html>
3.      <head>
4.          <meta charset="utf-8">
5.          <title>表格跨列合并</title>
6.      </head>
7.      <body>
8.          <p></p>
9.          <table border="1"cellspacing="0"width="300"  height="200"align
="center">
10.              <tr bgcolor="silver">
11.                  <th>课程编号</th><th>课程名称</th><th>学时</th><th>
学分</th><th>开课学期</th>
12.              </tr>
13.              <tr align="center">
14.                  <td>1356310001 </td><td>数据结构</td> <td> 64 </td>
<td>4 </td><td rowspan="2">秋季</td>
15.              </tr>
16.              <tr align="center">
17.                  <td>1356310002 </td><td>机器学习
</td> <td> 48 </td><td> 3 </td></tr>
18.              </tr>
19.              <tr align="center">
20.                  <td>1356310001 </td><td>计算机网络
</td> <td> 64 </td><td> 4 </td><td rowspan="2">春季</td>
21.              </tr>
22.              <tr align="center">
23.                  <td>1356310002 </td><td>深度学
及应用</td> <td> 48 </td><td> 3 </td>
24.              </tr>
25.              <tr align="center">
26.                  <td colspan="2">总计</td><td> 224 </td><td> 14 </td>
<td></td>
27.              </tr>
28.          </table>
29.      </body>
30. </html>
```

对单元格执行
跨列操作



课程编号	课程名称	学时	学分	开课学期
1356310001	数据结构	64	4	秋季
1356310002	机器学习	48	3	
1356310001	计算机网络	64	4	春季
1356310002	深度学习及应用	48	3	
总计		224	14	

图 2-20 表格的跨列合并

2.9 实例演练

使用记事本编写网页时,由于缺乏语法高亮、智能提示和结构模板,代码可读性差,错误排查困难,且每次需手动编写基本文档结构,开发效率较低。为此,推荐使用轻量高效、易于初学者学习的 HBuilderX 作为开发工具。它支持语法着色、代码补全、快速生成 HTML5 模板等功能,显著提升前端开发体验与效率。

(1) 在 HBuilderX 中创建项目。

依次点击【文件】->【新建】->【项目】,弹出新建项目对话框(如图 2-21 所示,不同版本界面可能不同)。选择“基本 html 项目”输入项目名称,设置或选择项目保存路径,并选择合适的模板,点击“创建”即可生成标准结构的项目文件。



图 2-21 HBuilderX 新建项目界面

(2) 创建完成, 自动生成基本的 HTML 文档模板, 如图 2-22 所示, 可以直接编写网页。

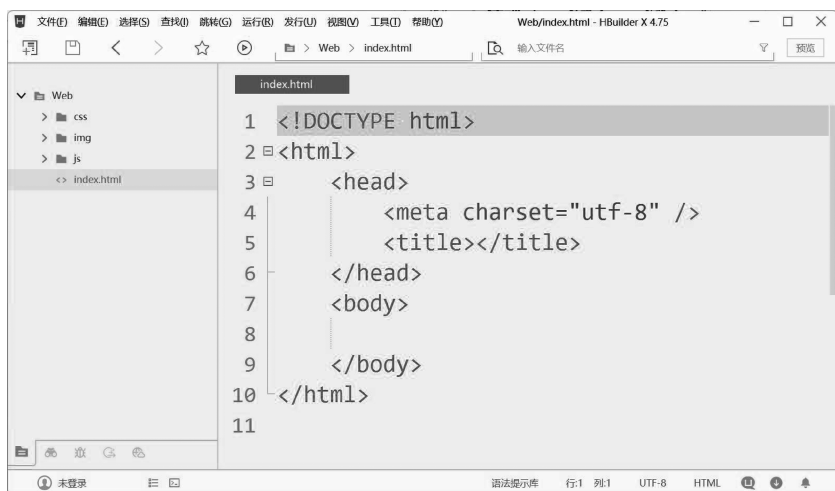


图 2-22 HTML 文档模板

(3) 参照给定的 HTML 代码, 应用 HBuilderX 设计 Web 网页, 页面运行效果如图 2-23 所示。

```
1. <!DOCTYPE html>
2. <html>
3.   <head>
4.     <meta charset="utf-8">
5.     <title> Web 前端开发技术初步应用</title>
6.     <style type="text / css">
7.       p{font-size:20px;color:red;}
8.       h3{font-size:24px;font-weight:bolder;color:#000099;}
9.     </style>
10.  </head>
11.  <body>
12.    <h3> Web 前端开发技术</h3>
13.    <p> HTML</p>
14.    <p> CSS</p>
15.    <p> JavaScript</p>
16.    <h3>网络学习资源</h3>
17.    <a href="http:// www.w3school.com.cn / html /"> W3C 教程</a>
18.  </body>
19. </html>
```

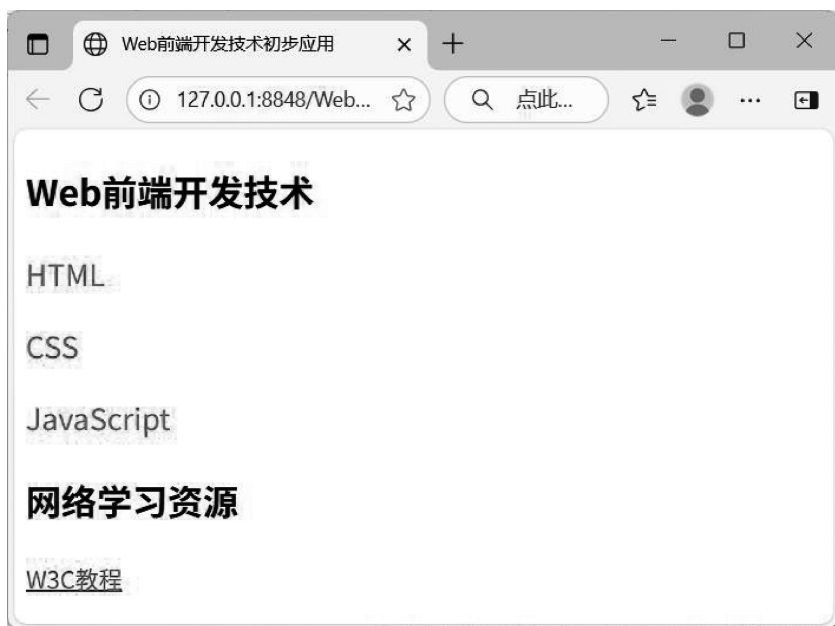



图 2-23 综合实例应用

上述代码第 3—10 行是 HTML 的头部,包含元信息、标题和样式的定义;第 11—18 行是 HTML 的主体,包含标题、段落和超链接。

2.10 思政融入

1. 规范意识与工匠精神:以严谨态度打造技术根基

HTML 作为网页的结构基础,其标签的正确使用、层级的合理嵌套、语义的准确表达,直接决定网页的质量与可维护性。一个缺少闭合标签、滥用 `<div>` 替代语义标签的页面,不仅影响样式与交互实现,还会降低搜索引擎友好性和无障碍访问能力。需要加强 HTML5 语义化标签(如 `<header>`、`<nav>`、`<article>` 等)的规范应用,养成结构清晰、命名合理、层次分明的编码习惯,把写好每一行 HTML 视为基本职业素养,为后续前端开发打下坚实的技术基础。

2. 信息责任与传播伦理:以技术守护信息价值

网页是信息传播的重要载体,HTML 的结构设计直接影响信息的组织方式与传达效果。通过案例教学,理解前端开发者不仅是“代码编写者”,更是“信息架构师”和“公共信息传播者”,用规范的技术手段促进信息公平传播。

2.11 本章小结

本章介绍了 HTML 的基础入门知识,重点介绍了常用的 HTML 标签,本节的重点是列表、链接、图像、表格和表单标签。HTML 的标签还有很多,标签的作用和含义可以参考 HTML 手册或 w3scholl 在线教程。

本章知识点如图 2-24 所示。

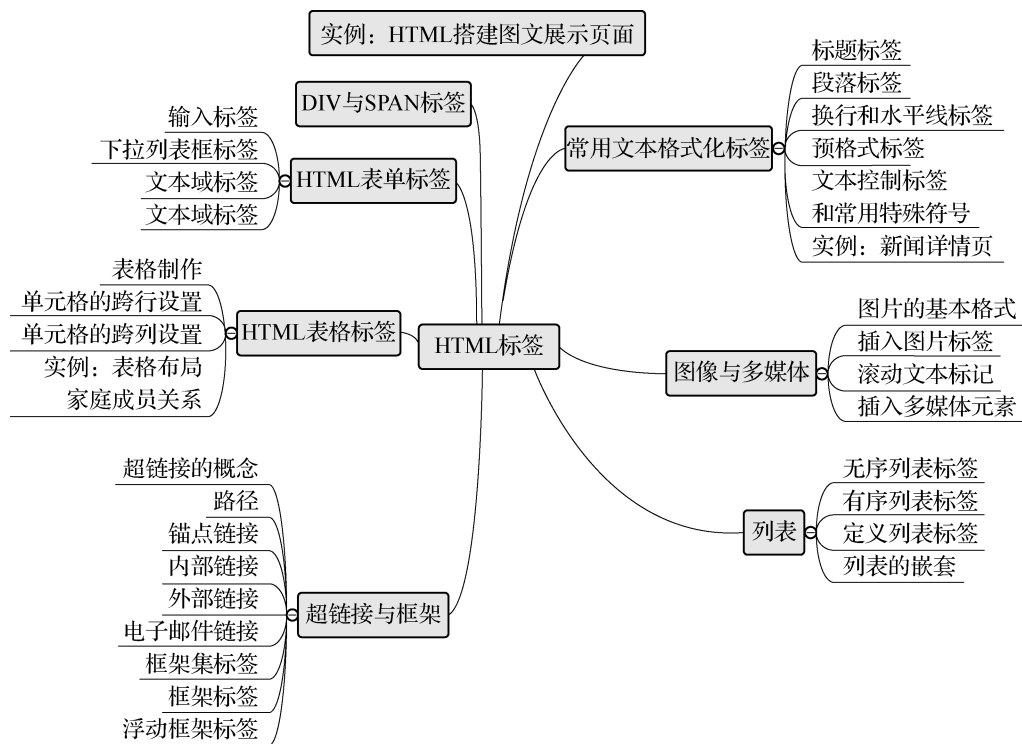


图 2-24 HTML 标签思维导图

2.12 拓展训练

(1) 以下说法错误的是()。

- A. HTML 与 CSS 配合使用,是为了内容与样式分离
- B. 如果只使用 HTML 而不使用 CSS,网页是不可能具有样式的
- C. JavaScript 可以嵌入在 HTML 语言中,作为网页源文件的一部分存在
- D. CSS 表示层叠样式表,可以添加页面的样式,规定网页的布局

(2) Web 标准的制定者是()。

- A. 微软
- B. 万维网联盟(W3C)
- C. 网景公司(Netscape)
- D. SUN 公司

(3) Web 的发明者是()。

- A. 冯·诺依曼
- B. 乔布斯
- C. 比尔·盖茨
- D. 蒂姆·伯纳斯·李

(4) 世界上第一个网页是()。

- A. <http://www.w3c.org>
- B. <http://info.cern.ch>
- C. <http://www.microsoft.com>
- D. <http://www.baidu.com>

(5) 网页是由 HTML 语言来实现的,HTML 语言是()。

- A. 大型数据库
C. 网络通信协议
- B. 网页源文件中出现的唯一一种语言
D. 超文本标记语言
- (6) 以下关于 HTML 标签叙述错误的是()。
- A. 可以单独出现,也可以成对出现
C. 标签可以带有属性,属性的顺序无关
- B. 必须正确嵌套
D. 标签和其属性构成了 HTML 元素
- (7) 正确表达页面注释格式的是()。
- A. `<!--注释信息 -->`
C. `<!--注释信息>`
- B. `<--注释信息 -->`
D. `<! comment>`
- (8) 以下不属于 meta 元信息标签的属性是()。
- A. name
B. color
C. content
D. http-equiv
- (9) `<!DOCTYPE>` 元素的作用是()。
- A. 用来定义文档类型
B. 用来声明命名空间
C. 用来向搜索引擎声明网站关键字
D. 用来向搜索引擎声明网站作者
- (10) 以下关于浏览器的描述错误的是()。
- A. 主流的浏览器有 Chrome、Firefox、IE 等
B. 不同浏览器厂商的浏览器,一定有不同的内核
C. 不同版本的浏览器差别可能很大,对 Web 技术的支持度也会不同
D. Chrome 浏览器可以在进行 Web 前端开发时,用于调试和测试
- (11) 创建最小的标题的文本标签是()。
- A. `<pre> </pre>`
B. `<h1> </h1>`
C. `<h6> </h6>`
D. `<b> </b>`
- (12) 下面标记中,()在标记的位置添加一个回车符。
- A. `<h1>`
B. `<enter>`
C. `<br />`
D. `<hr>`
- (13) 表示水平分割线的 HTML 代码是()。
- A. `<hr />`
B. `<br />`
C. `<tr />`
D. `<hr> </hr>`
- (14) 在 HTML 中,标记 `<pre>` 的作用是()。
- A. 标题标签
B. 预排版标签
C. 换行标签
D. 文字效果标签
- (15) 在网页中添加一个图片的 HTML 代码是()。
- A. `<bgcolor ="#FFFFFF">`
B. `<img src ="图片路径及文件名">`
C. `<img href ="图片路径及文件名">`
D. `<title> 图片 </title>`
- (16) 定义无序列表的 HTML 代码是()。
- A. `<li> <ui>... </li> </ul>`
B. `<ul> <li>... </li> </ul>`
C. `<ol> <li>... </li> </ol>`
D. `<li> <ol>... </li> </ol>`
- (17) 超文本标记语言 `<a href ="https:// www.cctv.com"> 央视国际 </a>` 的作用是()。
- A. 插入一段央视国际网站的文字
B. 插入一幅央视国际网站的图片
C. 创建一个指向央视国际网站的电子邮件
D. 创建一个指向央视国际网站的超链接

(18) 设置围绕表格的边框宽度的 HTML 代码是()。

- A. <table size =""> B. <table border ="">
C. <table bordersize =""> D. <tableborder ="">

(19) 用于设置文本输入框显示宽度的属性是()。

- A. size B. maxlength C. value D. length

(20) 下列 input 标签的类型属性取值表示单选按钮的是()。

- A. hidden B. checkbox C. radio D. select



【微信扫码】

本章参考答案 & 相关资源

南京大学出版社
版权所有

第三章

CSS 基础

CSS(Cascading Style Sheets)是控制网页外观与样式的核心技术,能够实现网页内容与表现的分离,提升页面的美观性、一致性和可维护性。本章系统介绍 CSS 的基本语法、样式表分类、选择器类型以及文字、文本、列表、表格等常用样式属性的使用方法。通过实例演练,掌握如何为 HTML 元素设置字体、颜色、间距、边框等视觉样式,理解层叠与继承机制,奠定网页美化与风格统一的技术基础。



教学目标

【知识目标】

- 掌握 CSS 的基本语法结构、三种样式表(行内、内部、外部)的使用场景与引入方式。
- 理解常用选择器(标签、类、ID 等)的优先级规则及文字、文本、列表、表格等核心样式的设置方法。

【能力目标】

- 能够运用 CSS 对 HTML 页面进行基础美化,实现字体、颜色、段落、边框等样式的精准控制。
- 能根据设计需求合理选择和组合选择器,编写结构清晰、可复用的样式代码。

【素养目标】

- 培养良好的代码组织习惯,注重样式与结构的分离,提升前端开发的工程化意识。
- 增强审美意识与用户导向思维,理解视觉设计在提升用户体验中的重要作用。

【思政元素】

- 工匠精神与细节意识:通过强调样式精度、代码规范与浏览器兼容性,培养学生精益求精、注重细节的职业态度。
- 自主创新与标准引领:结合我国在 Web 前端标准制定与国产浏览器发展中的进展,增强学生对技术自主可控的认同感与使命感。

3.1 CSS 概述

CSS(层叠样式表)是一种用于描述 HTML 或 XML(包括如 SVG、XHTML 等)文档外观和格式的样式语言。它作为网页设计的标准技术之一,实现了网页内容与表现样式的分离,提升了代码的可维护性与灵活性。CSS 由 Håkon Wium Lie 和 Bert Bos 于 1994 年共同提出,1996 年被万维网联盟(W3C)正式采纳并推荐使用。经过多年发展,CSS 已从最初的简单样式控制演进为功能强大的样式系统,目前广泛使用的版本为 CSS3。

在网页设计中应用 CSS,可以精确控制页面的布局、字体、颜色、背景、动画及其他视觉效果。通过修改统一的样式表文件,开发者能够高效地调整单个页面的局部样式,或批量更新整个网站的外观风格,极大地提升了开发效率与设计一致性。

3.2 CSS 的构成

3.2.1 CSS 的语法结构

CSS 样式由选择符和声明组成,声明由属性和值组成,如图 3-1 所示。

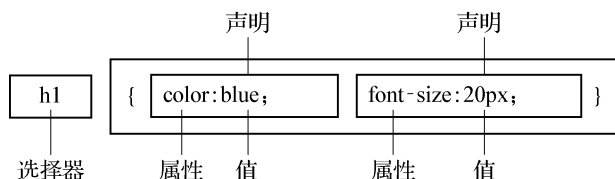


图 3-1 CSS 语法结构

选择器又称选择符,指明网页中要应用样式规则的元素;介于花括号{ }之间的内容都是声明,若需要定义多个声明,则每个声明之间使用分号分隔。

简单说来,CSS 的每一条语句都被认为是一个个的规则,这些 CSS 规则告诉浏览器如何去渲染 HTML 页面上的特定元素。CSS 规则中主要有三个核心部分,即选择器、属性、值,它们的主要作用如下所示:

- 选择器负责“选择”受此规则影响的 HTML 页面上的元素;
- 属性指的是选中的元素的样式外观,如文字颜色、边框样式、背景样式等;
- 值是给属性设置的准确的样式,如字号是 12 像素,颜色是红色;

注意:最后一条声明可以没有分号,但为了以后修改方便,一般也加上分号;为了使样式更加容易阅读,建议每行只描述一个属性。

3.2.2 注释语句

就像在 HTML 的注释一样,在 CSS 中也有注释语句:用“/* 注释语句 */”来标明(HTML 中使用“<!-- 注释语句 -->”)。

“/* 注释语句 */”这种格式可以单独一行书写,也可以写在语句的后面,可以注释一行,

也可以注释多行,但不能嵌套。

3.3 CSS 样式表分类

CSS 样式可以通过多种方式灵活地应用于 HTML 页面,具体选择取决于设计需求和代码组织方式。根据 CSS 代码在 HTML 中的位置,主要引入方式包括:行内样式、内部样式表和外部样式表,每种方式各有适用场景与优势。

3.3.1 行内样式

行内样式是引入 CSS 的较简单的一种方法。每个 HTML 标记都有一个 style 属性,行内式就是在标记的 style 属性中为元素添加 CSS 规则。基本语法:<标签名称 style ="样式属性 1:属性值 1; 样式属性 2:属性值 2;...">。

【例 3-1】 使用行内样式改变段落的颜色和左外边距。代码如下,页面显示效果如图 3-2 所示。

```
1. <!DOCTYPE html >
2. <html >
3.   <head>
4.     <meta charset ="utf - 8">
5.     <title>行内样式</title>
6.   </head>
7.   <body>
8.     <p style ="color:red; margin - left: 20px;">
9.       CSS 是一种描述 HTML 文档样式的语言。
10.    </p>
11.  </body>
12. </html>
```

样式放在 p 标签内部



图 3-2 行内样式

由于行内样式将表现(CSS)与内容(HTML)混杂在一起,违背了结构与样式分离的原则,从而失去了样式表在维护性、复用性和可读性方面的诸多优势。因此,应谨慎使用行内样式,建议在样式仅需对单个元素应用一次且无需复用的特殊场景下临时使用。一般情况下,不推荐采用行内样式。

3.3.2 内部样式

当定义的样式仅用于单个 HTML 页面时,通常采用嵌入式方式引入 CSS,也称为内部样式。顾名思义,内部样式是将 CSS 代码定义在网页文档内部,通过在 <head> 标签中使用 <style> 标签来声明样式表,从而实现样式与结构的分离,同时避免外部文件的引入,适用于页面独有样式的场景。

【例 3-2】 使用嵌入式改变段落的颜色和左外边距。代码如下,页面显示效果同图 3-2 所示。

```
1. <!DOCTYPE html>
2. <html>
3.   <head>
4.     <meta charset="utf-8">
5.     <title>内部样式</title>
6.     <style type="text / CSS">
7.       p {color:red; margin-left:20px;}
8.     </style>
9.   </head>
10.  <body>
11.    <p> CSS 是一种描述 HTML 文档样式的语言。</p>
12.  </body>
13. </html>
```



内部样式表常被用于当单个文档需要特殊的样式时使用。其中 type 属性规定被链接文档的类型。该属性最常见的类型是 "text / CSS",该类型描述样式表。

3.3.3 外部样式

链接式引入是指将一个独立的 CSS 文件通过 HTML 文档的头部使用 <link> 标签关联到网页中。在学习 CSS 或开发单个页面时,为便于调试,可采用行内样式或嵌入式样式;但在构建包含多个页面的网站时,为了统一页面风格、提升代码的可维护性与复用性,通常会将样式集中定义在一个或多个独立的 CSS 文件(以 .css 为扩展名)中,并通过链接方式引入到各个 HTML 页面中。

基本语法:<link rel="stylesheet"href="mysheet.css"type="text / css">

<link> 标签相关属性解析:

- rel 属性规定当前文档与被链接文档之间的关系。rel 的属性值为 stylesheet,指示被链接的文档是一个样式表。
- href 属性规定被链接文档的 URL 地址。可以是相对路径也可以是绝对路径。
- type 属性规定被链接文档的类型。"text / css"表示样式表。

使用记事本等编辑工具编写 CSS 文件 style1.css,代码编写如下:

```
p {color:red; margin-left:20px;}
```

【例 3-3】 使用链接式改变段落的颜色和左外边距。代码如下,页面显示效果同图 3-2 所示。

```
1. <!DOCTYPE html>
```



```
2. <html>
3.   <head>
4.     <meta charset="utf-8">
5.     <title>外部样式</title>
6.     <link href="../../css/style1.css" rel="stylesheet" type="text /
    CSS"/>
7.   </head>
8.   <body>
9.     <p> CSS 是一种描述 HTML 文档样式的语言。</p>
10.  </body>
11. </html>
```



链接式将 HTML 文件和 CSS 文件分成两个或多个文件,实现了页面框架 HTML 代码与美工 CSS 代码的完全分离,使得前期制作和后期维护都十分方便。如果要保持页面风格统一,只需要把这些公共的 CSS 样式定义在一个文件中,供不同的页面使用;如果需要改变网站风格,只需要修改公共 CSS 文件就可以了。

外部样式表最大的优势是应用起来更加方便、灵活,真正实现了“结构”与“表现”的分离。

如果上面三种方式中的两种同时作用于同一个页面,就会出现优先级的问题,三种样式的优先级从高至低为:行内式→嵌入式→链接式。

3.4 CSS 基本选择器

CSS 中提供了多种选择器,简单理解就是提供了几种方便快捷的选择元素的方式,开发者可根据需求以最直接的方式找到需要修饰的元素。下面我们介绍几种常用的选择器,其中 CSS 的基本选择器有:标签选择器、类选择器、id 选择器和通配符选择器。

3.4.1 标签选择器

一个完整的 HTML 页面由很多不同的标签组成,标签是元素固有的属性,CSS 标签选择器用来声明何种标签采用哪种 CSS 样式,因此,每一种 HTML 标签的名称都可以作为相应的标签选择器的名称。

【例 3-4】 标签选择器的应用。代码如下,页面显示效果如图 3-3 所示。

```
1. <!DOCTYPE html>
2. <html>
3.   <head>
4.     <meta charset="utf-8">
5.     <title> CSS 标签选择器</title>
6.     <style type="text / CSS">
7.       p {color:red;}
8.       div{background-color:#e6f7ff;border: 1px solid red;}
9.     </style>
10.  </head>
11.  <body>
```

南京大学出版社
版权所有

```
12.         <p>标签选择器</p>
13.         <div>一个完整的 HTML 页面是由不同的标签组成,标签选择器,则是决定哪些标
            签采用相应的 CSS 样式。</div>
14.     </body>
15. </html>
```



图 3-3 CSS 标签选择器

代码中第 6—9 行定义了 p 和 div 标签选择器,color 为属性,red 为属性值,该 CSS 将 HTML 中所有段落文字设置为红色。第 12 行 p 标记中的内容应用了标记选择器中定义的样式。

3.4.2 类选择器

标签选择器的样式一旦声明,页面中的所有同名标签的元素都会产生变化。例如,若声明 p{color:red;} ,则页面中所有的<p>元素都显示红色。如果希望其中某些<p>元素不是红色,而是另外一种颜色,这就需要将某些<p>元素定义为一类,使用类选择器选中这一类元素。

类选择器能够把相同的元素分类定义成不同的样式。类选择器以半角“.”开头,且类名称的第一位不能为数字,如图 3-4 所示。引用类选择器时为标签设置属性 class ="类名称"。

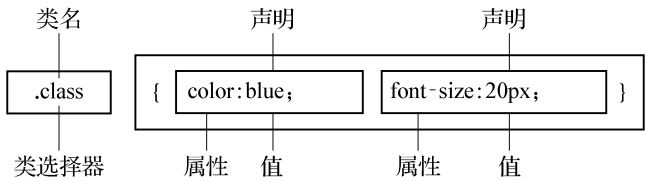


图 3-4 CSS 类选择器

【例 3-5】 类选择器的应用。代码如下,页面显示效果如图 3-5 所示。

```
1. <!DOCTYPE html>
2. <html>
3.     <head>
4.         <meta charset="utf-8">
5.         <title>类选择器</title>
6.         <style type="text/css">
7.             .one{
```

/* 类选择器 */

南京大学出版社
版权所有

```
8.             color:red;
9.             font-style:italic;}
10.            .two{                               /*类选择器*/
11.                text-indent:2em;}
12.        </style>
13.    </head>
14.    <body>
15.        <h3 class="one"> h3 标记,应用第一种类选择器样式</h3>
16.        <h4 class="one"> h4 标记,应用第一种类选择器样式</h4>
17.        <p class="one"> p 标记,应用第一种类选择器样式</p>
18.        <p class="two"> p 标记,应用第二种类选择器样式</p>
19.        <p class="one two"> p 标记,同时应用两种类选择器样式</p>
20.    </body>
21. </html>
```

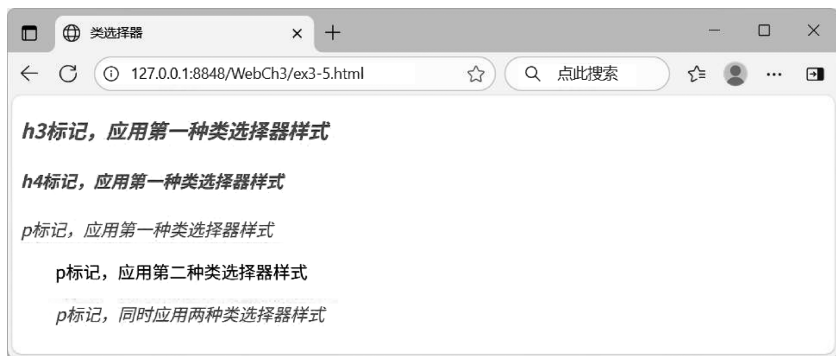


图 3-5 类选择器的应用

代码中第 6—12 行定义了 one 和 two 类选择器,第 15—17 行将标记 h3、标记 h4 和标记 p 都定义为同一类,类名为“one”,因此 h3 标签、h4 标签和第一个 p 标签中的文本颜色都为红色、斜体;第 18 行标签 p 定义为“two”类,因此首缩进 2em;第 19 行 p 标记使用 class = "one two"两种样式,文本为红色、斜体且段首缩进 2em。

3.4.3 id 选择器

id 选择器的使用方法与 class 选择器基本相同,不同之处在于一个 id 选择器只能应用于 HTML 文档中的一个元素,因此其针对性更强,而 class 选择器可以应用于多个元素。

id 选择器以半角“#”开头,且 id 名称的第一位不能为数字,如图 3-6 所示。

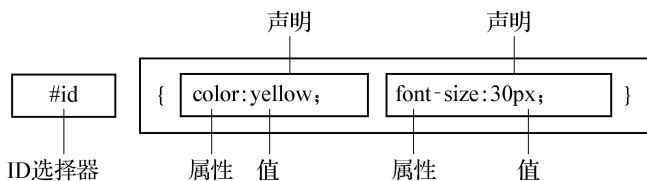


图 3-6 CSS id 选择器

【例 3-6】 id 选择器的应用。代码如下,页面显示效果如图 3-7 所示。

```
1. <!DOCTYPE html>
2. <html>
3.     <head>
4.         <meta charset="utf-8">
5.         <title> id 选择器</title>
6.         <style type="text/css">
7.             #one{ color:red;}           /* id 选择器 */
8.             #two{ font-style:italic;}    /* id 选择器 */
9.         </style>
10.    </head>
11.    <body>
12.        <h4 id="one"> h4 标记,应用第一种 id 选择器样式</h4>
13.        <p id="two"> p 标记,应用第二种类选择器样式</p>
14.    </body>
15. </html>
```

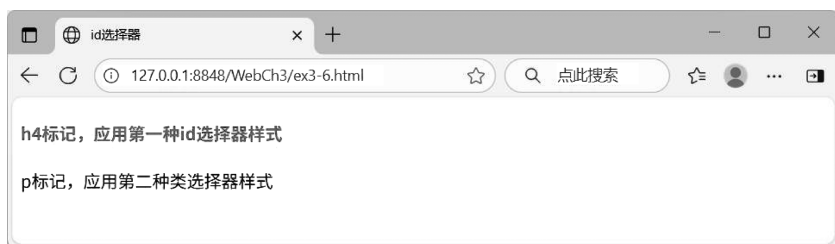


图 3-7 CSS id 选择器

代码中第 6—9 行定义了“one”和“two”两个 id 选择器,第 12 行应用了名为“one”的 id 选择器,第 13 行应用了名为“two”的 id 选择器,符合一个 id 选择器只能应用在一个元素上的规定。

id 与类选择器的重要区别:类可以复用,而每个 id 名只能被使用一次。id 选择器很多时候被用来标识标签的唯一身份,脚本或程序控制标签时用 id 可以很直接的找到。

3.4.4 通配符选择器

通配符选择器是功能最强大的选择器,它使用一个(*)号指定,它的作用是匹配 HTML 中所有标签。

比如,*{color:red;}可以将 HTML 中任意标签字体颜色全部设置为红色,通配符选择器虽然功能很强大,但除了必须对网页进行完全统一的样式设置时,一般很少用到。

3.4.5 实例:CSS 样式仿 LOGO 设计

【例 3-7】 综合实例,设计如图 3-8 所示的谷歌 LOGO。

(1) 案例分析

谷歌 LOGO 每个字母的颜色不同,但文字的字体和大小相同,在制作时应注意如何提高代码的效率,简化代码量。



图 3-8 综合实例

(2) 实现步骤

- 在 HTML 当中输入“Google”字母,因每个字母都要设置 CSS 样式,因此每个字母需要写到独立的 span 标签中,见代码 17—22 行。
- 插入内部样式表,使用标签选择器设置相同部分的 CSS 样式,设置文字大小为 120px,字体为 arial,见代码 7—9 行。
- 给不同的字母设置相应的颜色,这里有些字母颜色相同,如首字母“G”和单词中的“g”都为蓝色,使用 id 选择器来设置显然不合适。为了代码可以复用,使用类选择器,见代码 10—13 行。
- 给 span 添加 class 定义类名,颜色相同的字母,类名相同即可,见代码 17—22 行。

(3) 代码如下

```
1.  <!DOCTYPE html >
2.  <html >
3.      <head>
4.          <meta charset="utf - 8">
5.          <title></title>
6.          <style>
7.              span{
8.                  font - size: 120px;
9.                  font - family: arial;}
10.             .blue{color: blue;}
11.             .red{color:red;}
12.             .yellow{color: yellow;}
13.             .green{color: green;}
14.          </style>
15.      </head>
16.      <body>
17.          <span class ='blue'> G </span>
18.          <span class ='red'> o </span>
19.          <span class ='yellow'> o </span>
20.          <span class ='blue'> g </span>
21.          <span class ='green'> l </span>
```

```
22.         <span class='red'> e </span>
23.     </body>
24. </html>
```

3.5 CSS 派生选择器

在网页制作过程中只有几个基本的选择器不足以高效率的完成代码的编写,因此,CSS 还提供了一些由基本选择器派生的选择器。

3.5.1 后代选择器

后代选择器又称为包含选择器,可以选择某元素的后代元素。后代选择器的写法是把外层的标记写在前面,内层的标记写在后面,之间用空格隔开。例如,ul li 选择器表示“作为 ul 元素后代的任何 li 元素”。

【例 3-8】 后代选择器的应用。代码如下,页面显示效果如图 3-9 所示。

```
1.  <!DOCTYPE html>
2.  <html>
3.      <head>
4.          <meta charset="utf-8">
5.          <title></title>
6.          <style type="text/css">
7.              ul li ol li{font-style:italic;}
8.              ul li.special{font-weight:bold;}
9.          </style>
10.     </head>
11.     <body>
12.         <ul>
13.             <li>计算机工程学院
14.                 <ol>
15.                     <li class="special">计算机科学与技术专业</li>
16.                     <li>软件工程专业</li>
17.                     <li>物联网工程专业</li>
18.                 </ol>
19.             </li>
20.             <li>交通工程学院
21.                 <ol>
22.                     <li>交通运输专业</li>
23.                     <li>交通工程专业</li>
24.                     <li>车辆工程专业</li>
25.                 </ol>
26.             </li>
27.         </ul>
28.     </body>
```

29. </html>

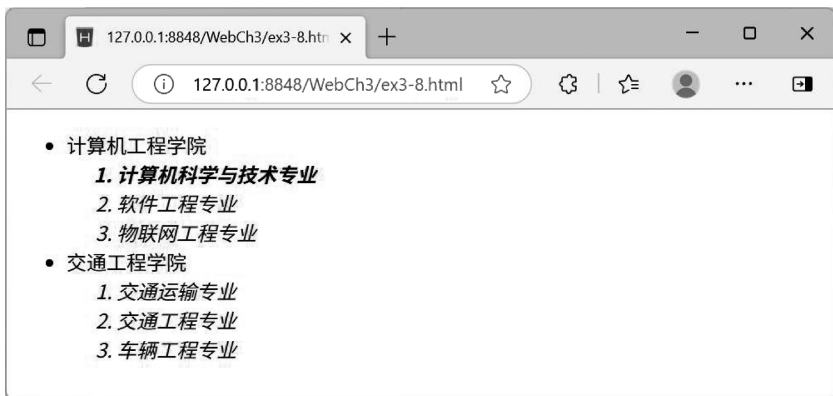


图 3-9 后代选择器

代码中第 6—9 行定义了“ul li ol li”后代选择器,可以将无序列表中的有序列表的 li 列表项文本定义为斜体显示,“ul li.special”后代选择器可以将 ul 标记中类为 special 的 li 标记中的文本加粗显示。

注意:使用后代选择器可以减少对 class 或 id 的声明。因此通常只给外层标记定义 class 或 id,内层标记可以通过后代选择器选中。

3.5.2 伪类选择器

CSS 伪类用于向某些选择器添加特殊的效果,如鼠标悬停或单击某元素。常用的伪类有 link(链接)、visited(已访问的链接)、hover(鼠标悬停状态)和 active(激活状态)。

伪类选择器必须指定标记名,标记和伪类之间用冒号“:”分隔,例如:

- a:link{color:#FF0000} /*未访问的链接 */
- a:visited{color:#00FF00} /*已访问的链接 */
- a:hover{color:#FF00FF} /*鼠标移动到链接上 */
- a:active{color:#0000FF} /*选定的链接 */

其中,前面两种称为链接伪类,只能应用于链接(a)元素,后两种称为动态伪类,一般任何元素都支持动态伪类,例如,li:hover、img:hover、div:hover 和 p:hover。

为了确保每次鼠标经过文本时的效果都相同,建议在定义样式时一定要按照 a:link、a:visited、a:hover 与 a:active 的顺序依次书写。

注意:在 CSS 定义中,a:hover 必须被置于 a:link 和 a:visited 之后,才是有效的。a:active 必须被置于 a:hover 之后,才是有效的。

3.5.3 实例:鼠标控制文字样式

【例 3-9】 综合实例,效果如图 3-10 所示。

(1) 案例分析

这是一个类似新闻类网站导航栏的效果,本案例中需要设置超链点击前的默认样式,突出显示的链接样式和鼠标移入时的链接样式。

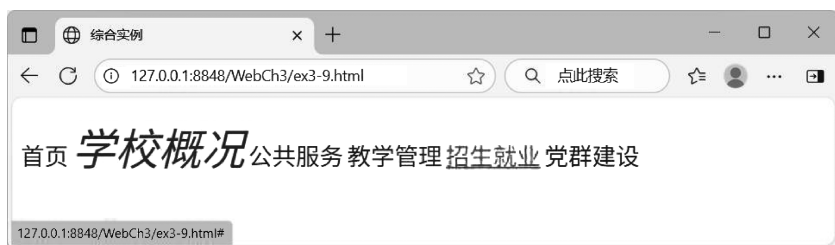


图 3-10 综合实例

(2) 实现步骤

- 新建 html 文件,输入段落标记和超链接及相关文本内容,见代码 25—33 行;
- 设置超链接访问前的样式,因访问后和访问前相同,我们可以使用合并样式的方式同时设置;设置超链接状态的文本颜色为深灰色,字体大小为 22px,并设置文本修饰为 none,去掉了下划线样式,见代码 7—10 行;
- 为需要突出显示的超链接设置类,定义相关的 CSS 样式,见 11—14 行;
- 设置鼠标移入超链接时的样式,显示下划线,修改文本颜色。最后将鼠标点击生效时的样式设置和访问前的样式相同;设置突出显示的链接样式为文本大小 40px,文本样式为斜体,文本颜色为橙色,见代码 15—21 行。

(3) 代码如下

```
1.  <! DOCTYPE html >
2.  <html >
3.      <head>
4.          <meta charset="utf - 8">
5.          <title>综合实例</title>
6.          <style type="text / css">
7.              a:link,a:visited{
8.                  color: #333;
9.                  font - size:22px;
10.                 text - decoration: none;}
11.             a.current{
12.                 font - size: 40px;
13.                 font - style: italic;
14.                 color: green;}
15.             a:hover{
16.                 color: orange;
17.                 text - decoration: underline;}
18.             a:active{
19.                 color: #333;
20.                 font - size:22px;
21.                 text - decoration: none;}
22.         </style>
23.     </head>
24.     <body>
```



```
25.         <p>
26.             <a href="#">首页</a>
27.             <a href="#"class="current">学校概况</a>
28.             <a href="#">公共服务</a>
29.             <a href="#">教学管理</a>
30.             <a href="#">招生就业</a>
31.             <a href="#">党群建设</a>
32.             <a href="#"></a>
33.         </p>
34.     </body>
35. </html>
```

注意:编写代码时,一定要遵循 link、visited、hover、active 的书写顺序。

3.6 CSS 单位

设置 CSS 属性值的难点在于单位的选用。它覆盖范围广,从长度单位到颜色单位,再到 URL 地址等。单位的取舍很大程度上依赖用户的显示器和浏览器,不恰当的使用单位会给页面布局带来很多麻烦,因此属性值的设置需要慎重考虑,合理使用。

CSS 中,单位可以划分为绝对单位和相对单位。

(1) 绝对单位

绝对单位在网页中很少使用,一般多用在传统平面印刷中,但在特殊的场合使用绝对单位很有必要。绝对单位包括英寸、厘米、毫米、磅和 pica。

- 英寸(in):是使用最广泛的长度单位。
- 厘米(cm):生活中最常用的长度单位。
- 毫米(mm):在研究领域使用广泛。
- 磅(pt):在印刷领域使用广泛,也称点。
- pica(pc):在印刷领域使用,1 pica 等于 12 磅,所以也称 12 点活字。

(2) 相对单位

相对单位与绝对单位相比显示大小不是固定的,它所设置的对象受屏幕分辨率、可视区域、浏览器设置以及相关元素的大小等多种因素影响。

- em 单位表示元素的字体高度,它能够根据字体的 font-size 属性值来确定单位的大小。例如:

```
p{
    font-size:12px;line-height:2em;}
```

代码中设置字体大小为 12px,行高为 2em,即是字体大小的 2 倍,所以行高为 24px。如果 font-size 的单位为 em,则 em 的值将根据父元素的 font-size 属性值来确定。

- ex 单位根据所使用的字体中小写字母 x 的高度作为参考。在实际使用中,浏览器将通过 em 的值除以 2 得到 ex 的值。
- px 单位是根据屏幕像素点来确定的。这样不同的显示分辨率就会使相同取值的 px

单位所显示出来的效果截然不同。实际设计中,建议网页设计师多使用相对长度单位 em,且在某一类型的单位上使用统一的单位。如设置字体大小,根据个人使用习惯,在一个网站中,可以统一使用 px 或 em。

- 百分比也是一个相对单位值。百分比值总是通过另一个值来计算,一般参考父对象中相同属性的值。例如,下面的代码设置段落的行高为字体高度的 150%。

```
p{
    font-size:12px; line-height:150 %;}
```

3.7 CSS 文字与文本属性

3.7.1 CSS 字体属性

CSS 字体属性主要设置字体类型、大小及粗细等样式,常用的字体属性如表 3-1 所示。

表 3-1 常用字体属性

属 性	说 明
font-size	设置字体的大小
font-style	设置字体的风格
font-family	设置字体名
font-weight	设置字体的粗细
font	设置文字各种属性的简捷方式

(1) 字体大小 font-size

font-size 属性用于设置文本字体的大小,其取值可以是绝对或者相对值,如表 3-2 所示。

表 3-2 font-size 取值

值	描 述
xx-small、 x-small、 small、 medium、large、x-large、xx-large	把字体的尺寸设置为不同的尺寸,从 xx-small 到 xx-large。默认值:medium
smaller	把 font-size 设置为比父元素更小的尺寸
larger	把 font-size 设置为比父元素更大的尺寸
length	把 font-size 设置为一个固定的值
%	把 font-size 设置为基于父元素的一个百分比值
inherit	规定应该从父元素继承字体尺寸

例如,下面的代码分别设置标签 p、标签 a 中文本字体的大小。

```
p{font-size:24px;}
a{font-size:larger;}
```

(2) 字体风格 font-style

该属性设置使用斜体(italic)、倾斜(oblique)或正常(normal)字体。

(3) 字体系列 font-family

font-family 用于设置字体名称,如宋体、黑体、Arial 等。基本语法:font-family: 字体 1, 字体 2,字体 3,...,字体 n。浏览器依次查找字体,只要存在就使用。不存在继续找下去,直到最后一个,如果都不存在使用默认字体。示例如下:

```
p{
    font-family:"Times New Roman",Times,serif; }
```

注意:若属性的某个值不是一个单词,则值要加引号,例如,上面代码中的:"Times New Roman";若为一个属性设置多个候选值,则每个值之间用逗号分隔。

(4) font-weight 属性

font-weight 属性用于设置字体的粗细,实现对一些字体的加粗显示。语法:font-weight:字体粗度值。font-weight 属性取值如表 3-3 所示。

表 3-3 font-weight 取值

值	描 述
normal	缺省值,定义标准的字符
bold	定义粗体字符
bolder	定义更粗的字符
lighter	定义更细的字符
100、200、300、400、500、600、700、800、900	定义由粗到细的字符。400 等同于 normal,而 700 等同于 bold

(5) font 属性

font 可以同时设置字符的各种属性,包括 font-style、font-weight、font-size、line-height、font-family。各属性间用空格隔开。如果同时设置 font-size 和 line-height,这两属性值间以“/”隔开。

例如,将标记 p 中的文本设置为隶书、字号 12 像素、斜体、加粗、行高 30 像素。

```
p{
    font:italic bold 12px / 30px 隶书; }
```

3.7.2 CSS 文本属性

CSS 文本属性可定义文本的外观。通过文本属性,可以改变文本的颜色、字符间距,对齐文本,装饰文本,对文本进行缩进等。常用的文本属性如表 3-4 所示。

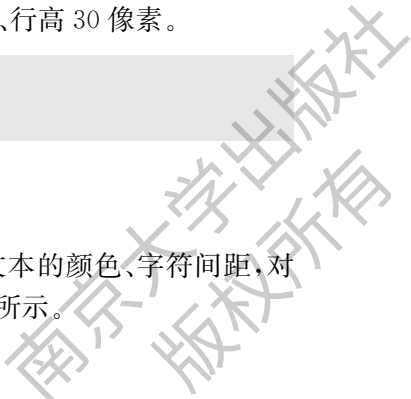


表 3-4 常用文本属性

属 性	说 明
color	设置文本颜色
text-align	对齐方式
text-decoration	文本的修饰
text-indent	首行文本缩进
line-height	设置行高
text-transform	控制文本的大小写
letter-spacing	字符间距
word-spacing	设置单词间距

(1) 文本颜色 color

color 属性用于设置文本的颜色，所有浏览器都支持 color 属性。语法:color: 颜色代码;。颜色取值可以是颜色关键字(如 yellow)，也可以是 RGB 表示或十六进制表示。

● 颜色关键字

例如，“red”为颜色关键字，可以被 CSS 识别的颜色约 140 种。

```
p{
    color:red; }
```

● RGB 表示

可以通过设置 RGB 三色的值描述颜色。使用 rgb(r,g,b)或 rgb(r %,g %,b %)。前者参数的取值为 0~255 之间的整数，后者是介于 0 到 100 之间的整数。其中,r 表示红色分量值,g 表示绿色分量值,b 表示蓝色分量值，数值越小，亮度越低，数值越大，亮度越高。

例如,RGB(0,0,0)为黑色(亮度最低),RGB(255,255,255)为白色(亮度最高),RGB(255,0,0)为红色。

例如,将 p 标记中的文本颜色设置为蓝色。

```
p{
    color:rgb(0,0,255); }
```

● 十六进制表示

在实际应用中,最常使用十六进制设置颜色值,是将 RGB 颜色数值转换成十六进制的数字,表示方式为:#rrggbb 或 #rgb,其取值范围是 00~FF 或 0~F,对应十进制的范围是 0~255。

例如:

```
h1{color:#00ff00;} /* 将 h1 标记中的文本颜色设置为绿色 */
p{color:#f00;} /* 将 p 标记中的文本颜色设置为红色 */
```

(2) 文本水平对齐方式 text-align

text-align 属性设置元素中的文本的水平对齐方式,取值如下:

- left: 默认值, 文本左对齐。
- right: 文本右对齐。
- center: 文本中间对齐。
- justify: 文本两端对齐。

(3) 文本装饰 text-decoration

text-decoration 属性主要用于对文本进行修饰,如设置文本是否有上划线、下划线或删除线,取值如下:

- none: 默认值, 为标准文本。
- underline: 文本有下划线。
- overline: 文本有上划线。
- line-through: 文本中间有一条删除线。
- blink: 表示文字闪烁效果,这一属性值只有在 Netscape 浏览器中才能正常显示。

例如,下面的规则可以去除 a 标签默认的字符下划线。

```
a {  
    text-decoration: none;}
```

(4) 文本首行缩进 text-indent

text-indent 属性用于设置 HTML 中块级元素(如 p、div)第一行的缩进数量,常用于设置段落的首行缩进。text-indent 可以使用所有长度单位,包括百分比值。

例如,下面的规则设置所有 p 标签首行缩进 2 个字符。

```
p{  
    text-indent: 2em;  
}
```

(5) 行高 line-height

line-height 用于控制一行文本的高度,类似 word 文档中的行距。

例如,下面的规则使所有 p 标签的行间距为 2 倍(字号大小)的间距。

```
p{  
    font-size: 14px;  
    text-height: 2em;}
```

(6) text-transform 属性

text-transform 属性可以控制文本的大小写,取值如下:

- none: 默认值。
- uppercase: 设置文本中所有字母为大写。
- lowercase: 设置文本中所有字母为小写。
- capitalize: 设置文本中的每个单词首字母为大写。

例如,下面的规则使 h1 标签中所有字母大写。

```
h1{
    text - transform: uppercase;}
```

(7) letter-spacing 属性

letter-spacing 属性可以设置文本中字符间的距离,该属性的设置多用于英文文本。

例如,下面的规则使 h1 标记的字符间距为 2px,h4 标记的字符间距设置为-3px,字符之间会出现相互重叠的效果。

```
h1{letter - spacing: 2px;}
h2{letter - spacing: - 3px;}
```

(8) word-spacing 属性

word-spacing 属性可以设置文本中单词之间的距离,该属性的设置多用于英文文本。

【例 3 - 10】 CSS 文字与文本属性应用示例。代码如下,页面显示效果如图 3 - 11 所示。

```
1.  <!doctype html >
2.  <html >
3.    <head>
4.      <meta charset ="UTF - 8" >
5.      <title>文字与文本属性</title>
6.      <style type ="text / css" >
7.          p{font-family:隶书,微软雅黑; }
8.      </style>
9.    </head>
10.   <body>
11.     <h3> CSS </h3>
12.     <h6> CSS 文字与文本属性</h6>
13.     <p style ="color: red;" > color 属性用于设置文本的颜色。</p>
14.     <p style ="text - align: right;" > text - align 属性设置元素中的文本
      的水平对齐方式</p>
15.     <p style ="text - decoration: underline" > text - decoration 属性主
      要用于对文本进行修饰</p>
16.     <p style ="text - indent: 2em" > text - indent 属性用于设置 HTML 中块
      级元素的第一行的缩进数量</p>
17.     <p style ="line - height: 2em" > line - height 用于控制一行文本的高度</
      p>
18.   </body>
19. </html >
```



图 3 - 11 文字与文本设置显示效果

3.8 CSS 列表属性

CSS 列表属性设置列表项的样式,包括符号、缩进等。list-style-type 属性用于设置列表项标记的类型,list-style-position 属性用于设置列表项标记的位置,list-style-image 属性用于将图像设置为列表项标记。

常用的列表属性如表 3 - 5 所示。

表 3 - 5 列表属性

属 性	属性值	描 述
list-style-type	none	无符号
	disc	实体的小圆点
	circle	空心的小圆点
	square	实心的小方块
	decimal	1,2,3...
	lower-roman	i,ii,iii...
	upper-roman	I,II,III...
	lower-alpha	a,b,c,d,e...
	upper-alpha	A,B,C,D,E...
list-style-position	inside	清单项目较往右移
	outside	清单项目正常显示
list-style-image	URL	使用图像作为列表项目符号
	none	不指定图像

【例 3-11】 使用 CSS 字体属性、文本属性、背景属性以及列表属性对文章排版,代码如下,页面显示效果如图 3-12 所示。

```
1.  <html>
2.      <head>
3.          <meta charset="UTF-8">
4.          <style type="text/css">
5.              body{
6.                  background:url(img/bg.jpg) no-repeat top right;}
7.              h2{
8.                  color:#333;
9.                  text-align:center;}
10.             h4{
11.                 text-align:right;
12.                 font-style:italic;
13.                 text-decoration:underline;
14.                 text-transform:capitalize;
15.                 word-spacing:12px;}
16.             p{
17.                 font-family:黑体, 隶书;
18.                 text-indent:2em;
19.                 line-height:150%;
20.                 color:#333;
21.                 font-size:14px; }
22.             li{
23.                 font-size:12px;
24.                 list-style-image:url(../img/arrow.gif); }
25.         </style>
26.     </head>
27.     <body>
28.         <h2> CSS 的特点</h2>
29.         <h4> cascading style sheets</h4>
30.         <p>样式通常保存在外部的 .css 文件中。通过仅仅编辑一个简单的 CSS 文档,外部样式表使你有能力同时改变站点中所有页面的布局 and 外观。</p>
31.         <p>在页面中插入样式表的方法有:</p>
32.         <ul>
33.             <li>行内样式表</li>
34.             <li>内部样式表</li>
35.             <li>链入外部样式表</li>
36.             <li>导入外部样式表</li>
37.         </ul>
38.     </body>
39. </html>
```

background 简写属性浏览器右上角不重复显示图片

h4 中的内容右对齐、斜体显示、加下划线 …

列表标记为动态图片



图 3-12 CSS 显示效果

3.9 CSS 表格属性

使用 CSS 表格属性,可以极大地改善表格的外观。常用的表格属性如表 3-6 所示。

表 3-6 表格属性

属 性	描 述
border	设置表格边框
border-collapse	设置是否把表格边框合并为单一的边框
width	设置表格的宽度
height	设置表格的高度
text-align	文本水平对齐方式
vertical-align	文本垂直对齐方式
padding	表格中内容与边框的距离

3.9.1 表格边框

可以使用 border 属性设置表格的边框,border 属性的用法在之前已经介绍过。

【例 3-12】 在下面的代码中使用并集选择器“table,th,td”设置表格以及单元格的边框,页面显示效果如图 3-13 所示。

```
1. <!DOCTYPE html>
2. <html>
3.     <head>
4.         <meta charset="utf-8">
5.         <style type="text/css">
6.             table,th,td{
7.                 border:1px solid blue;
```

南京大学出版社
版权所有

```
8.         text-align:center;}
9.         table{width:300px;height:100px;}
10.        caption{
11.            font-weight:bold;
12.            font-size:20px;}
13.        </style>
14.    </head>
15.    <body>
16.        <table>
17.        <caption>教材一览表</caption>
18.        <tr>
19.            <th>书名</th>
20.            <th>作者</th>
21.            <th>价格</th>
22.        </tr>
23.        <tr>
24.            <td> Java Web 应用开发</td>
25.            <td>李永飞</td>
26.            <td> 86 元</td>
27.        </tr>
28.        <tr>
29.            <td> Web 编程基础</td>
30.            <td>任平红</td>
31.            <td> 82 元</td>
32.        </tr>
33.        <tr>
34.            <td> Vue.js 3.0 从入门到精通</td>
35.            <td>李小威</td>
36.            <td> 70 元</td>
37.        </tr>
38.        </table>
39.    </body>
40. </html>
```



图 3-13 应用 border 属性

3.9.2 折叠边框

在上面的例子中,表格具有双线条边框,这是由于 table、th 以及 td 元素都有独立的边框。如果需要把表格显示为单线条边框,可以使用 border-collapse 属性。在上面的例子中添加以下代码,页面显示效果如图 3-14 所示。

```
table {  
    border-collapse: collapse; }
```



图 3-14 应用 border-collapse 属性

3.10 实例演练

【例 3-13】 制作如图 3-15 所示的隔行变色的表格。

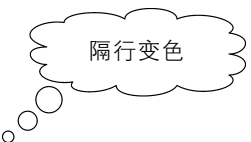
网页中经常使用表格展示数据,当数据表格的行比较多,用户查看数据时极易看错行,这时可以使用隔行变色显示。为奇数行或偶数行添加一个类,并定义该类的背景色即可实现隔行变色,代码如下:

```
1. <html>  
2.     <head>  
3.         <meta charset="UTF-8">  
4.         <style type="text/css">  
5.             #customers  
6.             {  
7.                 font-family:"Trebuchet MS", Arial;  
8.                 width:100 %;  
9.                 border-collapse:collapse;  
10.            }  
11.            #customers td, #customers th  
12.            {  
13.                font-size:1em;  
14.                border:1px solid #98bf21;
```

折叠边框

南京大学出版社
版权所有

```
15.         padding:3px 7px 2px 7px;
16.     }
17.     #customers th
18.     {
19.         font-size:1.1em;
20.         text-align:left;
21.         padding-top:5px;
22.         padding-bottom:4px;
23.         background-color:#A7C942;
24.         color:#ffffff;
25.     }
26.     #customers tr.alt td
27.     {
28.         color:#98bf21;
29.     }
30. </style>
31. </head>
32. <body>
33.     <table id="customers">
34.         <caption>教材一览表</caption>
35.         <tr>
36.             <th>书名</th>
37.             <th>作者</th>
38.             <th>价格</th>
39.         </tr>
40.         <tr class="alt">
41.             <td>Java Web 应用开发</td>
42.             <td>李永飞</td>
43.             <td>86 元</td>
44.         </tr>
45.         <tr>
46.             <td>Web 编程基础</td>
47.             <td>任平红</td>
48.             <td>82 元</td>
49.         </tr>
50.         <tr class="alt">
51.             <td>Vue.js 3.0 从入门到精通</td>
52.             <td>李小威</td>
53.             <td>70 元</td>
54.         </tr>
55.     </table>
56. </body>
57. </html>
```



隔行变色



图 3-15 隔行变色表格

3.11 思政融入

1. 工匠精神与细节意识：于细微处见真章

网页样式的精良与否,往往体现在像素级的对齐、颜色对比的舒适度、字体间距的合理性等细节之中。一个微小的 margin 偏差可能破坏整体布局,一段未重置的默认样式可能导致跨浏览器显示异常。通过强调 CSS 代码的规范性、选择器的合理性以及样式兼容性的测试,养成严谨细致的工作作风,杜绝“差不多”心态。

2. 自主创新与标准引领：增强技术自主的责任感

我国一些知名企业也积极参与 CSS 新特性的提案与实现,国产浏览器(如 360、QQ 浏览器、华为浏览器)在标准兼容性方面持续进步,部分团队还推出了自主渲染引擎和 UI 框架。在未来的职业发展中,增强对技术自主可控的认知,不仅做标准的使用者,更努力成为标准的参与者和推动者。

3.12 本章小结

本章重点介绍的是 CSS 控制网页内容常用的 CSS 属性,包括字体属性(font)、文本属性(text)、背景属性(background)、列表属性(list-style)和表格属性(table)。

本章知识点如图 3-16 所示。

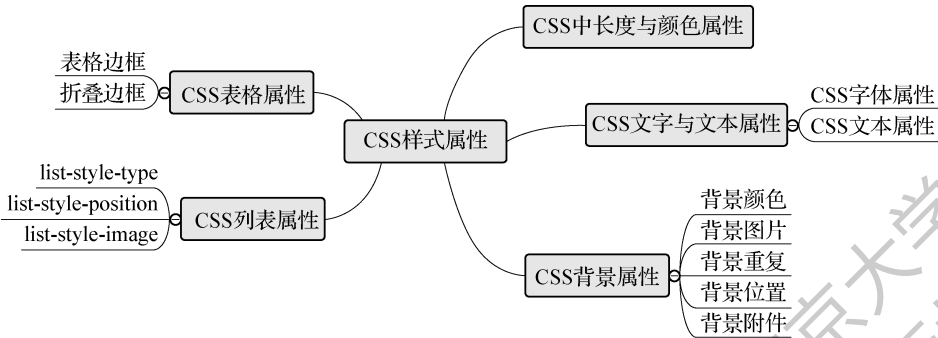


图 3-16 CSS 样式属性思维导图

3.13 拓展训练

(1) CSS 是()的缩写。

A. Colorful Style Sheets

B. Computer Style Sheets

C. Cascading Style Sheets

D. Creative Style Sheets

(2) 下列()是定义样式表的正确格式。

A. {body;color = black(body)}

B. body;color = black

C. body {color: black}

D. {body;color:black}

(3) 下列()是定义样式表中的注释语句。

A. /*注释语句 */

B. // 注释语句

C. // 注释语句 //

D. '注释语句

(4) 表示内嵌样式的元素是()。

A. <style>

B. <css>

C. <script>

D. <link>

(5) 如果要在不同的网页中应用相同的样式表定义,应该()。

A. 直接在 HTML 的元素中定义样式表

B. 在 HTML 的<head>标记中定义样式表

C. 通过一个外部样式表文件定义样式表

D. 以上都可以

(6) CSS 中的选择器不包括()。

A. 超文本标记选择器

B. 标签选择器

C. ID 选择器

D. 类选择器

(7) 样式表定义#title {color:red} 表示()。

A. 网页中的标题是红色的

B. 网页中某一个 id 为 title 的元素中的内容是红色的

C. 网页中元素名为 title 的内容是红色的

D. 以上任意一个都可以

(8) 样式表定义.outer {background-color:red} 表示()。

A. 网页中某一个 id 为 outer 的元素的背景色是红色

B. 网页中含有 class="outer"元素的背景色是红色

C. 网页中元素名为 outer 元素的背景色是红色

D. 以上任意一个都可以

(9) 下面选项中是合法的类样式的是()。

A. .word

B. #word

C. word

D. \$ word

(10) 以下关于 class 和 id 的说法错误的是()。

A. class 的定义方法是:.类名{样式};

B. id 的应用方法:<指定标签 id="id 名">

C. class 的应用方法:<指定标签 class="类名">

D. id 和 class 只是在写法上有区别,在应用和意义上没有区别

(11) 下列选项中不属于 CSS 文本属性的是()。

南京大学出版社
版权所有

- A. font-size
C. line-height
(12) 下列()表示 p 元素中的字体是粗体。
A. p {text-size: bold}
C. <p style="text - size:bold">
(13) 下列哪个 CSS 属性可以更改字体大小()。
A. text-size
C. text-style
(14) 下列哪个 CSS 属性能够更改文本字体()。
A. text
C. font-family
(15) 下列()表示 a 元素中的内容没有下划线。
A. a{text-decoration:no underline}
B. a{underline:none}
C. a{text-decoration:none}
D. a{decoration:no underline}
(16) 下面哪个 CSS 属性是用来更改背景颜色的()。
A. back
C. color
(17) 怎样给所有的<h1>标签添加背景颜色()。
A. h1 {color:#FFFFFF}
B. h1 {background - color:#FFFFFF;}
C. #h1 {background - color:#FFFFFF}
D. h1.all {background - color:#FFFFFF}
(18) 以下对列表的描述,()是不正确的。
A. list-style-type
C. list-style-position
(19) 在 CSS 语言中,下列()是“列表项标号图像”的语法。
A. width: <值>
C. list - style - image: <值>
(20) 以下说法错误的是()。
A. list-style-type:square 表示列表项符号是小方块
B. list-style-color 不是列表属性
C. 设置 table 标签的 border 属性,默认可以看到两层边框线
D. 可以使用 table-collapse 属性将表格边框和单元格边框重合在一起显示
- B. text-align
D. text-decoration
B. p {font-weight:bold}
D. <p style="font - size:bold">
B. font-size
D. font-style
B. line-height
D. text-decoration
B. a{underline:none}
D. a{decoration:no underline}
B. bgcolor
D. background-color
B. h1 {background - color:#FFFFFF;}
D. h1.all {background - color:#FFFFFF}
B. list-style-color
D. list-style-image:url
B. height: <值>
D. list - style - picture: <值>



【微信扫码】
本章参考答案 & 相关资源