高等院校计算机应用系列教材

Web 前端开发技术

胡　斌　谢玫秀　刘超超　主　编
黄旭义　王晓涵　熊诗颜　谭　淞　副主编

清華大學出版社
北　京

内 容 简 介

本书系统讲解了 Web 前端开发技术的基础知识，全面覆盖了 HTML、CSS 与 JavaScript 三大 Web 前端开发核心技术。本书内容从基础的 HTML、CSS 延伸至 JavaScript，再到 DOM 和 BOM 对象，同时结合 HTML5、CSS3 及现代 Web 开发实践，体系完整且覆盖面广，能帮助读者从零开始构建完整的Web 开发知识体系。

本书共 15 章，采用“基础知识→核心语法→实战应用”的教学模式，循序渐进，能够让读者快速入门，并掌握 Web 前端开发的基础核心技术，也为进一步学习前端框架(如 Vue、React)和后端开发(如 Node.js)奠定坚实基础。

本书适合作为高等院校计算机科学与技术、软件工程、网络工程、数据科学与大数据技术、数字媒体技术、电子商务等计算机及相关专业的 Web 前端开发教材；同时，也适合 Web 前端开发初学者用于自学；此外，还可作为其他 IT 相关人员学习 Web 前端开发的参考用书。

图书在版编目(CIP)数据

Web 前端开发技术 / 胡斌, 谢玫秀, 刘超超主编.
北京 : 清华大学出版社, 2026. 5. -- (高等院校计算机应用系列教材). -- ISBN 978-7-302-71281-7

Ⅰ. TP312.8；TP393.092.2

中国国家版本馆 CIP 数据核字第 2026SP2318 号

责任编辑：刘金喜
封面设计：高娟妮
版式设计：思创景点
责任校对：成凤进
责任印制：宋　林

出版发行：清华大学出版社
网　　址：https://www.tup.com.cn，https://www.wqxuetang.com
地　　址：北京清华大学学研大厦 A 座　　邮　　编：100084
社 总 机：010-83470000　　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

印 装 者：北京瑞禾彩色印刷有限公司
经　　销：全国新华书店
开　　本：185mm×260mm　　**印　　张：**20.75　　**字　　数：**558 千字
版　　次：2026 年 5 月第 1 版　　**印　　次：**2026 年 5 月第 1 次印刷
定　　价：78.00 元

产品编号：111489-01

前　言

2023 年 Stack Overflow 开发者调查报告显示，JavaScript 连续十年成为最常用的编程语言，HTML/CSS 稳居技术使用率前三甲，这一数据充分证明了 Web 前端技术在现代软件开发中的基础性地位。同时，从简单的静态网页到复杂的单页应用，从 PC 端到移动端，前端技术正持续不断地演进，对开发者的专业能力要求也越来越高。为帮助读者(尤其是普通本科在校生)快速入门 Web 前端开发领域，扎实掌握 Web 前端开发的基础核心技术，为后续学习前端框架(如 Vue、React)和后端开发(如 Node.js)奠定坚实基础，我们编写了本教材。

编写背景与目的

在数字化时代，Web 应用已深度渗透到人们生活的方方面面。无论是电子商务平台、在线教育系统，还是社交网络软件、企业管理平台，都离不开优质的 Web 界面提供支持。作为连接用户与后端服务的关键“桥梁”，前端开发的重要性日益凸显。然而，面对日新月异的前端技术栈，许多初学者(特别是普通本科在校生)往往感到无从下手，难以找到清晰的学习方向。本教材的编写，旨在为 Web 前端开发学习者提供一条清晰、系统的学习路径。我们摒弃了市面上部分教材“求全求新”的片面做法，转而回归技术本质，着重培养读者扎实的 HTML、CSS 和 JavaScript 基本功。这些基础技术不仅是进一步学习前端框架和后端开发的基本前提，更是成为优秀前端开发者的必经之路。

主要编写特色

- 内容新颖全面：本书全面覆盖 Web 前端开发的 HTML、CSS 和 JavaScript 三大核心技术，从基础的 HTML 语法、CSS 样式设计到 JavaScript 编程，再到 DOM(文档对象模型)和 BOM(浏览器对象模型)操作，以及最新的 HTML5 和 CSS3 特性及现代 Web 开发实践，内容兼具新颖性与全面性。
- 思政元素丰富：本书融入了传统文化、大好河山、我爱家乡等众多主题的思政元素及相关案例，在传授技术知识的同时，引导读者树立正确价值观。
- 项目驱动学习：每章均配有精心设计的实战案例，最后一章还专门提供了一个综合项目，帮助读者将各章节所学知识融会贯通，提升实战应用能力。
- 讲解图文并茂：书中使用大量图表对知识点进行归纳与分析，直观呈现技术原理与应用方法，有效提高学习效率与教学效果。
- 代码统一规范：全书提供风格统一、格式规范的源代码，帮助读者从入门阶段就养成良好的编程习惯，为后续职业发展打下基础。

- 丰富的教学资源：本书配套提供教学大纲、电子教案、PPT 课件、源代码和习题答案等资源，既方便教师开展教学工作，也便于读者自主学习。

主要编写内容

本书共分为 15 章，内容编排严格遵循由浅入深、循序渐进的原则，具体分为以下四个部分。

第一部分(第 1～7 章)：聚焦 HTML 基础，从 Web 技术概述开始，逐步讲解 HTML 文档基本结构、文本格式化标签、超链接应用、列表设计、表格制作、表单开发等核心内容。与同类教材相比，我们特别强化了语义化 HTML 的讲解，帮助读者建立符合行业标准的编码习惯。

第二部分(第 8～11 章)：深入讲解 CSS 技术，不仅涵盖选择器、盒模型等基础概念，还详细介绍了 Flexbox、Grid 等现代网页布局技巧。本部分采用“原理+案例”的教学方式，让读者在理解 CSS 工作机制的同时，快速掌握实际应用能力。

第三部分(第 12 章)：重点介绍 HTML5 的发展历程和核心基础知识，包括 HTML5 文档结构优化、HTML5 新增表单元素与属性，以及 HTML5 视频与音频的嵌入与控制方法。

第四部分(第 13～15 章)：系统讲解 JavaScript 编程，涵盖 JavaScript 基础语法、变量与数据类型、函数与对象、DOM 操作、事件处理等核心内容，为读者构建完整的前端交互开发知识体系。

教学资源

为方便教师开展教学工作与读者自主学习，本书提供了大量配套资源与实例代码。书中所有教学案例均采用统一格式命名，例如，“exam2_1_1”表示第 2 章 2.1 节的第 1 个案例；每章资源以独立子目录形式存放，例如，“ch2”目录存放第 2 章的教学资源，包含该章所有教学案例、图片素材、音视频文件等。本书提供的辅助教学资源主要包括：

(1) 详细的教学大纲；
(2) 一套完整的教学 PPT 课件；
(3) 系统的电子教案；
(4) 一套完整的教学案例源代码；
(5) 一套完整的教学与实验所需图片、文字、音视频素材；
(6) 一套完整的作业练习及参考答案。

上述资源可通过扫描下方二维码下载。

教学大纲　PPT 课件　电子教案

案例源代码　实验素材　习题与答案

本书的编写与出版得到了清华大学出版社及相关工作人员的大力支持，在此对他们表示衷心的感谢。同时，在编写过程中，作者参阅了大量 HTML、CSS、JavaScript、HTML5 及 CSS3 领域的相关书籍与主流网络媒体资源，在此也向这些书籍与资源的贡献者致以诚挚的谢意。

由于移动互联网技术发展迅速，且受作者自身水平所限，书中难免存在疏漏与欠妥之处，恳请各位专家与读者批评指正。

服务邮箱：476371891@qq.com

作　者

2026 年 1 月

目　录

第 1 章

概　述

1.1 Web 概述

Web(World Wide Web，万维网)是一个由无数网页和超文本链接组成的信息网络。它通过互联网连接全球的计算机系统，使得信息能够轻松地被访问、分享和交互。

Web 的初始设计目的是作为静态信息资源的发布平台，通过超文本标记语言(hyper text markup language，HTML)来描述信息资源，利用统一资源标识符(uniform resource identifier，URI)来定位信息资源，以及通过超文本传输协议(hyper text transfer protocol，HTTP)来请求信息资源，它们构成了 Web 的核心架构，是支撑 Web 运行的基础。随着时间的推移，Web 技术不断演进，从最初的静态页面发展到动态网页，再到现在的 Web 应用和移动互联网，其功能和应用范围不断扩大，已成为现代社会不可或缺的一部分。

简而言之，客户端(通常是浏览器)通过 URI 定位网站，发起 HTTP 请求，服务器接收到请求后返回 HTML 页面。由此可见，Web 是建立在 TCP/IP 协议之上的。TCP/IP 协议将计算机相互连接起来，而 Web 则在此协议族之上进一步将计算机的信息资源互联，从而形成了万维网。开发的 Web 应用本质上是能够提供信息或功能的 Web 资源，它们构成了这个全球超大规模分布式系统的一部分。随着技术的发展，Web 应用已不再局限于简单的信息展示，而是扩展到了电子商务、在线教育、社交媒体、云计算等多个领域，极大地丰富了人们的生活和工作方式。

1.1.1 Web 的起源

1989 年，Tim Berners-Lee(见图 1-1)在欧洲核子研究中心(CERN)工作时，提出了构建基于超文本的分布式系统来解决信息丢失问题的想法，并于 1990 年基于 NeXTSTEP 计算机开发出了世界上第一个 Web 服务器和第一个 Web 客户机，标志着 Web 的诞生。虽然最初的 Web 服务器功能简单，用户只能查询电话号码等信息，但它为后来的 Web 发展奠定了基础。

1991 年 8 月 6 日，Tim Berners-Lee 在 alt.hypertext 新闻组发布了一份关于 World Wide Web 的简要概述，这标志着 Web 页面首次在互联网上亮相，同年，第一个万维网 WWW 网站http://info.cern.ch/建成。Tim Berners-Lee 创建了构成 Web 基础的关键技术和标准，包括 HTML、HTTP 和 URL 等。HTML 用于定义网页的内容和结构；HTTP 是一种协议，用于在 Web 服务器和浏览器之间传输数据；URL 则提供了一种在 Internet 上定位资源的方法。

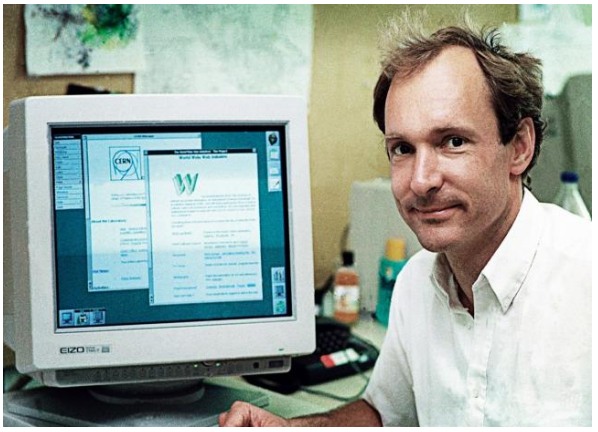

图 1-1　Tim Berners-Lee

为了让更多的人了解和使用 Web，Tim Berners-Lee 积极向 CERN 的同事和互联网社区展示他的发明，并于 1993 年正式将 Web 开放给公众。此后，他还创办了万维网联盟(World Wide Web consortium，W3C)，致力于制定 Web 技术标准，推动 Web 技术的标准化和互操作性，使得不同开发者开发的软件能够更好地协同工作。随着浏览器的普及和 W3C 的推动，Web 上可访问的资源逐渐增多，此时，Web 的主要功能是浏览器向服务器请求静态 HTML 信息。

Tim Berners-Lee 一直强调 Web 的开放性、去中心化和平等性，他认为 Web 应是一个任何人都可以访问和使用的信息共享平台，而不是被少数利益集团所控制。Tim Berners-Lee 的这种理念影响了 Web 的发展方向，使得 Web 成为全球范围内自由交流和信息传播的重要基础设施。

1.1.2　Web 的特点

1. Web 的图形化

Web 的图形化是其最显著的特点之一。与传统的纯文本界面相比，Web 能够在一个页面上同时展示色彩丰富的图形、图像、文本等多种元素，为用户提供了更加直观、生动的视觉体验。例如，新闻网站可以通过图片和视频展示新闻事件现场，让用户更直观地了解事件的情况；电商网站可以通过精美的产品图片和详细的产品介绍，增强用户的购买欲望。

除了丰富的视觉元素，Web 还具有易于导航的界面。通过超链接，用户可以方便地在不同的网页之间跳转，快速找到自己所需的信息。这种导航方式类似于图书馆的目录索引，用户可以根据自己的需求选择不同的链接，快速定位到相关的网页内容。同时，浏览器还提供了搜索功能、书签功能等辅助工具，进一步提高了用户的导航效率。

2. Web 的与平台无关性

Web 的另一个重要特点是与平台无关性。无论用户使用何种操作系统，如 Windows、Linux、MacOS 等，都可以通过浏览器访问 Web 上的各种资源。这是因为 Web 采用了统一的网络协议和标准，使得不同平台的设备能够相互通信和交互。例如，用户可以在 Windows 操作系统的计算机上访问一个网站，然后在手机上继续浏览相同的网站，而不会遇到兼容性问题。

浏览器是实现 Web 与平台无关性的关键软件，它作为用户与 Web 之间的“桥梁”，负责解析 HTML 等网页语言，并将网页内容呈现给用户。常见的浏览器有 Chrome、Firefox、Safari 等，它们

都具有跨平台的特性，可以在多种操作系统上运行。浏览器的发展也推动了 Web 技术的不断进步，使得 Web 应用在不同平台上的表现越来越一致。

3. Web 的分布式

Web 是一个分布式系统，大量信息资源分散存储在全球各地的服务器上。这种分布式结构使得 Web 能够容纳海量的信息，并且具有较高的可扩展性和可靠性。当某个服务器出现故障时，其他服务器仍然可以正常运行，不会影响用户对整个 Web 的访问。例如，一个大型的电商平台可能将其商品信息、用户数据等分别存储在不同的服务器上，以提高系统的性能和稳定性。

尽管信息在物理上是分散存储的，但在逻辑上却是一体化的，通过超链接和 URL(uniform resource locator，统一资源定位符)将不同的网页和资源链接在一起，形成一个庞大的信息网络。用户可以在该网络中自由浏览和切换，感觉就像在一个整体的系统中一样。例如，一个新闻网站的首页可能会链接到各种新闻分类页面、专题报道页面等，用户可以通过点击链接轻松获取不同类型的新闻信息。

4. Web 的动态性

Web 的动态性体现在信息的实时更新上。各 Web 站点的信息包含站点本身的信息，信息提供者可经常更新站点内容，以保证信息的时效性和准确性，如新闻网站会及时发布最新的新闻资讯、社交媒体平台会实时更新用户的动态信息、企业网站会及时更新产品信息和促销活动等。

除了信息的实时更新，Web 还可以根据用户的需求和操作生成动态的内容。例如，当用户在搜索引擎中输入关键词进行搜索时，搜索引擎会根据其算法和数据库中的信息生成相应的搜索结果页面；当用户在电商网站上选择商品并添加到购物车时，系统会自动更新购物车中的商品信息和总价等。这种动态内容的生成能力使得 Web 能够更好地满足用户的个性化需求。

5. Web 的交互性

Web 的交互性首先表现在超链接上。用户可以通过点击超链接在不同的网页之间进行跳转，自由地浏览自己感兴趣的内容。超链接的存在使得信息的获取变得更加便捷和高效，用户无须记住复杂的网址，只需单击鼠标即可访问相关网页。同时，超链接还能根据用户的操作动态改变颜色或状态，以提示用户是否已经访问过该链接。

除了超链接，Web 还通过表单实现了用户与服务器之间的交互。用户可以在表单中填写信息，如注册账号、登录密码、搜索关键词、留言评论等，然后将表单提交给服务器。服务器会根据用户提交的信息进行处理，并返回相应的结果或进行响应。例如，当用户在论坛中发表帖子时，需要填写标题、内容等信息，然后提交表单，服务器会将用户发表的帖子保存到数据库中，并在页面上显示出来供其他用户查看和评论等。

1.1.3　Web 的工作原理

Web 的工作原理基于客户端-服务器模型，通过 HTTP 协议传输数据，并借助 HTML、CSS、JavaScript 等技术实现页面的展示和交互功能。

当用户在浏览器中输入一个 URL 或点击链接时，浏览器会向服务器发送一个 HTTP 请求，包括请求方法(如 GET、POST)、请求头信息(如 User-Agent、Accept)等。浏览器首先会检查本地缓存，若缓存中存在有效响应，则直接从缓存中获取数据，否则会继续向 DNS 服务器请求域名解析。服务器接收到客户端的请求后，会进行处理并返回响应。响应包括响应状态码(如 200 表示请求成功、

404 表示未找到资源等)、响应头信息(如 Content-Type 表示内容类型、Content-Length 表示内容长度等)和响应体(即实际的内容)。服务器将响应数据发送给客户端，然后通过 HTTP 或 HTTPS 协议进行传输。HTTP 是无状态协议，即每次请求都是独立的，没有上下文关联。为了提高性能，可以使用 HTTP 缓存、压缩、持久连接等技术。HTTPS 是 HTTP 的安全版本，通过 SSL/TLS 协议对数据进行加密，确保数据在传输过程中的安全性。SSL/TLS 协议通过证书验证服务器的身份，并使用对称加密算法对数据进行加密。浏览器会验证服务器的证书，如果证书无效，则会显示警告信息。浏览器接收到服务器的响应后，会解析并渲染内容：首先，浏览器会解析 HTML 文档，构建 DOM(document object model，文档对象模型)树；其次，浏览器会解析 CSS 样式，构建 CSSOM(CSS object model，CSS 对象模型)树；接着，浏览器会将 DOM 树和 CSSOM 树合并，生成渲染树(render tree)；最后，浏览器会根据渲染树进行布局(layout)和绘制(painting)，将内容显示在屏幕上。

1.1.4 Web 的相关概念

1. 统一资源定位器

统一资源定位器(uniform resource locator，URL)是 Web 上用于标识和定位信息资源的字符串。它提供了一种标准化的方式，让用户能够准确地找到并访问网络上的各种资源，如网页、图片、视频等。

URL 的结构通常包括协议类型、服务器地址、端口号(可选)、路径及查询字符串(可选)等部分。其构成如下所示。

协议类型://服务器地址(端口号)/路径/文件名

其中，第一部分是协议类型，如表 1-1 所示；第二部分是服务器地址和端口号，HTTP 的默认端口号是 80；第三部分是主机资源的具体地址，如文件名等。

表 1-1 协议类型

序号	服务(协议)类型	含义
1	http	超文本传输协议
2	https	用加密传送的超文本传输协议
3	ftp	文件传输协议
4	mailto	电子邮件地址
5	ldap	轻型目录访问协议搜索
6	news	Usenet 新闻组
7	file	当地计算机或网上分享的文件
8	tel	打电话协议
9	sms	发短信协议

第一部分和第二部分之间用“://”隔开，第二部分和第三部分之间用“/”隔开，其中第三部分可以省略，如 https://www.edu.cn/、https://www.edu.cn/rd/meeting/xinwen/202501/t20250113_2650895.shtml 等。

2. Web 服务器

Web 服务器，也称 WWW(World Wide Web)服务器、HTTP 服务器或 HTTP Server，其是一种驻

留在因特网上的计算机程序，用于向请求终端提供网页信息浏览服务。以下是关于 Web 服务器的详细介绍。

1) 主要功能

(1) 静态内容服务。Web 服务器可提供 HTML 文件、图像、CSS 样式表和 JavaScript 脚本等静态资源。当用户请求这些资源时，Web 服务器会直接将文件返回给客户端浏览器。

(2) 动态内容服务。Web 服务器可处理动态内容，通过与其他服务器或数据库交互，生成动态页面内容，并将其返回给客户端浏览器。这样可以根据用户请求动态地生成不同的内容。

(3) 网站托管。Web 服务器支持虚拟主机功能，允许在同一台服务器上托管多个网站。每个虚拟主机拥有独立的域名和资源，使得多个网站可以在同一服务器上共享硬件资源，并实现资源的高效利用。同时，通过域名解析可将用户的域名映射到相应的网站根目录中，确保用户在输入域名时能够访问正确的网站内容。

(4) 脚本解析。Web 服务器能够解析服务器端脚本语言(如 PHP、Python、Node.js 等)，通过执行脚本生成动态内容，并将其返回给客户端浏览器。

(5) 数据库连接。Web 服务器可以连接到数据库服务器，从数据库中检索数据并在动态页面中呈现，实现个性化和动态的网站内容。

(6) 安全性保障。Web 服务器通过启用 HTTPS 协议，使用 SSL/TLS 加密传输数据，保障数据在传输过程中的安全性，防止数据被窃取或篡改。同时，还可以设置访问权限，限制特定用户或 IP 地址的访问，增强网站的安全性。

2) 常见类型

(1) Apache。Apache 是目前全球使用最多的 Web 服务器之一，其市场占有率达 60%左右，具有源代码开放、跨平台应用、支持多种操作系统、高度可移植性等特点。

(2) Nginx。Nginx 由俄罗斯人 Lgor Sysoev 为俄罗斯第二大站点 Rambler.ru 开发，是一个高性能的 HTTP 和反向代理服务器。其以稳定性、丰富的功能集、示例配置文件和低系统资源消耗而闻名。

(3) IIS。IIS(Internet Information Services，互联网信息服务)是微软公司提供的基于 Microsoft Windows 运行的互联网基本服务。其支持 ASP.NET、PHP、FTP、NNTP 和 SMTP 等协议。

(4) Tomcat。Tomcat 是一个开放源代码的基于 Java 的 Web 应用容器，用于运行 Servlet 和 JSP Web 应用程序。虽然其不能直接处理静态 HTML 文件或进行高并发的处理，但可以通过与 Apache 等 Web 服务器配合使用来实现完整的功能。

(5) Lighttpd。Lighttpd 是由德国人 Jan Kneschke 编写的开源 Web 服务器软件，具有占内存少、并发能力强等特点。

1.2 Web 前端开发技术

在当今 Web 前端开发领域，HTML、CSS 和 JavaScript 三大核心技术已成为构建现代网页不可或缺的基础。它们各自扮演着独特角色，相互协作，共同助力网页功能的丰富和视觉效果的优化。HTML 是构建网页内容结构的基础，它通过一系列标签定义了网页的“骨架”和内容框架。CSS 通过层叠样式表的应用，为网页增添了色彩与布局，使得网页更加美观且易于阅读；它不仅美化了网页，还通过响应式设计确保网页在不同设备和屏幕尺寸下的适应性。而 JavaScript，则赋予了网页动

态交互的能力，使得用户能够与网页进行实时互动，从而提升了用户体验的互动性和参与感。

1.2.1 HTML

HTML(hyper text markup language，超文本标记语言)是一种用于创建网页结构和内容的标签语言。它由一系列的标签组成，每个标签都具有特定的语义和功能。这些标签通过成对出现的尖括号(< >)来标识，例如<h1>和</h1>分别代表一级标题的起始和结束。HTML 标签不仅可以构建基本的网页元素，还能通过属性如 id、class、src 等，为元素赋予特定的标识或指定元素的来源，如图片的 URL 地址。这些属性使网页内容的组织和管理变得更加灵活和高效。

浏览器可以读取 HTML 文档，并以网页的形式显示出它们。例如，在 Microsoft Edge 浏览器的 URL 中输入百度网址，所看到的网页就是浏览器对 HTML 文件进行解释的结果，如图 1-2 所示。

图 1-2　百度首页

右击后，选中“查看网页源代码”，可以查看其网页源码，如图 1-3 所示。

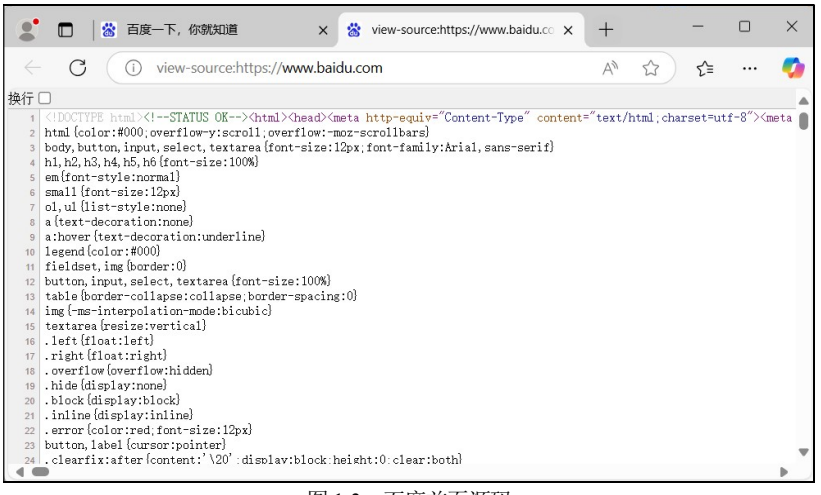

图 1-3　百度首页源码

1.2.2　CSS

CSS(cascading style sheets，层叠样式表)是一种用于为 HTML 文档添加样式和布局的技术。通过 CSS，可以调整网页中元素的外观和排列方式，从而实现网页的美化和布局优化。CSS 采用选择器来定位 HTML 元素，并通过属性值对其进行样式定义。例如，可以选择一个标题元素<h1>，然后通过 CSS 设置其颜色、字体大小、边距等样式。CSS 的层叠特性允许开发者为同一元素设置多个样式规则，并通过优先级来解决冲突，以确保最终的样式效果。此外，CSS 还支持响应式设计，通过媒体查询等技术，使得网页在不同设备和屏幕尺寸下展现不同的样式与布局，从而确保了网页在各种设备上均能提供优质的用户体验。

1.2.3　JavaScript

JavaScript 是一种被广泛使用的脚本语言，主要被设计用来为网页添加交互性和动态功能。它能让网页以一种非常直观的方式响应用户的操作，如点击、滚动和输入等。此外，JavaScript 还能够处理各种数据，包括用户输入的数据，以及与服务器进行各种形式的通信。通过 JavaScript，开发者可以操作网页上的各种元素，如修改文本内容、处理表单数据、创建各种动画效果、发送网络请求等。JavaScript 具备非常强大的编程功能，它支持面向对象编程、异步操作和模块化开发等先进的编程范式，并且能够与 HTML 和 CSS 紧密集成，为网页提供更加丰富和复杂的交互性功能与视觉效果。例如，JavaScript 可以实现表单验证、动态内容加载、页面动画等高级功能。这些功能的实现，极大地提升了网页的互动性和用户体验，使得网页不仅仅是一个静态的展示平台，而且是一个能够与用户进行实时互动的动态环境。JavaScript 的灵活性和强大的功能使其成为前端开发不可或缺的一部分，它让网页开发者能够创造出既美观又功能强大的网页应用，满足了现代网络用户对网页内容和交互的高要求。

1.2.4　AJAX

AJAX(asynchronous JavaScript and XML，异步 JavaScript 和 XML)是一种用于创建动态、交互式网页应用的技术，允许网页在不重新加载整个页面的情况下，与服务器进行数据交换和更新部分内容。它通过 XMLHttpRequest 对象在后台与服务器进行通信，从而实现了网页的异步更新。AJAX 的应用使得网页能够更快速地响应用户的操作，减少了用户的等待时间，提升了用户体验。

AJAX 的工作原理基于客户端与服务器之间的异步通信。当用户触发某个事件(如点击按钮、输入文本等)时，JavaScript 代码会捕获该事件，并创建一个 XMLHttpRequest 对象。该对象会向服务器发送一个请求，包含需要更新的数据或操作。服务器接收到请求后，会进行相应的处理，并返回一个响应。该响应可以是 XML 格式的数据，也可以是 JSON、HTML 等其他格式的数据。JavaScript 代码接收到服务器的响应后，会解析这些数据，并动态地更新网页的部分内容。这种通信方式避免了整个页面的重新加载，使得网页的交互更加流畅和高效。

AJAX 技术在 Web 开发中有着广泛的应用。例如，在在线购物网站中，当用户浏览商品列表时，可以通过 AJAX 技术实现商品的异步加载和分页显示，从而提高了页面的加载速度和用户的浏览体验；在社交媒体平台中，AJAX 技术也被用于实现实时更新用户动态、评论和点赞等功能。此外，AJAX 技术还可以用于实现表单验证、地图应用、实时数据更新等场景。

1.2.5 jQuery

jQuery 是一个快速、小巧、功能丰富的 JavaScript 库。它通过简化 HTML 文档的遍历与操作、事件处理、动画和 AJAX 交互，极大地提高了 Web 前端开发的效率和便捷性。jQuery 的设计理念是“写更少的代码，做更多的事情”，它提供了一整套丰富的预定义函数和插件，使得开发者可以更加高效地进行 DOM 操作、事件处理和数据交互。通过使用 jQuery，开发者可以轻松实现各种复杂的交互效果，如轮播图、手风琴效果、动画过渡等，从而显著提升用户体验和网页的吸引力。同时，jQuery 还具备良好的兼容性和扩展性，能够在不同的浏览器和平台上稳定运行，并支持开发者根据自己的需求进行定制和扩展。

1.3 Web 前端开发工具

1.3.1 Visual Studio Code

Visual Studio Code 是一款由微软公司开发的轻量级且功能强大的源代码编辑器。它不仅支持多种编程语言和标签语言，如 HTML、CSS、JavaScript 等，而且特别适合 Web 前端开发人员使用。Visual Studio Code 提供了丰富的扩展和插件，如代码质量检查工具、代码格式化工具等，能够协助开发者维护代码的一致性和可读性。此外，Visual Studio Code 还支持与版本控制系统(如 Git)的集成，从而简化了代码版本控制和团队协作流程，其界面设计简洁明了，功能强大，因此在 Web 前端开发领域获得了广泛的认可和欢迎。Visual Studio Code 的这些特性使得它成为开发者工具箱中不可或缺的一部分，无论是初学者还是经验丰富的开发者都能从中受益。

Visual Studio Code 程序界面如图 1-4 所示。

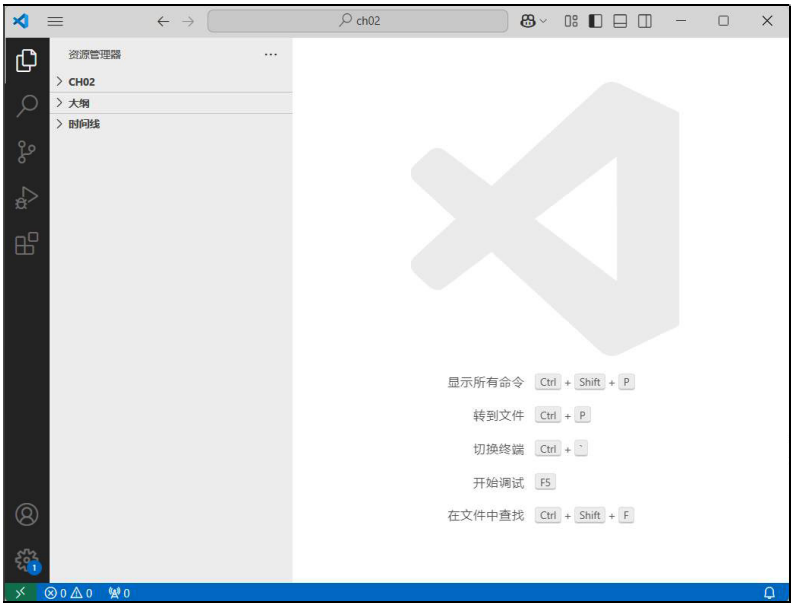

图 1-4 Visual Studio Code 程序界面

1.3.2　HBuilder X

HBuilder X 是一款专为前端开发者打造的集成开发环境(integrated development environment，IDE)，以其高效、智能的特点深受开发者喜爱。它内置了许多实用工具和插件，旨在帮助开发者快速构建并调试 Web 应用。HBuilder X 支持多种编程语言和框架，如 HTML、CSS、JavaScript 及 Vue、React 等，使开发者能够在统一开发环境中处理各种前端技术栈。此外，HBuilder X 还提供了智能的代码补全、语法高亮、实时预览等功能，极大地提升了开发效率和代码质量。其强大的调试工具能够帮助开发者快速定位和解决代码中的问题，从而节省了大量的时间和精力。对于需要频繁进行代码修改和调试的前端开发任务，HBuilder X 的这些特性无疑是一个巨大的助力。

HBuilder X 程序界面如图 1-5 所示。

图 1-5　HBuilder X 程序界面

1.3.3　WebStorm

WebStorm 是由 JetBrains 公司开发的一款功能强大的 JavaScript 集成开发环境(IDE)，它专为 Web 和移动开发者设计，提供了丰富的工具和功能，以简化开发流程并提高生产效率。WebStorm 支持多种编程语言和框架，包括 HTML、CSS、JavaScript、TypeScript、React、Vue、Angular 等，使得开发者能够在同一个开发环境中处理各种前端技术栈。

WebStorm 具备智能的代码编辑功能，如代码补全、语法高亮、实时错误检查等，这些功能能够帮助开发者快速编写高质量的代码。其内置的调试工具支持断点调试、变量监视、堆栈跟踪等，使得开发者能够轻松定位和解决代码中的问题。此外，WebStorm 还提供了版本控制系统集成，如 Git 等，方便开发者进行代码版本控制和团队协作。

WebStorm 还具有强大的重构和导航功能，开发者可以轻松重构代码，如重命名变量、提取方法、内联变量等。其内置的搜索和导航工具，如全局搜索、文件搜索、类搜索等，可帮助开发者快速找到所需的代码或资源。这些功能使得开发者能够更加高效地管理和维护大型项目。

WebStorm 还支持各种插件的扩展，开发者可以根据自己的需求安装和使用这些插件，以增强 IDE 的功能。这些插件涵盖了代码质量检查、代码格式化、代码生成、UI 设计等多个方面，为开发者提供了更加丰富的工具和选择。

1.4 浏览器工具

浏览器作为 Web 前端开发的核心工具，不仅用于浏览和展示网页，更是开发者调试和测试网页的重要平台。现代浏览器提供了丰富的开发者工具，如元素检查器、控制台、网络监视器等，这些工具极大地提高了开发效率和问题排查能力。开发者可以利用这些工具查看和修改网页的 HTML 结构、CSS 样式及 JavaScript 代码，实时观察网页的变化，从而快速定位并解决问题。此外，浏览器还支持各种扩展和插件，为开发者提供了更多的功能和便利，如代码格式化、代码高亮、实时预览等，进一步提升了开发体验。

1.4.1 Microsoft Edge

Microsoft Edge 是一款由微软公司开发的现代浏览器，它不仅具备快速、安全的浏览体验，还提供了丰富的开发者工具和功能，使 Web 前端开发变得更加高效、便捷。Microsoft Edge 浏览器内置了强大的调试工具，如 F12 开发者工具，包括元素检查器、控制台、网络监视器、性能分析器等，为开发者提供了全面的网页调试和测试能力。开发者可以利用这些工具查看网页的 HTML 结构、CSS 样式、JavaScript 代码及网络请求等信息，实时观察网页的变化，从而快速定位和解决问题。

在元素检查器中，开发者可以直观地查看和编辑网页的 DOM 结构和 CSS 样式，实时预览修改效果。控制台则用于输出 JavaScript 代码的执行结果和错误信息，帮助开发者调试代码。网络监视器可以记录网页发出的所有网络请求和响应，包括请求头、响应头、请求数据、响应数据等，方便开发者分析网络性能和排查网络问题。性能分析器则可以帮助开发者分析网页的性能瓶颈，优化网页的加载速度和响应性能。

除了内置的调试工具，Microsoft Edge 还支持各种扩展和插件，可为开发者提供更多的功能和便利。例如，一些代码格式化插件可以帮助开发者自动格式化 HTML、CSS 和 JavaScript 代码，提高代码的可读性和一致性；实时预览插件则可以在开发者编写代码时实时展示网页效果，提高开发效率。

另外，Microsoft Edge 最初是基于 Trident 内核开发的，这也是 Internet Explorer 所使用的渲染引擎。然而从 2019 年起，Microsoft Edge 浏览器的 79 及之后的版本均基于 Chromium 内核开发构建，这意味着它如今使用与 Google Chrome 相同的内核引擎。

Microsoft Edge 首页如图 1-6 所示。

图 1-6　Microsoft Edge 首页

1.4.2　Google Chrome

Google Chrome 是一款由 Google 开发的网页浏览器，以其高速、安全和丰富的功能而受到广大开发者的喜爱。它同样内置了强大的开发者工具，包括控制台、元素检查器、网络监视器和性能分析器等。控制台用于输出 JavaScript 代码的执行结果、调试信息和错误信息，帮助开发者快速定位和解决代码中的问题。元素检查器则允许开发者直接查看和修改网页的 HTML 和 CSS 代码，实时预览修改效果，极大地提高了开发效率。网络监视器可以记录并分析网页发出的所有网络请求和响应，包括请求头、响应头、请求数据和响应数据等，为开发者提供了全面的网络性能分析手段。性能分析器则可以帮助开发者深入分析网页的性能瓶颈，优化网页的加载速度和响应性能。

Google Chrome 首页如图 1-7 所示。

图 1-7　Google Chrome 首页

1.4.3 Safari

苹果公司倾力打造的 Safari 网页浏览器，以其简约而优雅的设计理念和众多卓越功能，在 Web 前端开发领域确立了其独特地位。该浏览器不仅为用户带来了顺畅的浏览体验，而且在前端开发者的工具库中占据了不可替代的位置。Safari 内置的开发者工具集成了众多实用功能，对前端开发者而言，这些功能是不可或缺的。例如，控制台功能允许开发者输出 JavaScript 代码的执行结果，并提供调试信息与错误信息，极大地方便了代码问题的发现和故障排除。元素检查器功能则使开发者能够直观地审视和编辑网页的 HTML 结构与 CSS 样式，并实时预览更改效果，这对于迅速定位和修正页面布局及样式问题极为有益。此外，Safari 还配备了网络监视器和性能分析器等高级工具，能够记录和分析网页的网络请求与响应，并深入剖析网页性能瓶颈，从而协助开发者优化网页加载速度和响应性能，进一步提升用户体验。

除了其卓越的功能性，Safari 浏览器的开发者工具还因其简洁直观的设计和易于掌握的操作而受到开发者的青睐。开发者能够通过简单的操作步骤和直观的界面，迅速定位并解决网页中的问题，这显著提升了开发效率和工作流程的顺畅性。此外，Safari 浏览器还支持众多扩展和插件，为开发者提供了额外的功能和便利，如代码格式化、代码高亮、实时预览等，进一步增强了开发体验和工作效率。因此，Safari 浏览器成为许多前端开发者日常工作中不可或缺的工具之一，它在提升工作效率和优化开发流程方面发挥着关键作用。

1.5 习题

1. 单选题

(1) Web 的初始设计目的是作为(　　)。

A. 动态信息资源的发布平台　　B. 静态信息资源的发布平台

C. 电子商务平台　　D. 社交媒体平台

(2) HTML、URI 和 HTTP 3 个规范构成了 Web 的(　　)。

A. 核心架构　　B. 辅助架构　　C. 扩展架构　　D. 安全架构

(3) Tim Berners-Lee 在(　　)年基于 NeXTSTEP 计算机开发出了世界上第一个 Web 服务器和第一个 Web 客户机。

A. 1989　　B. 1990　　C. 1991　　D. 1992

(4) Web 的图形化特点主要是(　　)。

A. 纯文本界面　　B. 色彩丰富的图形、图像、文本等多种元素

C. 仅文字链接　　D. 黑白色调

(5) 下列中不是 Web 特点的是(　　)。

A. 与平台无关性　　B. 分布式　　C. 静态性　　D. 动态性

(6) 统一资源定位器(URL)用于标识和定位(　　)。

A. 用户信息　　B. 网络设备　　C. 信息资源　　D. 网络安全

(7) Web 服务器的主要功能不包括(　　)。

A. 提供静态内容服务　　B. 处理动态内容

C. 执行操作系统更新　　D. 网站托管

(8) 下列技术中，允许网页在不重新加载整个页面的情况下与服务器进行数据交换和更新部分内容的是(　　)。

A. AJAX　　B. CSS　　C. HTML　　D. JavaScript

(9) Visual Studio Code 是由(　　)公司开发的。

A. Google　　B. Microsoft　　C. Apple　　D. JetBrains

(10) HBuilder X 是一款专为(　　)打造的集成开发环境(IDE)。

A. 后端开发者　　B. 全栈开发者　　C. 前端开发者　　D. UI 设计师

2. 多选题

(1) Web 技术不断演进，从最初的静态页面发展到现在的(　　)等方面。

A. 动态网页　　B. Web 应用　　C. 移动互联网　　D. 虚拟现实

(2) Web 的动态性包括(　　)。

A. 信息的实时更新　　B. 根据用户需求生成动态内容

C. 仅展示静态页面　　D. 支持用户交互

(3) Web 前端开发技术中，HTML、CSS 和 JavaScript 各自的作用是(　　)。

A. HTML 定义网页结构　　B. CSS 控制网页样式

C. JavaScript 实现网页交互　　D. HTML 提供动态内容

(4) 现代浏览器提供的开发者工具包括(　　)。

A. 元素检查器　　B. 控制台　　C. 网络监视器　　D. 性能分析器

(5) Web 服务器的常见类型有(　　)。

A. Apache　　B. Nginx　　C. IIS　　D. Lighttpd

(6) AJAX 技术在 Web 开发中的应用场景包括(　　)。

A. 在线购物网站的异步加载商品列表　　B. 社交媒体平台的实时更新用户动态

C. 地图应用的实时数据更新　　D. 表单验证

第 2 章

HTML基础

2.1 HTML 基本结构

HTML 文档的基本结构由以下 4 个关键部分组成，这些部分共同定义了网页的内容和布局。

1. 文档类型声明

文档类型声明(DOCTYPE)位于文档的最顶部，用于告知浏览器该文档使用的 HTML 版本。这有助于确保浏览器以正确的方式解析和渲染网页内容。

2. html 标签

html 标签是整个 HTML 文档的根元素。所有的其他 HTML 元素都应该被包含在 html 标签内。

3. head 标签

head 标签包含了关于文档的元数据，如文档的标题(title)、字符集声明、样式表链接、脚本引用等。这些信息对于网页的呈现和浏览器对网页的处理至关重要。特别是 title 标签，它定义了网页的标题，该标题会显示在浏览器的标签页上。

4. body 标签

body 标签包含了网页的可见内容，如文本、图像、链接、表格、表单等。这些内容构成了用户浏览网页时看到的主要部分。

【例 2-1-1】HTML 基本结构展示，代码如下所示。

```
<!-- exam_2_1_1.html -->
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
        <meta name="viewport" content="width=device-width, initial-scale=1.0">
     <title>第一个网页</title>
</head>
<body>
        <p>你好！欢迎来到 Web 的世界</p>
</body>
</html>
```

结果如图 2-1 所示。

图 2-1　第一个网页

2.2 头部 head

2.2.1 标题 title 标签

基本语法：

```
<title>标题信息显示在浏览器的标题栏上</title>
```

语法说明：

title 标签是双标签，<title>是开始标签，</title>是结束标签，两者之间的内容为显示在浏览器标题栏上的信息。

【例 2-2-1】标题 title 标签的应用，代码如下所示。

```
1    <!-- exam_2_2_1.html -->
2    <html>
3      <head>
4        <title> Web 页面标题</title>
5      </head>
6      <body>
7      </body>
8    </html>
```

上述代码的运行效果，如图 2-2 所示。

图 2-2　标题 title 标签的应用

2.2.2 元信息 meta 标签

meta 标签是 HTML 文档中用于定义网页元数据的关键部分。元数据是关于数据的数据，它提供了有关网页的各种信息，这些信息虽然不会直接显示在网页的内容中，但对于网页的呈现、搜索引擎优化(SEO)及社交媒体分享等方面都至关重要。

meta 标签通常位于 head 标签内，用于指定网页的字符集、页面描述、关键词、作者信息、页

面刷新时间等。这些信息对搜索引擎了解网页的内容和上下文、社交媒体平台在分享网页时展示的信息及浏览器对网页的渲染方式都具有重要影响。

基本语法：

```
<meta name="属性名" content="属性值">
<meta http-equiv="" content="">
```

属性说明：

name 属性与 content 属性用于定义网页的描述信息。其中，name 属性用于指定元数据的名称，如 description(页面描述)、keywords(关键词)、author(作者信息)等。content 属性则提供了对应名称的具体值。

例如：

```
<meta name="description" content="这是一个关于 Web 前端开发的介绍页面">
```

而 http-equiv 属性则用于模拟 HTTP 响应头的功能，如设置网页的字符集、页面自动刷新时间等。

例如：

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

用于指定网页的字符集为 UTF-8，这有助于确保网页内容在不同浏览器和设备上正确显示。

通过合理使用 meta 标签，开发者可以更好地控制网页在搜索引擎中的排名、社交媒体分享时的显示效果及浏览器的渲染方式，从而提升网页的可见性和用户体验。

meta 标签的属性、取值和说明如表 2-1 所示。

表 2-1　meta 标签的属性、取值和说明

属性	取值	说明
content	some_text	定义与 http-equiv 或 name 属性相关的元信息
http-equiv	content-type	内容类型
	expires	网页缓存过期时间
	refresh	刷新与跳转(重定向)页面
	set-cookie	如果网页过期，那么存盘的 cookie 将被删除
name	author	定义网页作者
	description	定义网页简短描述
	keywords	定义网页关键词
	generator	定义编辑器
scheme	some_text	定义用于翻译 content 属性值的格式

【例 2-2-2】元信息 meta 标签的应用，代码如下所示。

```
1   <!-- exam_2_2_2.html  -->
2   <!DOCTYPE html>
3   <html lang="zh">
4       <head>
5           <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
6           <title>中国知网</title>
7           <meta name="keywords" content="中国知网,数字出版,知识发现,知识服务,知识
```

```
8               管理,数字出版,增强出版,CAJ-N,网络首发,CNKI 首发,数字图书馆,学术文献,期
9               刊,博士论文,硕士论文,会议论文,报纸,年鉴,统计数据,专利,科技成果,标准,法
10               规,古籍,工具书,引文,图片搜索,外文文献" />
11                  <meta name="description" content="中国知网知识发现网络平台——面向海内外
12               读者提供中国学术文献、外文文献、学位论文、报纸、会议、年鉴、工具书等
13               各类资源统一检索、统一导航、在线阅读和下载服务。涵盖基础科学、文史哲、
14               工程科技、社会科学、农业、经济与管理科学、医药卫生、信息科技等十大领
15               域。" />
16                  <meta http-equiv="X-UA-Compatible" content="IE=edge,chrome=1" />
17            <link rel="stylesheet" href="https://piccache.cnki.net/kdn/index/kns8s/
18               nvsmcss_min/index.min.css?v=1.2" />
19            <script type="text/javascript" src="https://piccache.cnki.net/kdn/index/kns8s/
20               nvsmscripts/jquery.min.js?v=1.12.4"></script>
21         </head>
22         <body>
23         </body>
24      </html>
```

在上述代码中，第 5 行设置了网页的字符集为 UTF-8，这确保了网页内容能够正确地在不同浏览器和设备上显示中文和其他特殊字符。第 6 行定义了网页的标题为“中国知网”，该标题会显示在浏览器的标签页上。第 7 至 10 行和第 11 至 15 行分别通过 meta 标签的 name 属性设置了网页的关键词和描述信息，这些信息对于搜索引擎优化(SEO)至关重要，能够帮助搜索引擎更好地了解网页的内容和上下文，从而在用户搜索相关关键词时提高网页的排名和可见性。第 16 行则通过 meta 标签的 http-equiv 属性模拟 HTTP 响应头的功能，设置了 IE 浏览器的兼容模式，以优化网页在 IE 浏览器上的显示效果。此外，第 17 至 20 行还引入了外部的 CSS 样式表和 JavaScript 脚本文件，用于美化网页的外观和添加交互功能。

2.3 主体 body

在 HTML 文档结构中，body 标签扮演着承载网页核心内容与框架的角色。通过嵌入多样化的 HTML 元素，如段落、标题、列表、图像、表格、表单及多媒体内容等，构建网页的视觉呈现与交互功能。这些元素不仅扩充了网页的信息内容，而且通过恰当的布局与设计，进一步优化了用户的浏览体验。

2.3.1 body 标签

基本语法：

```
<body>…</body>
```

body 标签是双标签，<body>为开始标签，</body>为结束标签，这两个标签之间的内容即为网页的可见部分，包括文本、图像、链接、表格、表单等。在这一区域内，可以通过 HTML 标签和属性来构建网页的结构和样式，从而为用户呈现一个丰富、互动且易于导航的视觉界面。

2.3.2 body 标签的属性

body 标签不仅用于包裹网页的可见内容，还支持多种属性来进一步定义网页的显示特性。这些属性能够调整网页的背景颜色、文本颜色、链接样式等，为网页设计师提供了更多的自定义空间。

常用 body 标签的属性、取值和说明如表 2-2 所示。

表 2-2 常用 body 标签的属性、取值和说明

属性	取值	说明
text	rgb(R,G,B) rgb(R%,G%,B%) #RRGGBB 或#RGB Color name	规定文档中所有文本的颜色
bgcolor	同上	用于设置网页的背景颜色
alink	同上	用于设置网页中活动链接(即鼠标悬停或单击时的链接)的颜色
link	同上	用于设置网页中未访问链接的默认颜色
vlink	同上	用于设置网页中已访问链接的颜色
background	URL	用于设置网页的背景图像。通过指定图像的 URL，可以为网页添加视觉层次和吸引力
topmargin	Pixel	规定文档中上边距的大小
leftmargin	pixel	规定文档中左边距的大小

【例 2-3-1】body 标签属性的应用，代码如下所示。

```
<!-- exam_2_3_1.html -->
<!doctype html>
<html lang="en">
  <head>
      <meta charset="UTF-8">
      <title>简易网页设计</title>
  </head>
  <body text="green">
      <h3 align="center">Web 前端开发技术课程简介</h3>
       <hr color="red">
       <p>《Web 前端开发技术》通过对 Web 前端开发三大主流技术的学习和研究，让学
  生理解和掌握 HTML、JavaScript、CSS 等相关知识，通过实验培养学生设计与开发 Web
  站点的基本操作技能。</p>
    </body>
</html>
```

在上述代码中，第 8 至 14 行定义了网页的主体部分。第 8 行通过 body 标签的 text 属性将网页中所有文本的颜色设置为绿色，第 9 行使用了一个居中对齐的 h3 标题标签，用于展示网页的主要标题“Web 前端开发技术课程简介”，第 10 行插入了一个红色的水平线标签，第 11 至 13 行则包含了网页的正文内容，通过 p 段落标签进行了封装。

页面效果如图 2-3 所示。

图 2-3　body 标签属性应用页面效果

2.4 HTML 基本语法

2.4.1 标签的类型

HTML 标签是由尖括号包围的关键词，用于说明指定内容的外貌和特征，也称为标签(Tag)。<html>、<head>、<body>等都是标签。标签通常分为单个标签和成对标签两种类型。

1. 单个标签

单个标签是指不需要结束标签的标签，它们在 HTML 中的作用主要是提供命令或说明。

基本语法：

```
<标签名称>或<标签名称/>
```

常用的单个标签有
、<hr>。
、
表示换行，<hr>、<hr/>表示水平分隔线。这些标签在 HTML 代码中独立存在，不需要闭合标签来完成其功能。

2. 成对标签

成对标签由一个开始标签和一个结束标签组成，它们之间的内容即为该标签所定义的范围。开始标签和结束标签之间的内容会被浏览器解析并显示出来，或者根据其属性进行特定的处理。

基本语法：

```
<标签名称>内容</标签名称>
```

其中，“内容”是被这对标签施加作用的部分。例如， 和 是一对用于定义段落的标签，它们之间的文本会被视为一个独立的段落进行处理；<a>和是一对用于定义超链接的标签，它们之间的文本或图像会成为可点击的链接，指向另一个网页或网页内的某个位置。

此外，HTML 标签还可以根据其功能进行分类，如文本格式化标签、列表标签、表格标签、表单标签等。这些标签在 HTML 文档中扮演着不同的角色，共同构建出网页的结构和内容。

2.4.2 HTML 属性

HTML 使用标签来描述网页，浏览器会根据标签解释其中包含内容的显示效果。每一个标签都定义了默认的显示效果，这些默认效果是通过标签的属性来定义的。如果要修改某一效果，则需要修改该标签的附加信息。

基本语法：

```
<标签名称  属性 1="属性值 1"  属性 2="属性值 2" … 属性 n="属性值 n">
```

属性应在开始标签(首标签)内定义，并且和标签名之间有一个空格分隔。

例如：

```
<hr size="3"  color="red"  align="center">
```

其中，align 为属性，center 为属性值，属性值可以直接书写，也可以使用双引号括起来。

【例 2-4-1】标签语法及属性语法的应用，代码如下所示。

```
1    <!--exam2_4_1.html-->
2    <!doctype html>
3    <html lang="en">
4     <head>
5     <meta charset="UTF-8">
6     <title>Web 介绍</title>
7     </head>
8     <body background="" text="blue">
9       <h2 align="center">Web 前 端 开 发 技 术</h2>
10          <hr size="3" color="#6600ff" width="50%"/>
11          <p align="left">Web 前端开发技术是一门综合性较强的课程</p>
12          <p align="left">随着技术的发展，不断涌现新的框架和工具，要保持学习的热情，
13                  及时掌握这些新技术，为自己的职业发展打下坚实的基础。</p>
14     </body>
15    </html>
```

上述代码中，第 8 至 14 行定义了网页的主体部分，并对网页的显示效果进行了自定义设置。第 8 行通过 text 属性将网页中所有文本的颜色设置为蓝色，第 9 行使用了一个居中对齐的 h2 标题标签，展示了网页的副标题“Web 前端开发技术”，第 10 行插入了一个自定义的水平线标签，其中 size 属性设置了线条的粗细为 3，color 属性通过十六进制颜色代码设置了线条的颜色为紫色(#6600ff)，width 属性则限制了线条的宽度为网页宽度的 50%，第 11 至 13 行包含了网页的正文内容，通过 p 段落标签进行了封装，并使用了 align 属性将段落文本对齐方式设置为左对齐。

页面效果如图 2-4 所示。

图 2-4　标签语法及属性语法应用的页面效果

2.5 注释

在 HTML 代码中，注释是一种非常重要的元素，它不仅可以帮助开发者在编写代码时添加说明和备注，提高代码的可读性，还可以在代码调试和维护过程中发挥关键作用。注释的内容不会被浏览器解析和显示，因此它不会影响网页的最终呈现效果。

HTML 注释的基本语法非常简单，“<!--”为标签注释的开始，“-->”为标签注释的结束。在这两个标签之间的任何内容都会被视为注释，不会被浏览器渲染。例如：

```
<!--这是一个注释-->
```

或者对于多行注释，也可以这样写：

```
<!--
这是一个多行注释
可以包含多行文本
-->
```

在团队合作或项目维护中，合理使用注释可以极大提高代码的可维护性和可读性。开发者可以通过注释来说明代码的功能、逻辑、算法或注意事项，这样即使后续有其他开发者接手项目，也能够快速理解代码的意图和结构。

此外，在 HTML 文档的开发过程中，注释还可以被用来临时禁用某些代码段，以便进行调试或测试。当需要恢复这些代码段时，只需删除注释标签即可，非常方便。

需要注意的是，虽然注释在 HTML 代码中非常重要，但过多的注释也可能会导致代码变得冗长和难以管理。因此，在使用注释时，应该遵循简洁明了的原则，只添加必要的说明和备注。

2.6 HTML 文档编写规范

2.6.1 HTML 代码书写规范

在进行 HTML 编码时，除了确保代码功能的正确性，遵循特定的编码规范亦至关重要。这不仅能够显著提升代码的可读性、可维护性，还能在团队协作中提高效率。

1. 缩进与对齐

为使 HTML 标签的嵌套关系更为清晰，应采用统一的缩进风格，通常选择两个或四个空格作为缩进单位。同时，保持代码对齐亦至关重要，其有助于确保相关标签和属性处于同一层级，便于阅读与修改。

2. 标签闭合

编码时，务必确保所有 HTML 标签正确闭合。成对出现的标签必须包含开始标签和结束标签。对于无须结束标签的单个标签，也应确保其使用恰当，并符合 HTML 标准规范。例如：

<head> <title> ... </title> </head>　<!- -这是正确的书写格式- ->

<head> <title> ... </head> </title>　<!- -这是错误的书写格式- ->

3. 属性顺序

HTML 标签中属性的排列顺序应保持一致，一般是先编写全局属性(如 id、class)，再添加特定属性(如 href、src 等)。

4. 属性值引号

为 HTML 标签的属性值添加引号是一种良好的编码实践，无论是单引号还是双引号均可使用，即使属性值中不含空格或特殊字符，也建议添加引号。这样做可避免潜在的解析错误，并能增强代码的可读性。

5. 注释使用

在代码中合理使用注释，能有效解释代码的功能、逻辑或需注意的事项。注释应简洁明了，避免冗长和无关的信息。同时，要确保注释不会影响代码的正常执行。

6. 空行与空格

在代码中恰当使用空行和空格，以分隔不同的代码块和标签，可大幅提高代码的可读性，如在不同的 HTML 部分(如 head、body)之间添加空行、在标签和属性之间添加适当的空格等。

7. 避免内联样式和脚本

尽量避免在 HTML 标签中直接使用内联样式和脚本。相反，应使用外部的 CSS 文件和 JavaScript 文件来管理样式和脚本。这种做法有助于将内容与表现形式分离，从而提高代码的可维护性和团队协作的效率。

8. 遵循命名规范

为 HTML 标签、属性、类名和 ID 等选择有意义的命名至关重要。命名应简洁明了，能准确反映其功能和用途。同时，应遵循团队或项目的命名规范，以保持代码的一致性。

遵循上述 HTML编码规范，将有助于编写出更整洁、可读和易于维护的 HTML 文档，从而在开发过程中提升效率与质量。

2.6.2 HTML 文档的命名规则

在命名 HTML 文档时，应注意以下几点。

(1) 文档的扩展名为 html 或 htm，建议统一用 html 作为文件名的后缀。

(2) 文档名中只能由英文字母、数字或下画线组成，建议以字母或下画线开始。

(3) 文档名中不能包含特殊符号，如空格、$、&等。

(4) 文档名区分大小写。

(5) Web 服务器主页一般是 index.html 或 default.html。

2.7 HTML 文档的类型

2.7.1 !doctype 标签

!doctype 标签用于声明 HTML 文档的类型和版本，它告诉浏览器以何种标准来解析和渲染网页

内容。在 HTML5 中，!doctype 标签被简化为“<!doctype html>”，这一声明不区分大小写，用于指示浏览器是一个 HTML5 文档。

基本语法：

```
<!DOCTYPE element-name DTD-type DTD-name DTD-url>
```

<!DOCTYPE>用于告知浏览器或解析器当前文档所遵循的文档类型定义(DTD)是什么。其中 DOCTYPE 是一个关键字，它标志着文档类型声明的开始。

element-name 用于明确指定该 DTD 的根元素名称。该根元素是构成文档结构的基础，定义了文档的最外层结构。

DTD-type 用于区分该 DTD 是属于标准公用还是私人制定。标准公用的 DTD 通常由标准化组织提供，而私人制定的 DTD 则可能是某个组织或个人根据特定需求创建的。

DTD-name 用于指定该 DTD 的文件名称。该名称通常与 DTD 文件的实际文件名相对应，便于在文档中引用和识别。

DTD-url 用于指定该 DTD 文件所在的 URL 地址，这样浏览器或解析器就可以通过网络访问并获取该 DTD 文件，确保文档的正确解析。

>标志着文档类型声明的结束，它告诉浏览器或解析器 DTD 声明已经完成，接下来的内容将遵循之前声明的 DTD 规则。

2.7.2　HTML5 的 DTD 定义

HTML5 对文档类型声明(DTD)进行了简化，仅需使用“<!doctype html>”即可明确标识一个 HTML5 文档。这种简化的声明方式使得 HTML5 文档的结构更为清晰，并且便于开发人员编写与理解。相较于早期的 HTML 版本，HTML5 的 DTD 定义摒弃了对复杂 DTD 文件的依赖，转而采用一种更为直接且简洁的方法，以向浏览器明确指出文档的类型与版本。

2.8　综合案例

【例 2-8-1】以传统美德故事“孔融让梨”设计网页，代码如下所示。

```
1   <!--exam2_8_1.html -->
2   <!doctype html>
3   <html lang="en">
4     <head>
5       <meta charset="UTF-8">
6       <title>孔融让梨</title>
7     </head>
8     <body bgcolor="#FEFEFE" leftmargin="100px" rightmargin="100px">
9       <h2 align="center">孔融让梨</h2>
10       <hr size="1" color="red" width="100%" />
11       <div id="" class="">
12         <img src="picture1.jpg">
```

```
13                <p>孔融让梨的故事讲述的是孔融四岁时候，与兄弟一起吃梨，但他一直
14    拿最小的梨吃，父亲奇怪地询问他，他回答说：“我是小孩子，按理应该拿小的。”孔融
15    的宗族因而对他感到惊奇。孔融是东汉末年一代名儒，继蔡邕为文章宗师，亦擅诗歌。
16    魏文帝曹丕十分欣赏孔融文辞，在他死后曾悬赏征募他的文章，把孔融与王粲、陈琳、
17    徐干、阮瑀、应玚、刘桢六位文学家相提并论，列为“建安七子”。 </p>
18                <hr size="1" color="red" width="100%" />
19            </div>
20        </body>
21    </html>
```

上述代码中，第 2 至 21 行构成了一个完整的 HTML5 文档，用于展示传统美德故事“孔融让梨”。第 2 行声明了该文档遵循 HTML5 标准，第 3 至 21 行定义了 HTML 文档的根元素，其中包含了文档的头部(head)和主体(body)部分。

头部部分(第 4 至 7 行)通过 meta 标签设置了文档的字符编码为 UTF-8，通过 title 标签设置了网页的标题为“孔融让梨”。

主体部分(第 8 至 20 行)定义了网页的显示内容和布局。第 8 行设置了网页的背景颜色为浅灰色(#FEFEFE)，并通过 leftmargin 和 rightmargin 属性设置了网页左右两侧的边距。第 9 行使用了一个居中对齐的 h2 标题标签，展示了网页的主标题“孔融让梨”。第 10 行插入了一个自定义的水平线标签，其中 size 属性设置了线条的粗细为 1，color 属性通过颜色名称设置了线条的颜色为红色，width 属性则限制了线条的宽度为网页宽度的 100%。

第 11 至 19 行定义了一个 div 容器，用于封装网页的正文内容和图片。第 12 行通过 img 标签插入了一张与孔融让梨故事相关的图片(picture1.jpg)，第 13 至 17 行包含了网页的正文内容，通过 p 段落标签进行了封装，并讲述了孔融让梨的故事及其背景信息。最后，第 18 行再次插入了一个与第 10 行相同的水平线标签，用于分隔网页的正文内容和可能存在的其他内容或元素。

页面效果如图 2-5 所示。

图 2-5　孔融让梨效果图

2.9 习题

1. 多选题

(1) 下列中，(　　)标签不是 HTML 文档的根元素。

A. <head>　　B. <html>　　C. <body>　　D. <title>

(2) HTML文件的基本结构包括(　　)。

A. DOCTYPE 声明　　B. html 标签

C. head 标签　　D. body 标签

(3) 在 HTML中，不能用于设置网页背景颜色的 body 标签属性的是(　　)。

A. Text　　B. bgcolor　　C. alink　　D. vlink

2. 编程题

编写一个 HTML文档，实现一个简单的个人简历页面的展示效果。

第 3 章

HTML格式化文本、段落和图像

3.1 Web 页面初步设计

Web 页面设计需要遵循简约、一致性和易读性原则。在设计 Web 页面时，首先需明确页面的主题与目标受众，再根据这些信息来确定页面的布局、色彩搭配和字体选择。

良好的 Web 页面设计能够提升用户体验，使用户更加愿意停留在页面上，从而提高网站的访问量和转化率。接下来，我们将详细介绍如何向 Web 页面中添加文字信息、使用标题字标签及添加空格与特殊符号，这些都是 Web 页面初步设计中的重要环节。通过合理使用这些 HTML 元素，可使页面内容更加丰富、有序，同时也能提升页面的美观度和可读性。

3.1.1 向 Web 页面中添加文字信息

基本语法：

```
<body>向这里添加内容</body>
```

body 元素定义文档的主体，它包含文档的所有内容，如文本、超链接、图像、表格和列表等。body 标签所包含的内容会显示在页面上。

在实际应用过程中，我们应将向用户呈现的内容置于<body>与</body>标签之间。例如，若需在页面上展示“欢迎来到我的网站”，则应在<body>标签内输入该段文字。需要指出的是，HTML 对字母大小写不具有区分性，故<body>与<BODY>具有相同的效果。然而，为了维护代码的一致性和提高可读性，建议在开发实践中采用全部小写的形式。

3.1.2 标题字标签

在 HTML 中，标题使用<h1>到<h6>标签来表示，这些标签分别代表六个级别的标题。其中，<h1>表示最高级别的标题，通常用于页面的主标题；而<h6>表示最低级别的标题，通常用于页面中的次要或辅助标题。使用标题字标签不仅可以使页面内容更加结构化，还可以提高页面的可读性和 SEO 效果。搜索引擎通常会根据标题字标签来判断页面的主题和内容，并给予相应的权重。因此，在设计 Web 页面时，合理使用标题字标签是非常重要的。

基本语法：

```
<h1 align="left|center|right| justify " >1 号标题文字</h1>
```

align 属性用于设置标题的对齐方式，可选值包括 left(左对齐)、center(居中对齐)、right(右对齐)和 justify(两端对齐)。

3.1.3　添加空格与特殊符号

在 HTML 文档中，连续的空格或制表符通常会被浏览器忽略，只显示为一个空格。为了满足页面设计的需要，我们可以使用特定的 HTML 实体来表示空格和特殊符号。

基本语法：

```
<body>
     &lt;&reg;&times;
</body>
```

注意，在 body 标签内无论输入多少空格、回车符，页面在显示时都会忽略。

在 HTML 文件中插入的特殊符号(与空格相同)如表 3-1 所示。

表 3-1　特殊符号

显示结果	说明	符号代码	编码
	显示一个空格		
<	小于	&lt;	&#60;
>	大于	&gt;	&#62;
&	&符号	&amp;	&#38;
"	双引号	&quot;	&#34;
©	版权	&copy;	&#169;
®	注册商标	&reg;	&#174;
×	乘号	&times;	&#215;
÷	除号	&divide;	&#247;

【例 3-1-1】Web 页面初步设计，代码如下所示。

```
1   <!-- exam3_1_1.html -->
2   <!DOCTYPE html>
3   <html lang="en">
4     <head>
5       <meta charset="utf-8">
6       <title>Web 页面初步设计</title>
7     </head>
8     <body>
9       <h1 align="center">学好 Web 能干什么</h1>
10        <h2 align="left">就业前景</h2>
11            目前在 IT 公司中，前端的岗位越来越不可或缺，
12        前端的地位也愈见明显，很多网页设计都离不开这项技术
```

```
13            <hr color="blue">
14        </body>
15    </html>
```

第 9 至 13 行展示了如何使用 HTML 的基本语法来构建一个简单的 Web 页面。在<body>标签内，首先使用<h1>标签居中显示了一个主标题“学好 Web 能干什么”；其次使用<h2>标签左对齐显示了一个次级标题“就业前景”；在次级标题下方，通过使用 实体插入了连续的空格，以实现段落的缩进效果，并随之展示了一段关于 Web 前端开发就业前景的描述性文字；最后使用<hr>标签插入了一条蓝色的水平分隔线，以在视觉上区分页面的不同部分。

页面效果如图 3-1 所示。

图 3-1　Web 页面初步设计效果图

3.2　格式化文本标签

HTML 中提供了很多格式化文本的标签，如文字加粗、斜体、下画线、底纹、上下标等。

3.2.1　文本修饰标签

常见的文本修饰标签如表 3-2 所示。

表 3-2　常见的文本修饰标签

标签	说明
加粗	加粗
斜体	斜体
<u>下画线</u>	下画线
<s>删除线</s>	删除线
^{上标}	上标
_{下标}	下标

【例 3-2-1】文本修饰标签的应用，代码如下所示。

```
1    <!-- exam3_2_1.html -->
2    <!doctype html>
```

```
<html lang="en">
  <head>
    <meta charset="UTF-8">
    <title>文本修饰标签的应用</title>
  </head>
  <body>
    <b>Web 真是一门有意思的课程</b><br>
    <i>Web 真是一门有意思的课程</i><br>
    <u>Web 真是一门有意思的课程</u><br>
    <s>Web 真是一门有意思的课程</s><br>
    <sup>Web</sup>真是一门有趣的<sub>课程</sub>
  </body>
</html>
```

页面效果如图 3-2 所示。

图 3-2　文本修饰标签的应用

3.2.2　字体标签

在不指定任何样式的情况下，IE 浏览器会把字体显示为“3 号、黑色、宋体”。因此，在设计网页时，需根据需求更改字体。在 HTML5 中，可以使用 CSS 的字体属性作为替代方案。

font 标签规定了文本的字体系列、字体尺寸、字体颜色，所有浏览器均支持 font 标签。

基本语法：

```
<font face="字体" size="尺寸" color="颜色">文本</font>
```

在该标签中，face 属性负责定义字体样式，size 属性用于调整字体尺寸，而 color 属性则用于指定字体颜色。值得注意的是，尽管 font 标签在早期 HTML 规范中被广泛采纳，但在 HTML5 标准中，推荐使用 CSS 来取代 font 标签进行文本样式的设置，因为 CSS 提供了更加多样和灵活的文本格式化选项。

属性说明：

- color。

定义：规定元素中文本的颜色，可使用 rgb 函数、十六进制数和颜色英文名称。

示例：This is some text!，将文本颜色设置为红色。

- face。

定义：规定元素中文本的字体。

示例：This is some text!，将文本字体设置为 Arial。

- size。

定义：规定元素中文本的尺寸大小。其值可以是从 1 到 7 的数字，浏览器默认值为 3。

示例：This is some text!，将文本大小设置为 5。

【例 3-2-2】字体标签的应用，代码如下所示。

```
<!-- exam3_2_2.html -->
<!doctype html>
<html lang="en">
  <head>
      <meta charset="UTF-8">
    <title>登鹳雀楼</title>
  </head>
  <body>
    <strong>登鹳雀楼</strong><br>
     <u>王之涣</u><br>
     <font face="宋体" size="3" color="#ff5533">白日依山尽</font><br>
     <font face="黑体" size="-1" color="#000fff">黄河入海流</font><br>
     <font face="仿宋" size="-3" color="#ff5533">欲穷千里目</font><br>
     <font face="黑体" size="-6" color="#0055ff">更上一层楼</font><br>
  </body>
</html>
```

页面效果如图 3-3 所示。

图 3-3　字体标签的应用

3.3　段落与排版标签

3.3.1　段落标签

段落标签使用 标签来定义文本段落。在 HTML 中，每个段落均应由 标签起始，并以 标签终止。浏览器会自动在段落前后添加空行，以确保文本内容的清晰度和易读性。当我们在网页上展示连续的文本段落时，恰当运用 标签能够有效地组织和分隔内容，使页面布局更为整洁、有序。

基本语法：

```
<p align="left|center|right">段落正文内容</p>
```

align 属性可用于设置段落的对齐方式，其可选值包括 left(左对齐)、center(居中对齐)及 right(右对齐)。需要注意的是，与标题字标签相似，尽管 HTML 对 align 属性的值不区分大小写，但为了保持代码的一致性和提高可读性，在编写代码时建议采用全部小写字母的形式。

3.3.2　换行标签

在 HTML 中，若想在某个位置实现换行，且无须开启新的段落，可以使用换行标签
或</br>。该标签是一个空标签，即它没有结束标签。使用换行标签后，浏览器会在该标签所在的位置插入一个换行符，从而改变文本的显示流。

基本语法：

```
<br>或</br>
```

换行标签同样支持 align 属性，借助该属性我们能够指定换行后文本的对齐方式，包括 left(左对齐)、center(居中对齐)和 right(右对齐)。不过，与段落标签<p>的 align 属性一样，随着 HTML5 标准的推广，推荐使用 CSS 来控制文本的对齐，因为 CSS 提供了更加灵活且强大的布局选项。

3.3.3　水平分隔线标签

在 HTML 中，可使用<hr>或</hr>标签来插入水平分隔线，它呈现为一条水平线，通常用于分隔内容或创建视觉上的间隔。水平分隔线标签是一个空标签，这意味着它不需要结束标签。借助水平分隔线标签，能够在网页上便捷地添加水平分隔线，从而增强页面的可读性和视觉效果。

基本语法：

```
<hr>或</hr>
```

此外，水平分隔线标签还支持多个属性，通过这些属性可以自定义分隔线的样式，包括宽度(width)、颜色(color)、对齐方式(align)、阴影大小(size)等。不过，随着 HTML5 标准的推广，推荐使用 CSS 来控制这些样式属性，因为 CSS 提供了更加多样和精细的控制选项。

3.3.4　段落缩进标签

在 HTML 文档中，为了提升美观度和可读性，有时需要对段落文本进行缩进处理，而段落缩进标签正是为此而设计的。

基本语法：

```
<blockquote></blockquote>
```

<blockquote>标签用于实现段落的缩进效果，通过将需要缩进的文本内容包裹在<blockquote>和</blockquote>标签之间，浏览器会自动为这些文本添加缩进，从而在视觉上区分出不同的段落或引用内容。这种缩进效果通常用于表示引用、注释或强调某些重要的文本信息。

3.3.5 预格式化标签

预格式化标签使用<pre>标签来定义。该标签的主要功能是保留文本中的空格和换行符，并原样呈现文本内容。这在展示程序代码、ASCII 艺术或其他需要精确布局和格式的内容时非常有用。使用<pre>标签后，浏览器会保留文本中的所有空白字符，包括空格、制表符和换行符，从而确保内容的准确呈现。

基本语法：

```
<pre>预格式化的文本内容</pre>
```

例如，若我们想在网页上展示一段 HTML 代码，并希望浏览器能够按照代码的原始格式显示，则可以使用<pre>标签。这样，代码中的缩进、空格和换行都会被保留下来，使得代码更易于阅读和理解。

需要注意的是，虽然<pre>标签能够为我们提供预格式化的文本显示功能，但在使用时仍需谨慎。因为过多的预格式化文本可能会影响页面的整体布局和美观性。因此，在设计网页时，我们需要根据实际需求和内容特点来合理地使用<pre>标签。

【例 3-3-1】段落与排版标签的应用，代码如下所示。

```
<!-- exam3_3_1.html -->
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8">
    <title>将进酒</title>
  </head>
  <body>
    <h1 align="center">将进酒</h1>
     <pre>
                    将 进 酒
                      唐·李白
        君不见，黄河之水天上来，奔流到海不复回。
        君不见，高堂明镜悲白发，朝如青丝暮成雪。
            人生得意须尽欢，莫使金樽空对月。
            天生我材必有用，千金散尽还复来。

      </pre>
      <hr>
      <p>《将进酒》原是汉乐府短箫铙歌的曲调，题目意绎即“劝酒歌”，故古词有“将进酒，乘大白”云。<br>人生快事莫若置酒会友，作者又正值“抱用世之才而不遇合”(萧士赟)之际，于是满腔不合时宜借酒兴诗情，来了一次淋漓尽致的抒发。</p>
  </body>
</html>
```

在上述代码中，我们使用了<pre>标签来展示《将进酒》这首诗，通过预格式化标签保留了文本中的空格和换行，使得诗歌的原始格式得以准确呈现。在<pre>标签内部，所有的空格、制表符和

换行符都会被浏览器保留并显示出来，这对于展示程序代码、ASCII 艺术或其他需要精确布局的内容来说是非常有用的。

上述代码的运行效果，如图 3-4 所示。

图 3-4　段落与排版标签应用效果图

3.4 图像

在网页设计中，图像扮演着举足轻重的角色，其能增强页面的视觉效果、传达丰富的信息，并能提升用户体验。HTML 提供了一系列标签和属性，使得在网页中插入和管理图像变得简便而高效。

3.4.1　插入图像

在 HTML中，插入图像的基本标签是<img>。该标签是一个单标签，意味着它不需要结束标签。通过使用<img>标签，我们可以轻松地在网页上添加图片，从而丰富页面的内容和视觉效果。

基本语法：

```
<img src="URL" alt="替代文本" width="宽度" height="高度" border="边框" align="对齐方式" hspace="水平间距" vspace="垂直间距">
```

在这些属性中，src 属性用于指定图像的来源 URL，是必需的。alt 属性用于提供图像的替代文本，当图像无法加载时，这段文本将被显示出来，同时也有助于提升网页的可访问性。width 和 height 属性用于设置图像的宽度和高度，可以像素(px)或百分比(%)为单位。border 属性用于指定图像边框的宽度，而 align 属性则用于设置图像的对齐方式，包括 left(左对齐)、center(居中对齐)、right(右对齐)等。此外，hspace 和 vspace 属性分别用于设置图像周围的水平间距和垂直间距。

3.4.2　设置图像的替代文本

替代文本是通过 alt 属性指定的内容，在图像无法显示时尤为重要。它不仅能向用户传达图像的基本信息，还能提升网页的可访问性，使得屏幕阅读器等辅助技术可以正确解读页面内容。因此，在编写 HTML 代码时，为图像添加准确且描述性的替代文本是一项不可忽视的任务。