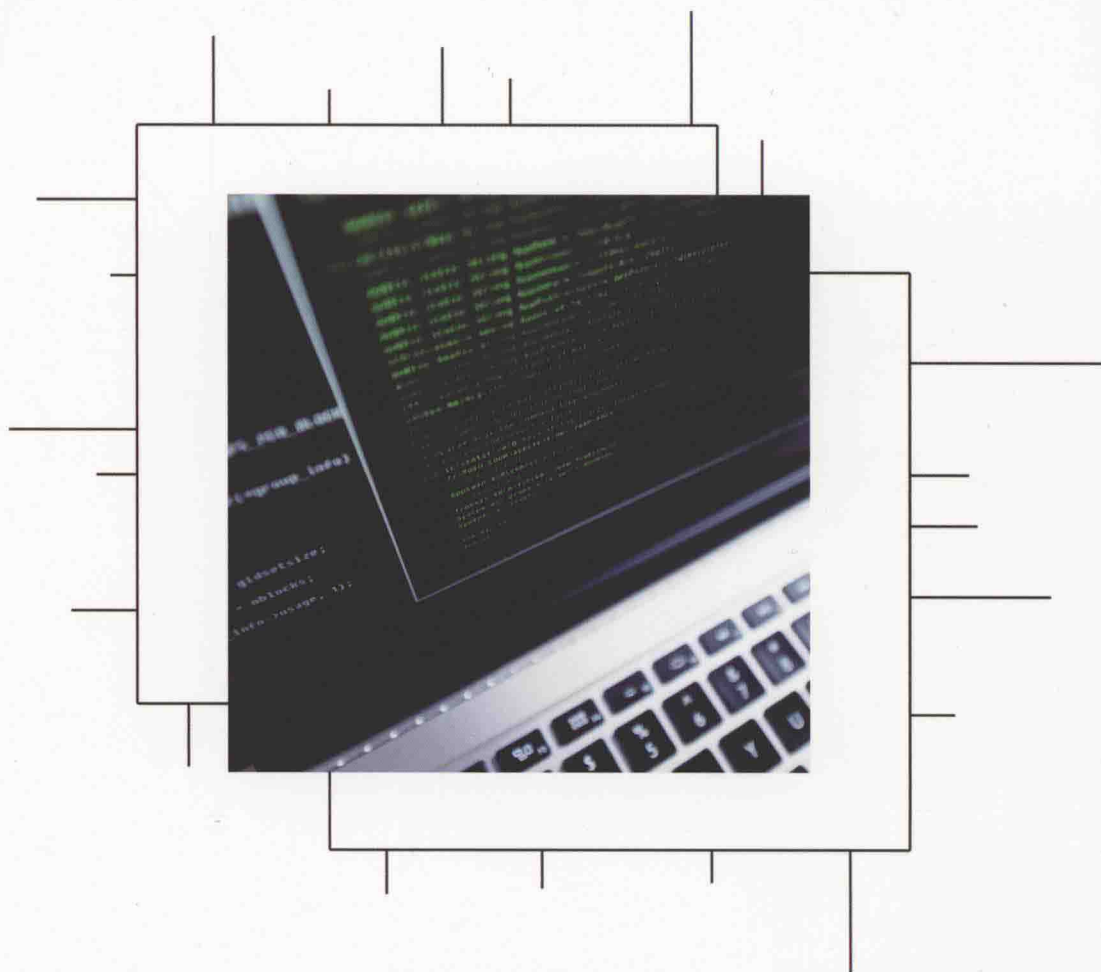




高等院校计算机类规划教材

全国高等院校计算机基础教育研究会立项项目成果



# HTML5、CSS3、JavaScript 程序设计实用教程

主 编 周子程 王志海



北京邮电大学出版社  
www.buptpress.com



高等院校计算机类规划教材

全国高等院校计算机基础教育研究会立项项目成果

# HTML5、CSS3、JavaScript 程序设计实用教程

主编 周子程 王志海



北京邮电大学出版社  
[www. buptpress. com](http://www.buptpress.com)

## 内 容 简 介

本书介绍了 HTML5、CSS3 与 JavaScript 相关的程序设计方法,书中对知识点做了详细的分析与解读,通俗易懂,便于学习。全书共分 9 章,第 1~2 章介绍了 HTML,包括 HTML5 概述、通过 HTML 创建表格与表单;第 3~5 章介绍了 CSS,包括 CSS 概述和 CSS 应用;第 6~8 章介绍了 JavaScript 基础与对象、DOM 对象与 BOM 对象;第 9 章介绍了基于框架的程序设计。书中的知识点都结合了案例进行介绍,并对案例做了适当的说明,方便读者快速掌握网页程序设计方法。

本书可以作为大学计算机及相关专业的网页设计基础类课程的教材,也适用于有一定程序设计基础的初学者。

### 图书在版编目 (CIP) 数据

HTML5、CSS3、JavaScript 程序设计实用教程 / 周子程,王志海主编. -- 北京:北京邮电大学出版社, 2022. 6

ISBN 978-7-5635-6636-5

I. ①H… II. ①周… ②王… III. ①超文本标记语言—程序设计—教材②JAVA 语言—程序设计—教材 IV. ①TP312. 8

中国版本图书馆 CIP 数据核字(2022)第 067932 号

策划编辑:马晓仟 刘纳新 责任编辑:廖娟 封面设计:七星博纳

出版发行:北京邮电大学出版社

社 址:北京市海淀区西土城路 10 号

邮政编码:100876

发行部:电话:010-62282185 传真:010-62283578

E-mail: publish@bupt. edu. cn

经 销:各地新华书店

印 刷:保定市中华美凯印刷有限公司

开 本:787 mm×1 092 mm 1/16

印 张:13

字 数:335 千字

版 次:2022 年 6 月第 1 版

印 次:2022 年 6 月第 1 次印刷

ISBN 978-7-5635-6636-5

定价:36.00 元

· 如有印装质量问题,请与北京邮电大学出版社发行部联系 ·

# 前 言

随着互联网技术的不断发展,以互联网为背景的行业随之兴起,不同主题的 Web 应用也愈来愈多,而 Web 前端程序设计作为 Web 应用中的重要一环,往往是计算机科学、软件工程、信息管理、电子商务等计算机相关专业的的基础核心课程之一。

全书共 9 章,内容主要分为 HTML5、CSS3 和 JavaScript 三部分。由于三者关系并不完全独立,后续章节中的案例往往需要结合前面章节中的知识点演示,因此为了帮助初学者快速掌握网页程序设计方法,本书在内容的编排上做了适当的取舍。其中,第 1~2 章介绍 HTML,主要内容包括 HTML 的文档结构与编写方法、HTML 常用标签以及如何利用 HTML 创建表格与表单等。第 3~5 章介绍 CSS,主要内容包括 CSS 用法,如何使用 CSS 的各种选择器并通过修改样式属性来美化网页元素,盒子模型,各种网页布局方法等。第 6~8 章介绍 JavaScript,考虑到大部分高校的计算机相关专业会将 C 语言或 Java 语言作为程序设计的入门语言,所以第 6 章对 JavaScript 基础中与其他语言相似的部分做了适当的取舍,以便读者能够更早地结合 HTML5 与 CSS3 等内容体会网页设计的乐趣。第 7~8 章讲解了 JavaScript 中的内置对象、BOM 对象和 DOM 对象,并结合实例介绍了对象的用法。第 9 章介绍了部分 JavaScript 框架及用法,以帮助读者更高效地进行网页设计。

本书在编写过程中得到了多位老师的帮助与指导,在此表示衷心的感谢。本书的出版由全国高等院校计算机基础教育研究会 2019 年度计算机基础教育教学研究项目(项目编号:2019-AFCEC-005)资助。

由于编者水平有限,书中难免存在不当之处,请读者批评指正。

编 者

# 目 录

<b>第 1 章 HTML5 概述</b> .....	1
1.1 HTML5 概述 .....	1
1.2 HTML 文档结构和编写方法 .....	2
1.2.1 HTML 文档结构 .....	2
1.2.2 HTML 文档的编写方法 .....	2
1.3 HTML 常用标签与全局属性 .....	4
1.3.1 头部标签 .....	4
1.3.2 主体标签 .....	7
1.3.3 文本标签 .....	8
1.3.4 列表标签 .....	10
1.3.5 图像标签 .....	12
1.3.6 超级链接标签 .....	15
1.3.7 热点标签 .....	18
1.3.8 区块元素标签与内联元素标签 .....	19
1.3.9 注释标签 .....	20
1.3.10 全局属性 .....	20
本章小结 .....	20
练习题 .....	21
<b>第 2 章 创建表格、表单与框架</b> .....	22
2.1 表格 .....	22
2.1.1 表格的结构 .....	22
2.1.2 完整的表格标签 .....	25
2.1.3 使用表格布局 .....	28
2.2 表单 .....	29
2.3 框架 .....	39
本章小结 .....	40
练习题 .....	41

<b>第 3 章 CSS 概述</b> .....	42
3.1 CSS 介绍 .....	42
3.2 CSS 语法 .....	42
3.3 CSS 用法 .....	43
3.3.1 行内样式 .....	44
3.3.2 内部样式表 .....	45
3.3.3 外部样式表 .....	45
3.3.4 导入样式 .....	47
3.4 CSS 选择器 .....	48
3.4.1 标签选择器 .....	48
3.4.2 类选择器 .....	48
3.4.3 ID 选择器 .....	49
3.4.4 伪类选择器 .....	51
3.4.5 属性选择器 .....	55
3.4.6 选择器的组合 .....	57
3.4.7 选择器的分组与继承 .....	59
本章小结 .....	60
练习题 .....	60
<b>第 4 章 CSS 属性与应用</b> .....	61
4.1 文本与背景样式 .....	61
4.1.1 文本效果 .....	61
4.1.2 背景效果 .....	67
4.1.3 列表样式 .....	72
4.2 表格与表单样式 .....	74
4.2.1 边框与表格 .....	74
4.2.2 表单样式 .....	80
本章小结 .....	83
练习题 .....	83
<b>第 5 章 盒子模型与网页布局</b> .....	85
5.1 盒子模型 .....	85
5.1.1 外边距 .....	85
5.1.2 内填充 .....	89
5.1.3 定位属性 .....	90

5.1.4 浮动属性	96
5.2 页面布局	97
5.2.1 圣杯布局	97
5.2.2 双飞翼布局	99
5.2.3 Flex Box 布局	101
本章小结	108
练习题	109
<b>第 6 章 JavaScript 基础</b>	<b>110</b>
6.1 JavaScript 简介	110
6.1.1 JavaScript 的构成	110
6.1.2 使用 JavaScript	110
6.2 JavaScript 语法基础	113
6.2.1 数据类型	113
6.2.2 变量	119
6.2.3 运算符	121
6.2.4 流程控制	122
6.2.5 函数	125
6.3 事件	128
本章小结	139
练习题	139
<b>第 7 章 JavaScript 对象</b>	<b>141</b>
7.1 自定义对象	141
7.2 String 对象	142
7.3 Array 对象	150
7.4 Math 对象	157
7.5 Number 对象	159
7.6 Date 对象	160
本章小结	162
练习题	162
<b>第 8 章 BOM 与 DOM</b>	<b>163</b>
8.1 BOM	163
8.1.1 Window 对象	163
8.1.2 Location 对象	172

8.1.3 History 对象 .....	173
8.1.4 Screen 对象 .....	173
8.2 DOM .....	173
8.2.1 HTML DOM 简介 .....	174
8.2.2 Document 对象 .....	174
8.2.3 操作元素 .....	176
本章小结 .....	184
练习题 .....	185
<b>第 9 章 基于框架的程序设计</b> .....	<b>186</b>
9.1 jQuery .....	186
9.1.1 使用 jQuery .....	186
9.1.2 jQuery 用法 .....	187
9.2 ECharts .....	192
9.2.1 使用 ECharts .....	192
9.2.2 ECharts 用法 .....	193
本章小结 .....	196
练习题 .....	196
<b>参考文献</b> .....	<b>197</b>

# 第 1 章 HTML5 概述

Web 应用程序可以利用网页文件、数据交互技术与多媒体资源等向用户呈现丰富的信息。HTML 是一种标记语言,HTML5 是在 HTML 基础上的新标准,增加了一些新的标签与特性。在网页文件中,如用户登录注册信息的页面,会显示输入框和按钮等控件,这些可以通过 HTML 标签来实现,在第 2 章中会有更详细的说明。本章将介绍 HTML 基础与 HTML5 中增加的新标签,使读者进一步了解 HTML,从而为网页设计打下基础。

## 1.1 HTML5 概述

我们常见的网站是由网页组成的,网页和网页的链接都有对应的 URL(统一资源定位符,用以定位互联网上的资源)。如同计算机中的一张图片素材,我们可以通过它的物理路径找到它,如图 1-1 所示。

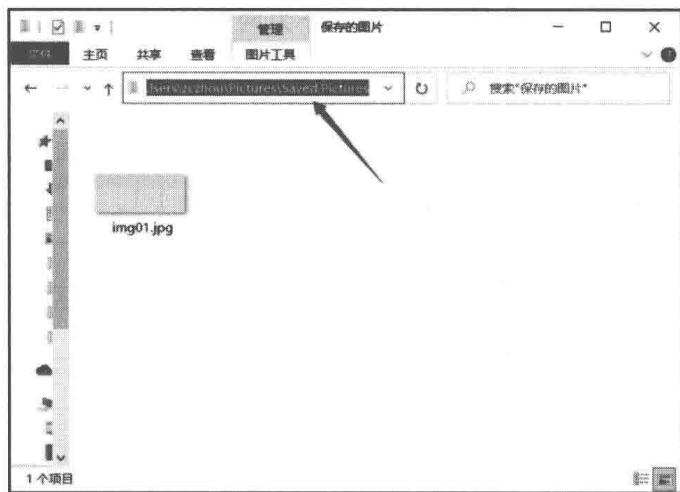


图 1-1 图片路径

网页被分类为静态网页和动态网页。Web 1.0 时代,人们访问的大多是静态网页,这类网页中的数据只能单方面向用户流动,所以网站的主题较为局限,用户访问的网站类别常常局限于纯粹的新闻类或资讯类。

发展到 Web 2.0 时代,网站更注重用户交互,并且用户既是网站内容的消费者,也是内容的创造者,如微博、优酷等社交类的网站,这种网站采用动态网页技术,即以数据库技术为基础完成大量的数据存取等操作。Web 3.0 时代则更好地将人工智能、数据挖掘等一系列技术融入其中,对数据进行定向挖掘,从而塑造用户人物特征,并试图从互联网上的海量信息中搜寻适合用户的部分并反馈,但无论是 1.0 还是 3.0 都需要遵循 Web 标准。

Web 标准是将各部分标准组合在一起,这些标准大部分由 W3C 负责制定,也有一些标准

由其他标准组织制定的,如 ECMA 的 ECMAScript 标准等。本书围绕 HTML5、CSS3、JavaScript 介绍,它们三者的关系如图 1-2 所示。



图 1-2 网页开发技术结构关系图

## 1.2 HTML 文档结构和编写方法

### 1.2.1 HTML 文档结构

静态网页文档的扩展名为 .html 或者 .htm,可以采用任何的文本编辑器进行开发。一个纯 HTML 文档是由一系列的标签组成的,HTML 的标签用来限定元素在文档的位置。这些标签常利用<>符号和标签名组合而成,其语法格式为:<标签名>数据</标签名>。

一个 HTML 文档以< HTML >为开始,以</HTML >为结束。值得注意的是,HTML 中有些标签并不是成对出现的。HTML 文档又可分为两部分,分别是文档头和文档体,文档头使用< HEAD ></HEAD >定义,在文档头中可以指定文档的某些属性;文档体使用< BODY ></BODY >定义,用以指定文档中要显示的内容和结构,是文档的主要部分。标签之间常采用并列、嵌套等关系来完成结构设计,虽然标签不区分大小写,但作为网页开发者最好将其统一。

**【例 1-1】** 采用基本结构标记文档的 HTML 文档 code 1-1。

```
<html >
<head >
    <title >第一个 html 文档</title >
</head >
<body >
    <h1 >欢迎学习 html 基础</h1 >
</body >
</html >
```

HTML 从初期到现在经历了很多版本,我们需要利用<! DOCTYPE >声明来告知浏览器目前的 HTML 文档是基于哪一个版本,<! DOCTYPE >声明位于文档最前面的位置,处于<html >标签之前。HTML5 的文档声明为<! DOCTYPE html >。而 HTML 4.01 中,<! DOCTYPE >声明需引用 DTD(文档类型声明),分别是 Strict、Transitional 和 Frameset。这里的建议是今后在编辑 HTML 文档时,尽量都加上<! DOCTYPE >声明来确保浏览器可以预先明确文档类型,以便文档可以正确显示。

### 1.2.2 HTML 文档的编写方法

常用的编写方法分为两种:一种是手动编写 HTML 文档,这种方式有助于初学者掌握常

用的标签;另一种是采用 HTML 编辑工具编写,这种方式更高效,适用于有一定基础的网页开发人员。

### 1. 采用记事本编写 HTML 文档

具体步骤如下。

1) 在 Windows 桌面右击,创建文本文档 hello.txt。打开记事本后,输入例 1-1 中的代码,如图 1-3 所示。

2) 编辑完 HTML 代码后,使用“CTRL+S”组合键保存文件,单击“文件”→“另存为”,在对话框中将“保存类型”选择为“所有文件”,修改 hello.txt 为 hello.html 并保存,如图 1-4 所示。

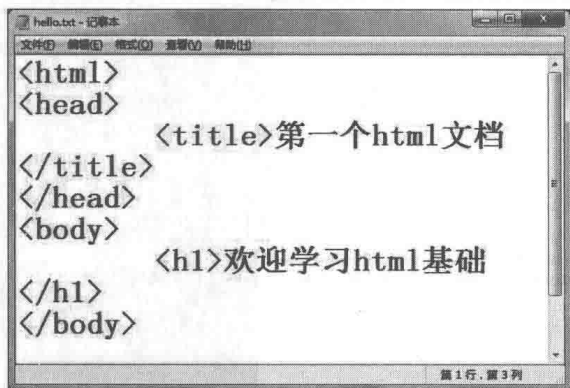


图 1-3 编辑代码



图 1-4 修改文件类型

3) 在桌面找到 hello.html 文档,使用浏览器打开后的效果如图 1-5 所示。



图 1-5 网页浏览效果

如果想要再次编辑当前网页,可以在桌面找到 hello.html 文档右击,选择“打开方式”,在弹出的菜单中选择“选择默认程序”,然后在弹出的对话框中找到“记事本”即可。将编辑好的代码保存后,在浏览器中重新刷新即可浏览。

2. 采用 Dreamweaver CC 2019 编写 HTML 文档  
具体操作步骤如下。

- 打开 Dreamweaver CC 2019, 初次进入软件可以根据向导进一步了解软件的使用方法。
- 了解完向导后, 单击“文件”→“新建”→“新建文档”→“HTML”, 文档类型可以选择其他标准, 这里我们不做选择, 采用默认的 HTML5 标准, 单击“创建”, 进入编辑界面。如图 1-6 所示。

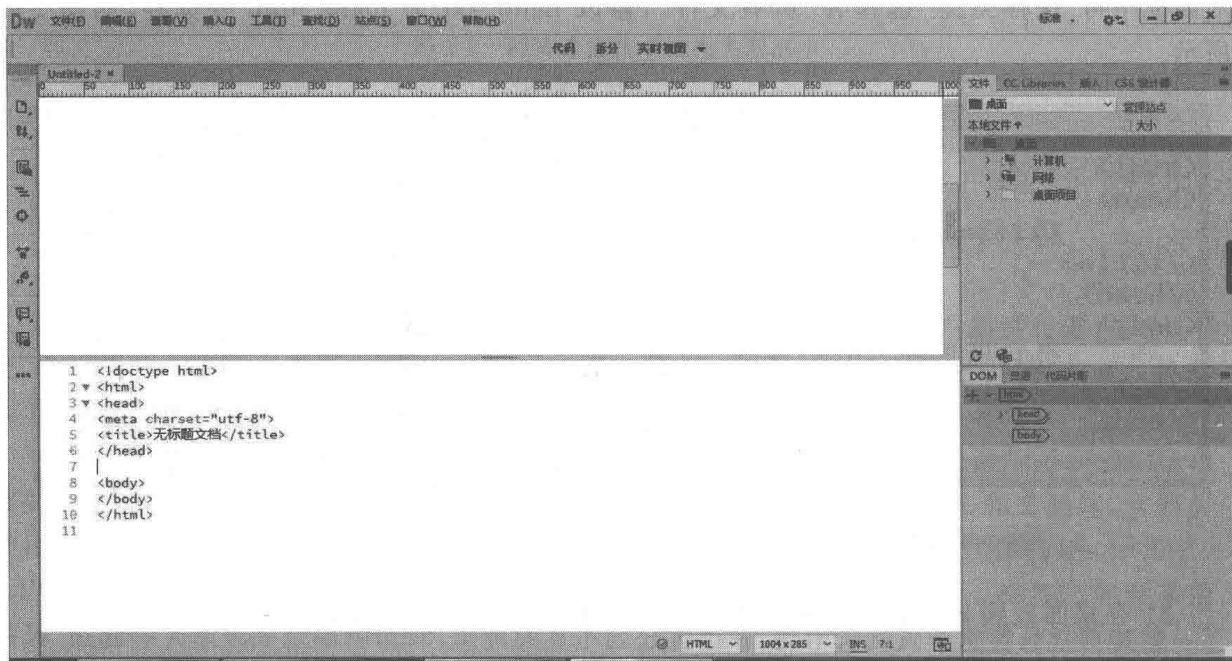


图 1-6 代码编辑界面

- 在当前界面, 默认采用拆分的形式显示, 上半部分是设计区域, 会实时显示代码编辑的效果; 下半部分属于代码编辑区域。请根据例 1-1 修改代码, 观察实时显示的结果。

由于一些辅助开发工具支持对代码的自动化生成, 在未完全掌握文档结构与常用标签前, 不建议使用 Dreamweaver 或类似的工具进行 HTML 代码编辑。

## 1.3 HTML 常用标签与全局属性

HTML 的常用标签有头部标签、主体标签、文本标签、列表标签、图像标签、超级链接标签、热点标签、作为容器的区块和内联标签、注释标签等。

### 1.3.1 头部标签

`<head>` 作为容器标签包含了所有的头部标签。在 `<head>` 元素中可以插入 JavaScript 脚本 (scripts)、CSS 样式文件 (style) 及各种 meta 信息。

可以添加在头部区域的单标签包括 `<meta>` 和 `<link>`, 双标签包括 `<title></title>`、`<style></style>`、`<script></script>`、`<noscript></noscript>` 和 `<base></base>`。

## 1. <title>标签

<title>定义了不同文档的标题信息。当添加当前网页到收藏夹时,默认标题会显示在收藏夹中。执行例 1-1 代码后,谷歌浏览器中标题显示位置如图 1-7 所示。



图 1-7 标题显示位置

## 2. <meta>标签

<meta>元素通常用于指定网页的描述、关键词,文件的最后修改时间、作者和其他元数据。元数据不显示在页面上,但会被浏览器解析。元数据可以用于浏览器(如何显示内容或重新加载页面)、搜索引擎(关键词)或其他 Web 服务。值得注意的是,在 HTML 中,<meta>元素没有结束标签</meta>且永远位于 head 元素内部。

HTML 标签可以拥有属性。属性提供了有关 HTML 元素的更多的信息。属性总是通过键值对的形式出现,如 `color = "red"`。meta 元素相关属性的简单介绍如表 1-1 所示。

表 1-1 <meta>标签中的属性与对应值

属性	属性值	说明
charset	character_encoding	定义文档的字符编码
content	text	定义与 http-equiv 或 name 属性相关的元信息
http-equiv	content-type default-style refresh	把 content 属性关联到 HTTP 头部
name	application-name author description generator keywords	把 content 属性关联到一个名称

### 1) charset 属性

charset 属性可以设定 HTML 文档的字符编码。浏览器在获取 HTML 文档之后,会根据

指定的编码方式对文档解码,如果文档的编码方式与指定的编码方式不一致,则会出现乱码,中英文混编常用的编码格式为 utf-8 和 gb2312,其使用方法如下。

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

## 2) content 属性

content 属性提供了名称/值对中的值。该值可以是任何有效的字符串,并且 content 属性始终要和 name 属性或 http-equiv 属性一起使用。如例 1-2 所示。

**【例 1-2】** content 属性用法 code 1-2。

```
<!DOCTYPE html >
<html >
<head >
<meta http-equiv = "Content-Type" content = "text/html; charset = gb2312" >
<meta name = "keywords" content = "HTML, CSS, ,JavaScript " >
</head >
<body >

<p> The meta elements of this document describe the document and its keywords.</p>

</body >
</html >
```

在本例中,将 meta 中的属性 name 设置为 keywords,在早些时候,meta keywords 关键字对搜索引擎的排名会产生一定的影响,也是很多设计者对网页搜索优化的选择,例子中关键字为 HTML、CSS、JavaScript,这是我们在浏览网页时无法看到的。

## 3) http-equiv 属性

http-equiv 属性用于指定头部协议类型,我们可以利用它向浏览器传回一些有用的信息,以帮助正确和精确地显示网页内容,content 属性用于指定头部协议类型的值。其用法如下。

```
<meta http-equiv = "参数" content = "参数变量值">
```

如表 1-1 所示,http-equiv 可选取的值主要包括 content-type、default-style 和 refresh,其中 content-type 属性值搭配 content 使用可用于定义用户的浏览器或相关终端以哪种方式加载数据,原因在于 http 协议采用的是请求/响应模型,客户端向服务器端发送一个请求,该请求由请求行、请求头和请求体组成,其中请求头包含请求的方法、URI、协议版本,以及请求修饰符、客户信息和内容的类似于 MIME 的消息结构。服务器以一个状态行作为响应,相应的内容包括消息协议的版本、成功或者错误编码、服务器信息、实体元信息以及可能的实体内容。Content-Type 是返回消息中非常重要的内容,表示后面的文档属于哪种 MIME 类型。示例如下。

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

可用于以网页形式打开该资源,并设定网页的编码格式为 UTF-8,这里的 UTF-8 是针对

Unicode 的一种可变长度字符编码。MIME 类型有很多种,如以 Office 文件方式加载、XML 方式加载和二进制流数据加载等。

default-style 属性值可以用于设定要使用的预定义的样式表。示例如下。

```
<meta http-equiv = "default-style" content = "the document 's preferred stylesheet">
```

refresh 属性值可以用于定义文档自动刷新的时间间隔。如例 1-3 所示。

**【例 1-3】** refresh 属性用法 code 1-3。

```
<!DOCTYPE html >
<html >
<head >
<meta http-equiv = "refresh" content = "3;url = 'https://www.baidu.com'"/>
  <title>refresh 用法</title>
</head >
<body >
  <h1 >3 秒后将自动刷新跳转到百度首页</h1 >
</body >
</html >
```

例 1-3 中指定在 3 秒后跳转到 URL 对应的页面,其中 URL 对应的值也可以是本地文件,但要注意路径问题。不指定跳转页面可以用如下写法。

```
<meta http-equiv = "refresh" content = "3"/>
```

#### 4) name 属性

关于 name 属性对应属性值的说明如表 1-2 所示。

表 1-2 name 属性对应属性值及说明

属性值	说明
application-name	设定页面所代表的 Web 应用程序的名称
author	设定文档的作者的名字
description	设定页面的描述。搜索引擎会把这个描述显示在搜索结果中
generator	设定用于生成文档的一个软件包(不用于手写页面)
keywords	设定一个逗号分隔的关键词列表,具体用法如例 1-2 所示

### 1.3.2 主体标签

在网页中,用于显示内容的元素都会放在主体标签内部,形成嵌套结构,可以将其想象成一张很大的屏幕,图片、视频和文字等内容都可在屏幕中展示。屏幕可大可小,屏幕的底色可白可灰,这些都可以通过 body 的属性来控制。主体标签的语法格式如下。

```
<body 属性 = "属性值">元素...</body >
```

**【例 1-4】** body 属性用法 code 1-4。

```
<!DOCTYPE html >
<html >
<head ></head >
<body bgcolor = "yellow"><h1>修改 body 背景颜色为黄色</h1 ></body >
</html >
```

虽然可以通过标签的属性对样式进行修改,但在学习 CSS 部分后,更推荐使用 CSS 来完成样式的设定。

### 1.3.3 文本标签

文本和图片是网页中内容输出的主要元素,如同在 Office Word 中对文本格式化一样,在 HTML 中也可以设置文字的字体、颜色、大小等。本节将文本标记分为三个部分,分别是文本标记、文本样式标记和文本排版标记。由于篇幅关系,这里仅列出部分标签,如表 1-3 所示。

表 1-3 文本标签及说明

类型	标签名	说明
文本标记	font	字体标签,H5 不支持,使用 CSS 代替
文本样式标记	h1~h6	标题标签,h1 做主标题,h2 做次级标题,顺次排列
	b	定义加粗文本
	i	定义斜体文本
	sub/sup	定义下标/上标
文本排版标记	p	定义段落标记
	br	定义换行标记,单体标签

#### 1. 标题标签

通过<h1>~<h6>标签来定义标题,<h1>相当于 Office Word 中的一级标题,<h6>相当于最小的标题。需要注意的是,标题不仅能让文本形成加粗和增大字号的效果,而且搜索引擎会使用标题为网页的结构和内容编制索引,所以可以用标题来调整文本结构。

**【例 1-5】** 标题标签用法 code 1-5。

```
<!DOCTYPE html >
<html >
<head >
  <meta http-equiv = "content-type" content = "text/html; charset = UTF-8">
  <title>标题标签用法</title >
</head >
<body >
  <h1 ><i >一级标题</i ></h1 >
  <h2 >二级标题</h2 >
  <h3 >三级标题</h3 >
  <h4 >四级标题</h4 >
```

```

<h5>五级标题</h5>
<h6>六级标题</h6>
</body>
</html>

```

例 1-5 执行的效果如图 1-8 所示。

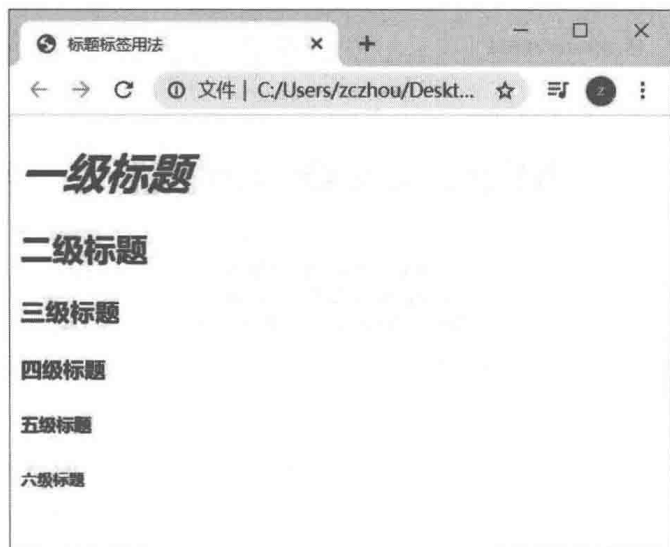


图 1-8 标题标签用法执行结果

## 2. 文本排版标记

换行标签 `<br>` 是一个独立的标签,没有结束标记,它能强制换行一次。我们可以利用 `<p></p>` 标签来定义一个段落。值得注意的是,段落标签是有结束标记的,如果希望在文档中有比较明确的分割线,我们可以使用 `<hr>` 标签来完成, `<hr>` 标签可定义 HTML 页面中的一条水平线,用以分割新内容。 `<center></center>` 标签可以定义居中格式,例 1-6 中将上述标签做了综合演示。

**【例 1-6】** 文本排版标记 code 1-6。

```

<!DOCTYPE html >
<html >
<head >
  <meta http-equiv = "content-type" content = "text/html; charset = UTF-8">
  <title>文本排版标签用法</title>
</head >
<body >
  <center >< font color = "red">< h1 >Microsoft Office 软件</h1 ></font ></center >
  <hr >
  <p align = "center">
    Microsoft Office Word <br >
    Microsoft Office Excel <br >
    Microsoft Office PowerPoint <br >

```