HTML5+CSS3笔记

|  |  |
| --- | --- |
| 班级： | 183110701 |
| 学号： | 18311070114 |
| 姓名： | 蔺明超 |

目录

[第一章 HTML5+CSS3网页设计概述 1](#_Toc87373537)

[1.1 网页概述 1](#_Toc87373538)

[1.1.1 认识网页 1](#_Toc87373539)

[1.1.2 网页名词解释 1](#_Toc87373540)

[1.1.3 Web标准 2](#_Toc87373541)

[1.2 网页制作技术入门 2](#_Toc87373542)

[1.2.1 HTML 2](#_Toc87373543)

[1.2.2 CSS 2](#_Toc87373544)

[1.2.3 JavaScript 2](#_Toc87373545)

[1.2.4 浏览器 3](#_Toc87373546)

[1.2.5什么是浏览器内核 3](#_Toc87373547)

[第二章 初识HTML5 4](#_Toc87373548)

[2.1 HTML5的优势 4](#_Toc87373549)

[2.2 HTML5全新的结构 4](#_Toc87373550)

[2.3 标签概述 4](#_Toc87373551)

[2.3.1 标签的分类 4](#_Toc87373552)

[2.3.2 标签的关系 5](#_Toc87373553)

[2.3.3 标签的属性 5](#_Toc87373554)

[2.3.4 HTML5文档头部相关标签 5](#_Toc87373555)

[2.4 文本控制标签 5](#_Toc87373556)

[2.4.1 页面格式化标签 5](#_Toc87373557)

[2.4.2 文本样式标签 6](#_Toc87373558)

[2.4.3 文本格式化标签 6](#_Toc87373559)

[2.4.4 文本语义标签 7](#_Toc87373560)

[2.5 图像标签 8](#_Toc87373561)

[2.5.1 图像标签<img /> 8](#_Toc87373562)

[2.5.3 相对路径和绝对路径 8](#_Toc87373563)

[第三章 初识CSS3 10](#_Toc87373564)

[3.1 结构与表现分离 10](#_Toc87373565)

[3.2 CSS的优势 10](#_Toc87373566)

[3.3 CSS核心基础 10](#_Toc87373567)

[3.3.1 CSS样式规则 10](#_Toc87373568)

[3.3.2 引入CSS样式表 11](#_Toc87373569)

[3.3.3 CSS基础选择器 11](#_Toc87373570)

[3.4 设置文本样式 12](#_Toc87373571)

[3.4.1 CSS字体样式属性 12](#_Toc87373572)

[3.4.2 CSS文本外观属性 13](#_Toc87373573)

[3.5 高级特性 14](#_Toc87373574)

[3.5.1 CSS复合选择器 14](#_Toc87373575)

[3.5.2 CSS层叠性和继承性 15](#_Toc87373576)

[第四章 CSS4选择器 15](#_Toc87373577)

[4.1 属性选择器 15](#_Toc87373578)

[4.1.1 E[att^=value] 属性选择器 15](#_Toc87373579)

[4.1.2 E[att$=value]属性选择器 15](#_Toc87373580)

[4.1.3 E[att\*=value]属性选择器 16](#_Toc87373581)

[4.2 关系选择器 16](#_Toc87373582)

[4.2.1 子元素选择器 16](#_Toc87373583)

[4.2.2 相邻兄弟选择器（+、~） 16](#_Toc87373584)

[4.3 结构化伪类选择器 16](#_Toc87373585)

[4.3.1 :root选择器 17](#_Toc87373586)

[4.3.2 :not选择器 17](#_Toc87373587)

[4.3.3 :only-child选择器 17](#_Toc87373588)

[4.3.4 :first-child和:last-child选择器 17](#_Toc87373589)

[4.3.5 :nth-child(n)和:nth-last-child(n)选择器 17](#_Toc87373590)

[4.3.6 :nth-of-type(n)和:nth-last-of-type(n)选择器 17](#_Toc87373591)

[4.3.7 :empty选择器 18](#_Toc87373592)

[4.4 伪元素选择器 18](#_Toc87373593)

[第五章 盒子模型 18](#_Toc87373594)

[5.1 盒子模型相关属性 18](#_Toc87373595)

[5.1.1 边框属性 18](#_Toc87373596)

[5.1.2 内边距属性 20](#_Toc87373597)

[5.1.3 外边距属性 21](#_Toc87373598)

[5.1.4 背景属性 21](#_Toc87373599)

[5.1.5 盒子的宽与高 23](#_Toc87373600)

[5.2 盒子模型新增属性 23](#_Toc87373601)

[5.2.1 颜色透明度 23](#_Toc87373602)

[5.2.2 圆角 23](#_Toc87373603)

[5.2.3图片边框 24](#_Toc87373604)

[5.2.4 阴影 24](#_Toc87373605)

[5.2.5 渐变 25](#_Toc87373606)

[5.2.6 多背景图像 26](#_Toc87373607)

[5.2.7 修剪背景图像 27](#_Toc87373608)

[5.3 元素的类型和转换 28](#_Toc87373609)

[5.3.1 元素的类型 28](#_Toc87373610)

[5.3.2 <div>和<span>标签 28](#_Toc87373611)

[5.3.3 元素类型的转换 29](#_Toc87373612)

[5.4 块元素垂直外边距的合并 29](#_Toc87373613)

[5.4.1 相邻块元素垂直外边距的合并 29](#_Toc87373614)

[第六章 列表与超链接 29](#_Toc87373615)

[6.1 列表标签 29](#_Toc87373616)

[6.2 CSS控制列表样式 32](#_Toc87373617)

[6.3 超链接标签 34](#_Toc87373618)

[6.4 链接伪类控制超链接 35](#_Toc87373619)

[第七章 表格与表单 36](#_Toc87373620)

[7.1 表格 36](#_Toc87373621)

[7.1.1 创建表格 36](#_Toc87373622)

[7.1.2 <table>标签的属性 36](#_Toc87373623)

[7.1.3 <tr>标签的属性 38](#_Toc87373624)

[7.1.4 <td>标签的属性 38](#_Toc87373625)

[7.1.5 <th>标签及其属性 39](#_Toc87373626)

[7.1.6 表格结构 39](#_Toc87373627)

[7.2 CSS控制表格样式 40](#_Toc87373628)

[7.2.1 CSS控制表格边框 40](#_Toc87373629)

[7.2.2 CSS控制单元格边距 40](#_Toc87373630)

[7.2.3 CSS控制单元格的宽高 41](#_Toc87373631)

[7.3 表单 41](#_Toc87373632)

[7.3.1 表单的构成 41](#_Toc87373633)

[7.3.2 创建表单 41](#_Toc87373634)

[7.4 表单控件 42](#_Toc87373635)

[7.4.1 input控件 42](#_Toc87373636)

[7.4.2 textarea控件 46](#_Toc87373637)

[7.4.3 select控件 47](#_Toc87373638)

[7.5 HTML5表单新属性 48](#_Toc87373639)

[7.5.1 全新的form属性 48](#_Toc87373640)

[7.5.2 全新的表单控件 48](#_Toc87373641)

[7.5.3 全新的input控件类型 48](#_Toc87373642)

[7.5.4 全新的input属性 51](#_Toc87373643)

[7.6 CSS控制表单样式 55](#_Toc87373644)

[第八章 DIV+CSS布局 55](#_Toc87373645)

[8.1 布局概述 55](#_Toc87373646)

[8.2 布局常用属性 55](#_Toc87373647)

[8.2.1 标签的浮动属性 55](#_Toc87373648)

[8.2.2 标签的定位属性 56](#_Toc87373649)

[8.3 布局其他属性 59](#_Toc87373650)

[8.3.1 overflow属性 59](#_Toc87373651)

[8.3.2 Z-index标签层叠 59](#_Toc87373652)

[8.4 布局类型 59](#_Toc87373653)

[8.4.1 单列布局 59](#_Toc87373654)

[8.4.2 两列布局 60](#_Toc87373655)

[8.4.3 三列布局 60](#_Toc87373656)

[8.4.4 通栏布局 60](#_Toc87373657)

[8.4.5 全新的HTML5结构元素 60](#_Toc87373658)

[8.4.6 网页模块的命名规范 62](#_Toc87373659)

[第九章 多媒体嵌入 64](#_Toc87373660)

[9.1视频音频嵌入技术概述 64](#_Toc87373661)

[9.2视频文件和音频文件的格式 64](#_Toc87373662)

[9.3 嵌入视频和音频 64](#_Toc87373663)

[9.3.1 在HTML5中嵌入视频 64](#_Toc87373664)

[9.3.2 在HTML5中嵌入音频 64](#_Toc87373665)

[9.3.3 视频音频文件的兼容性问题 65](#_Toc87373666)

[9.3.4 调用网络音频视频文件 65](#_Toc87373667)

[9.4 CSS控制视频的宽高 65](#_Toc87373668)

[第十章 CSS3高级应用 66](#_Toc87373669)

[10.1 过渡 66](#_Toc87373670)

[10.1.1 transition-property属性 66](#_Toc87373671)

[10.1.2 transition-duration属性 66](#_Toc87373672)

[10.1.3 transition-timing-function属性 66](#_Toc87373673)

[10.1.4 transition-delay属性 67](#_Toc87373674)

[10.1.5 transition属性 67](#_Toc87373675)

[10.2 变形 67](#_Toc87373676)

[10.2.1认识transform 67](#_Toc87373677)

[10.2.2 2D转换 68](#_Toc87373678)

[10.2.3 3D转换 70](#_Toc87373679)

[10.3 动画 72](#_Toc87373680)

[10.3.1 @keyframes 72](#_Toc87373681)

[10.3.2 animation-name属性 72](#_Toc87373682)

[10.3.3 animation-duration属性 72](#_Toc87373683)

[10.3.4 animation-timing-function属性 73](#_Toc87373684)

[10.3.5animation-delay属性 73](#_Toc87373685)

[10.3.6 animation-iteration-count属性 73](#_Toc87373686)

[10.3.7 animation-direction属性 74](#_Toc87373687)

[10.3.8 animation属性 74](#_Toc87373688)

[第十一章 绘图和数据存储原理 75](#_Toc87373689)

[11.1 简单的JavaScript 75](#_Toc87373690)

[11.1.1 JavaScript的引入 75](#_Toc87373691)

[11.1.2 变量 76](#_Toc87373692)

[11.1.3 document对象 76](#_Toc87373693)

[11.2 HTML5画布 77](#_Toc87373694)

[11.2.1 认识画布 77](#_Toc87373695)

[11.2.2 使用画布 77](#_Toc87373696)

[11.2.3 绘制线 77](#_Toc87373697)

[11.2.4 线的样式 78](#_Toc87373698)

[11.2.5 线的路径 79](#_Toc87373699)

[11.2.7 绘制圆 80](#_Toc87373700)

# 第一章 HTML5+CSS3网页设计概述

## 1.1 网页概述

**网页**可以看做承载各种网站应用和信息的容器，所有可视化的内容都会通过网页展示给用户。

### 1.1.1 认识网页

网页主要由文字、图像和超链接（超链接为单击可以跳转的网页元素）等元素构成。当然除了这些元素，网页中还可以包含音频、视频以及动画等。

**静态网页：**

* 用户无论何时何地访问，网页都会显示固定的信息，除非网页源代码被重新修改上传。
* 静态网页更新不方便，但是访问速度快。

**动态网页**

* 显示的内容则会随着用户操作和时间的不同而变化。
* 动态网页可以和服务器数据库进行实时的数据交换。

### 1.1.2 网页名词解释

|  |  |
| --- | --- |
| 名词 | 名词释义 |
| Internet网络 | 就是通常所说的互联网，是由一些使用公用语言互相通信的计算机连接而成的网络。 |
| WWW | WWW（英文World Wide Web的缩写）中文译为“万维网”是Intertnet提供的一种网页浏览服务。 |
| URL | URL（英文Uniform Resource Locator的缩写）中文译为“统一资源定位符” URL其实就是Web地址，俗称“网址”。 |
| DNS | DNS （英文Domain Name System的缩写）是域名解析系统。 |
| HTTP和HTTPS | HTTP (英文Hypertext transfer protocol的缩写) 中文译为超文本传输协议，是一种详细规定了浏览器和万维网服务器之间互相HTTPS协议是由SSL+HTTP协议构建的可进行加密传输、身份认证的网络协议，要比HTTP协议安全。  通信的规则。 |
| Web | Web通常指互联网的使用环境。但对于网站制作者来说，它是一系列技术的复合总称，通常称之为网页。 |
| W3C组织 | W3C（英文World Wide Web Consortium的缩写）中文译为“万维网联盟”。万维网联盟是国际最著名的标准化组织。 |

### 1.1.3 Web标准

**（1）什么是Web标准？**

**Web标准**不是某一个标准，而是由W3C和其他标准化组织制定的一系列标准的集合。 包含我们所熟悉的HTML、XHTML、CSS、Javascript等等。

由于不同的浏览器解析出来的效果可能不一致，开发者常常需要为多版本的开发而艰苦工作。

**结构标准：结构标准用于对网页元素进行整理和分类，主要包括：HTML、XML和XHTML。**

**表现标准：表现标准用于设置网页元素的版式、颜色、大小等外观样式，主要指的是CSS。**

**行为标准: 行为标准是指网页模型的定义及交互的编写，主要包括两个部分：DOM和ECMAScript。**

## 1.2 网页制作技术入门

HTML、CSS和JavaScript是网页制作的标准语言，要想学好、学会网页制作技术，首先需要对它们有一个整体的认识。

### 1.2.1 HTML

**HTML(英文Hyper Text Markup Language的缩写)中文译为“超文本标签语言”，主要是通过HTML标签对网页中的文本、图片、声音等内容进行描述。**

### 1.2.2 CSS

**CSS通常称为CSS样式或样式表，主要用于设置HTML页面中的文本内容（字体、大小、对齐方式等）、图片的外形（宽高、边框样式、边距等）以及版面的布局等外观显示样式。**

### 1.2.3 JavaScript

**JavaScript是网页中的一种脚本语言，其前身叫做LiveScript，由Netscape(网景)公司开发。后来在Sun公司推出著名的Java语言之后，Netscape公司和Sun公司于1995年一起重新设计了LiveScript，并把它改名为JavaScript。**

### 1.2.4 浏览器

IE、火狐和谷歌是目前互联网上的三大浏览器，其他常用的浏览器还有苹果的Safari浏览器和欧朋浏览器等。对于一般的网站，只要兼容IE浏览器、火狐浏览器和谷歌浏览器，就能满足绝大多数用户的需求。

### 1.2.5什么是浏览器内核

浏览器内核是浏览器最核心的部分，负责对网页语法的解释并渲染网页（也就是显示网页效果），是渲染引擎（标准叫法）的通俗叫法。渲染引擎决定了浏览器如何显示网页的内容以及页面的格式信息。不同的浏览器内核对网页编写语法的解释也不同，因此同一网页在不同的内核的浏览器里的渲染（显示）效果也可能不同。

**常见**

**的浏览器**

**内核**

# 第二章 初识HTML5

## 2.1 HTML5的优势

HTML5作为HTML的最新版本，是HTML的传递和延续。经历HTML4.0、XHTML再到HTML5从某种意义上讲，这是HTML超文本标签语言的更新与规范过程。因此，HTML5并没有给用户带来多大的冲击，大部分标签在HTML5版本中依然适用。

**HTML5的优势主要体现在兼容、合理、易用三个方面**

## 2.2 HTML5全新的结构

**<!DOCTYPE>标签:** **位于文档的最前面，用于向浏览器说明当前文档使用哪种 HTML 或 XHTML 标准规范。**

**<html>标签**：DOCTYPE> 标签之后，也称为根标签，用于告知浏览器其自身是一个 HTML 文档。

**<head>标签：定义HTML文档的头部信息，也称为头部标签，紧跟在<html>标签之后，主要用来封装其他位于文档头部的标签。**

**<body>标签：定义HTML文档所要显示的内容，也称为主体标签。浏览器中显示的所有文本、图像、音频和视频等信息都必须位于<body>标签内。**

## 2.3 标签概述

在HTML页面中，带有“< >”符号的元素被称为HTML标签，如上面提到的<html>、<head>、<body>都是HTML标签。所谓标签就是放在“< >”符号中表示某个功能的编码命令，也称为HTML标签或HTML元素。

### 2.3.1 标签的分类

**（1）双标签：双标签也称体标签，是指由开始和结束两个标签符组成的标签。例如：<标签名>内容</标签名>**

**（2）单标签:** **单标签也称空标签，是指用一个标签符号即可完整地描述某个功能的标签。L例如：< 标签名 />**

**（3）为什么要有单标签？**

**HTML标签的作用原理就是选择网页内容，从而进行描述，也就是说需要描述哪个元素，就选择哪个元素，所以才会有双标签的出现，用于定义标签作用的开始与结束。而单标签本身就可以描述一个功能，不需要选择。例如，水平线标签<hr />,按照双标签的语法，它应该写成“<hr></hr>”，但是水平线标签不需要选择，本身就代表一条水平线，此时写成双标签就显得有点多余，但是又不能没有结束符号，所以在标签名称后面加一个关闭符，即<标签名 />。**

### 2.3.2 标签的关系

**1、嵌套关系**

**2、并列关系**

**并列关系也称为兄弟关系，就是两个标签处于同一级别，并且没有包含关系。例如在HTML5的结构代码中，<head>标签和<body>标签就是并列关系。在HTML标签中，无论是单标签还是双标签，都可以拥有并列关系。**

### 2.3.3 标签的属性

**<标记名 属性1="属性值1" 属性2="属性值2" …> 内容 </标记名>**

**认识键值对**

**“键值对”可以理解为对“属性”设置“属性值”。键值对有多种表现形式，例如color="red" 、width:200px;等，其中color和width即为“键值对”中的“键”（英文key），red和200px为“键值对”中的“值”（英文value）。“键值对”广泛地应用于编程中，HTML属性的定义形式“属性="属性值"”只是“键值对”中的一种。**

### 2.3.4 HTML5文档头部相关标签

**<title>网页标题名称</title>**

**<meta/>：设置网页关键字、设置网页描述、设置网页作者、设置字符集、设置页面自动刷新与跳转**

## 2.4 文本控制标签

**HTML中提供了一系列文本控制标签，如标题标签<h1>~<h6>、段落标签<P>等**

### 2.4.1 页面格式化标签

**1、标题标记**

**HTML提供了6个等级的标题，即<h1>、<h2>、<h3>、<h4>、<h5>和<h6>，从<h1>到<h6>标题的重要性依次递减。**

**2、段落标签**

**在网页中使用<p>标签来定义段落。<p>标签是HTML文档中最常见的标签，默认情况下，文本在一个段落中会根据浏览器窗口的大小自动换行。**

**3、水平线标签**

**在网页中常常看到一些水平线将段落与段落之间隔开，使得文档结构清晰，层次分明。水平线可以通过<hr />标签来定义。**

|  |  |  |
| --- | --- | --- |
| **属性名** | **含义** | **属性值** |
| **align** | 设置水平线的对齐方式 | 可选择left、right、center三种值，默认为center，居中对齐显示。 |
| **size** | 设置水平线的粗细 | 以像素为单位，默认为2像素。 |
| **color** | 设置水平线的颜色 | 可用颜色名称、十六进制#RGB、rgb(r,g,b)。 |
| **width** | 设置水平线的宽度 | 可以是确定的像素值，也可以是浏览器窗口的百分比，默认为100%。 |

**4、换行标签**

**在网页中，如果想要将某段文本强制换行显示，就需要使用换行标签<br />。**

### 2.4.2 文本样式标签

**文本样式标记用来控制网页中文本的字体、字号和颜色**

**基本语法格式：<font 属性="属性值">文本内容</font>**

|  |  |
| --- | --- |
| 属性名 | 含义 |
| face | 设置文字的字体，例如微软雅黑、黑体、宋体等 |
| size | 设置文字的大小，可以取1到7之间的整数值 |
| color | 设置文字的颜色 |

### 2.4.3 文本格式化标签

|  |  |
| --- | --- |
| 标记 | 显示效果 |
| <b></b>和<strong></strong> | 文字以粗体方式显示（XHTML推荐使用strong） |
| <i></i>和<em></em> | 文字以斜体方式显示（XHTML推荐使用em） |
| <s></s>和<del></del> | 文字以加删除线方式显示（XHTML推荐使用del） |
| <u></u>和<ins></ins> | 文字以加下划线方式显示（XHTML不赞成使用u） |

### 2.4.4 文本语义标签

**1、time标签**

**1、time标签用于定义时间或日期，可以代表24小时中的某一时间。**

* **datetime：用于定义相应的时间或日期。取值为具体时间（例如：14:00）或具体日期（例如：2015-09-01），不定义该属性时，由文本的内容给定日期或时间。**
* **pubdate：用于定义time标签中的文档（或article元素）发布日期。取值一般为“pubdate”。**

**2、mark标签**

**mark标签的主要功能是在文本中高亮显示某些字符，该元素的用法与em标签和strong标签有相似之处，但是使用mark标签在突出显示样式时更随意灵活。**

**3、cite标签**

**cite标签可以创建一个引用，用于对文档引用参考文献的说明，一旦在文档中使用了该标签，被标注的文档内容将以斜体的样式展示在页面中，以区别于段落中的其他字符。**

**2.4.5 特殊字符标签**

|  |  |  |
| --- | --- | --- |
| 特殊字符 | 描述 | 字符的代码 |
|  | 空格符 | &nbsp; |
| < | 小于号 | &lt; |
| > | 大于号 | &gt; |
| & | 和号 | &amp; |
| ￥ | 人民币 | &yen; |
| © | 版权 | &copy; |
| ® | 注册商标 | &reg; |
| ° | 摄氏度 | &deg; |
| ± | 正负号 | &plusmn; |
| × | 乘号 | &times; |
| ÷ | 除号 | &divide; |
| ² | 平方2（上标2） | &sup2; |
| ³ | 立方3（上标3） | &sup3; |

## 2.5 图像标签

### 2.5.1 图像标签<img />

**基本语法格式**

**<img src="图像URL" />**

**alt：图像的替换文本属性，在图像无法显示时告诉用户该图片的内容。**

**width、height:** **用来定义图片的宽度和高度，通常我们只设置其中的一个，另一个会按原图等比例显示。**

**border:** **为图像添加边框、设置边框的宽度。但边框颜色的调整仅仅通过HTML属性是不能够实现的。**

**vspace、hspace:** **HTML中通过vspace和hspace属性可以分别调整图像的垂直边距和水平边距。**

**align:** **图像的对齐属性align。用于调整图像的位置**

|  |  |  |
| --- | --- | --- |
| 属性 | 属性值 | 描述 |
| src | URL | 图像的路径 |
| alt | 文本 | 图像不能显示时的替换文本 |
| title | 文本 | 鼠标悬停时显示的内容 |
| width | 像素（XHTML不支持%页面百分比） | 设置图像的宽度 |
| height | 像素（XHTML不支持%页面百分比） | 设置图像的高度 |
| border | 数字 | 设置图像边框的宽度 |
| vspace | 像素 | 设置图像顶部和底部的空白（垂直边距） |
| hspace | 像素 | 设置图像左侧和右侧的空白（水平边距） |
| align | left | 将图像对齐到左边 |
| right | 将图像对齐到右边 |
| top | 将图像的顶端和文本的第一行文字对齐，其他文字居图像下方 |
| middle | 将图像的水平中线和文本的第一行文字对齐，其他文字居图像下方 |

### 2.5.3 相对路径和绝对路径

**绝对路径:** **绝对路径一般是指带有盘符的路径，例如“D: \chapter02\img\logo.gif,或完整的网络地址，例如“**<http://www.itcast.cn/images/logo.gif>**”。**

**相对路径:** **相对路径不带有盘符，通常是以HTML网页文件为起点，通过层级关系描述目标图像的位置。**

**(1)** **图像文件和html文件位于同一文件夹：**

**只需输入图像文件的名称即可，如<img src=“logo.gif” />。**

**(2)** **图像文件位于html文件的下一级文件夹：**

**输入文件夹名和文件名，之间用“/”隔开，如<img src="img/img01/logo.gif" />。**

**(3)** **图像文件位于html文件的上一级文件夹：**

**在文件名之前加入“../” ，如果是上两级，则需要使用 “../ ../”，以此类推，如<img src="../logo.gif" />。**

# 第三章 初识CSS3

## 3.1 结构与表现分离

**什么是结构与表现分离？**

**结构与表现相分离是指在网页设计中，HTML标签只用于搭建网页的基本结构，不使用标签属性设置显示样式，所有的样式交由CSS来设置。**

## 3.2 CSS的优势

**CSS3是CSS规范的最新版本，在CSS2.1的基础上增加了很多强大的新功能，以帮助开发人员解决一些实际面临的问题。使用CSS3不仅可以设计炫酷美观的网页，还能提高网页性能。**

**与传统的CSS相比，CSS3最突出的优势主要体现在节约成本和提高性能两方面。**

**节约成本：CSS3的新功能帮我们摒弃了冗余的代码结构，远离很多Javascript脚本或者Flash代码。网页设计者不再需要花大把时间去写脚本，极大的节约了开发成本。**

**提高性能：由于功能的加强，CSS3能够用更少的图片或脚本制作图形化网站。在进行网页设计时，减少标签的嵌套和图片的使用数量，网页页面加载也会更快。**

## 3.3 CSS核心基础

### 3.3.1 CSS样式规则

**选择器{属性1:属性值1; 属性2:属性值2; 属性3:属性值3;}**

**在上面的样式规则中，选择器用于指定CSS样式作用的HTML对象，花括号内是对该对象设置的具体样式。其中，属性和属性值以“键值对”的形式出现，用英文“:”连接，多个“键值对”之间用英文“;”进行区分。**



### 3.3.2 引入CSS样式表

**行内式：行内式也称为内联样式，是通过标签的style属性来设置元素的样式，其基本语法格式如下:**

**<标签名 style="属性1:属性值1; 属性2:属性值2; 属性3:属性值3;"> 内容 </标签名>**

**内嵌式：内嵌式是将CSS代码集中写在HTML文档的<head>头部标签中，并且用<style>标签定义，其基本语法格式如下:**

**<head>**

**<style type="text/css">**

**选择器 {属性1:属性值1; 属性2:属性值2; 属性3:属性值3;}**

**</style>**

**</head>**

**链入式：链入式是将所有的样式放在一个或多个以.css为扩展名的外部样式表文件中，通过<link />标签将外部样式表文件链接到HTML文档中，其基本语法格式如下:**

**<head>**

**<link href="CSS文件的路径" type="text/css" rel="stylesheet" />**

**</head>**

**导入式：导入式针对外部样式表文件。对HTML头部文档应用style标签，并在<style>标签内的开头处使用@import语句，即可导入外部样式表文件。其基本语法格式如下:**

**<style type="text/css" >**

**@import url(css文件路径);或 @import "css文件路径";**

**/\* 在此还可以存放其他CSS样式\*/**

**</style>**

### 3.3.3 CSS基础选择器

**标签选择器：标签选择器是指用HTML标签名称作为选择器，按标签名称分类，为页面中某一类标签指定统一的CSS样式。其基本语法格式如下：**

**标签名{属性1:属性值1; 属性2:属性值2; 属性3:属性值3; }**

**类选择器: 类选择器使用“.”（英文点号）进行标识，后面紧跟类名，其基本语法格式如下：**

**.类名{属性1:属性值1; 属性2:属性值2; 属性3:属性值3; }**

**id选择器: id选择器使用“#”进行标识，后面紧跟id名，其基本语法格式如下：**

**#id名{属性1:属性值1; 属性2:属性值2; 属性3:属性值3; }**

**通配符选择器: 通配符选择器用“\*”号表示，它是所有选择器中作用范围最广的，能匹配页面中所有的元素。其基本语法格式如下：**

## 3.4 设置文本样式

### 3.4.1 CSS字体样式属性

**(1)font-size属性用于设置字号.**

p{ font-family:"微软雅黑";}

**(2)** **font-family属性用于设置字体。**

**(3)** **font-weight属性用于定义字体的粗细。**

|  |  |
| --- | --- |
| **值** | **描述** |
| normal | 默认值。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。 |
| 100~900（100的整数倍） | 定义由细到粗的字符。其中400等同于normal，700等同于bold，值越大字体越粗。 |

**(4) font-style属性用于定义字体风格。**

**normal：默认值，浏览器会显示标准的字体样式;**

**italic：浏览器会显示斜体的字体样式;**

**oblique：浏览器会显示倾斜的字体样式;**

**(5) font综合属性用于综合设置字体样式。**

**选择器{font: font-style font-variant font-weight font-size/line-height font-family;}**

**(6)** **@font-face属性是CSS3的新增属性，用于定义服务器字体。**

**@font-face{**

**font-family:字体名称;**

**src:字体路径;**

**}**

**(7)** **word-wrap属性用于实现长单词和URL地址的自动换行。**

### 3.4.2 CSS文本外观属性

**color: color属性用于定义文本的颜色，其取值方式有如下3种：**

* **预定义的颜色值，如red，green，blue等。**
* **十六进制，如#FF0000，#FF6600，#29D794等。实际工作中，十六进制是最常用的定义颜色的方式。**
* **RGB代码，如红色可以表示为rgb(255,0,0)或rgb(100%,0%,0%)。**

**letter-spacing: letter-spacing属性用于定义字间距，所谓字间距就是字符与字符之间的空白。**

**letter-spacing属性，其属性值可为不同单位的数值，允许使用负值，默认为normal。**

**word-spacing: word-spacing属性用于定义英文单词之间的间距，对中文字符无效。**

**word-spacing属性用于定义英文单词之间的间距，和letter-spacing一样，其属性值可为不同单位的数值，允许使用负值，默认为normal。**

**line-height: line-height属性用于设置行间距，所谓行间距就是行与行之间的距离，即字符的垂直间距，一般称为行高。**

**line-height常用的属性值单位有三种，分别为像素px，相对值em和百分比%，实际工作中使用最多的是像素px。**

**text-align: text-align属性用于设置文本内容水平对齐，相当于html中的align对齐属性。**

**其可用属性值如下：**

* **left：左对齐（默认值）**
* **right：右对齐。**
* **center：居中对齐。**

**text-transform: text-transform属性用于控制英文字符的大小写。**

**其可用属性值如下：**

* **none：不转换（默认值）。**
* **capitalize：首字母大写。**
* **uppercase：全部字符转换为大写。**
* **lowercase：全部字符转换为小写。**

**text-decoration: text-decoration属性用于设置文本的下划线，上划线，删除线等装饰效果。**

**其可用属性值如下：**

* **none：没有装饰（正常文本默认值）。**
* **underline：下划线。**
* **overline：上划线。**
* **line-through：删除线。**

**text-indent: text-indent属性用于设置首行文本的缩进。**

**其属性值可为不同单位的数值、em字符宽度的倍数、或相对于浏览器窗口宽度的百分比%，允许使用负值, 建议使用em作为设置单位。**

**white-space: white-space属性可设置空白符的处理方式。**

**其属性值如下：**

* **normal：常规（默认值），文本中的空格、空行无效，满行（到达区域边界）后自动换行。**
* **pre：预格式化，按文档的书写格式保留空格、空行原样显示。**
* **nowrap：空格空行无效，强制文本不能换行，除非遇到换行标签<br />。内容超出元素的边界也不换行，若超出浏览器页面则会自动增加滚动条。**

**text-shadow: text-shadow属性可以为页面中的文本添加阴影效果。**

**h-shadow用于设置水平阴影的距离，v-shadow用于设置垂直阴影的距离，blur用于设置模糊半径，color用于设置阴影颜色。**

**text-overflow: text-overflow属性用于处理溢出的文本。**

* **clip：修剪溢出文本，不显示省略标签“…”。**
* **ellipsis：用省略标签“…”替代被修剪文本，省略标签插入的位置是最后一个字符。**

**word-wrap: word-wrap属性用于实现长单词和URL地址的自动换行。**

* **normal：只在允许的断字点换行（浏览器保持默认处理）。**
* **break-word：在长单词或 URL 地址内部进行换。**

## 3.5 高级特性

### 3.5.1 CSS复合选择器

**标签指定式选择器: 标签指定式选择器又称交集选择器，由两个选择器构成，其中第一个为标签选择器，第二个为class选择器或id选择器，两个选择器之间不能有空格，如h3.special或p#one。**

**后代选择器: 后代选择器用来选择元素或元素组的后代，其写法就是把外层标签写在前面，内层标签写在后面，中间用空格分隔。当标签发生嵌套时，内层标签就成为外层标签的后代。**

**并集选择器: 并集选择器是各个选择器通过逗号连接而成的，任何形式的选择器都可以作为并集选择器的一部分。若某些选择器定义的样式完全或部分相同，可利用并集选择器为它们定义相同的样式。**

### 3.5.2 CSS层叠性和继承性

**层叠性:** **所谓层叠性是指多种CSS样式的叠加。**

**继承性:** **所谓继承性是指书写CSS样式表时，子标签会继承父标签的某些样式，如文本颜色和字号。**

**下面的属性不具有继承性：**

**边框属性、定位属性、内/外边距属性、布局属性、背景属性、元素宽高属性**

# 第四章 CSS4选择器

## 4.1 属性选择器

**属性选择器可以根据元素的属性及属性值来选择元素。CSS3中新增了3种属性选择器。**

### 4.1.1 E[att^=value] 属性选择器

**概念：E[att^=value] 属性选择器是指选择名称为E的标记，且该标记定义了att属性，att属性值包含前缀为value的子字符串。**

**例如：div[id^=section]**

**表示匹配包含id属性，且id属性值是以“section”字符串开头的div元素。**

### 4.1.2 E[att$=value]属性选择器

**概 念：E[att$=value] 属性选择器是指选择名称为E的标记，且该标记定义了att属性，att属性值包含后缀为value的子字符串。与E[att^=value]选择器一样，E元素可以省略，如果省略则表示可以匹配满足条件的任意元素。**

**例如：div[id$=section]**

**表示匹配包含id属性，且id属性值是以“section”字符串结尾的div元素。**

### 4.1.3 E[att\*=value]属性选择器

**概 念：E[att\*=value]选择器用于选择名称为E的标记，且该标记定义了att属性，att属性值包含value子字符串。该选择器与前两个选择器一样，E元素也可以省略，如果省略则表示可以匹配满足条件的任意元素。**

**例如：div[id\*=section]**

**表示匹配包含id属性，且id属性值包含“section”字符串的div元素。**

## 4.2 关系选择器

**关系选择器和前面讲的复合选择器类似，但关系选择器可以更精确的控制元素样式。CSS3中的关系选择器主要包括子元素选择器和相邻兄弟选择器，其中子元素择器由符号“>”连接，兄弟选择器由符号“+”和“~”连接**

### 4.2.1 子元素选择器

**概 念：子元素择器主要用来选择某个元素的第一级子元素**

**例如：希望选择只作为 h1 元素子元素的 strong 元素，可以这样写：h1 > strong。**

### 4.2.2 相邻兄弟选择器（+、~）

**概 念：兄弟选择器用来选择与某元素位于同一个父元素之中，且位于该元素之后的兄弟元素。**

**临近兄弟选择器：该选择器使用加号“+”来链接前后两个选择器。选择器中的两个元素有同一个父亲，而且第二个元素必须紧跟第一个元素。**

**普通兄弟选择器：普通兄弟选择器使用 “~”来链接前后两个选择器。选择器中的两个元素有同一个父亲，但第二个元素不必紧跟第一个元素。**

## 4.3 结构化伪类选择器

**常用的结构化伪类选择器有**

* **:root选择器**
* **:not选择器**
* **:only-child 选择器**
* **:first-child和:last-child选择器**
* **:nth-child(n)和:nth-last-child(n)选择器**
* **:nth-of-type(n)和:nth-last-of-type(n)选择器**
* **:empty选择器**

### 4.3.1 :root选择器

**:root选择器：:root选择器用于匹配文档根元素，在HTML中，根元素始终是html元素。也就是说使用“:root选择器”定义的样式，对所有页面元素都生效。对于不需要该样式的元素，可以单独设置样式进行覆盖。**

### 4.3.2 :not选择器

**:not选择器：如果对某个结构元素使用样式，但是想排除这个结构元素下面的子结构元素，让它不使用这个样式，可以使用:not选择器。**

### 4.3.3 :only-child选择器

**:only-child 选择器：:only-child 选择器用于匹配属于某父元素的唯一子元素的元素，也就是说，如果某个父元素仅有一个子元素，则使用“:only-child 选择器”可以选择这个子元素。**

### 4.3.4 :first-child和:last-child选择器

**:first-child和:last-child选择器：:first-child选择器和:last-child选择器分别用于为父元素中的第一个或者最后一个子元素设置样式。**

### 4.3.5 :nth-child(n)和:nth-last-child(n)选择器

**:nth-child(n)和:nth-last-child(n)选择器：使用:first-child选择器和:last-child选择器可以选择某个父元素中第一个或最后一个子元素，但是如果用户想要选择第2个或倒数第2个子元素，这两个选择器就不起作用了。为此，CSS3引入了:nth-child(n)和:nth-last-child(n)选择器，它们是:first-child选择器和:last-child选择器的扩展。**

### 4.3.6 :nth-of-type(n)和:nth-last-of-type(n)选择器

**:nth-of-type(n)和:nth-last-of-type(n)选择器：:nth-of-type(n)和:nth-last-of-type(n)选择器用于匹配属于父元素的特定类型的第 n 个子元素和倒数第n个子元素。**

### 4.3.7 :empty选择器

**:empty选择器：:empty选择器用来选择没有子元素或文本内容为空的所有元素。**

## 4.4 伪元素选择器

**伪元素选择器，是针对CSS中已经定义好的伪元素使用的选择器。CSS中常用的伪元素选择器有:before伪元素选择器和:after伪元素选择器。**

**:before选择器:** **:before伪元素选择器用于在被选元素的内容前面插入内容，必须配合content属性来指定要插入的具体内容。**

**例如：<元素>:before**

**{**

**content:文字/url();**

**}**

**:after选择器：:after伪元素选择器用于在某个元素之后插入一些内容，使用方法与:before选择器相同。**

**例如：<元素>:after**

**{**

**content:文字/url();**

**}**

# 第五章 盒子模型

## 5.1 盒子模型相关属性

### 5.1.1 边框属性

**为了分割页面中不同的盒子，常常需要给元素设置边框效果。**

|  |  |  |
| --- | --- | --- |
| 设置内容 | 样式属性 | 常用属性值 |
| 边框样式 | border-style:上边 [右边 下边 左边]; | none无（默认）、solid单实线、dashed虚线、dotted点线、double双实线 |
| 边框宽度 | border-width:上边 [右边 下边 左边]; | 像素值 |
| 边框颜色 | border-color:上边 [右边 下边 左边]; | 颜色值、#十六进制、rgb(r,g,b)、rgb(r%,g%,b%) |
| 综合设置边框 | border:四边宽度 四边样式 四边颜色; |  |
| 圆角边框 | border-radius:水平半径参数/垂直半径参数; | 像素值或百分比 |
| 图片边框 | border-images:图片路径 裁切方式/边框宽度/边框扩展距离 重复方式; |  |

**边框样式（border-style）属性值**

**none：没有边框**

**solid：边框为单实线**

**dashed：边框为虚线**

**dotted：边框为点线**

**double：边框为双实线**

**border-style综合属性**

**border-style:solid ; /\*四边均为实线\*/**

**border-style:solid dotted ; /\*上下实线、左右点线\*/**

**border-style:solid dotted dashed; /\*上实线、左右点线、下虚线\*/**

**边框宽度（border-width）属性值**

**在设置边框宽度时，必须同时设置边框样式，如果未设置样式或设置为none，则不论宽度设置为多少都无效。**

**border-width综合属性**

**border-width:5px; /\*四边宽度均为5像素\*/**

**border-width:5px 3px ; /\*上下边框5像素宽度、左右边框3像素宽度\*/**

**border-width:5px 3px 4px; /\*上边框5像素宽、左右边框3像素宽度、下边框4像素宽度\*/**

**边框颜色（border-color）**

**颜色值，例：red、green**

**#十六进制色值，例：#ccc 实际工作中最常用**

**rgb(r,g,b)，例：rgb(30,0,0)**

**rgb(r%,g%,b%)**

**border-color综合属性**

**border-color:#f00; /\*四边均为红色\*/**

**border-color:#f00 #00f ; /\*上下红色、左右蓝色\*/**

**border-color:#f00 #00f #0f0; /\*上红色、左右蓝色、下绿色\*/**

**CSS3在原边框颜色属性（border-color）的基础上派生了4个边框颜色属性。**

**border-top-colors**

**border-right-colors**

**border-bottom-colors**

**border-left-colors**

**例如对段落文本<p>添加渐变边框效果，示例代码如下：**

**p{**

**border-style:solid;**

**border-width:10px;**

**-moz-border-top-colors:#a0a #909 #808 #707 #606 #505 #404 #303;**

**-moz-border-right-colors:#a0a #909 #808 #707 #606 #505 #404 #303;**

**-moz-border-bottom-colors:#a0a #909 #808 #707 #606 #505 #404 #303;**

**-moz-border-left-colors:#a0a #909 #808 #707 #606 #505 #404 #303;**

**}**

**边框综合属性（border）**

**border-top:上边框宽度 样式 颜色;**

**border-right:右边框宽度 样式 颜色;**

**border-bottom:下边框宽度 样式 颜色;**

**border-left:左边框宽度 样式 颜色;**

**border:四边宽度 样式 颜色;**

### 5.1.2 内边距属性

**padding-top:上边距;**

**padding-right:右边距;**

**padding-bottom:下边距;**

**padding-left:左边距;**

**padding：四边内边距;**

**注意：内边距padding不允许使用负值**

**padding:5px /\*四个方向内边距为5像素宽度\*/**

**padding:5px 3px /\*上下内边距为5像素，左右内边距为3像素\*/**

**padding:5px 3px 4px /\*上内边距为5像素，左右内边距为3像素，下内边距为4像素\*/**

### 5.1.3 外边距属性

**margin-top:上外边距;**

**margin-right:右外边距;**

**margin-bottom:下外边距;**

**margin-left:左外边距;**

**margin:四边外边距;**

**注意：和内边距不同，外边距margin允许使用负值**

**margin:5px /\*四边外边距为5像素宽度\*/**

**margin:5px 3px /\*上下外边距为5像素，左右外边距为3像素\*/**

**margin:5px 3px 4px /\*外边距：上为5像素，左右为3像素，下为4像素\*/**

**当对块级元素应用宽度属性width，并将左右的外边距都设置为auto，可使块级元素水平居中，实际工作中常用这种方式进行网页布局。**

**margin:0 auto /\* 利用margin实现块元素水平居中\*/**

**margin:5px auto /\* 利用margin实现块元素水平居中，并且上下拉开5像素边距\*/**

**内外边距清除**

**为了更方便地控制网页中的元素，制作网页时，通常先清除元素的默认内外边距。**

**\*{**

**padding:0; /\*清除内边距\*/**

**margin:0; /\*清除外边距\*/**

**}**

### 5.1.4 背景属性

**1、设置背景颜色**

**background-color:背景颜色属性**

**2、设置背景图像**

**background-image :背景颜色属性**

**3、设置背景图像平铺**

**background-repeat图像平铺属性**

|  |  |
| --- | --- |
| 平铺属性值 | 含义 |
| repeat | 沿水平和竖直两个方向平铺（默认值） |
| no-repeat | 不平铺（图像位于元素的左上角，只显示一次） |
| repeat-x | 只沿水平方向平铺 |
| repeat-y | 只沿竖直方向平铺 |

**4、设置背景图像的位置**

**background-position图像位置属性**

|  |  |
| --- | --- |
| **位置属性取值** | **含义** |
| **单位数值** | **设置图像左上角在元素中的坐标，例如background-position:20px 20px;** |
| **预定义的关键字** | **水平方向值：left、center、right。** |
| **垂直方向值：top、center、bottom。** |
| **百分比** | **0% 0% ：图像左上角与元素的左上角对齐。** |
| **50% 50%：图像50% 50%中心点与元素50% 50%的中心点对齐。** |
| **20% 30%：图像20% 30%的点与元素20% 30%的点对齐。** |
| **100% 100%：图像右下角与元素的右下角对齐，而不是图像充满元素。** |

**5、设置背景图像固定**

**background-attachment图像固定属性**

|  |  |
| --- | --- |
| **固定属性取值** | **含义** |
| **scroll** | **图像随页面元素一起滚动（默认值）。** |
| **fixed** | **图像固定在屏幕上，不随页面元素滚动。** |

**6、综合设置元素的背景**

**在CSS中背景属性也是一个复合属性，可以将背景相关的样式都综合定义在一个复合属性background中。**

**background:背景色 url("图像") 平铺 定位 固定;**

**background: url(img/wdjl.jpg) no-repeat 50px 80px fixed;**

### 5.1.5 盒子的宽与高

* **盒子的总宽度= width+左右内边距之和+左右边框宽度之和+左右外边距之和**
* **盒子的总高度= height+上下内边距之和+上下边框宽度之和+上下外边距之和**

## 5.2 盒子模型新增属性

### 5.2.1 颜色透明度

通过引入RGBA模式和opacity属性，对**背景与图片**设置**不透明度**

rgba(r,g,b,alpha);

例如：p{background-color:rgba(255,0,0,0.5);}

opacity：opacityValue;

opacity属性用于定义标签的不透明度，参数opacityValue表示不透明度的值，它是一个介于0~1之间的浮点数值。其中，0表示完全透明，1表示完全不透明，而0.5则表示半透明。

### 5.2.2 圆角

**概 念：在网页设计中，经常会看到一些圆角的图形，如按钮、头像图片等，运用CSS3中的border-radius属性可以将矩形边框四角圆角化，实现圆角效果。**

border-radius:水平半径参数1 水平半径参数2 水平半径参数3 水平半径参数4/垂直半径参数1 垂直半径参数2 垂直半径参数3 垂直半径参数4;



需要注意的是，当应用值复制原则设置圆角边框时，如果“垂直半径参数”省略，则会默认等于“水平半径参数”的参数值。此时圆角的水平半径和垂直半径相等。

### 5.2.3图片边框

**概 念：在网页设计中，我们还可以使用图片作为元素的边框。运用CSS3中的border-image属性可以轻松实现这个效果。**

**border-image: border-image-source/ border-image-slice/ border-image-width/ border-image-outset/ border-image-repeat;**

|  |  |
| --- | --- |
| 属性 | 说明 |
| border-image-source | 指定图片的路径 |
| border-image-slice | 指定边框图像顶部、右侧、底部、左侧内偏移量。 |
| border-image-width | 指定边框宽度 |
| border-image-outset | 指定边框背景向盒子外部延伸的距离。 |
| border-image-repeat | 指定背景图片的平铺方式 |

### 5.2.4 阴影

**概 念：在网页制作中，经常需要对盒子添加阴影效果。使用CSS3中的box-shadow属性可以轻松实现阴影的添加。**

**box-shadow属性也可以改变阴影的投射方向以及添加多重阴影效果。**

图片包含 图形用户界面

描述已自动生成

**例如：box-shadow:5px 5px 10px 2px #999 inset,-5px -5px 10px 2px #333 inset;**

### 5.2.5 渐变

**1、线性渐变**

**在线性渐变过程中，起始颜色会沿着一条直线按顺序过渡到结束颜色。运用CSS3中的 “background-image:linear-gradient（参数值）;”样式可以实现线性渐变效果。**

**background-image:linear-gradient(渐变角度,颜色值1,颜色值2...,颜色值n);**

**linear-gradient用于定义渐变方式为线性渐变，括号内用于设定渐变角度和颜色值。**

**渐变角度：渐变角度指水平线和渐变线之间的夹角，可以是以deg为单位的角度 数值或关键词。**

**颜色值：颜色值用于设置渐变颜色，其中“颜色值1”表示起始颜色，“颜色值n”表示结束颜色，起始颜色和结束颜色之间可以添加多个颜色值，各颜色值之间用“,”隔开。**

**2、径向渐变**

**径向渐变同样是网页中一种常用的渐变，在径向渐变过程中，起始颜色会从一个中心点开始，按照椭圆或圆形形状进行扩张渐变。运用CSS3中的“background-image:radial-gradient（参数值）;”样式可以实现径向渐变效果。**

**background-image:radial-gradient(渐变形状 圆心位置,颜色值1,颜色值2...,颜色值n);**

**radial-gradient用于定义渐变的方式为径向渐变，括号内的参数值用于设定渐变形状、圆心位置和颜色值。**

**渐变形状：渐变形状用来定义径向渐变的形状，其取值即可以是定义水平和垂直半径的像素值或百分比，也可以是相应的关键词。**

**圆心位置：圆心位置用于确定元素渐变的中心位置，使用“at”加上关键词或参数值来定义径向渐变的中心位置。**

**颜色值：“颜色值1”表示起始颜色，“颜色值n”表示结束颜色，起始颜色和结束颜色之间可以添加多个颜色值，各颜色值之间用“,”隔开。**

**3、重复渐变**

**重复线性渐变**

**在CSS3中，通过“background-image:repeating-linear-gradient（参数值）;”样式可以实现重复线性渐变的效果**

**background-image:repeating-linear-gradient(渐变角度,颜色值1,颜色值2...,颜色值n);**

**重复径向渐变**

**在CSS3中，通过“background-image:repeating-radial-gradient（参数值）;”样式可以实现重复线性渐变的效果。**

**background-image:repeating-radial-gradient(渐变形状 圆心位置,颜色值1,颜色值2...,颜色值n);**

### 5.2.6 多背景图像

**概述：在CSS3中，通过background-image、background-repeat、background-position和background-size等属性提供多个属性值可以实现多重背景图像效果，各属性值之间用逗号隔开。**

**图形用户界面

描述已自动生成**

**例如：background-image:**

**url(images/caodi.png),**

**url(images/taiyang.png),**

**url(images/tiankong.png);**

### 5.2.7 修剪背景图像

**1、设置背景图像的大小**

**在CSS3中，新增了background-size属性用于控制背景图像的大小**

**background-size:属性值1 属性值2;**

|  |  |
| --- | --- |
| **属性值** | **说 明** |
| **像素值** | **设置背景图像的高度和宽度。第一个值设置宽度，第二个值设置高度。如果只设置一个值，则第二个值会默认为auto；** |
| **百分比** | **以父元素的百分比来设置背景图像的宽度和高度。第一个值设置宽度，第二个值设置高度。如果只设置一个值，则第二个值会默认为auto；** |
| **cover** | **把背景图像扩展至足够大，使背景图像完全覆盖背景区域。背景图像的某些部分也许无法显示在背景定位区域中；** |
| **contain** | **把图像扩展至最大尺寸，以使其宽度和高度完全适应内容区域；** |

**2、设置背景图像的显示区域**

**运用CSS3中的background-origin属性可以自行定义背景图像的相对位置**

**background-origin:属性值;**

**在上面的语法格式中，background-origin属性有三种取值，分别表示不同的含义，具体解释如下。**

* **padding-box：背景图像相对于内边距区域来定位。**
* **border-box：背景图像相对于边框来定位。**
* **content-box：背景图像相对于内容框来定位。**

**3、设置背景图像的裁剪区域**

**在CSS样式中，background-clip属性用于定义背景图像的裁剪区域**

**background-clip:属性值;**

**在语法格式上，background-clip属性和background-origin 属性的取值相似，但含义不同，具体解释如下。**

* **border-box：默认值，从边框区域向外裁剪背景。**
* **padding-box：从内边距区域向外裁剪背景。**
* **content-box：从内容区域向外裁剪背景。**

## 5.3 元素的类型和转换

### 5.3.1 元素的类型

**块元素**

* **在页面中以区域块的形式出现。**
* **每个块元素通常都会独自占据一整行或多整行。**
* **可以对其设置宽度、高度、对齐等属性。**

**行内元素**

* **不占有独立的区域。**
* **仅仅靠自身的字体大小和图像尺寸来支撑结构。**
* **一般不可以设置宽度、高度、对齐等属性。**

**常见的块元素：<h1>—<h6>、<p>、<li>、<div>、<ul>、<ol>**

**其中<div>标签是最典型的块元素**

**常见的行内元素：<strong>、<a>、<u>、<b>、<em>、<span>**

**其中<span>标签最典型的行内元素。**

### 5.3.2 <div>和<span>标签

**1、div标签**

* **<div>标记是一个块容器标记。**
* **可以将网页分割为独立的部分，以实现网页的规划和布局。**
* **大多数HTML标记都可以嵌套在<div>标记中，<div>中还可以嵌套多层<div>。**
* **可以替代大多数的块级文本标记。**

**2、span标签**

* **<span>标记是一个行内标记。**
* **<span>与</span>之间只能包含文本和各种行内标记。**
* **<span>标记常用于定义网页中某些特殊显示的文本，配合class属性使用。**
* **当其他行内标记都不合适时，就可以使用<span>标记。**

### 5.3.3 元素类型的转换

**display**

**inline：此元素将显示为行内元素（行内元素默认的display属性值）。**

**block：此元素将显示为块元素（块元素默认的display属性值）。**

**inline-block:** **此元素将显示为行内块元素，可以对其设置宽高和对齐等属性，但是该元素不会独占一行。**

**none:** **此元素将被隐藏，不显示，也不占用页面空间。**

## 5.4 块元素垂直外边距的合并

### 5.4.1 相邻块元素垂直外边距的合并

**相邻元素垂直外边距的合并:** **相邻块元素之间的垂直间距不是margin-bottom与margin-top之和，而是两者中的较大者。**

**嵌套元素垂直外边距的合并:** **对于两个嵌套关系的块元素，如果父元素没有上内边距及边框，则父元素的上外边距会与子元素的上外边距发生合并，合并后的外边距为两者中的较大者。**

# 第六章 列表与超链接

## 6.1 列表标签

**无序列表:** **网页中最常用的列表，其各个列表项之间没有顺序级别之分，通常是并列的。**

**有序列表: 有排列顺序的列表，其各个列表项按照一定的顺序排列。**

**定义列表:** **用于对术语或名词进行解释和描述。与无序和有序列表不同，定义列表的列表项前没有任何项目符号。**

**无序列表**

**格式：**

**<ul>**

**<li>列表项1</li>**

**<li>列表项2</li>**

**<li>列表项3</li>**

**......**

**</ul>**

**<ul></ul>标签用于定义无序列表**

**<li></li>标签用于描述具体的列表项**

**注意：每对<ul></ul>中至少应包含一对<li></li>。**

**<ul>和<li>都拥有type属性，用于指定列表项目符号。**

**无序列表的type属性值**

|  |  |
| --- | --- |
| **type属性值** | **显示效果** |
| **disc（默认值）** | **●** |
| **circle** | **○** |
| **square** | **■** |

**注意：不赞成使用无序列表的type属性，一般通过CSS样式属性替代。**

**<li>与</li>之间相当于一个容器，可以容纳所有的元素。但是<ul></ul>中只能嵌套<li></li>，直接在<ul></ul>标签中输入文字的做法是不被允许的。**

**有序列表**

**概念：有序列表即为有排列顺序的列表。网页中常见的歌曲排行榜、游戏排行榜等都可以通过有序列表来定义。**

**格式：**

**<ol>**

**<li>列表项1</li>**

**<li>列表项2</li>**

**<li>列表项3</li>**

**......**

**</ol>**

**<ol></ol>标签用于定义无有序列表**

**<li></li>标签用于描述具体的列表项**

**注意：每对<ol></ol>中至少应包含一对<li></li>。**

**有序列表中，除了type属性之外，还可以为<ol>定义start属性、为<li>定义value属性。**

|  |  |  |
| --- | --- | --- |
| **属性** | **属性值** | **描述** |
| **type** | **1（默认）** | **项目符号显示为数字1 2 3…** |
| **a或A** | **项目符号显示为英文字母a b c d…或A B C…** |
| **i或I** | **项目符号显示为罗马数字i ii iii…或I II III…** |
| **start** | **数字** | **规定项目符号的起始值** |
| **value** | **数字** | **规定项目符号的数字** |

**注意：各浏览器对有序列表的type和value属性的解析不同。**

**不赞成使用<ol>、<li>的type、start和value属性，一般通过CSS样式属性替代。**

**定义列表**

**定义列表常用于图文混排，其中<dt></dt>标签中插入图片，<dd></dd>标签中放入对图片解释说明的文字。**

**格式：**

**<dl>**

**<dt>名词1</dt>**

**<dd>名词1解释1</dd>**

**<dd>名词1解释2</dd>**

**...**

**<dt>名词2</dt>**

**<dd>名词2解释1</dd>**

**<dd>名词2解释2</dd>**

**...**

**</dl>**

**<dl></dl>标签用于指定定义列表**

**<dt></dt>标签用于指定术语名词**

**<dd></dd>标签用于对名词进行解释和描述**

**列表嵌套**

**在使用列表时，列表项中也有可能包含若干子列表项。这种在列表项中定义子列表项的方法被称为列表的嵌套。**

文本

中度可信度描述已自动生成

子分类

商品分类

**格式：**

**<ul>**

**<li>咖啡**

**<ol>**

**<li>拿铁</li>**

**<li>摩卡</li>**

**</ol>**

**</li>**

**<li>茶**

**<ul>**

**<li>碧螺春</li>**

**<li>龙井</li>**

**</ul>**

**</li>**

**</ul>**

## 6.2 CSS控制列表样式

**定义无序或有序列表时，可以通过标签的属性控制列表的项目符号，但是这种方式实现的效果并不理想，因此需要使用CSS中的列表样式属性。**

**list-style-type属性：list-style-type属性用于控制无序和有序列表的项目符号。**

**list-style-image属性：list-style-image属性可以为各个列表项设置项目图像，使列表的样式更加美观。**

**list-style-position属性：list-style-position属性用于控制列表项目符号的位置。**

**list-style-type的属性值及显示效果**

|  |  |  |
| --- | --- | --- |
| **列表类型** | **属性值** | **显示效果** |
| **无序列表（ul）** | **disc** | **●** |
| **circle** | **○** |
| **square** | **■** |
| **有序列表（ol）** | **decimal** | **阿拉伯数字1、2、3......** |
| **upper-alpha** | **大写英文字母A、B、C......** |
| **lower-alpha** | **小写英文字母a、b、c......** |
| **upper-roman** | **大写罗马数字I、II、III......** |
| **lower-roman** | **小写罗马数字i、ii、iii......** |
| **<ul>、<ol>公共属性** | **none** | **不显示任何符号** |

**注意：因为各个浏览器对list-style-type属性的解析不同。因此，在实际网页制作过程中不推荐使用list-style-type属性。**

**list-style-image属性**

**基本格式**

**ul{list-style-image:url(图片路径);}**

**例如：**

**<style type="text/css">**

**ul{list-style-image:url(images/book.png);}**

**</style>**

**list-style-position属性**

**inside：列表项目符号位于列表文本以内。**

**outside：列表项目符号位于列表文本以外。**

**图形用户界面

低可信度描述已自动生成**

**例如：**

**<style type="text/css">**

**.in{list-style-position:inside;}**

**.out{list-style-psition:outside;}**

**li{ border:1px solid #CCC;}**

**</style>**

**复合属性：列表样式复合属性list-style**

**list-style:列表项目符号 列表项目符号的位置 列表项目图像;**

**在实际网页制作过程中，为了更高效地控制列表项目符号，通常将list-style的属性值定义为none，然后通过为<li>设置背景图像的方式实现不同的列表项目符号。**

**使用复合属性list-style时，通常按上面语法格式中的顺序书写，各个样式之间以空格隔开，不需要的样式可以省略。**

## 6.3 超链接标签

**<a href="跳转目标" target="目标窗口的弹出方式">文本或图像</a>**

**href用于指定链接目标的url地址，当为<a>标签应用href属性时，它就具有了超链接的功能。**

**target：用于指定链接页面的打开方式，其取值有\_self和\_blank两种，其中\_self为默认值，意为在原窗口中打开，\_blank为在新窗口中打开。**

**注意：1、暂时没有确定链接目标时，通常将<a>标签的href属性值定义为“#”**

**(即href="#")，表示该链接暂时为一个空链接。**

**2、不仅可以创建文本超链接，在网页中各种网页元素，如图像、 表格、**

**音频、视频等都可以添加超链接。**

**3、当给图像添加超链接时，图像会自动加上边框，通常需要清除超链接**

**图像的边框。**

**锚点链接：HTML语言提供了一种特殊的链接——锚点链接，通过创建锚点链接，用户能够快速定位到目标内容。**

**创建锚点链接：  
使用“<a href="#id名">链接文本</a>”创建链接文本**

**使用相应的id名标注跳转目标的位置**

**例如：**

**<a href="#one">平面广告设计</a>**

**.**

**<h3 id="one">平面广告设计</h3>/跳转目标的位置**

## 6.4 链接伪类控制超链接

**超链接标签<a>的伪类**

|  |  |
| --- | --- |
| **超链接标签<a>的伪类** | **含义** |
| **a:link{ CSS样式规则; }** | **未访问时超链接的状态** |
| **a:visited{ CSS样式规则; }** | **访问后超链接的状态** |
| **a:hover{ CSS样式规则; }** | **鼠标经过、悬停时超链接的状态** |
| **a:active{ CSS样式规则; }** | **鼠标点击不动时超链接的状态** |

**例如：**

**图形用户界面, 文本, 应用程序, 聊天或短信

描述已自动生成**

**同时使用链接的4种伪类时，通常按照a:link、a:visited、a:hover和a:active的顺序书写，否则定义的样式可能不起作用。**

**除了文本样式之外，链接伪类还常常用于控制超链接的背景、边框等样式。**

# 第七章 表格与表单

## 7.1 表格

### 7.1.1 创建表格

**<table></table> 定义一个表格**

**<tr></tr> 定义表格中的一行，嵌套在<table></table>中**

**<td></td> 定义表格中的单元格，嵌套在<tr></tr>中**

### 7.1.2 <table>标签的属性

|  |  |  |
| --- | --- | --- |
| **属性** | **描述** | **常用属性值** |
| **border** | **设置表格的边框（默认border="0"为无边框）** | **像素值** |
| **cellspacing** | **设置单元格与单元格边框之间的空白间距** | **像素值（默认为2像素）** |
| **cellpadding** | **设置单元格内容与单元格边框之间的空白间距** | **像素值（默认为1像素）** |
| **width** | **设置表格的宽度** | **像素值** |
| **height** | **设置表格的高度** | **像素值** |
| **align** | **设置表格在网页中的水平对齐方式** | **left、center、right** |
| **bgcolor** | **设置表格的背景颜色** | **预定义的颜色值、十六进制#RGB、rgb(r,g,b)** |
| **background** | **设置表格的背景图像** | **url地址** |

**<table>标签属性在实体表格中的表现如下图所示：**

**图形用户界面

描述已自动生成**

**<table>标签属性在实体表格中的表现如下图所示：**

**日历

描述已自动生成**

**<table>标签属性在实体表格中的表现如下图所示：**

**图形用户界面, 应用程序

描述已自动生成**

### 7.1.3 <tr>标签的属性

|  |  |  |
| --- | --- | --- |
| 属性 | 描述 | 常用属性值 |
| height | 设置行高度 | 像素值 |
| align | 设置一行内容的水平对齐方式 | left、center、right |
| valign | 设置一行内容的垂直对齐方式 | top、middle、bottom |
| bgcolor | 设置行背景颜色 | 预定义的颜色值、十六进制#RGB、rgb(r,g,b) |
| background | 设置行背景图像 | url地址 |

**学习<tr>的属性时需要注意以下几点：**

**<tr>标签无宽度属性width，其宽度取决于表格标签<table>。**

**对<tr>标签应用valign属性，用于设置一行内容的垂直对齐方式。**

**虽然可以对<tr>标签应用background属性，但是在<tr>标签中此属性兼容问题严重。**

### 7.1.4 <td>标签的属性

|  |  |  |
| --- | --- | --- |
| **属性名** | **含义** | **常用属性值** |
| **width** | 设置单元格的宽度 | 像素值 |
| **height** | 设置单元格的高度 | 像素值 |
| **align** | 设置单元格内容的水平对齐方式 | left、center、right |
| **valign** | 设置单元格内容的垂直对齐方式 | top、middle、bottom |
| **bgcolor** | 设置单元格的背景颜色 | 预定义的颜色值、十六进制#RGB、rgb(r,g,b) |
| **background** | 设置单元格的背景图像 | url地址 |
| **colspan** | 设置单元格横跨的列数（用于合并水平方向的单元格） | 正整数 |
| **rowspan** | 设置单元格竖跨的行数（用于合并竖直方向的单元格） | 正整数 |

**学习<td>的属性时需要注意以下几点：**

**在<td>标签的属性中，重点掌握colspan和rolspan，其他的属性了解即可，不建议使用，均可用CSS样式属性替代。**

**当对某一个<td>标签应用width属性设置宽度时，该列中的所有单元格均会以设置的宽度显示。**

**当对某一个<td>标签应用height属性设置高度时，该行中的所有单元格均会以设置的高度显示。**

### 7.1.5 <th>标签及其属性

**<th></th>标签用来设置表头，其文本默认加粗居中显示。**

**手机屏幕的截图

描述已自动生成**

### 7.1.6 表格结构

**头部：定义表格的头部,包含网页的logo和导航等头部信息。<thead></thead>**

**主体：定义表格的页脚，包含网页底部的企业信息等。<tfoot></ tfoot >**

**页脚：定义表格的主体，包含网页中除头部和底部之外的内容。<tbody></tbody>**

**学习表格的结构时需要注意以下几点**

**一个表格只能定义一对<thead></thead>、一对<tfoot></ tfoot >和多对<tbody></ tbody >，它们必须按<thead></thead>、<tfoot></tfoot >和<tbody></tbody >的顺序使用。之所以将 <tfoot></ tfoot >置于<tbody></ tbody >之前，是为了使浏览器在收到全部数据之前即可显示页脚。**

**使用表格的结构划分标记后，搜索引擎可以更好地理解网页内容，但表格的实际显示效果并不会改变。**

## 7.2 CSS控制表格样式

### 7.2.1 CSS控制表格边框

**图形用户界面, 应用程序, 表格

描述已自动生成**

**CSS代码如下所示：**

**table{**

**width:280px;**

**height:280px;**

**border:1px solid #F00; /\*设置table的边框\*/**

**}**

**td,th{ border:1px solid #F00; } /\*为单元格单独设置边框\*/**

**学习CSS控制表格边框时需要注意以下几点：**

**border-collapse属性的属性值除了collapse（合并）之外，还可以为separate（分离），默认为separate。**

**当表格的border-collapse属性设置为collapse时， HTML中设置的cellspacing属性值无效。**

**行标签<tr>无border样式属性，本书不再做具体的演示，初学者可以自己测试加深理解。**

### 7.2.2 CSS控制单元格边距

**知识概述：设置单元格内容与边框之间的距离，可以对<td>标签应用内边距样式属性padding，或对<table>标签应用HTML标签属性cellpadding。**

**th,td{**

**border:1px solid #30F; /\*为单元格单独设置边框\*/**

**padding:20px; /\*为单元格内容与边框设置20px的内边距\*/**

**margin:20px; /\*为单元格与单元格边框之间设置20px的外边距\*/**

**}**

**学习CSS控制单元格边距时需要注意以下几点：**

**行标签<tr>无内边距属性padding和外边距属性margin。**

**外边距属性margin对单元格无效，要想设置相邻单元格边框之间的距离，只能对<table>标签应用HTML标签属性cellspacing。**

### 7.2.3 CSS控制单元格的宽高

**知识概述：对单元格标签<td>应用width和height属性，可以控制单元格宽度和高度。**

**td{**

**width:20px;**

**Height:20px;**

**}**

**对同一行中的单元格定义不同的高度，或对同一列中的单元格定义不同的宽度时，最终的宽度或高度将取其中的较大者。**

## 7.3 表单

### 7.3.1 表单的构成

**提示信息：一个表单中通常需要包含一些说明性的文字，提示用户进行填写和操作。**

**表单域：相当于一个容器，用来容纳所有的表单控件和提示信息。**

**表单控件：包含了具体的表单功能项，如单行文本输入框、密码输入框、复选框、提交按钮、重置按钮等。**

### 7.3.2 创建表单

**知识概述：<form></form>标签用来创建表单。**

**<form action="url地址" method="提交方式" name="表单名称">**

**各种表单控件**

**</form>**

**结论：<form>与</form>之间的表单控件是由用户自定义的，action、method和name为表单标签<form>的常用属性。**

**表单属性**

**action属性:** **action属性用于指定接收并处理表单数据的服务器程序的url地址。**

**method属性:** **method属性用于设置表单数据的提交方式，其取值为get或post。**

**name属性:** **name属性用于指定表单的名称，以区分同一个页面中的多个表单**

## 7.4 表单控件

**图形用户界面, 应用程序

描述已自动生成**

**大致可分为input控件、textarea控件、select控件三大类。**

### 7.4.1 input控件

**知识概述:** **<input />元素是表单中最常见的元素，网页中常见的单行文本框、单选按钮、复选框等都是通过它定义的。**

**<input type="控件类型"/>**

**<input />标签为单标签，type属性为其最基本的属性，其取值有多种，用于指定不同的控件类型。**

|  |  |  |
| --- | --- | --- |
| 属性 | 属性值 | 描述 |
| type | text | 单行文本输入框 |
| password | 密码输入框 |
| radio | 单选按钮 |
| checkbox | 复选框 |
| button | 普通按钮 |
| submit | 提交按钮 |
| reset | 重置按钮 |
| image | 图像形式的提交按钮 |
| hidden | 隐藏域 |
| file | 文件域 |
| name | 由用户自定义 | 控件的名称 |
| value | 由用户自定义 | input控件中的默认文本值 |
| size | 正整数 | input控件在页面中的显示宽度 |
| readonly | readonly | 该控件内容为只读（不能编辑修改） |
| disabled | disabled | 第一次加载页面时禁用该控件（显示为灰色） |
| checked | checked | 定义选择控件默认被选中的项 |
| maxlength | 正整数 | 控件允许输入的最多字符数 |

**在HTML中，<input>标签拥有多个type属性值，用于定义不同的控件类型。**

**图形用户界面, 应用程序

描述已自动生成**

**图形用户界面, 应用程序

描述已自动生成**

**图形用户界面

描述已自动生成**

**图形用户界面

描述已自动生成**

**图片包含 图形用户界面

描述已自动生成**

**图片包含 图形用户界面

描述已自动生成**

**图片包含 图表

描述已自动生成**

**图形用户界面, 应用程序

描述已自动生成**

**图片包含 图形用户界面

描述已自动生成**

**图形用户界面

描述已自动生成**

### 7.4.2 textarea控件

**格式：**

**<textarea cols="每行中的字符数" rows="显示的行数">**

**文本内容**

**</textarea>**

**结论：cols和rows为<textarea>标记的必须属性，其中cols用来定义多行文本输入框每行中的字符数，rows用来定义多行文本输入框显示的行数，取值均为正整数。**

|  |  |  |
| --- | --- | --- |
| **属性** | **属性值** | **描述** |
| **name** | **由用户自定义** | **控件的名称** |
| **readonly** | **readonly** | **该控件内容为只读（不能编辑修改）** |
| **disabled** | **disabled** | **第一次加载页面时禁用该控件（显示为灰色）** |

### 7.4.3 select控件

**浏览网页时，经常会看到包含多个选项的下拉菜单。**

**图形用户界面

描述已自动生成**

**格式：**

**<select>**

**<option>选项1</option>**

**<option>选项2</option>**

**<option>选项3</option>**

**...**

**</select>**

|  |  |  |
| --- | --- | --- |
| **标签名** | **常用属性** | **描述** |
| **<select>** | **size** | **指定下拉菜单的可见选项数（取值为正整数）。** |
| **multiple** | **定义multiple="multiple"时，下拉菜单将具有多项选择的功能，方法为按住Ctrl键的同时选择多项。** |
| **<option>** | **selected** | **定义selected =" selected "时，当前项即为默认选中项。** |

# 7.5 HTML5表单新属性

### 7.5.1 全新的form属性

**autocomplete属性：autocomplete属性用于指定表单是否有自动完成功能**

**novalidate属性：novalidate属性指定在提交表单时取消对表单进行有效的检查。**

### 7.5.2 全新的表单控件

**datalist控件**

**网页中的列表通过datalist内的option进行创建。如果用户不希望从列表中选择某项，也可以自行输入其他内容。datalist控件通常与input控件配合使用，来定义input的取值。在使用<datalist>控件时，需要通过id属性为其指定一个唯一的标识，然后为input控件指定list属性，将该属性值设置为datalist对应的id属性值即可。**

**图形用户界面, 文本, 应用程序

描述已自动生成**

### 7.5.3 全新的input控件类型

**在HTML5中，增加了一些新的input控件类型，通过这些新的控件，可以丰富表单功能，更好的实现表单的控制和验证。**

**图形用户界面

描述已自动生成**

**图形用户界面, 应用程序

描述已自动生成**

**图形用户界面

描述已自动生成**

**图形用户界面, 应用程序

描述已自动生成**

**图形用户界面, 应用程序

描述已自动生成**

**图形用户界面, 应用程序

描述已自动生成**

**图形用户界面

中度可信度描述已自动生成**

**图片包含 图形用户界面

描述已自动生成**

### 7.5.4 全新的input属性

**1、autofocus属性**

**图形用户界面, 文本, 应用程序

描述已自动生成**

**2、form属性**

**图形用户界面, 文本, 应用程序, 聊天或短信

描述已自动生成**

**3、list属性**

**图形用户界面, 文本, 应用程序

描述已自动生成**

**4、multiple属性**

**图形用户界面, 文本, 应用程序

描述已自动生成**

**5、min、max和step属性**

**图形用户界面, 应用程序

描述已自动生成**

**6、pattern属性**

**图形用户界面, 文本, 应用程序

描述已自动生成**

**pattern属性常用的正则表达式和说明**

|  |  |
| --- | --- |
| **正则表达式** | **说明** |
| **^[0-9]\*$** | **数字** |
| **^\d{n}$** | **n位的数字** |
| **^\d{n,}$** | **至少n位的数字** |
| **^\d{m,n}$** | **m-n位的数字** |
| **^(0|[1-9][0-9]\*)$** | **零和非零开头的数字** |
| **^([1-9][0-9]\*)+(.[0-9]{1,2})?$** | **非零开头的最多带两位小数的数字** |
| **^(\-|\+)?\d+(\.\d+)?$** | **正数、负数、和小数** |
| **^\d+$ 或 ^[1-9]\d\*|0$** | **非负整数** |
| **^-[1-9]\d\*|0$ 或 ^((-\d+)|(0+))$** | **非正整数** |
| **^[\u4e00-\u9fa5]{0,}$** | **汉字** |
| **^[A-Za-z0-9]+$ 或 ^[A-Za-z0-9]{4,40}$** | **英文和数字** |
| **^[A-Za-z]+$** | **由26个英文字母组成的字符串** |
| **^[A-Za-z0-9]+$** | **由数字和26个英文字母组成的字符串** |

|  |  |
| --- | --- |
| **正则表达式** | **说明** |
| **^\w+$ 或 ^\w{3,20}$** | **由数字、26个英文字母或者下划线组成的字符串** |
| **^[\u4E00-\u9FA5A-Za-z0-9\_]+$** | **中文、英文、数字包括下划线** |
| **^\w+([-+.]\w+)\*@\w+([-.]\w+)\*\.\w+([-.]\w+)\*$** | **Email地址** |
| **[a-zA-z]+://[^\s]\* 或 ^http://([\w-]+\.)+[\w-]+(/[\w-./?%&=]\*)?$** | **URL地址** |
| **^\d{15}|\d{18}$** | **身份证号(15位、18位数字)** |
| **^([0-9]){7,18}(x|X)?$ 或 ^\d{8,18}|[0-9x]{8,18}|[0-9X]{8,18}?$** | **以数字、字母x结尾的短身份证号码** |
| **^[a-zA-Z][a-zA-Z0-9\_]{4,15}$** | **帐号是否合法(字母开头，允许5-16字节，允许字母数字下划线)** |
| **^[a-zA-Z]\w{5,17}$** | **密码(以字母开头，长度在6~18之间，只能包含字母、数字和下划线)** |

**7、placeholder属性**

**图形用户界面, 文本, 应用程序, 聊天或短信

描述已自动生成**

**8、required属性**

**图形用户界面, 文本

描述已自动生成**

## 7.6 CSS控制表单样式

**使用CSS可以轻松地控制表单控件的样式，主要体现在控制表单控件的字体、边框、背景和内边距等。**

# 第八章 DIV+CSS布局

## 8.1 布局概述

**最简单的页面布局，主要由头部（header）、导航（nav）、焦点图（banner）、内容（content）、页面底部（footer）五部分组成。**

## 8.2 布局常用属性

**在使用div+css进行网页布局时，经常会使用一些属性对标签进行控制，常见的属性有浮动属性（float属性）和定位属性（position属性）。**

### 8.2.1 标签的浮动属性

**所谓元素的浮动是指设置了浮动属性的元素会脱离标准文档流的控制，移动到其父元素中指定位置的过程。**

**基本语法格式**

**选择器{float:属性值;}**

|  |  |
| --- | --- |
| **属性值** | **描述** |
| **left** | **元素向左浮动** |
| **right** | **元素向右浮动** |
| **none** | **元素不浮动（默认值）** |

**运用clear属性清除浮动**

**格式：选择器{clear:属性值;}**

**常用属性值**

|  |  |
| --- | --- |
| **属性值** | **描述** |
| **left** | **不允许左侧有浮动元素（清除左侧浮动的影响）** |
| **right** | **不允许右侧有浮动元素（清除右侧浮动的影响）** |
| **both** | **同时清除左右两侧浮动的影响** |

**空标签：在浮动元素之后添加空标签，并对该标签应用“clear:both”样式，可清除浮动。这个空标签可以为<div>、<p>、<hr />等任何标签。**

**overflow：“overflow:hidden;”样式，也可以清除浮动对该元素的影响，该方法弥补了空标签清除浮动的不足。**

**after伪对象:** **使用after伪对象也可以清除浮动，但是该方法只适用于IE8及以上版本浏览器和其他非IE浏览器。**

### 8.2.2 标签的定位属性

**定位模式:** **position属性用于定义元素的定位模式，其基本语法格式如下：**

**选择器{position:属性值；}**

**position属性的常用值有四个,**

|  |  |
| --- | --- |
| **值** | **描述** |
| **static** | **自动定位（默认定位方式）** |
| **relative** | **相对定位，相对于其原文档流的位置进行定位** |
| **absolute** | **绝对定位，相对于其上一个已经定位的父元素进行定位** |
| **fixed** | **固定定位，相对于浏览器窗口进行定位** |

**边偏移:** **通过边偏移属性top、bottom、left或right，来精确定义定位元素的位置，其取值为不同单位的数值或百分比，如下表所示**

|  |  |
| --- | --- |
| **边偏移属性** | **描述** |
| **top** | **顶端偏移量，定义元素相对于其父元素上边线的距离** |
| **bottom** | **底部偏移量，定义元素相对于其父元素下边线的距离** |
| **left** | **左侧偏移量，定义元素相对于其父元素左边线的距离** |
| **Right** | **右侧偏移量，定义元素相对于其父元素右边线的距离** |

**静态定位:** **元素的默认定位方式，当position属性的取值为static时，可以将元素定位于静态位置。 所谓静态位置就是各个元素在HTML文档流中默认的位置。任何元素在默认状态下都会以静态定位来确定自己的位置，所以当没有定义position属性时，并不说明该元素没有自己的位置，它会遵循默认值显示为静态位置。在静态定位状态下，无法通过边偏移属性（top、bottom、left或right）来改变元素的位置。**

**相对定位是将元素相对于它在标准文档流中的位置进行定位**

**图形用户界面

描述已自动生成**

**图片包含 图形用户界面

描述已自动生成**

**绝对定位:** **将元素依据最近的已经定位（绝对、固定或相对定位）的父元素进行定位，若所有父元素都没有定位，则依据body根元素（浏览器窗口）进行定位。**

**图形用户界面, 应用程序

描述已自动生成**

**图片包含 日程表

描述已自动生成**

**固定定位:绝对定位的一种特殊形式，它以浏览器窗口作为参照物来定义网页元素。当position属性的取值为fixed时，即可将元素的定位模式设置为固定定位。**

## 8.3 布局其他属性

### 8.3.1 overflow属性

**overflow属性可以解决溢出问题，其基本语法格式如下：**

**选择器{overflow:属性值;}**

**overflow属性的常用值有四个**

|  |  |
| --- | --- |
| **属性值** | **描述** |
| **visible** | **内容不会被修剪，会呈现在元素框之外（默认值）** |
| **hidden** | **溢出内容会被修剪，并且被修剪的内容是不可见的** |
| **auto** | **在需要时产生滚动条，即自适应所要显示的内容** |
| **scroll** | **溢出内容会被修剪，且浏览器会始终显示滚动条** |

### 8.3.2 Z-index标签层叠

**图示

描述已自动生成**

## 8.4 布局类型

### 8.4.1 单列布局

**“单列布局”是网页布局的基础，所有复杂的布局都是在此基础上演变而来的。**

### 8.4.2 两列布局

**“两列布局”和“一列布局”类似，只是网页内容被分为了左右两部分**

### 8.4.3 三列布局

**对于一些大型网站，特别是电子商务类网站，由于内容分类较多，通常需要采用“三列布局”的页面布局方式。**

### 8.4.4 通栏布局

**无论布局类型是单列布局、两列布局或者多列布局，为了网站的美观，网页中的一些模块，例如头部、导航、焦点图或页面底部等经常需要通栏显示。**

**图形用户界面, 应用程序

描述已自动生成**

**设置为通栏后，无论页面放大或缩小，该模块都将横铺于浏览器窗口中。**

### 8.4.5 全新的HTML5结构元素

**1、header元素**

**HTML5中的header元素是一种具有引导和导航作用的结构元素，该元素可以包含所有通常放在页面头部的内容。**

**格式：**

**<header>**

**<h1>网页主题</h1>**

**...**

**</header>**

**2、nav标签**

**nav元素用于定义导航链接，是HTML5新增的元素，该元素可以将具有导航性质的链接归纳在一个区域中，使页面元素的语义更加明确。**

**格式：**

**<nav>**

**<ul>**

**<li><a href="#">首页</li>**

**<li><a href="#">公司概况</li>**

**<li><a href="#">产品展示</li>**

**<li><a href="#">联系我们</li>**

**</ul>**

**</nav>**

**图片包含 图示

描述已自动生成**

**3、footer标签**

**footer元素用于定义一个页面或者区域的底部，它可以包含所有通常放在页面底部的内容。**

**例如：**

**<article>**

**文章内容**

**<footer>**

**文章分页列表**

**</footer>**

**</article>**

**<footer>**

**页面底部**

**</footer>**

**4、article标签**

**article元素代表文档、页面或者应用程序中与上下文不相关的独立部分，该元素经常被用于定义一篇日志、一条新闻或用户评论等。**

**例如：**

**图形用户界面, 文本, 应用程序

描述已自动生成**

**5、section标签**

**section元素用于对网站或应用程序中页面上的内容进行分块，一个section元素通常由内容和标题组成。**

**注意：**

* **不要将section元素用作设置样式的页面容器，那是div的特性。**
* **如果article元素、aside元素或nav元素更符合使用条件，那么不要使用section元素。**
* **没有标题的内容区块不要使用section元素定义。**

**6、aside标签**

**aside元素用来定义当前页面或者文章的附属信息部分，它可以包含与当前页面或主要内容相关的引用、侧边栏、广告、导航条等其他类似的有别于主要内容的部分。**

**aside元素的用法主要分为两种：**

* **被包含在article元素内作为主要内容的附属信息。**
* **在article元素之外使用，作为页面或站点全局的附属信息部分。**

### 8.4.6 网页模块的命名规范

**如果没有统一的命名规范进行必要的约束，随意地命名，就会使整个网站的后续工作很难进行。**

**命名需要遵循以下几个原则：**

* + **避免使用中文字符命名（例如id=“导航栏”）。**
  + **不能以数字开头命名（例如id=“1nav”）。**
  + **不能占用关键字（例如id=“h3”）。**
  + **用最少的字母达到最容易理解的意义。**

**网页中，常用的命名方式：**

**驼峰式：除了第一个单词外后面的单词首写字母都要大写（例如：partOne）。**

**帕斯卡：每一个单词之间用“\_”连接（例如：content\_one）。**

|  |  |  |  |
| --- | --- | --- | --- |
| **相关模块** | **命名** | **相关模块** | **命名** |
| **头** | **header** | **内容** | **content/container** |
| **导航** | **nav** | **尾** | **footer** |
| **侧栏** | **sidebar** | **栏目** | **column** |
| **左边、右边、中间** | **left right center** | **登录条** | **loginbar** |
| **标志** | **logo** | **广告** | **banner** |
| **页面主体** | **main** | **热点** | **hot** |
| **新闻** | **news** | **下载** | **download** |
| **子导航** | **subnav** | **菜单** | **menu** |
| **子菜单** | **submenu** | **搜索** | **search** |
| **友情链接** | **frIEndlink** | **版权** | **copyright** |
| **滚动** | **scroll** | **标签页** | **tab** |
| **文章列表** | **list** | **提示信息** | **msg** |
| **小技巧** | **tips** | **栏目标题** | **title** |
| **加入** | **joinus** | **指南** | **guild** |
| **服务** | **service** | **注册** | **regsiter** |
| **状态** | **status** | **投票** | **vote** |
| **合作伙伴** | **partner** |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **CSS文件** | **命名** | **CSS文件** | **命名** |
| **主要样式** | **master** | **基本样式** | **base** |
| **模块样式** | **module** | **版面样式** | **layout** |
| **主题** | **themes** | **专栏** | **columns** |
| **文字** | **font** | **表单** | **forms** |
| **打印** | **print** |  |  |

# 第九章 多媒体嵌入

## 9.1视频音频嵌入技术概述

**运用HTML5的video和audio标签可以在页面中嵌入视频或音频文件，如果想要这些文件在页面中加载播放，还需要设置正确的多媒体格式。**

## 9.2视频文件和音频文件的格式

**视频格式：Ogg、MPEG4、WebM**

**音频格式：Ogg、MP3、Wav**

## 9.3 嵌入视频和音频

### 9.3.1 在HTML5中嵌入视频

**格式：<video src="视频文件路径" controls="controls"></video>**

|  |  |  |
| --- | --- | --- |
| **属性** | **值** | **描述** |
| **autoplay** | **autoplay** | **当页面载入完成后自动播放视频。** |
| **loop** | **loop** | **视频结束时重新开始播放。** |
| **preload** | **preload** | **如果出现该属性，则视频在页面加载时进行加载，并预备播放。如果使用 "autoplay"，则忽略该属性。** |
| **poster** | **url** | **当视频缓冲不足时，该属性值链接一个图像，并将该图像按照一定的比例显示出来。** |

### 9.3.2 在HTML5中嵌入音频

**格式：<audio src="音频文件路径" controls="controls"></audio>**

|  |  |  |
| --- | --- | --- |
| **属性** | **值** | **描述** |
| **autoplay** | **autoplay** | **当页面载入完成后自动播放音频。** |
| **loop** | **loop** | **音频结束时重新开始播放。** |
| **preload** | **preload** | **如果出现该属性，则音频在页面加载时进行加载，并预备播放。如果使用 "autoplay"，则忽略该属性。** |

### 9.3.3 视频音频文件的兼容性问题

**在HTML5中，运用source元素可以为video元素或audio元素提供多个备用文件。**

**格式：**

**<audio controls="controls">**

**<source src="音频文件地址" type="媒体文件类型/格式">**

**<source src="音频文件地址" type="媒体文件类型/格式">**

**……**

**</audio>**

**src用于指定媒体文件的URL地址。**

**type指定媒体文件的类型。**

### 9.3.4 调用网络音频视频文件

**例如：**

**<video src="http://www.w3school.com.cn/i/movie.ogg" controls="controls">调用网络视频文件**

**</video>**

## 9.4 CSS控制视频的宽高

**在HTML5中，经常会通过为video元素添加宽高的方式给视频预留一定的空间，这样浏览器在加载页面时就会预先确定视频的尺寸，为其保留合适的空间，使页面的布局不产生变化。**

# 第十章 CSS3高级应用

## 10.1 过渡

### 10.1.1 transition-property属性

**transition-property 属性用于指定应用过渡效果的CSS属性的名称。**

**格式：**

**transition-property: none | all | property;**

|  |  |
| --- | --- |
| 属性值 | 描述 |
| none | 没有属性会获得过渡效果。 |
| all | 所有属性都将获得过渡效果。 |
| property | 定义应用过渡效果的CSS属性名称，多个名称之间以逗号分隔。 |

### 10.1.2 transition-duration属性

**transition-duration属性用于定义过渡效果花费的时间，默认值为0，常用单位是秒（s）或者毫秒（ms）。**

**格式：transition-duration:time;**

### 10.1.3 transition-timing-function属性

**transition-timing-function属性规定过渡效果的速度曲线，默认值为“ease“。**

**格式：**

**transition-timing-function:linear|ease|ease-in|ease-out|ease-in-out|cubic-bezier(n,n,n,n);**

|  |  |
| --- | --- |
| 属性值 | 描述 |
| linear | 指定以相同速度开始至结束的过渡效果，等同于cubic-bezier(0,0,1,1)）。 |
| ease | 指定以慢速开始，然后加快，最后慢慢结束的过渡效果，等同于cubic-bezier(0.25,0.1,0.25,1）。 |
| ease-in | 指定以慢速开始，然后逐渐加快（淡入效果）的过渡效果，等同于cubic-bezier(0.42,0,1,1）。 |
| ease-out | 指定以慢速结束（淡出效果）的过渡效果，等同于cubic-bezier(0,0,0.58,1）。 |
| ease-in-out | 指定以慢速开始和结束的过渡效果，等同于cubic-bezier(0.42,0,0.58,1）。 |
| cubic-bezier(n,n,n,n) | 定义用于加速或者减速的贝塞尔曲线的形状，它们的值在0~1之间。 |

### 10.1.4 transition-delay属性

**transition-delay属性规定过渡效果何时开始，默认值为0，常用单位是秒（s）或者毫秒（ms）。**

**格式：transition-delay:time;**

**正数:过渡动作会延迟触发**

**负数：过渡动作会从该时间点开始，之前的动作被截断。**

### 10.1.5 transition属性

**transition属性是一个复合属性，用于在一个属性中设置transition-property、transition-duration、transition-timing-function、transition-delay四个过渡属性**

**格式：transition：property duration timing-function delay;**

**使用transition属性设置多个过渡效果时，它的各个参数必须按照顺序进行定义，不能颠倒。**

**例如：transition:border-radius 5s ease-in-out 2s;**

**transition属性：无论是单个属性还是简写属性，使用时都可以实现多个过渡效果。如果使用transition简写属性设置多种过渡效果，需要为每个过渡属性集中指定所有的值，并且使用逗号进行分隔。**

## 10.2 变形

### 10.2.1认识transform

**CSS3的变形（transform）属性可以让元素在一个坐标系统中变形。**

**格式：transform：none | transform-functions;**

**transform属性的默认值为none，适用于内联元素和块元素，表示不进行变形。transform-function用于设置变形函数，可以是一个或多个变形函数列表。**

**transform-function函数**

**matrix（）：定义矩形变换，即基于X和Y坐标重新定位元素的位置。**

**translate（）：移动元素对象，即基于X和Y坐标重新定位元素。**

**scale（）：缩放元素对象，可以使任意元素对象尺寸发生变化，取值包括正数、负数和小数。**

**rotate（）：旋转元素对象，取值为一个度数值。**

**skew（）：倾斜元素对象，取值为一个度数值。**

### 10.2.2 2D转换

**平移：使用translate（）方法能够重新定义元素的坐标，实现平移的效果**

**格式：transform:translate（x-value,y-value）;**

**x-value指元素在水平方向上移动的距离，y-value指元素在垂直方向上移动的距离。如果省略了第二个参数，则取默认值0。当值为负数时，表示反方向移动元素。**

**图示, 示意图

描述已自动生成**

**缩放：scale() 方法用于缩放元素大小，该函数包含两个参数值，分别用来定义宽度和高度的缩放比例。**

**格式：transform:scale(x-axis,y-axis);**

**x-axis和y-axis参数值可以是正数、负数和小数。正数值表示基于指定的宽度和高度放大元素。负数值不会缩小元素，而是反转元素（如文字被反转），然后再缩放元素。如果第二个参数省略，则第二个参数等于第一个参数值。**

**图表, 箱线图

描述已自动生成**

**倾斜：skew()方法能够让元素倾斜显示，该函数包含两个参数值，分别用来定义X轴和Y轴坐标倾斜的角度。**

**格式：transform:skew(x-angle,y-angle);**

**参数x-angle和y-angle表示角度值，第一个参数表示相对于X轴进行倾斜，第二个参数表示相对于Y轴进行倾斜，如果省略了第二个参数，则取默认值0。**

**图示

描述已自动生成**

**旋转：rotate()方法能够旋转指定的元素对象，主要在二维空间内进行操作。该方法中的参数允许传入负值，这时元素将逆时针旋转。**

**格式：transform:rotate(angle);**

**参数angle表示要旋转的角度值。如果角度为正数值，则按照顺时针进行旋转，否则按照逆时针旋转。**

**卡通人物

中度可信度描述已自动生成**

**变形操作都是以元素的中心点为基准进行的，如果需要改变这个中心点，可以使用transform-origin属性。**

**格式：transform-origin: x-axis y-axis z-axis;**

**transform-origin属性包含三个参数，其默认值分别为50% 50% 0，各参数的具体含义**

|  |  |
| --- | --- |
| **参数** | **描述** |
| **x-axis** | **定义视图被置于X轴的何处。可能的值有：**   * **left** * **center** * **right** * **length** * **%** |
| **y-axis** | **定义视图被置于Y轴的何处。可能的值有：**   * **top** * **center** * **bottom** * **length** * **%** |
| **z-axis** | **定义视图被置于Z轴的何处。可能的值有：**   * **length** |

### 10.2.3 3D转换

**rotateX() 方法：rotateX()函数用于指定元素围绕X轴旋转。**

**格式：transform:rotateX(a);**

**参数a用于定义旋转的角度值，单位为deg，其值可以是正数也可以是负数。如果值为正，元素将围绕X轴顺时针旋转；反之，如果值为负，元素围绕X轴逆时针旋转。**

**rotateY() 方法：rotateY()函数指定一个元素围绕Y轴旋转**

**格式：transform:rotateY(a);**

**参数a与rotateX(a)中的a含义相同，用于定义旋转的角度。如果值为正，元素围绕Y轴顺时针旋转；反之，如果值为负，元素围绕Y轴逆时针旋转。**

**rotate3D（）方法：在三维空间里，除了rotateX()、rotateY()和rotateZ()函数可以让元素在三维空间中旋转之外，还有一个rotate3d()函数。**

**格式：rotate3d(x,y,z,angle);**

**x、y、z可以取值0或1，当要沿着某一轴转动，就将该轴的值设置为1，否则设置为0。Angle为要旋转的角度。**

**perspective属性：perspective属性可以简单的理解为视距，主要用于呈现良好的3D透视效果。例如我们前面设置的3D旋转果并不明显，就是没有设置perspective的原因。**

**格式：perspective:参数值;**

**perspective属性参数值可以为none或者数值（一般为像素），其透视效果由参数值决定，参数值越小，透视效果越突出。**

**在CSS3中包含很多转换的属性，通过这些属性可以设置不同的转换效果。**

|  |  |  |
| --- | --- | --- |
| **属性名称** | **描述** | **属性值** |
| **transform-style** | **用于保存元素的3D 空间** | **flat：子元素将不保留其 3D 位置。（默认属性）** |
| **preserve-3d子元素将保留其 3D 位置。** |
| **backface-visibility** | **定义元素在不面对屏幕时是否可见** | **visible：背面是可见的。** |
| **hidden：背面是不可见的。** |

**CSS3中还包含很多转换的方法，运用这些方法可以实现不同的转换效果。**

|  |  |
| --- | --- |
| **方法名称** | **描述** |
| **translate3d(x,y,z)** | **定义3D位移** |
| **translateX(x)** | **定义3D位移，仅使用用于 X 轴的值** |
| **translateY(y)** | **定义3D位移，仅使用用于 Y 轴的值** |
| **translateZ(z)** | **定义3D位移，仅使用用于Z 轴的值** |
| **scale3d(x,y,z)** | **定义 3D 缩放** |
| **scaleX(x)** | **定义 3D 缩放，通过给定一个 X 轴的值** |
| **scaleY(y)** | **定义 3D 缩放，通过给定一个 Y 轴的值** |
| **scaleZ(z)** | **定义 3D 缩放，通过给定一个 Z 轴的值** |

## 10.3 动画

### 10.3.1 @keyframes

**格式：**

**@keyframes animationname {**

**keyframes-selector{css-styles;}**

**}**

**css-styles：定义执行到当前关键帧时对应的动画状态，由CSS样式属性进行定义，多个属性之间用分号分隔，不能为空。**

**keyframes-selector：关键帧选择器，即指定当前关键帧要应用到整个动画过程中的位置，值可以是一个百分比、from或者to。其中，from和0%效果相同表示动画的开始，to和100%效果相同表示动画的结束。**

**animationname：表示当前动画的名称，它将作为引用时的唯一标识，因此不能为空。**

### 10.3.2 animation-name属性

**animation-name属性用于定义要应用的动画名称。**

**格式：animation-name: keyframename | none;**

**animation-name 属性初始值为none，适用于所有块元素和行内元素。keyframename参数用于规定需要绑定到选择器的keyframe的名称，如果值为none，则表示不应用任何动画，通常用于覆盖或者取消动画。**

### 10.3.3 animation-duration属性

**animation-duration属性用于定义整个动画效果完成所需要的时间，以秒或毫秒计。**

**格式：animation-duration: time;**

**animation-duration 属性初始值为0，适用于所有块元素和行内元素。time参数是以秒（s）或者毫秒（ms）为单位的时间，默认值为0，表示没有任何动画效果。当值为负数时，则被视为0。**

### 10.3.4 animation-timing-function属性

**animation-timing-function用来规定动画的速度曲线，可以定义使用哪种方式来执行动画效果。**

**格式：animation-timing-function:value;**

**animation-timing-function包括linear、ease-in、ease-out、ease-in-out、cubic-bezier(n,n,n,n)等常用属性值。**

**animation-timing-function的常用属性值**

|  |  |
| --- | --- |
| **属性值** | **描述** |
| **linear** | **动画从头到尾的速度是相同的。** |
| **ease** | **默认。动画以低速开始，然后加快，在结束前变慢。** |
| **ease-in** | **动画以低速开始。** |
| **ease-out** | **动画以低速结束。** |
| **ease-in-out** | **动画以低速开始和结束。** |
| **cubic-bezier(n,n,n,n)** | **在cubic-bezier函数中自己的值。可能的值是从0到1的数值。** |

### 10.3.5animation-delay属性

**animation-delay属性用于定义执行动画效果之前延迟的时间，即规定动画什么时候开始。**

**格式：animation-delay:time;**

**参数time用于定义动画开始前等待的时间，其单位是秒或者毫秒，默认属性值为0。animation-delay属性适用于所有的块元素和行内元素。**

### 10.3.6 animation-iteration-count属性

**animation-iteration-count属性用于定义动画的播放次数**

**格式：animation-iteration-count: number | infinite;**

**animation-iteration-count属性初始值为1，适用于所有的块元素和行内元素。如果属性值为number，则用于定义播放动画的次数；如果是infinite，则指定动画循环播放。**

### 10.3.7 animation-direction属性

**animation-direction属性定义当前动画播放的方向，即动画播放完成后是否逆向交替循环。**

**格式：animation-direction: normal | alternate;**

**animation-direction 属性初始值为normal，适用于所有的块元素和行内元素。该属性包括两个值，默认值normal表示动画每次都会正常显示。如果属性值是"alternate"，则动画会在奇数次数（1、3、5 等等）正常播放，而在偶数次数（2、4、6 等）逆向播放。**

### 10.3.8 animation属性

**格式：**

**animation: animation-name animation-duration animation-timing-function animation-delay animation-iteration-count animation-direction;**

**使用animation属性时必须指定animation-name和animation-duration属性，否则持续的时间为0，并且永远不会播放动画。**

# 第十一章 绘图和数据存储原理

## 11.1 简单的JavaScript

### 11.1.1 JavaScript的引入

**1、 行内式**

**行内式是将JavaScript代码作为HTML标签的属性值使用。**

**例如：单击“test”时，弹出一个警告框提示“Happy”。**

**<a href="javascript:alert('Happy');"> test </a>**

**单击网页中的一个按钮时，就会触发按钮的单击事件。**

**<input type="button" onclick="alert('Happy'); " value="test" >**

**2、 嵌入式**

**在HTML中运用<script>标签及其相关属性可以嵌入JavaScript脚本代码。**

**格式：**

**<head>**

**<script type="text/javascript">**

**// 此处为JavaScript代码**

**</script>**

**</head>**

**在编写JavaScript代码时可以省略type属性。**

**3、 外链式**

**外链式是将所有的JavaScript代码放在一个或多个以.js为扩展名的外部JavaScript文件中，通过<src >标签将这些JavaScript文件链接到HTML文档中**

**格式：**

**<script type="text/Javascript" src="脚本文件路径" >**

**</script>**

**外链式的优势：**

**（1）利于后期修改和维护**

**（2）减轻文件体积、加快页面加载速度**

### 11.1.2 变量

**变量的命名**

**必须以字母或下划线开头，中间可以是数字、字母或下划线。**

**变量名不能包含空格、加、减等符号**

**不能使用JavaScript中的关键字作为变量名，如var int。**

**JavaScript的变量名严格区分大小写，如UserName与username代表两个不同的变量。**

**声明变量基本语法格式**

**var 变量名;**

**声明变量时，需要遵循的规则如下**

**1. 可以使用一个关键字var同时声明多个变量。**

**例如：var a,b,c //同时声明a、b和c三个变量**

**2.** **可以在声明变量的同时对其赋值，即初始化。**

**例如：var a=1,b=2,c=3； //同时声明a、b和c三个变量，并分别对其进行初始化**

3. var语句可以用作for循环和for/in循环的一部分

**4.** **使用var语句多次声明同一个变量，就相当于对变量的重新赋值。**

**例如：**

**var unit, room; // 声明变量**

**var unit = 3; // 为变量赋值**

**var room = 1001; // 为变量赋值**

**var fname = 'Tom', age = 12; // 声明变量的同时赋值**

### 11.1.3 document对象

**如果我们想要在JavaScript中操作某个标签，首先要获取该标签的属性。在JavaScript中通过document对象及其方法可以获取标签属性，如id、name和class等属性。**

|  |  |
| --- | --- |
| 方法 | 说明 |
| document.getElementById() | 返回对拥有指定id名的第一个对象的引用（简单理解为获取指定id名的标签） |
| document.getElementsByName() | 返回带有指定name属性名的对象集合（简单理解为获取指定name名的标签） |
| document.getElementsByTagName() | 返回带有指定标签名的对象集合（简单理解为获取标签名） |
| document.getElementsByClassName() | 返回带有指定类名的对象集合（简单理解为获取指定class名的标签） |

## 11.2 HTML5画布

### 11.2.1 认识画布

**说到画布，其实大家并不陌生，在美术课上，我们可以用画笔在画布上绘画和涂鸦。在网页中，我们把用于绘制图形的特殊区域也称为“画布”，网页设计师可以在该区域义绘制自定的图形样式。**

### 11.2.2 使用画布

**1、创建画布**

**使用HTML5中的canvas标签可以在网页中创建画布。**

**格式：**

**<canvas id="画布名称" width="数值" height="数值">**

**您的浏览器不支持canvas**

**</canvas>**

**2、获取画布**

**要想在JavaScript中控制画布，首先要获取画布。使用getElementById()方法可以获取网页中的画布对象。**

**格式：var canvas = document.getElementById('cavs');**

**3、准备画笔**

**有了画布之后，要开始绘图，还需要准备一只画笔，这支画笔就是context对象。context对象也被称为绘制环境，通过该对象，可以在画布中绘制图形。**

**格式：canvas.getContext('2d');**

**注意：在JavaScript中，我们通常会定义一个变量来保存获取的context对象。**

**var context = canvas.getContext('2d');**

### 11.2.3 绘制线

**一条最简单的线由三部分组成，分别为初始位置、连线端点以及描边**

**图示

描述已自动生成**

**1、初始位置**

**在绘制图形时，我们首先需要确定从哪里下“笔”，这个下“笔”的位置就是初始位置。**

**图示, 文本, 信件

描述已自动生成**

**例如：**

**var cas = document.getElementById('cas');**

**var context = cas.getContext('2d');**

**context.moveTo(100,100);**

**2、连线端点**

**连线端点用于定义一个端点，并绘制一条从该端点到初始位置的连线。在画布中使用lineTo(x,y)方法来定义连线端点。**

**格式：context.lineTo(100,100);**

**3、描边**

**通过初始位置和连线端点可以绘制一条线，但这条线并不能被看到。这时我们需要为线添加描边，让线变得可见。使用画布中的stroke()方法，可以实现线的可视效果。**

**格式：context.stroke();**

### 11.2.4 线的样式

**1、宽度**

**使用画布中的lineWidth属性可以定义线的宽度，该属性的取值为数值（不带单位），以像素为计量。**

**格式：context.lineWidth='10';**

**2、描边颜色**

**使用画布中的strokeStyle属性可以定义线的描边颜色，该属性的取值为十六进制颜色值或颜色英文。**

**例如：**

**context.strokeStyle='#f00';**

**context.strokeStyle='red';**

**3、端点形状**

**默认情况下，线的端点是方形的，通过画布中的lineCap属性可以改变端点的形状。**

**格式：lineCap=’属性值’**

****

|  |  |
| --- | --- |
| **属性值** | **显示效果** |
| **butt（默认值）** | **默认效果，无端点，显示直线方形边缘。** |
| **round** | **显示圆形端点。** |
| **square** | **显示方形端点。** |

### 11.2.5 线的路径

**1、重置路径**

**在同一画布中，我们添加再多的连线端点也只能有一条路径，如果想要开始新的路径，就需要使用beginPath()方法，当出现beginPath()即表示路径重新开始。**

**2、闭合路径**

**闭合路径就是将我们绘制的开放路径，进行封闭处理，多点的路径闭合后会形成特定的形状。在画布中，使用closePath()方法闭合路径。**

**11.2.6 填充路径**

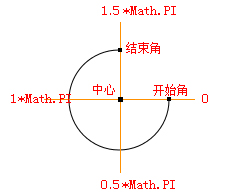
**图形用户界面, 文本, 应用程序, 聊天或短信

描述已自动生成**

### 11.2.7 绘制圆

**在画布中，使用arc()方法可以绘制圆或弧线**

格式：arc(x,y,r,开始角,结束角,方向)



* x和y：x和y表示圆心在x轴和y轴的坐标位置。
* r：表示圆形或弧形的半径，用于确定图形的大小。
* 开始角：表示初始弧点位置。其中弧点使用数值和“Math.PI”（圆周率，可以理解为180度）表示。
* 结束角：结束的弧点位置，初始角的设置方式一致。

方向：分为顺时针和逆时针绘。