

A decorative border with intricate floral and scrollwork patterns in a dark brown color, framing the central text.

李炎恢HTML/ CSS教程

书栈(BookStack.CN)

目 录

致谢

第 1 章 HTML5 概述

第 2 章 基本格式

第 3 章 文本元素

第 4 章 超链接和路径

第 5 章 分组元素

第 6 章 表格元素

第 7 章 文档元素

第 8 章 嵌入元素

第 9 章 音频和视频

第 10 章 表单元素[上]

第 10 章 表单元素[中]

第 10 章 表单元素[下]

第 11 章 全局属性和其他

第 12 章 CSS 入门

第 13 章 CSS 选择器[上]

第 14 章 CSS 颜色与度量单位

第 15 章 CSS 文本样式[上]

第 15 章 CSS 文本样式[下]

第 16 章 CSS 盒模型[上]

第 16 章 CSS 盒模型[下]

第 17 章 CSS 边框与背景[上]

第 17 章 CSS 边框与背景[下]

第 18 章 CSS 表格与列表

第 19 章 CSS 其他样式

第 20 章 CSS3 前缀和 rem

第 21 章 CSS3 文本效果

第 21 章 CSS3 文本效果

第 23 章 CSS3 边框图片效果

第 24 章 CSS3 变形效果[下]

第 25 章 CSS3 过渡效果

第 26 章 CSS3 动画效果

第 27 章 CSS 传统布局[上]

第 27 章 CSS 传统布局[下]

第 28 章 CSS3 多列布局

第 29 章 CSS3 弹性伸缩布局[上]

[第 29 章 CSS3 弹性伸缩布局\[中\]](#)

[第 29 章 CSS3 弹性伸缩布局\[下\]](#)

[第 30 章 使用 Emmet 插件](#)

致谢

当前文档 《李炎恢HTML/CSS教程》 由 进击的皇虫 使用 书栈(BookStack.CN) 进行构建，生成于 2018-07-11。

书栈(BookStack.CN) 仅提供文档编写、整理、归类等功能，以及对文档内容的生成和导出工具。

文档内容由网友们编写和整理，书栈(BookStack.CN) 难以确认文档内容知识点是否错漏。如果您在阅读文档获取知识的时候，发现文档内容有不恰当的地方，请向我们反馈，让我们共同携手，将知识准确、高效且有效地传递给每一个人。

同时，如果您在日常工作、生活和学习中遇到有价值有营养的知识文档，欢迎分享到 书栈(BookStack.CN) ，为知识的传承献上您的一份力量！

如果当前文档生成时间太久，请到 书栈(BookStack.CN) 获取最新的文档，以跟上知识更新换代的步伐。

文档地址：<http://www.bookstack.cn/books/liyanhui-html-css>

书栈官网：<http://www.bookstack.cn>

书栈开源：<https://github.com/TruthHun>

分享，让知识传承更久远！ 感谢知识的创造者，感谢知识的分享者，也感谢每一位阅读到此处的读者，因为我们都将成为知识的传承者。

第 1 章 HTML5 概述

第 1 章 HTML5 概述

学习要点：

1.HTML5 的历史

2.HTML5 的功能

3.HTML5 的特点

4.课程学习问题

主讲教师：李炎恢

HTML5 是继 HTML4.01 和 XHTML1.0 之后的超文本标记语言的最新版本。它是由一群自由思想者组成的团队设计出来，并最终实现多媒体支持、交互性、更加智能的表单，以及更好的语义化标记。

HTML5 并不仅仅是 HTML 规范的最新版本，而是一系列用来制作现代富 Web 内容的相关技术的总称，其中最重要的三项技术分别为：HTML5 核心规范（标签元素）、CSS（层叠样式表第三代）、和 JavaScript。

一. HTML5 的历史

1993 年 HTML 首次以因特网草案的形式发布，然后经历了 2.0、3.2 和 4.0，直到 1999 年的 HTML4.01 版本稳定下来。由于发展缓慢，逐渐的被更加严格的 XHTML 取代。

XHTML 的兴衰史

自从 HTML4.01 版本之后，掌握着 HTML 规范的万维网联盟（W3C）组织没有再发布新的标准，而是围绕着 XHTML1.0 以及之后的 XHTML2.0 展开工作。XHTML 是基于 XML、致力于实现更加严格并且统一的编码规范的 HTML 版本，解决之前 HTML4.01 版本时，由于编码不规范导致浏览器的各种古怪行为。所以，Web 开发者对 XHTML 非常的拥护。XHTML 极大的好处，就是强迫开发者养成良好的编码习惯，放弃 HTML 的凌乱写法，最终降低了浏览器解析页面的难度，方便移植到更多平台。

可是，越是想往好的方面发展，往往可能是带来的却是毁灭性的灾难，世间万物就是如此。XHTML2.0 规范了更严格的错误处理规则，强制要求浏览器拒绝无效的 XHTML2 页面，强制 Web 开发者写出绝对正确规范的代码，同时不得向下兼容，摒弃 HTML 遗留的怪异行为和编码习惯。按理说，取其精华、舍其糟粕应该是好事。但是，这样的话，数亿的页面将无法兼容，Web 开发者的难度又被加大，并且制定这个标准又太过久远，最终被抛弃。

HTML5 的回归

2008 年 W3C 发布了 HTML5 的工作草案，2009 年停止了 XHTML2 计划。又过去大概一年，HTML5 规范进一步解决了诸多非常实际的问题，各大浏览器厂商开始对旗下的产品进行升级，以便支持 HTML5。这样，得益于浏览器的实验反馈，HTML5 规范得到了持续的进步和完善，从而迅速融入到 Web 平台的实质性改进中。

和 XHTML2.0 不同，制定 HTML5 规范的一群人并不想挑出以往 HTML 的各种毛病为其改正，而是尽可能的补全 Web 开发者急需的各种功能。这些功能包括更强大的 CSS3、表单验证、音频视频、本地存储、地理定位、绘画

(Canvas)、Web 通信等等。

二. HTML5 的功能

HTML5 到底涵盖了哪些功能？这些功能到底在主流的浏览器支持情况如何？

1.HTML5 核心：这部分主要由 W3C 官方的规范组成，涉及新的语义元素、新的增强的Web 表单、音频和视频、以及通过 JavaScript 绘图的 Canvas。这部分大多数主流浏览器均得到很好的支持；

2.曾经的 HTML5 标准：这部分主要来自于最初制定的 HTML5 规范，其中大多数功能需要 JavaScript 且支持富 Web 应用开发。比如：本地数据存储、离线应用和消息传递；**3.非 HTML5 标准：**这部分通常指下一代功能，虽然从未进入 HTML5 标准，但人们还是会把它认做 HTML5 的一部分。这些包括最为常见的 CSS3，以及很热门的地理定位。

对于最为常用且实用的部分，基本上主流的浏览器都支持的比较好。而那些特殊需求的部分，则需要根据不同的浏览器检测才能知道是否支持自己想要的功能。

三. HTML5 的特点

在 HTML5 发展的同时，XHTML2.0 也在不断发展，那么到底是哪些特点导致 HTML5 取得最终的胜利呢？

1.向下兼容

对于 XHTML2.0 要求遵循规则，否则不予显示的方式，HTML5 却实行“不破坏 Web”的原则。也就是说，以往已存在的 Web 页面，还可以保持正确的显示。

当然，面对开发者，HTML5 规范要求摒弃过去那些编码坏习惯和废弃的标签元素；而面对浏览器厂商，要求它们兼容 HTML 遗留的一切，以做到向下兼容。

2.用户至上

HTML5 遵循“用户至上”的原则，在出现具体问题时，会把用户放在第一位，其次是开发者，然后是浏览器厂商，最后才是规范制定者。比如，开发者在编码时不严谨导致本该出现警告或错误时，却正常显示了页面。

3.化繁为简

HTML5 对比之前的 XHTML，做了大量的简化工作。具体如下：

- (1).以浏览器的原生能力代替复杂的 JavaScript；
- (2).DOCTYPE 被简化到极致；
- (3).字符集声明被简化；
- (4).简单强大的 API。

4.无插件范式

在 HTML5 出现之前，很多功能只能通过插件或 hack（如绘图 API）来实现，但 HTML5原生提供了这些支持。使用插件有很多问题，具体如下：

- (1).插件安装容易失败；

- (2). 插件被浏览器或软件禁用屏蔽（如 Flash 插件）；
- (3). 插件经常会被爆出漏洞被利用攻击；
- (4). 插件不容易与 HTML 文档其他部分集成（比如整体透明化等）。

5. 访问通用性

这个原则分为三个概念：

- (1). 可访问性：比如更加利于残障人士的阅读方案；
- (2). 媒体中立：比如 HTML5 的媒体播放在不同设备或平台均能正常运行；
- (3). 支持所有语种：比如新元素<ruby>。

6. 引入语义

HTML5 引入了一些用来区分不同含义和内容的标记元素。这种方式极大的提供的编码人员的可读性和代码区域查询的便利性。

7. 引入原生媒体支持

HTML5 的一次大改进救生衣支持在浏览器中直接播放视频和音频文件，以前都需要借助插件才能实现此类功能。

8. 引入可编程内容

HTML5 最大的变化就是引入了需要通过 JavaScript 编程才能完全的各种效果，而这些很多都是 HTML5 原生的。那么现在 HTML5 可以理解为 HTML + CSS + JavaScript 的总称。

四. 课程学习问题

学习 HTML5 需要一些测试用的浏览器、编码用的开发工具、以及建议推荐的学习方法。

1. 浏览器选择

IE9+

Firefox 3.5+

Chrome 3.0+

Safari 3.0+

Opera 10.5+

这里重点要说明一下 IE 浏览器。由于历史和系统绑定原因，还有相当一部分电脑残留IE9 以下版本的浏览器。虽然微软已经开始发表声明逐步不再维护 IE8，但这部分群体还占有一定的份额。所以，是否要迎合这部分用户，取决于个人对市场的判断和成本的考量。

2. 开发工具

本课程我们使用 Sublime Text3 作为 HTML5 课程的编码工具。使用了 Soda Dark 3作为软件界面的主题。

3. 学习方式

本课程原则上是零基础、初学者可学，但如果你已经有 XHTML 课程基础，那么学习起来将非常轻松。当然，虽然我们已录制了 XHTML 基础，在录制 HTML5 课程时，还是将所有学员当作刚接触的初学者来对待。再当然，这里所说的零基础和初学者也是需要一定经验，因为长期的教学工作，我们接触到很多连 QQ 不会用、邮件不会发送、迅雷不会下载的学员。这些学员是初学者之前的、负基础的学员，所以，如果是初学者一般问题不大。

原文: <https://wizardforce1.gitbooks.io/liyanhui-tutorials/content/2.html>

第 2 章 基本格式

第 2 章 基本格式

学习要点：

1. HTML5 文档结构

2. 文档结构解析

3. 元素标签探讨

主讲教师：李炎恢

本章主要先从 HTML5 的文档结构谈起。 这些基础元素确定了 HTML 文档的轮廓以及浏览器的初始环境。几乎所有页面都必须首先键入这些元素。

一. HTML5 文档结构

1. 第一步：打开 Sublime Text 3，打开指定文件夹；
2. 第二步：保存 index.html 文件到磁盘中，.html 是网页后缀；
3. 第三步：开始编写 HTML5 的基本格式。

```
1. <!DOCTYPE html> //文档类型声明 <html lang="zh-cn"> //表示 HTML 文档开始 <head> //包含文档元数据开始 <meta
  charset="utf-8"> //声明字符编码 <title>基本结构</title> //设置文档标题 </head> //包含文档元数据结束 <body> //表示 HTML
  文档内容 <a href="http://www.baidu.com">百度</a> //一个超链接元素（标签） </body> //表示 HTML </html>
  //表示 HTML 文档结束
```

二. 文档结构解析

1. Doctype

文档类型声明 (Document Type Declaration, 也称 Doctype)。它主要告诉浏览器所查看的文件类型。 在以往的 HTML4.01 和 XHTML1.0 中， 它表示具体的 HTML 版本和风格。而如今 HTML5 不需要表示版本和风格了。

```
1. <!DOCTYPE html> //不区分大小写
```

2. html 元素

首先，元素就是标签的意思，html 元素即 html 标签。html 元素是文档开始和结尾的元素。它是一个双标签，头尾呼应，包含内容。这个元素有一个属性和值：lang="zh-cn"，表示文档采用语言为：简体中文。

```
1. <html lang="zh-cn"> //如果是英文则为 en
```

3.head 元素

用来包含元数据内容，元数据包括：<link>、<meta>、<noscript>、<script>、<style>、<title>。这些内容用来浏览器提供信息，比如 link 提供 CSS 信息，script 提供 JavaScript 信息，title 提供页面标题等。

```
1. <head>...</head> //这些信息在页面不可见
```

4.meta 元素

这个元素用来提供关于文档的信息，起始结构有一个属性为：charset="utf8"。表示告诉浏览器页面采用的什么编码，一般来说我们就用 utf8。当然，文件保存的时候也是utf8，而浏览器也设置 utf8 即可正确显示中文。

```
1. <meta charset="utf-8"> //除了设置编码，还有别的
```

5.title 元素

这个元素很简单，顾名思义：设置浏览器左上角的标题。

```
1. <title>基本结构</title>
```

6.body 元素

用来包含文档内容的元素，也就是浏览器可见区域部分。所有的可见内容，都应该在这个元素内部进行添加。

```
1. <body>...</body>
```

7.a 元素

一个超链接，后面会详细探讨。

```
1. <a href="http://www.baidu.com">百度</a>
```

三. 元素标签探讨

HTML 是一种标记语言，刚才的结构我们已经详细探讨过。这里，我们再剖析一下这些“标记”或者叫“标签”，书面上经常称作为“元素”的东西是怎么构成的。

1.元素

元素就是一组告诉浏览器如何处理一些内容的标签。每个元素都有一个关键字，比如<body>、<title>、<meta>都是元素。不同的标签名称代表不同的意义，后面将会涉及到段落标签、文本标签、链接标签、图片标签等。

元素一般分为两种：单标签（空元素）和双标签。单标签一般用于声明或者插入某个元素，比如声明字符编码就用<meta>，插入图片就用；双标签一般用于设置一段区域的内容，将其包含起来，比如段落<p>...</p>。

2.属性和值

元素除了有单双之分，元素的内部还可以设置属性和值。这些属性值用来改变元素某些方面的行为。比如超链接：`<a>`中的 `href` 属性，里面替换网址即可链接到不同的网站。当然一个元素里面可以设置多个属性，甚至自定义属性。

原文：<https://wizardforce1.gitbooks.io/liyanhui-tutorials/content/3.html>

第 3 章 文本元素

第 3 章 文本元素

学习要点：

1. 文本元素总汇

2. 文本元素解析

主讲教师：李炎恢

本章主要探讨 HTML5 中的文本元素，所谓文本元素，就是将一段文本设置成相匹配的结构和含义。

一．文本元素总汇

HTML5 规范指出：使用元素应该完全从元素的语义出发。但是由于历史遗留及用户至上的原则，这种语义会宽松许多。

元素名称	说明
a	生成超链接
br	强制换行
wbr	可安全换行
b	标记一段文字但不强调
strong	表示重要
i	表示外文或科学术语
em	表示强调
code	表示计算机代码
var	表示程序输出
samp	表示变量
kdb	表示用户输入
abbr	表示缩写
cite	表示其他作品的标题
del	表示被删除的文字
s	表示文字已不再确认
dfn	表示术语定义
mark	表示与另一段上下文有关的内容
q	表示引自他处的内容
rp	与 ruby 元素结合使用，标记括号
rt	与 ruby 元素结合使用，标记括号

ruby	表示位于表意文字上方或右方的注音符号
bdo	控制文字的方向
small	表示小号字体内容
sub	表示下标字体
sup	表示上标字体
time	表示时间或日期
u	标记一段文字但不强调
span	通用元素，没有任何语义。一般配合 CSS 修饰。

从上面这张表格中，我们发现文本元素还是非常多的。但实际上，在现实应用中，真正常用的也就是那么几个，绝大部分是针对英文的。

二. 文本元素解析

1. 表示关键字和产品名称

```
1. <b>HTML5</b>
```

解释：元素实际作用就是加粗。从语义上来看，就是标记一段文字，但并不是特别强调或重要性。比如：一段文本中的某些关键字或者产品的名称。

2. 表示重要的文字

```
1. <strong>HTML5</strong>
```

解释：元素实际作用和一样，就是加粗。从语义上来看，就是强调一段重要的文本。

3.
强制换行、<wbr>安全换行

```
1. <br> Thisabc<wbr>dkekslakdj<wbr>fkdlisakd is apple.
```

解释：在任意文本位置键入
都会被换行，而在英文单词过长时使用<wbr>会根据浏览器的宽度适当的裁切换行。

4. <i>表示外文词汇或科技术语

```
1. <i>HTML5</i>
```

解释：<i>元素实际作用就是倾斜。从语义上来看，表示区分周围内容，并不是特别强调或重要性。

5. 加以强调

```
1. <em>HTML5</em>
```

解释：元素实际作用和<i>一样，就是倾斜；从语义上来看，表示对一段文本的强调。

6. <s>表示不准确或校正

```
1. <s>HTML5</s>
```

解释：<s>元素实际作用就是删除线；从语义上来看，表示不准确的删除。

7.表示删除文字

```
1. <del>HTML5</del>
```

解释：元素实际作用和<s>一样，就是删除线；从语义上来看，表示删除相关文字。

8.<u>表示给文字加上下划线

```
1. <u>HTML5</u>
```

解释：<u>元素实际作用就是加一条下划线；从语义上来看，并不强调此段文本。

9.<ins>添加一段文本

```
1. <ins>HTML5</ins>
```

解释：<ins>元素实际作用和<u>一样，加一条下划线；从语义上来看，是添加一段文本，起到强调的作用。

10.<small>添加小号字体

```
1. <small>HTML5</small>
```

解释：<small>元素实际作用就是将文本放小一号。从语义上来看，用于免责声明和澄清声明。

11.<sub><sup>添加上标和下标

```
1. <sub>5</sub>
2.
3. <sup>5</sup>
```

解释：<sub>和<sup>元素实际作用就是数学的上标和下标。语义也是如此。

12.<code>等表示输入和输出

```
1. <code>HTML5</code>
2.
3. <var>HTML5</var>
4.
5. <samp>HTML5</samp>
6.
7. <kdb>HTML5</kdb>
```

解释：<code>表示计算机代码片段；<var>表示编程语言中的变量；<samp>表示程序或计算机的输出；<kdb>表示

用户的输入。由于这属于英文范畴的，必须将 `lang="en"` 英语才能体现效果。

13. <abbr>表示缩写

```
1. <abbr>HTML5</abbr>
```

解释：<abbr>元素没有实际作用；从语义上看，是一段文本的缩写。

14. <dfn>表示定义术语

```
1. <dfn>HTML5</dfn>
```

解释：<dfn>元素就是一般性的倾斜；从语义上看，表示解释一个词或短语的一段文本。

15. <q>引用来自他处的内容

```
1. <q>HTML5</q>
```

解释：<q>元素实际作用就是加了一对双引号。从语义上来看，表示引用来自其他地方的内容。

16. <cite>引用其他作品的标题

```
1. <cite>HTML5</cite>
```

解释：<cite>元素实际作用就是加粗。从语义上来看，表示引用其他作品的标题。

17. <ruby>语言元素

```
1. <ruby> 饕<rp>(</rp><rt>tāo</rt><rp>)</rp> 餮<rp>(</rp><rt>tiè</rt><rp>)</rp> </ruby>
```

解释：<ruby>用来为非西方语言提供支持。<rp><rt>用来帮助读者掌握表意语言文字正确发音。比如：汉语拼音在文字的上方。但目前 Firefox 还不支持此特性。

18. <bdo>设置文字方向

```
1. <bdo dir="rtl">HTML5</bdo>
```

解释：<bdo>必须使用属性 `dir` 才可以设置，一共两个值：`rtl`（从右到左）和 `ltr`（从左到右）。一般默认是 `ltr`。还有一个<bdi>元素也是处理方向的，由于是特殊语言的特殊效果，且主流浏览器大半不支持，忽略。

19. <mark>突出显示文本

```
1. <mark>HTML5</mark>
```

解释：<mark>实际作用就是加上一个黄色的背景，黑色的字；从语义上来看，与上下文相关而突出的文本，用于记号。

20.<time>表示日期和时间

```
1. <time>2015-10-10</time>
```

解释：<time>元素没有实际作用；从语义上来看，表示日期和时间。

21.表示一般性文本

```
1. <span>HTML5</span>
```

解释：元素没有实际作用；语义上就是表示一段文本，我们经常用它来设置 CSS 等操作。

原文：<https://wizardforcel.gitbooks.io/liyanhui-tutorials/content/4.html>

第 4 章 超链接和路径

第 4 章 超链接和路径

学习要点：

- 1. 超链接的属性
- 2. 相对与绝对路径
- 3. 锚点设置

主讲教师：李炎恢

本章主要探讨 HTML5 中文本元素最重要的一个超链接，探讨它自身的属性以及路径问题。

一．超链接的属性

<a>元素属于文本元素，有一些私有属性或者叫局部属性。那么，相对应的还有通用属性或叫做全局属性。这方面的知识，后面会详细探讨。

属性名称	说明
href	指定<a>元素所指资源的 URL
hreflang	指向的链接资源所使用的语言
media	说明所链接资源用于哪种设备
rel	说明文档与所链接资源的关系类型
target	指定用以打开所链接资源的浏览环境
type	说明所链接资源的 MIME 类型（比如 text/html）

在这几个属性当中，只有 href 和 target 一般比较常用，而 href 是必须要用的。其他几个属性，在<a>元素使用较少，将在 CSS 章节再探讨。

1.href 属性

```
1. <a href="http://www.baidu.com">百度</a>
```

解释：href 是必须属性，否则<a>元素就变成空元素了。如果属性值是 http://开头的 URL，意味着点击跳转到指定的外部网站。

2.target 属性

```
1. <a href="http://www.baidu.com" target="_blank">百度</a>
```

解释：target 属性告诉浏览器希望将所链接的资源显示在哪里。

属性名称	说明
_blank	在新窗口或标签页中打开文档
_parent	在父窗框组 (frameset) 中打开文档
_self	在当前窗口打开文档 (默认)
_top	在顶层窗口打开文档

这四种最常用的是 _blank，新建一个窗口。而 _self 是默认，当前窗口打开。_parent 和 _top 是基于框架页面的，分别表示在父窗口打开和在整个窗口打开。而 HTML5 中，框架基本被废弃，只能使用 <iframe> 元素，且以后大量结合 JavaScript 和 PHP 等语言配合，框架用的就很少了。

二. 相对与绝对路径

所谓相对路径，就是相对于链接页面而言的另一个页面的路径。而绝对路径，就是直接从 file:/// 磁盘符开始的完整路径。我们在同一个目录下做上两个页面，其中一个页面链接到另一个页面。

1. 绝对路径

```
1. <a href="file:///D:/备课/HTML5 第一季/code/index2.html">index2</a>
```

解释：首先是 file:/// 开头，然后是磁盘符，然后是一个个的目录层次，找到相应文件。这种方式最致命的问题是，当整个目录转移到另外的盘符或其他电脑时，目录结构一旦出现任何变化，链接当即失效。

2. 相对路径

```
1. <a href="index2.html">index2</a>
```

解释：相对路径的条件是必须文件都在一个磁盘或目录下，如果是在同一个目录下，直接属性值就是被链接的文件名。后缀名。如果在同一个主目录下，有多个子目录层次，那就需要使用目录结构语法。

3. 目录语法

同一个目录：index2.html 或 ./index2.html；

在子目录：xxx/index2.html；

在孙子目录：xxx/xxx/index2.html；

在父目录：../index2.html；

在爷爷目录：../../index2.html；

三. 锚点设置

超链接也可用来将同一个文档中的另一个元素移入视野。通过属性 id 或 name 实现锚点定位。

// 链接

```
1. <a href="#1">第一章</a>
2. <a href="#2">第二章</a>
```

3. `第三章` //锚点
4. ``

原文: <https://wizardforce1.gitbooks.io/liyanhui-tutorials/content/5.html>

第 5 章 分组元素

第 5 章 分组元素

学习要点：

1. 分组元素总汇

2. 分组元素解析

主讲教师：李炎恢

本章主要探讨 HTML5 中分组元素的用法。所谓分组，就是用来组织相关内容的 HTML5 元素，清晰有效的进行归类。

一．分组元素总汇为了页面的排版需要，HTML5 提供了几种语义的分组元素。

元素名称	说明
p	表示段落
div	一个没有任何语义的通用元素，和 span 是对应元素
blockquote	表示引自他出的大段内容
pre	表示其格式应被保留的内容
hr	表示段落级别的主题转换，即水平线
ul,ol	表示无序列表，有序列表
li	用于 ul,ol 元素中的列表项
dl,dt,dd	表示包含一系列术语和定义说明的列表。dt 在 dl 内部表示术语，一般充当标题；dd 在 dl 内部表示定义，一般是内容。
figure	表示图片
figcaption	表示 figure 元素的标题

二．分组元素解析

1.<p>建立段落

```
1. <p> 这是一个段落 </p>
2. <p> 这也是一个段落 </p>
```

解释：<p>元素实际作用就是将内部包含的文本形成一个段落；而段落和段落之间保持一定量的空隙。

2.<div>通用分组

```
1. <div> 这是一个通用分组 </div>
2. <div> 这是又一个通用分组 </div>
```

解释：<div>元素在早期的版本中非常常用，通过<div>这种一般性分组元素进行布局。

而在 HTML5 中，由于语义的缘故，被其他各种文档元素所代替。和<p>段落的区别就是，两段文本的上下空隙是没有的，空隙间隔和
换行一样。

3.<blockquote>引用大段他出内容

1. <blockquote> 这是一个大段引自他出内容 </blockquote>
2. <blockquote> 这是另一个大段引自他出内容 </blockquote>

解释：<blockquote>元素实际作用除了和<p>元素一样，有段落空隙的功能，还包含了首尾缩进的功能。语义上表示，大段的引用他出的内容。

4.<pre>展现格式化内容

1. <pre> #####
- 2.
3. #####
- 4.
5. #####
- 6.
7. #####
- 8.
9. ##### </pre>

解释：<pre>元素实际作用就是编辑器怎么排版的，原封不动的展现出来。当然，这种只适合简单的排版，复杂的排版就无法满足要求了。

5.<hr>添加分隔

1. <hr>

解释：<hr>元素实际作用就是添加一条分割线，意图呈现上下文主题的分割。

6.添加无序列表

1.
2. 张三
3. 李四
4. 王五
5. 马六
6.

解释：元素表示无序列表，而元素则是内部的列表项。

7.添加有序列表

1.
2. 张三

```
3.      <li> 李四 </li>
4.      <li> 王五 </li>
5.      <li> 马六 </li>
6. </ol>
```

解释：元素表示有序列表，而元素则是内部的列表项。元素目前支持三种属性

ol 元素属性

属性名称	说明
start	从第几个序列开始统计：<ol start="2">
reversed	是否倒序排列：<ol reversed>，一半主流浏览器不支持
type	表示列表的编号类型，值分别为：1、a、A、i、I

li 元素属性

属性名称	说明
value	强行设置某个项目的编号。

```
1. <li value="7">王五</li>
```

8.<dl><dt><dd>生成说明列表

```
1. <dl>
2.   <dt> 这是一份文件 </dt>
3.   <dd> 这里是这份文件的详细内容 1 </dd>
4.   <dd> 这里是这份文件的详细内容 2 </dd>
5. </dl>
```

解释：这三个元素是一个整体，但<dt>或<dd>并非都必须出现。

9.<figure><figcaption>使用插图

```
1. <figure>
2.   <figcaption> 这是一张图 </figcaption>
3.   
4. </figure>
```

解释：这两个元素一般用于图片的布局。

原文： <https://wizardforce1.gitbooks.io/liyanhui-tutorials/content/6.html>

第 6 章 表格元素

第 6 章 表格元素

学习要点：

1. 表格元素总汇

2. 构建表格解析

主讲教师：李炎恢

本章主要探讨 HTML5 中表格元素的用法。表格的主要用途是以网格的形式显示二维数据。

一．表格元素总汇

表格的基本构成最少需要三个元素：<table>、<tr>、<td>，其他的一些作为可选辅助存在。

元素名称	说明
table	表示表格
thead	表示标题行
tbody	表示表格主体
tfoot	表示表脚
tr	表示一行单元格
th	表示标题行单元格
td	表示单元格
col	表示一列
colgroup	表示一组列
caption	表示表格标题

二．构建表格解析

1.<table><tr><td>构建基础表格

```
1. <table border="1">
2.   <tr>
3.     <td>张三</td>
4.     <td>男</td>
5.     <td>未婚</td>
6.   </tr>
7.   <tr>
8.     <td>李四</td>
9.     <td>女</td>
10.    <td>已婚</td>
11.  </tr>
```

```
12. </table>
```

解释：<table>元素表示一个表格的声明，<tr>元素表示表格的一行，<td>元素表示一个单元格。默认情况下表格是没有边框的，所以，在<table>元素增加一个 border 属性，设置为 1 即可显示边框。

2.<th>为表格添加标题单元格

```
1. <table border="1" style="width:300px;">
2.   <tr>
3.     <th>姓名</th>
4.     <th>性别</th>
5.     <th>婚姻</th>
6.   </tr>
7.   <tr>
8.     <td>张三</td>
9.     <td>男</td>
10.    <td>未婚</td>
11.  </tr>
12.  <tr>
13.    <td>李四</td>
14.    <td>女</td>
15.    <td>已婚</td>
16.  </tr>
17. </table>
```

解释：<th>元素主要是添加标题行的单元格，实际作用就是将内部文字居中且加粗。这里使用了一个通用属性 style，主要用于 CSS 样式设置，以后会涉及到。<th><td>均属于单元格，包含两个合并属性：colspan、rowspan 等。

3.<thead>添加表头

```
1. <thead>
2.   <tr>
3.     <th>姓名</th>
4.     <th>性别</th>
5.     <th>婚姻</th>
6.   </tr>
7. </thead>
```

解释：<thead>元素就是限制和规范了表格的表头部分。尤其是以后动态生成表头，它的位置是不固定的，使用此元素可以限定在开头位置。

4.<tfoot>添加表脚

```
1. <tfoot>
2.   <tr>
3.     <td colspan="3">统计：共两名</td>
4.   </tr>
5. </tfoot>
```


解释：<tfoot>元素为表格生成表脚，限制在表格的底部。

5.<tbody>添加表主体

```
1. <tbody>
2.   <tr>
3.     <td>张三</td>
4.     <td>男</td>
5.     <td>未婚</td>
6.   </tr>
7.   <tr>
8.     <td>李四</td>
9.     <td>女</td>
10.    <td>已婚</td>
11.  </tr>
12. </tbody>
```

解释：<tbody>元素主要是包含住非表头表脚的主体部分，有助于表格格式的清晰，更加有助于后续 CSS 和 JavaScript 的控制。

6.<caption>添加表格标题

```
1. <caption>这是一个人物表</caption>
```

解释：<caption>元素给表格添加一个标题。

7.<colgroup>设置列

```
1. <colgroup span="2" style="background:red;">
```

解释：<colgroup>元素是为了处理某个列，span 属性定义处理哪些列。1 表示第一列，2 表示前两列。如果要单独设置第二列，那么需要声明两个，先处理第一个，将列点移入第二位，再处理第二个即可。

8.<col>更灵活的设置列

```
1. <colgroup>
2.   <col>
3.   <col style="background:red;" span="1">
4. </colgroup>
```

解释：<col>元素表示单独一列，一个表示一列，控制更加灵活。如果设置了 span 则，控制多列。

原文：<https://wizardforce1.gitbooks.io/liyanhui-tutorials/content/7.html>

第 7 章 文档元素

第 7 章 文档元素

学习要点：

1. 文档元素总汇

2. 文档元素解析

主讲教师：李炎恢

本章主要探讨 HTML5 中文档元素，文档元素的主要作用是划分各个不同的内容，让整个布局清晰明快。让整个布局元素具有语义，进一步替代 div。

一．文档元素总汇文档元素基本没有什么实际作用效果，主要目的是在页面布局时区分各个主题和概念。

元素名称	说明
h1~h6	表示标题
header	表示首部
footer	表示尾部
nav	表示有意集中在一起的导航元素
section	表示重要概念或主题
article	表示一段独立的内容
address	表示文档或 article 的联系信息
aside	表示与周边内容少有牵涉的内容
hgroup	将一组标题组织在一起
details	生成一个区域，用户将其展开可以获得更多细节
summary	用在 details 元素中，表示该元素内容的标题或说明

二．文档元素解析

文档元素的大部分标签，是没有任何效果的，完全是为了配合语义使用，以强调它的结构性。只有在后面的章节学习 CSS，配合使用才能起到布局和样式的效果。

1.<header>表示首部

```
1. <header> 这里部分一般是页面头部，包括：LOGO、标题、导航等内容 </header>
```

解释：<header>元素主要设置页面的标头部分。

2.<footer>表示尾部

1. `<footer>` 这里是页面的尾部，一般包括：版权声明、友情链接等内容 `</footer>`

解释：`<footer>`元素主要设置页面的尾部。

3. `<h1>~<h6>`添加标题

1. `<h1>`标题部分`</h1>`
- 2.
3. `<h4>`小标题部分`</h4>`;

解释：`<h1>~<h6>`实际作用就是加粗并改变字体大小。用于各种标题文档。

4. `<hgroup>`组合标题

1. `<hgroup>`
2. `<h1>`标题部分`</h1>`
3. `<h4>`小标题部分`</h4>`
4. `</hgroup>`

解释：`<hgroup>`元素的作用就是当多个标题出现，干扰到一对或多个本身需要整合的标题，这是使用此元素包含群组。

5. `<section>`文档主题

1. `<section>` 这里一般是存放文档主题内容。 `</section>`

解释：`<section>`元素的作用是定义一个文档的主题内容。

6. `<nav>`添加导航

1. `<nav>`这里存放文档的导航`</nav>`

解释：`<nav>`元素给文档页面添加一个导航。

7. `<article>`添加一个独立成篇的文档

1. `<article>`
2. `<header>`
3. `<nav></nav>`
4. `</header>`
5. `<section></section>`
6. `<footer></footer>`
7. `</article>`

解释：`<article>`元素表示独立成篇的文档，里面可以包含头、尾、主题等一系列内容。在比较大的页面中会使用到，比如一片博文的列表，每篇博文，都有自己的头、尾、主题等内容。和此相似的还有论坛的帖子、用户的评论、新闻等。

8.<aside>生成注释栏

```
1. <aside>这是是一个注释</aside>
```

解释：<aside>元素专门为某一段内容进行注释使用。

9.<address>表示文档或 article 元素的联系信息。

```
1. <address>联系信息</address>
```

解释：如果是在<body>元素下时，表示整个文档的联系信息。如果是在<article>元素下时，表示其下的联系信息。

10.<details>元素生成详情区域、<summary>元素在其内部生成说明标签

解释：由于大多数主流浏览器尚未支持，暂略。

原文：<https://wizardforcel.gitbooks.io/liyanhui-tutorials/content/8.html>

第 8 章 嵌入元素

第 8 章 嵌入元素

学习要点：

1. 嵌入元素总汇

2. 嵌入元素解析

主讲教师：李炎恢

本章主要探讨 HTML5 中嵌入元素，嵌入元素主要功能是把外部的一些资源插入到HTML 中。

一．嵌入元素总汇

这里所列出的元素，并非本节课全部涉及到，比如音频 audio、视频 video、以及动态图像 canvas 和媒体资源 source、track 等会在后面章节或季度讲解。

元素名称	说明
img	嵌入图片
map	定义客户端分区响应图
area	表示一个用户客户端分区响应图的区域
audio	表示一个音频资源
video	表示一个视频资源
iframe	嵌入一个文档
embed	用插件在 HTML 中嵌入内容
canvas	生成一个动态的图形画布
meter	嵌入数值在许可值范围背景中的图形表示
object	在 HTML 文档中嵌入内容
param	表示将通过 object 元素传递给插件的参数
progress	嵌入目标进展或任务完成情况的图形表示
source	表示媒体资源
svg	表示结构化矢量内容
track	表示媒体的附加轨道（例如字幕）

二．嵌入元素解析

1.**嵌入图像**

```
1. 
```

解释: 元素主要是插入一张外部的图片, 那么图片的路径问题和超链接一致。

img 的私有属性

属性名称	说明
src	嵌入图像的 URL
alt	当图片不加载时显示的备用内容
width	定义图片的长度(单位是像素)
height	定义图片的高度(单位是像素)
ismap	创建服务器端分区响应图
usemap	关联<map>元素

```
1. <a href="index.html">
2.  </a>
```

2.<map>**创建分区响应图**

```
1.  <map name="Map">
2. <area shape="rect" coords="31,28,112,100" href="http://www.baidu.com" target="_blank" alt="方形">
3. <area shape="circle" coords="187,142,47" href="http://www.google.com" target="_blank" alt="圆形">
4. <area shape="poly" coords="287,26,240,66,308,112" href="http://www.soso.com" target="_blank" alt="多边形">
5. </map>
```

解释: 通过图片中的热点进行超链接, 这里我们采用 Dreamweaver 进行操作生成的。

3.<iframe>**嵌入另一个文档**

```
1. <a href="index.html" target="in">index</a> | <a href="http://www.baidu.com" target="in">百度</a>
2.
3. &lt;iframe src="http://www.ycku.com" width="600" height="500" name="in"&gt;&lt;/iframe&gt;
```

解释: <iframe>表示内嵌一个 HTML 文档。其下的 src 属性表示初始化时显示的页面, width 和 height 表示内嵌文档的长度和高度, name 表示用于 target 的名称。

4.<embed>**嵌入插件内容**

```
1. <embed src="http://www.tudou.com/v/i4ZZvFwfluI/&bid=05&rpId=50797543&resourceId=50797543_05_05_99/v.swf"
type="application/x-shockwave-flash" width="480" height="400"></embed>
```

解释: <embed>元素是扩展浏览器的功能, 大部分用于添加对插件的支持。这里添加了一个土豆网的 flash 视频。type 类型表示被插入内容的类型, 这里不列出所有, 后面如果遇到其他类型的文件, 再继续探讨; width 和 height 表示宽高; src 表示文件路径。

5.<object>**和<param>元素**

解释: <object>元素和<embed>一样, 只不过 object 是 html4 的标准, 而 embed 是 html5 的标准。而

object 不但可以嵌入 flash，还可以嵌入图片等其他内容。由于图片、音频、视频、插件都有相应标签元素代替，我们这里暂时不对其详细讲解。

6.<progress>**显示进度**

```
1. <progress value="30" max="100"></progress>
```

解释：<progress>元素可以显示一个进度条，一般通过 JS 控制内部的值。IE9 以及更低版本不支持此元素。

7.<meter>**显示范围里的值**

```
1. <meter value="90" min="10" max="100" low="40" high="80" optimum="60"></meter>
```

解释：<meter>元素显示某个范围内的值。其下的属性包含：min 和 max 表示范围边界，low 表示小于它的值过低，high 表示大于它的值过高，optimum 表示最佳值，但不出现效果。IE 浏览器不支持此元素。

原文：<https://wizardforcel.gitbooks.io/liyanhui-tutorials/content/9.html>

第9章 音频和视频

第9章 音频和视频

学习要点：

1. 音频和视频概述

2. video 视频元素

3. audio 音频元素

主讲教师：李炎恢

本章主要探讨 HTML5 中音频和视频元素，通过这两个原生的媒体元素向 HTML 页面中嵌入音频和视频。

一．音频和视频概述

首先，我们要理解两个概念：容器（container）和编解码器（codec）。

1. **视频容器**

音频文件或视频文件，都只是一个容器文件。视频文件包含了音频轨道、视频轨道和其他一些元数据。视频播放时，音频轨道和视频轨道是绑定在一起的。元数据包含了视频的封面、标题、子标题、字幕等相关信息。主流视频容器支持的格式为：.avi、.flv、.mp4、.mkv、.ogg、.webm。

2. **编解码器**

音频和视频编码/解码是一组算法，用来对一段特定音频或视频进行解码和编码，以便音频和视频能够播放。原始的媒体文件体积非常巨大，如果不对其进行编码，那么数据量是非常惊人的，在互联网上传播则要耗费无法忍受的时间；如果不对其进行解码，就无法将编码后的数据重组为原始的媒体数据。主流的音频编解码器：AAC、MPEG-3、Ogg Vorbis，视频编解码器：H.264、VP8、Ogg Theora。

3. **浏览器支持情况**

起初，HTML5 规范本来打算指定编解码器，但实施起来极为困难。部分厂商已有自己的标准，不愿实现标准；而有一些编解码器受专利保护，需要支付昂贵费用。最终放弃了统一规范的要求，导致各个浏览器实现自己的标准。

容器格式	视频编解码	音频编解码	IE9+	Firefox5+	Chrome13+
WebM	VP8	Vorbis	×	√	√
OGG	Theora	Vorbis	×	√	√
MPEG-4	H.264	AAC	√	×	疑问？

除了上面三款浏览器，还有 Safari5+支持 MPEG-4, Opera11 支持 WebM 和 OGG，通过这组数据，只要备好 MP4 和 OGG 格式的即可，但对于新的高清标准 WebM，当然是非常必要的。因为 WebM 不但清晰度高，而且免费，不受限制许可的使用源码和专利权。

目前 Chrome 浏览器，为了推广 WebM 格式的视频。声称未来将放弃 H.264 编码的视频，所以有可能在以后的版本中无法播放 MP4 的视频。当然，目前演示的版本还是支持的。

二. ****video**** 视频元素

以往的视频播放，需要借助 Flash 插件才可以实现。但 Flash 插件的不稳定性经常让浏览器导致崩溃，因此很多浏览器或系统厂商开始抛弃它。而取代它的正是 HTML5 的 video 元素。

属性名称	说明
src	视频资源的 URL
width	视频宽度
height	视频高度
autoplay	设置后，表示立刻开始播放视频
preload	设置后，表示预先载入视频
controls	设置后，表示显示播放控件
loop	设置后，表示反复播放视频
muted	设置后，表示视频处于静音状态
poster	指定视频数据载入时显示的图片

1. ****嵌入一个 WebM视频****

```
1. <video src="test.webm" width="800" height="600"></video>
```

解释: <video>插入一个视频，主流的视频为 .webm, .mp4, .ogg 等。src 表示资源 URL; width 表示宽度; height 表示高度。

2. ****附加一些属性****

```
1. <video src="test.webm" width="800" height="600" autoplay controls loop muted></video>
```

解释: autoplay 表示自动开始播放; controls 表示显示播放控件; loop 表示循环播放; muted 表示静音。

3. ****预加载设置****

```
1. <video src="http://li.cc/test.webm" width="800" height="600" controls
2. preload="none"></video>
```

解释: preload 属性有三个值: none 表示播放器什么都不加载; metadata 表示播放之前只能加载元数据 (宽高、第一帧画面等信息); auto 表示请求浏览器尽快下载整个视频。

4. ****使用预览图****

```
1. <video src="http://li.cc/test.webm" width="800" height="600" controls poster="img.png"></video>
```

解释：poster 属性表示在视频的第一帧，做一张预览图。

5. **兼容多个浏览器**

```
1. <video width="800" height="600" controls poster="img.png">
2.     <source src="test.webm">
3.     <source src="test.mp4">
4.     <source src="test.ogg">
5.     <object>这里引入 flash 播放器实现 IE9 以下，但没必要了</object>
6. </video>
```

解释：通过<source>元素引入多种格式的视频，让更多的浏览器保持兼容。

二. **audio** 音频元素

和 video 元素一样，audio 元素用于嵌入音频内容，而音频元素的属性和视频元素类似。音频的支持度和视频类似，使用<source>元素引入多种格式兼容即可。主流的音频格式有：.mp3，.m4a，.ogg，.wav。

属性名称	说明
src	视频资源的 URL
autoplay	设置后，表示立刻开始播放视频
preload	设置后，表示预先载入视频
controls	设置后，表示显示播放控件

1. **嵌入一个音频**

```
1. <audio src="test.mp3" controls autoplay></audio>
```

解释：和嵌入视频一个道理。

2. **兼容多个浏览器**

```
1. <audio controls>
2.     <source src="test.mp3">
3.     <source src="test.m4a">
4.     <source src="test.wav">
5. </audio>
```

解释：略。

PS：更多设计到 API 的 JavaScript 控制，将在以后的基于 JavaScript 基础后讲解。

原文：<https://wizardforce1.gitbooks.io/liyanhui-tutorials/content/10.html>

第 10 章 表单元素[上]

第 10 章 表单元素[上]

学习要点：

- 1. 表单元素总汇
- 2. 表单元素解析

主讲教师：李炎恢

本章主要探讨 HTML5 中表单元素，表单元素用于获取用户的输入数据。

一．表单元素总汇

在 HTML5 的表单中，提供了各种可供用户输入的表单控件。

元素名称	说明
form	表示 HTML 表单
input	表示用来收集用户输入数据的控件
textarea	表示可以输入多行文本的控件
select	表示用来提供一组固定的选项
option	表示提供提供一个选项
optgroup	表示一组相关的 option 元素
button	表示可用来提交或重置的表单按钮（或一般按钮）
datalist	定义一组提供给用户的建议值
fieldset	表示一组表单元素
legend	表示 fieldset 元素的说明性标签
label	表示表单元素的说明标签
output	表示计算结果

二．表单元素解析

1.<form>**定义表单**

```
1. <form method="post" action="http://www.haosou.com/">
2.
3.     <button>提交</button>;
4. </form>;
```

解释：<form>元素主要是定义本身是一组表单。

--	--

元素名称	说明
action	表示表单提交的页面
method	表示表单的请求方式：有 POST 和 GET 两种，默认 GET
enctype	表示浏览器对发送给服务器的数据所采用的编码格式。有三种：application/x-www-form-urlencoded（默认编码，不能将文件上传到服务器）、multipart/form-data（用于上传文件到服务器）、text/plain（未规范的编码，不建议使用，不同浏览器理解不同）
name	设置表单名称，以便程序调用
target	设置提交时的目标位置：_blank、_parent、_self、_top
autocomplete	设置浏览器记住用户输入的数据，实现自动完成表单。默认为 on 自动完成，如果不想自动完成则设置 off。
novalidate	设置是否执行客户端数据有效性检查，后面课程讲解。

//使用 get 提交数据

```
1. method="get"
```

//丧失自动提示功能

```
1. autocomplete="off"
```

//使用_blank 新建目标

```
1. target="_blank"
```

2.<input>**表示用户输入数据**

```
1. <input name="user">
```

解释：<input>元素默认情况会出现一个单行文本框，有五个属性。

属性名称	说明
autofocus	让光标聚焦在某个 input 元素上，让用户直接输入
disabled	禁用 input 元素
autocomplete	单独设置 input 元素的自动完成功能
form	让表单外的元素和指定的表单挂钩提交
type	input 元素的类型，内容较多，将在下节课展开讲解
name	定义 input 元素的名称，以便接收到相应的值

//聚焦光标

```
1. <input name="user" autofocus>
```

//禁用 input

```
1. <input name="user" disabled>
```

//禁止自动完成

```
1. <input name="user" autocomplete="off">
```

//表单外的 input

```
1. <form method="get" id="register"> ... </form>
2. <input name="email" form="register">
```

3.<label>**添加说明标签**

```
1. <p><label for="user">用户名 : <input id="user" name="user"></label></p>
```

解释: <label>元素可以关联 input，让用户体验更好。且更加容易设置 CSS 样式。

4.<fieldset>**对表单进行编组**

```
1. <fieldset>...</fieldset>
```

解释: <fieldset>元素可以将一些表单元素组织在一起，形成一个整体。

属性名称	说明
name	给分组定义一个名称
form	让表单外的分组与表单挂钩
disabled	禁用分组内的表单

5.<legend>**添加分组说明标签**

```
1. <fieldset>
2.     <legend> 注册表单 </legend>
3. </fieldset>
```

解释: <legend>元素给分组加上一个标题。

6.<button>**添加按钮**

```
1. <button type="submit"></button>
```

解释: <button>元素添加一个按钮，type 属性有如下几个值：

值名称	说明

submit	表示按钮的作用是提交表单，默认
reset	表示按钮的作用是重置表单
button	表示按钮为一般性按钮，没有任何作用

//提交表单

```
1. <button type="submit">提交</button>
```

//重置表单

```
1. <button type="reset">重置</button>
```

//普通按钮

```
1. <button type="button">按钮</button>
```

对于 type 属性为 submit 时，按钮还会提供额外的属性。

属性名称	说明
form	指定按钮关联的表单
formaction	覆盖 form 元素的 action 属性
formenctype	覆盖 form 元素的 enctype 属性
formmethod	覆盖 form 元素的 method 属性
formtarget	覆盖 form 元素的 target 属性
formnovalidate	覆盖 form 元素的 novalidate 属性

//表单外关联提交

```
1. <button type="submit" form="register">提交</button>
```

原文: <https://wizardforce1.gitbooks.io/liyanhui-tutorials/content/11.html>

第 10 章 表单元素[中]

第 10 章 表单元素[中]

学习要点：

1.type 属性总汇

2.type 属性解析

主讲教师：李炎恢

本章主要探讨 HTML5 中表单中 input 元素的 type 属性，根据不同的值来显示不同的输入框。

一. **type** 属性总汇

input 元素可以用来生成一个供用户输入数据的简单文本框。在默认的情况下，什么样的数据均可以输入。而通过不同的属性值，可以限制输入的内容。

属性名称	说明
text	一个单行文本框，默认行为
password	隐藏字符的密码框
search	搜索框，在某些浏览器键入内容会出现叉标记取消
submit、reset、button	生成一个提交按钮、重置按钮、普通按钮
number、range	只能输入数值的框；只能输入在一个数值范围的框
checkbox、radio	复选框，用户勾选框；单选框，只能在几个中选一个
image、color	生成一个图片按钮，颜色代码按钮
email、tel、url	生成一个检测电子邮件、号码、网址的文本框
date、month、time、week、datetime、datetime-local	获取日期和时间
hidden	生成一个隐藏控件
file	生成一个上传控件

二. **input** 元素解析

1.type 为 text 值时**

```
1. <input type="text">
```

解释：当 type 值为 text 时，呈现的就是一个可以输入任意字符的文本框，这也是默认行为。并且，还提供了一些额外的属性。

属性名称	说明
list	指定为文本框提供建议值的 <code>datalist</code> 元素，其值为 <code>datalist</code> 元素的 <code>id</code> 值
maxlength	设置文本框最大字符长度
pattern	用于输入验证的正则表达式
placeholder	输入字符的提示
readonly	文本框处于只读状态
disabled	文本框处于禁用状态
size	设置文本框宽度
value	设置文本框初始值
required	表明用户必须输入一个值，否则无法通过输入验证

//设置文本框长度

```
1. <input type="text" size="50">
```

//设置文本框输入字符长度

```
1. <input type="text" maxlength="10">
```

//设置文本框的初始值

```
1. <input type="text" value="初始值">
```

//设置文本框输入提示

```
1. <input type="text" placeholder="请输入内容">
```

//设置文本提供的建议值

```
1. <input list="footlist">
2. <datalist id="footlist">
3.     <option value="苹果">苹果</option>
4.     <option value="桔子">桔子</option>
5.     <option value="香蕉" label="香蕉">
6.     <option value="梨子">
7. </datalist>
```

//设置文本框内容为只读，可以提交数据

```
1. <input type="text" readonly>
```

//设置文本框内容不可用，不可以提交数据


```
1. <input type="text" disabled>
```

2. type**为** password值时

```
1. <input type="password">
```

解释：当 type 值为 password 时，一般用于密码框的输入，所有的字符都会显示星号。密码框也有一些额外属性。

属性名称	说明
maxlength	设置密码框最大字符长度
pattern	用于输入验证的正则表达式
placeholder	输入密码的提示
readonly	密码框处于只读状态
disabled	文本框处于禁用状态
size	设置密码框宽度
value	设置密码框初始值
required	表明用户必须输入同一个值

这里除了正则和验证需要放在下一节，其余和文本框一致。

3. type**为** search时

```
1. <input type="search">
```

解释：和文本框一致，在除 Firefox 浏览器的其他现代浏览器，会显示一个叉来取消搜索内容。额外属性也和 text 一致。

4. type**为** number、range时

```
1. <input type="number">
2. <input type="range">
```

解释：只限输入数字的文本框，不同浏览器可能显示方式不同。生成一个数值范围文本框，只是样式是拖动式。额外属性如下：

属性名称	说明
list	指定为文本框提供建议值的 datalist 元素，其值为datalist 元素的 id 值
min	设置可接受的最小值
max	设定可接受的最大值
readonly	设置文本框内容只读
required	表明用户必须输入一个值，否则无法通过输入验证
step	指定上下调节值的步长

value	指定初始值
-------	-------

//范围和步长

```
1. <input type="number" step="2" min="10" max="100">
```

5.type**为** date系列时

```
1. <input type="date">
2. <input type="month">
3. <input type="time">
4. <input type="week">
5. <input type="datetime">
6. <input type="datetime-local">
```

解释：实现文本框可以获取日期和时间的值，但支持的浏览器不完整。我们测试 Chrome 和 Opera 支持，其他浏览器尚未支持。所以，在获取日期和时间，目前还是推荐使用 jQuery 等前端库来实现日历功能。额外属性和 number 一致。

6.type**为** color时

```
1. <input type="color">
```

解释：实现文本框获取颜色的功能，最新的现代浏览器测试后 IE 不支持，其余的都能显示一个颜色对话框提供选择。

7.type**为** checkbox、radio时

```
1. 音乐<input type="checkbox"> 体育<input type="checkbox">
2.
3. <input type="radio" name="sex" value="男"> 男 <input type="radio" name="sex" value="女"> 女
```

解释：生成一个获取布尔值的复选框或固定选项的单选框。额外属性如下：

属性名称	说明
checked	设置复选框、单选框是否为勾选状态
required	表示用户必须勾选，否则无法通过验证
value	设置复选框、单选框勾选状态时提交的数据。默认为 on

//默认勾选，默认值为 1

```
1. <input type="checkbox" name="music" checked value="1">
```

8.type**为** submit、reset和** button**时

```
1. <input type="submit">
```

解释：生成一个按钮，三种模式：提交、重置和一般按钮，和<button>元素相同。

值名称	说明
submit	生成一个提交按钮
reset	生成一个重置按钮
button	生成一个普通按钮

如果是 submit 时，还提供了和<button>元素一样的额外属性：formaction、formenctype、formmethod、formtarget 和 formnovalidate。

9.type**为** image时

```
1. <input type="image" src="img.png">
```

解释：生成一个图片按钮，点击图片就实现提交功能，并且传送了分区响应数据。图片按钮也提供了一些额外属性。

属性名称	说明
src	指定要显示图像的 URL
alt	提供图片的文字说明
width	图像的长度
height	图像的高度
提交额外属性	formaction、formenctype、formmethod、formtarget和 formnovalidate。

10.type**为** email、tel、url时

```
1. <input type="email">
2. <input type="tel">
3. <input type="url">
```

解释：email 为电子邮件格式、tel 为电话格式、url 为网址格式。额外属性和 text 一致。但对于这几种类型，浏览器支持是不同的。email 支持比较好，现在浏览器都支持格式验证；tel 基本不支持；url 支持一般，部分浏览器只要检测到 <http://>就能通过。

11.type**为** hidden时

```
1. <input type="hidden">
```

解释：生成一个隐藏控件，一般用于表单提交时关联主键 ID 提交，而这个数据作为隐藏存在。

12.type**为** file时

```
1. <input type="file">
```

解释：生成一个文件上传控件，用于文件的上传。额外提供了一些属性：

属性名称	说明
accept	指定接受的 MIME 类型
required	表明用户必须提供一个值，否则无法通过验证

```
1. accept="image/gif, image/jpeg, image/png"
```

原文: <https://wizardforce1.gitbooks.io/liyanhui-tutorials/content/12.html>

第 10 章 表单元素[下]

第 10 章 表单元素[下]

学习要点：

1. 其他元素

2. 输入验证

主讲教师：李炎恢

本章主要探讨 HTML5 中表单中剩余的其他元素，然后重点了解一下表单的输入验证功能。

一. 其他元素

表单元素还剩下几个元素没有讲解，包括下拉框列表 `select`、多行文本框 `textarea`、和 `output` 计算结果元素。

元素名称	说明
<code>select</code>	生成一个下拉列表进行选择
<code>optgroup</code>	对 <code>select</code> 元素进行编组
<code>option</code>	<code>select</code> 元素中的项目
<code>textarea</code>	生成一个多行文本框
<code>output</code>	表示计算结果

1. **生成下拉列表**

```
1. <select name="fruit">
2.   <option value="1">苹果</option>
3.
4.   <option value="2">橘子</option>
5.
6.   <option value="3">香蕉</option>
7. </select>
```

解释：<select>下拉列表元素至少包含一个<option>子元素，才能形成有效的选项列表。<select>元素包含两个子元素<option>项目元素和<optgroup>分组元素，还包含了一些额外属性。

属性名称	说明
<code>name</code>	设定提交时的名称
<code>disabled</code>	将下拉列表禁用
<code>form</code>	将表单外的下拉列表与某个表单挂钩
<code>size</code>	设置下拉列表的高度

multiple	设置是否可以多选
autofocus	获取焦点
required	选择验证，设置后必须选择才能通过

//设置高度并实现多选

```
1. <select name="fruit" size="30" multiple>
```

//默认首选

```
1. <option value="2" selected>橘子</option>
```

//使用 optgroup 进行分组，label 为分组名称，disabled 可以禁用分组

```
1. <optgroup label="水果类">
2.   <option value="1">苹果</option>
3.   <option value="2" selected>橘子</option>
4.   <option value="3" label="香蕉">香蕉</option>
5. </optgroup>
```

2.**多行文本框**

```
1. <textarea name="content">请留下您的建议！ </textarea>
```

解释：生成一个可变更大小的多行文本框。属性如下：

属性名称	说明
name	设定提交时的名称
form	将表单外的多行文本框与某个表单挂钩
readonly	设置多行文本框只读
disabled	将多行文本框禁用
maxlength	设置最大可输入的字符长度
autofocus	获取焦点
placeholder	设置输入时的提示信息
rows	设置行数
cols	设置列数
wrap	设置是否插入换行符，有 soft 和 hard 两种
required	设置必须输入值，否则无法通过验证

//设置行高和列宽，设置插入换行符

```
1. <textarea name="content" rows="20" cols="30" wrap="hard"></textarea>
```

3. 计算结果

```
1. <form oninput="res.value = num1.valueAsNumber * num2.valueAsNumber">
2.     <input type="number" id="num1"> x <input type="number" id="num2">
3.     <output for="num1 num2" name="res">
4. </form>
```

解释: output 就是计算两个文本框之间的值, 其实就是内嵌了 JavaScript 功能。

二. 输入验证

HTML5 对表单提供了输入验证检查方式, 但这种验证还是比较简陋的, 并且不同的浏览器支持的成熟度还不同。在大部分情况下, 可能还是要借助 jQuery 等前端库来实现丰富的验证功能和显示效果。

//必须填写一个值

```
1. <input type="text" required>
```

//限定在某一个范围内

```
1. <input type="number" min="10" max="100">
```

//使用正则表达式

```
1. <input type="text" placeholder="请输入区号+座机" required pattern="^[\\d]{2,4}\\-\\[\\d]{6,8}$">
```

//禁止表单验证

```
1. <form action="http://li.cc" novalidate>
```

原文: <https://wizardforce1.gitbooks.io/liyanhui-tutorials/content/13.html>

第 11 章 全局属性和其他

第 11 章 全局属性和其他

学习要点：

- 1. 实体
- 2. 元数据
- 3. 全局属性

主讲教师：李炎恢

本章主要探讨 HTML5 中的 HTML 实体、以及 HTML 核心构成的元数据，最后了解一下 HTML 中的全局属性。

一．实体

HTML 实体就是将有特殊意义的字符通过实体代码显示出来。



二．元数据

<meta>元素可以定义文档中的各种元数据，而且一个 HTML 页面可以包含多个<meta>元素。

1. **指定名/值元数据对**

```
1. <meta name="author" content="bnbbs">
2. <meta name="description" content="这是一个 HTML5 页面">
3.
4. &lt;meta name="keywords" content="html5,css3,响应式"&gt;
5.
6. &lt;meta name="generator" content="sublime text 3"&gt;
```

元数据名称	说明
author	当前页面的作者
description	当前页面的描述
keywords	当前页面的关键字
generator	当前页面的编码工具

2. 声明字符编码

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


3. 模拟 HTTP 标头字段

//5 秒跳转到指定 URL

```
1. <meta http-equiv="refresh" content="5;http://li.cc">
```

//另一种声明字符编码

```
1. <meta http-equiv="content-type" content="text/html charset=utf-8">
```

属性值	说明
refresh	重新载入当前页面，或指定一个 URL。单位秒。
content-type	另一种声明字符编码的方式

三. 全局属性

在此之前，我们涉及到的元素都讲解了它的局部数据，当然也知道一些全局属性，比如id。全局属性是所有元素共有的行为，HTML5 还提供了一些其他的全局属性。

1.id**属性**

```
1. <p id="abc">段落</p>
```

解释：id 属性给元素分配一个唯一标识符。这种标识符通常用来给 CSS 和 JavaScript 调用选择元素。一个页面只能出现一次同一个 id 名称。

2.class**属性**

```

1. <p class="abc">段落</p>
2.
3. &lt;p class="abc"&gt;段落&lt;/p&gt;
4.
5. &lt;p class="abc"&gt;段落&lt;/p&gt;
```

解释：class 属性用来给元素归类。通过是文档中某一个或多个元素同时设置 CSS 样式。

3.accesskey**属性**

```
1. <input type="text" name="user" accesskey="n">
```

解释：快捷键操作，windows 下 alt+指定键，前提是浏览器 alt 并不冲突。

4.contenteditable**属性**

```
1. <p contenteditable="true">我可以修改吗</p>
```

解释：让文本处于可编辑状态，设置 true 则可以编辑，false 则不可编辑。或者直接设置属性。

5.dir**属性**

```
1. <p dir="rtl">文字到右边了</p>
```

解释：让文本从左到右（ltr），还是从右到左（rtl）。

6.hidden**属性**

```
1. <p hidden>文字到右边了</p>
```

解释：隐藏元素。

7.lang**属性**

```
1. <p lang="en">HTML5</p>
```

解释：可以局部设置语言。

8.title**属性**

```
1. <p title="HTML5 教程">HTML5</p>
```

解释：对元素的内容进行额外的提示。

9.tabindex**属性**

```
1. <input type="text" name="user" tabindex="2">  
2.  
3. &lt;input type="text" name="user" tabindex="1"&gt;
```

解释：表单中按下 tab 键，实现获取焦点的顺序。如果是-1，则不会被选中。

10.style**属性**

```
1. <p style="color:red;">CSS 样式</p>
```

解释：设置元素行内 CSS 样式。

原文：<https://wizardforcel.gitbooks.io/liyanhui-tutorials/content/14.html>

第 12 章 CSS 入门

第 12 章 CSS 入门

学习要点：

1. 使用 CSS

2. 三种方式

3. 层叠和继承

主讲教师：李炎恢

本章主要探讨 HTML5 中 CSS（层叠样式表），它是用来对 HTML 文档外观的表现形式进行排版和格式化。

一．使用**CSS**

CSS 样式由一条或多条以分号隔开的样式声明组成。每条声明的样式包含着一个 CSS 属性和属性值。

```
1. <p style="color:red;font-size:50px;">这是一段文本</p>
```

解释：style 是行内样式属性。color 是颜色属性，red 是颜色属性值；font-size是字体大小属性，50px 是字体大小属性值。

二．三种方式

创建 CSS 样式表有三种方式：1. 元素内嵌样式；2. 文档内嵌样式；3. 外部引入样式。

1.**元素内嵌样式**

```
1. <p style="color:red;font-size:50px;">这是一段文本</p>
```

解释：即在当前元素使用 style 属性的声明方式。

2.**文档内嵌样式**

```
1. <style type="text/css"> p {    color: blue; font-size: 40px;
2. }
3. &lt;/style&gt;
4.
5. &lt;p&gt;这是一段文本&lt;/p&gt;
```

解释：在<head>元素之间创建<style>元素，通过选择器的方式调用指定的元素并设置相关 CSS。

3.**外部引用样式**

```
1. <link rel="stylesheet" type="text/css" href="style.css">
```

```
//style.css
```

```
1. @charset "utf-8";
2. p {    color: green;    font-size: 30px;
3. }
```

解释：很多时候，大量的 HTML 页面使用了同一个组 CSS。那么就可以将这些 CSS 样式保存在一个单独的.css 文件中，然后通过<link>元素去引入它即可。`@charset "utf-8"` 表明设置 CSS 的字符编码，如果不写默认就是 utf-8。如果有多个.css 文件，可以使用 `@import` 导入方式引入.css 文件。只不过，性能不如多个<link>链接。

三．层叠和继承

所谓的样式表层叠：指的是同一个元素通过不同方式设置样式表产生的样式重叠。样式表继承：指的是某一个被嵌套的元素得到它父元素样式。还有一种样式叫浏览器样式，是这个元素在这个浏览器运行时默认附加的样式。

1. **浏览器样式**

```
1. <b>这个元素隐含加粗样式</b>
2. <span style="font-weight:bold;">这个元素通过 style 加粗</span>
```

解释：元素就是具有加粗的隐含样式，而元素没有任何隐含样式，通过 style 属性设置样式。

2. **样式表层叠**

样式表层叠通过五种方式进行，如果样式相同，那么比如会产生冲突替换。这时，它的优先级顺序就显的比较重要。以下优先级从低到高：

- (1). 浏览器样式（元素自身携带的样式）；
- (2). 外部引入样式（使用<link>引入的样式）；
- (3). 文档内嵌样式（使用<style>元素设置）；
- (4). 元素内嵌样式（使用 style 属性设置）。

```
//元素内嵌
```

```
1. <p style="color:red;font-size:30px;">我将被三种方式叠加样式</p>
```

```
//文档内嵌
```

```
1. <style type="text/css">
2.
3. p {    color:blue; font-weight: bold;
4. }
5. &lt;/style&gt;
```

//外部引入

```
1. @charset "utf-8";
2. p {    color: green;    font-style: italic;
3. }
```

如果某一个样式被优先级高的给替换掉了，却又想执行这个样式方案，可以将这个方案标记成重要样式（important）。

//强行设置最高优先级

```
1. color: green !important;
```

3. **样式继承**

如果某一个元素并没有设置父元素相关的样式，那么就会使用继承机制将父元素的样式集成下来。

//元素继承了<p>元素的样式

```
1. <p style="color:red;">这是<b>HTML5</b></p>
```

样式继承只适用于元素的外观（文字、颜色、字体等），而元素在页面上的布局样式则不会被继承。如果继承这种样式，就必须使用强制继承：inherit。

//强制继承布局样式

```
1. <p>这是<b>HTML5</b></p>
2.
3. <style type="text/css">
4.
5. p {    border: 1px solid red;
6. } b {    border : inherit;
7. }
8. &lt;/style>
```

原文: <https://wizardforcel.gitbooks.io/liyanhui-tutorials/content/15.html>

第 13 章 CSS 选择器[上]

第 13 章 CSS 选择器[上]

学习要点：

- 1. 选择器总汇
- 2. 基本选择器
- 3. 复合选择器
- 4. 伪元素选择器

主讲教师：李炎恢

本章主要探讨 HTML5 中 CSS 选择器，通过选择器定位到想要设置样式的元素。目前 CSS 选择器的版本已经升级至第三代，即 CSS3 选择器。CSS3 选择器提供了更多、更丰富的选择器方式，主要分为三大类。

一．选择器总汇

本节课主要涉及到三种选择器：基本选择器、复合选择器和伪元素选择器，具体如下：

选择器	名称	说明	CSS 版本
*	通用选择器	选择所有元素	2
<type>	元素选择器	选择指定类型的元素	1
#<id>	id 选择器	选择指定 id 属性的元素	1
.<class>	class 选择器	选择指定 class 属性的元素	1
[attr]系列	属性选择器	选择指定 attr 属性的元素	2 ~ 3
s1,s2,s3...	分组选择器	选择多个选择器的元素	1
s1 s2	后代选择器	选择指定选择器的后代元素	1
s1 > s2	子选择器	选择指定选择器的子元素	2
s1 + s2	相邻兄弟选择器	选择指定选择器相邻的元素	2
s1 ~ s2	普通兄弟选择器	选择指定选择器后面的元素	3
::first-line	伪元素选择器	选择块级元素文本的首行	1
::first-letter	伪元素选择器	选择块级元素文本的首字母	1
::before	伪元素选择器	选择元素之前插入内容	2
::after	伪元素选择器	选择元素之后插入内容	2

二．基本选择器

使用简单且频率高的一些选择器归类为基本选择器。

1. **通用选择器**

```
1. * { border: 1px solid red;
2. }
```

解释：“*”号选择器是通用选择器，功能是匹配所有 html 元素的选择器包括<html>和<body>标签。可以使用如下元素标记测试效果：

```
1. <p>段落</p>
2.
3. <b>加粗</b>;
4.
5. <span>无</span>;
```

通用选择器会将所有元素匹配并配置样式，这是把双刃剑，好处就是非常方便，坏处就是将不必要的元素也配置了。目前为止，还不存在所有元素都必须配置的样式，所以，一般来说，不常用。

2. **元素选择器**

```
1. p { color: red;
2. }
```

```
1. <p>段落</p>
```

解释：直接使用元素名称作为选择器名即可。

3. ID**选择器**

```
1. #abc { font-size: 20px;
2. }
```

```
1. <p id="abc">段落</p>
```

解释：通过对元素设置全局属性 id，然后使用#id 值来选择页面唯一元素。

4. **类选择器**

```
1. .abc { border: 1px solid red;
2. }
```

```
1. <b class="abc">加粗</b>
2.
3. <span class="abc">无</span>;
```

解释：通过对元素设置全局属性 class，然后使用.class 值选择页面一个或多个元素。

```
1. b.abc { border: 1px solid red;
2. }
```

解释：也可以使用“元素.class 值”的形式，限定某种类型的元素。

```
1. <span class="abc edf">无</span>
```

解释：类选择器还可以调用多个样式，中间用空格隔开进行层叠。

5. **属性选择器**

//所需 CSS2 版本

```
1. [href] { color: orange;
2. }
```

解释：属性选择器，直接通过两个中括号里面包含属性名即可。当然，还有更多扩展的属性选择器。

//所需 CSS2 版本

```
1. [type="password"] { border: 1px solid red;
2. }
```

解释：匹配属性值的属性选择器。

//所需版本 CSS3

```
1. [href^="http"] { color: orange;
2. }
```

解释：属性值开头匹配的属性选择器。

//所需版本 CSS3

```
1. [href$=".com"] { color: orange;
2. }
```

解释：属性值结尾匹配的属性选择器。

//所需版本 CSS3

```
1. [href*="baidu"] { color: orange;
2. }
```

解释：属性值包含指定字符的属性选择器。

//所需版本 CSS2


```

1. [class~="edf"] { font-size: 50px;
2. }
```

解释：属性值具有多个值时，匹配其中一个值的属性选择器。

//所需版本 CSS2

```

1. [lang|="en"] { color: red;
2. }
```

解释：属性值具有多个值且使用“-”号连接符分割的其中一个值的属性选择器。比如

```

1. <i lang="en-us">HTML5</i>
```

二. 复合选择器

将不同的选择器进行组合形成新的特定匹配，我们称为复合选择器。

1. **分组选择器**

```

1. p,b,i,span { color: red;
2. }
```

解释：将多个选择器通过逗号分割，同时设置一组样式。当然，不但可以分组元素选择器，还可以使用 ID 选择器、类选择器、属性选择器混合使用。

2. **后代选择器**

```

1. p b { color: red;
2. }
```

解释：选择<p>元素内部所有元素。不在乎的层次深度。当然，后代选择器也可以混合使用 ID 选择器、类选择器、属性选择器。

3. **子选择器**

```

1. ul > li { border: 1px solid red;
2. }
```

```

1. <ul>
2.     <li> 我是儿子 <ol>
3.         <li> 我是孙子 </li>
4.         <li> 我是孙子 </li>
5.     </ol>
6. </li>
7. <li> 我是儿子 </li>
8. <li> 我是儿子 </li>
9. </ul>
```

解释：子选择器类似与后代选择器，而最大的区别就是子选择器只能选择父元素向下一级的元素，不可以再往下选择。

4. **相邻兄弟选择器**

```
1. p + b { color: red;
2. }
```

解释：相邻兄弟选择器匹配和第一个元素相邻的第二个元素。

5. **普通兄弟选择器**

```
1. p ~ b { color: red;
2. }
```

解释：普通兄弟选择器匹配和第一个元素后面的所有元素。

三. 伪元素选择器

伪选择器分为两种第一种是下节伪类选择器，还有一种就是伪元素选择器。这两种选择器特性上比较容易混淆，在 CSS3 中为了区分，伪元素前置两个冒号 (::)，伪类前置一个冒号 (:)。

1. ::first-line**块级首行**

```
1. ::first-line { color: red;
2. }
```

解释：块级元素比如<p>、<div>等的首行文本被选定。如果想限定某种元素，可以加上前置 p::first-line。

2. ::first-letter**块级首字母**

```
1. ::first-letter { color: red;
2. }
```

解释：块级元素的首行字母。

3. ::before**文本前插入**

```
1. a::before { content: '点击-';
2. }
```

解释：在文本前插入内容。

4. ::after**文本后插入**

```
1. a::after { content: '-请进';
2. }
```

解释：在文本后插入内容。

原文：<https://wizardforce1.gitbooks.io/liyanhui-tutorials/content/16.html>

第 14 章 CSS 颜色与度量单位

第 14 章 CSS 颜色与度量单位

学习要点：

1. 颜色表方案

2. 度量单位

主讲教师：李炎恢

本章主要探讨 HTML5 中 CSS 颜色和度量单位等问题，包括颜色的选取方式、相对长度和绝对长度等。

一. 颜色表方案

颜色的表现形式主要有三种方式：颜色名称、十六进制代码和十进制代码。

```
1. p { color: red;
2. }
```

解释：这是将一个段落内的文字设置为红色，采用的是英文颜色名称。问题是，其他各种颜色我们将如何设置？

在古老的 HTML4 时，颜色名称只有 16 种。

颜色名称	十六进制代码	十进制代码	含义
black	#000000	0,0,0	黑色
silver	#c0c0c0	192,192,192	银灰色
gray	#808080	128,128,128	灰色
white	#ffffff	255,255,255	白色
maroon	#800000	128,0,0	栗色
red	#ff0000	255,0,0	红色
purple	#800080	128,0,128	紫色
fuchsia	#ff00ff	255,0,255	紫红
green	#008000	0,128,0	绿色
lime	#00ff00	0,255,0	闪光绿
olive	#808000	128,128,0	橄榄色
yellow	#ffff00	255,255,0	黄色
navy	#000080	0,0,128	海军蓝
blue	#0000ff	0,0,255	蓝色
teal	#008080	0,128,128	水鸭色
aqua	#00ffff	0,255,255	浅绿色

当然，目前颜色名称远远不止这些，可以搜索更多的 [HTML 颜色表](#)或 [CSS 颜色表](#)查阅。这里提供一些页面如下：

<http://xh.5156edu.com/page/z1015m9220j18754.html>

<http://finle.me/colors.html>

http://www.w3school.com.cn/tags/html_ref_colornames.asp

在上面的表格中，我们也罗列出对应的十六进制和十进制颜色表示方法。使用方法如下：

//红色的十六进制方案

```
1. p { color: #ff0000;
2. }
```

十进制表示方法就比较多样化，有四种方案：

函数	说明	示例
rgb(r,g,b)	用 RGB 模型表示颜色	rgb(0,128,128)
rgba(r,g,b,a)	同上，a 表示透明度 0~1 之间	rgba(0,128,128,0.5)
hsl(h,s,l)	用 HSL 模型（色相、饱和度和透明度）来表示颜色	hsl(120,100%,30%)
hsla(h,s,l,a)	同上，a 表示透明度 0~1 之间	hsla(120,100%,30%,0.5)

```
1. p { color: rgb(112, 128, 114); color: rgba(0, 128, 128, 0.5); color: hsl(120, 100%, 30%); color: hsla(120,
100%, 30%, 0.5);
2. }
```

目前又有一个疑问，这些值从哪里获取。除了颜色表之外，想要微调自己的颜色值。我们可以使用 photoshop 等平面设计软件的调色板获取相应的值。



二. 度量单位

在 CSS 长度设置中，我们经常需要使用到度量单位，即以什么样的单位设计我们的字体或边框长度。而在 CSS 中长度单位又分为绝对长度和相对长度。

绝对长度指的是现实世界的度量单位，CSS 支持五种绝对长度单位。	
绝对长度单位	
单位标识符	说明
in	英寸
cm	厘米
mm	毫米
pt	磅
pc	pica

相对长度指的是依托其他类型的单位，也是五种。	
相对长度单位	
单位标识符	说明
em	与元素字号挂钩
ex	与元素字体的“x 高度”挂钩
rem	与根元素的字号挂钩
px	像素，与分辨率挂钩
%	相对另一值的百分比

下面我们使用一些常用的单位作为演示，而不做演示的基本用不到了。

//em 相对单位

```
1. p { margin: 0; padding: 0; background: silver; font-size: 15px; height: 2em;
2. }
```

解释：em 是相对单位，与字号大小挂钩，会根据字体大小改变自己的大小，灵活性很高。

//px 相对单位，绝对特性

```
1. p { margin: 0; padding: 0; background: silver; font-size: 15px; height: 55px;
2. }
```

解释：虽然 px 也是相对单位，但由于和分辨率挂钩，导致他其实就变成一个绝对单位了，自然灵活性没有 em 高，但是使用难度较低，且大量的开发者习惯性使用它。

//%百分比

```
1. p { margin: 0; padding: 0; background: silver; font-size: 200%; width: 50%;
2. }
```

解释：长度比较好理解，就是挂钩它所在区块的宽度。而 font-size 则是继承到的原始大小的百分比。

原文：<https://wizardforce1.gitbooks.io/liyanhui-tutorials/content/17.html>



第 15 章 CSS 文本样式[上]

第 15 章 CSS 文本样式[上]

学习要点：

- 1. 字体总汇
- 2. 字体设置
- 3. Web 字体

主讲教师：李炎恢

本章主要探讨 HTML5 中 CSS 文本样式，通过文本样式的设置，更改字体的大小、样式以及文本的方位。

一．字体总汇

本节课，我们重点了解一下 CSS 文本样式中字体的一些设置方法，样式表如下：

属性名	说明	CSS 版本
font-size	设置字体的大小	1
font-variant	设置英文字体是否转换为小型大写	1
font-style	设置字体是否倾斜	1
font-weight	设置字体是否加粗	1
font-family	设置 font 字体	1
font	设置字体样式复合写法	1 ~ 2
@font-face	设置 Web 字体	3

二．字体设置

我们可以通过 CSS 文本样式来修改字体的大小、样式以及形态。

1. font-size

```
1. p { font-size: 50px;
2. }
```

解释：设置文本的大小。属性值如下表：



//先设置父元素字体大小

```
1. body { font-size: 50px;
2. }
```

//再设置相对小一些

```
1. p { font-size: smaller;
2. }
```

2.font-variant

```
1. p { font-variant: small-caps;
2. }
```

解释：设置字体是否以小型大写字母显示。

值	说明
normal	表示如果以小型大写状态，让它恢复小写状态。
small-caps	让小写字母以小型大写字母显示。

//先让父元素设置小型大写

```
1. body { font-variant: small-caps;
2. }
```

//让子元素设置恢复小写

```
1. p { font-size: 50px; font-variant: normal;
2. }
```

3.font-style

```
1. p { font-style: italic;
2. }
```

解释：设置字体是否倾斜。

值	说明
normal	表示让倾斜状态恢复到正常状态。
italic	表示使用斜体。
oblique	表示让文字倾斜。区别在没有斜体时使用。

```
1. p { font-weight: bold;
2. }
```


解释：设置字体是否加粗。

值	说明
normal	表示让加粗的字体恢复正常。
bold	粗体
bolder	更粗的字体
lighter	轻细的字体
100 ~ 900 之间的数字	600 及之后是加粗，之前不加粗

在目前计算机和浏览器显示中，只有 bold 加粗，其他更粗更细，目前体现不出来。

5. font-family

```
1. p { font-family: 微软雅黑;
2. }
```

解释：使用指定字体名称。这里使用的字体是浏览者系统的字体。有时为了兼容很多浏览者系统的字体，可以做几个后备字体。

//备用字体

```
1. p { font-family: 楷体, 微软雅黑, 宋体;
2. }
```

6. font

```
1. p { font: 50px 楷体;
2. }
```

解释：字体设置简写组合方式。格式如下：[是否倾斜|是否加粗|是否转小型大写] 字体大小 字体名称；

三. **Web** 字体

虽说可以通过备用字体来解决用户端字体缺失问题，但终究用户体验不好，且不一定备用字体所有用户都安装了。所以，现在 CSS 提供了 web 字体，也就是服务器端字体。

//服务器提供字体

```
1. @font-face { font-family: abc; src: url('BrushScriptStd.otf');
2. } p { font-size: 50px; font-family: abc;
3. }
```

英文字体文件比较小，而中文则很大。所以，中文如果想用特殊字体可以使用图片。大面积使用特殊中文字体，就不太建议了。

原文：<https://wizardforce1.gitbooks.io/liyanhui-tutorials/content/18.html>

第 15 章 CSS 文本样式[下]

第 15 章 CSS 文本样式[下]

学习要点：

- 1. 文本总汇
- 2. 文本样式
- 3. 文本控制

主讲教师：李炎恢

本章主要探讨 HTML5 中 CSS 文本样式，通过文本样式的设置，更改字体的大小、样式以及文本的方位。

一．文本总汇

本节课，我们重点了解一下 CSS 文本样式中文本内容的一些设置方法，样式表如下：

属性名	说明	CSS 版本
text-decoration	装饰文本出现各种划线。	1
text-transform	将英文文本转换大小写。	1
text-shadow	给文本添加阴影	3
text-align	设置文本对齐方式	1, 3
white-space	排版中的空白处理方式	1
letter-spacing	设置字母之间的间距	1
word-spacing	设置单词之间的间距	1
line-height	设置行高	1
word-wrap	控制段词	3
text-indent	设置文本首行的缩进	1

二．文本样式

CSS 文本样式有三种：文本装饰、英文大小写转换和文本阴影。

1.text-decoration

```
1. p { text-decoration: underline;
2. }
```

解释：设置文本出现下划线。属性值如下表：

值	说明
---	----

none	让本身有划线装饰的文本取消掉
underline	让文本的底部出现一条下划线
overline	让文本的头部出现一条上划线
line-through	让文本的中部出现一条删除划线
blink	让文本进行闪烁，基本不支持了

//让本来有下划线的超链接取消

```
1. a { text-decoration: none;
2. }
```

2.text-transform

```
1. p { text-transform: uppercase;
2. }
```

解释：设置英文文本转换为大小写。

值	说明
none	将已被转换大小写的值恢复到默认状态
capitalize	将英文单词首字母大写
uppercase	将英文转换为大写字母
lowercase	将英文转换为小写字母

3.text-shadow

```
1. p { text-shadow: 5px 5px 3px black;
2. }
```

解释：给文本添加阴影。其中四个值，第一个值：水平偏移；第二个值：垂直偏移；第三个值：阴影模糊度（可选）；第四个值：阴影颜色（可选）。

三. 文本控制

CSS 文本样式中还有一组对文本进行访问、形态进行控制的样式。

1.text-align

```
1. p { text-align: center;
2. }
```

解释：指定文本的对齐方式。

值	说明
left	靠左对齐，默认

right	靠右对齐
center	居中对齐
justify	内容两端对齐
start	让文本处于结束的边界
end	让文本处于结束的边界

start 和 end 属于 CSS3 新增的功能，但目前 IE 和 Opera 尚未支持。

2.white-space

```
1. p { white-space: nowrap;
2. }
```

解释：处理空白排版方式。

值	说明
normal	默认值，空白符被压缩，文本自动换行
nowrap	空白符被压缩，文本不换行
pre	空白符被保留，遇到换行符则换行
pre-line	空白符被压缩，文本会在排满或遇换行符换行
pre-wrap	空白符被保留，文本会在排满或遇换行符换行

3.letter-spacing

```
1. p { letter-spacing: 4px;
2. }
```

解释：设置文本之间的间距。

值	说明
normal	设置默认间距
长度值	比如：“数字”+“px”

4.word-spacing

```
1. p { word-spacing: 14px;
2. }
```

解释：设置英文单子之间的间距。

值	说明
normal	设置默认间距
长度值	比如：“数字”+“px”

5. line-height

```
1. p { line-height: 200%;  
2. }
```

解释：设置段落行高。

值	说明
normal	设置默认间距
长度值	比如：“数字”+“px”
数值	比如：1, 2, 3
%	比如：200%

6. word-wrap

```
1. p { word-wrap: break-word;  
2. }
```

解释：让过长的英文单词断开。

值	说明
normal	单词不断开
break-word	断开单词

7. text-indent

```
1. p { text-indent: 20px;  
2. }
```

解释：设置文本首行的缩进。

值	说明
normal	设置默认间距
长度值	比如：“数字”+“px”

原文：<https://wizardforce1.gitbooks.io/liyanhui-tutorials/content/19.html>

第 16 章 CSS 盒模型[上]

第 16 章 CSS 盒模型[上]

学习要点：

- 1.元素尺寸
- 2.元素内边距
- 3.元素外边距
- 4.处理溢出

主讲教师：李炎恢

本章主要探讨 HTML5 中 CSS 盒模型，学习怎样了解元素的外观配置以及文档的整体布局。

一．元素尺寸

CSS 盒模型中最基础的就是设置一个元素的尺寸大小。有三组样式来配置一个元素的尺寸大小，样式表如下：

属性	值	说明	CSS 版本
width	auto、长度值或百分比	设置元素的宽度	1
height	auto、长度值或百分比	设置元素的高度	1
min-width	auto、长度值或百分比	设置元素最小宽度	2
min-height	auto、长度值或百分比	设置元素最小高度	2
max-width	auto、长度值或百分比	设置元素最大宽度	2
max-height	auto、长度值或百分比	设置元素最大高度	2

//设置元素尺寸

```
1. div { width: 200px; height: 200px;
2. }
```

解释：设置元素的固定尺寸。

//限制元素尺寸

```
1. div { min-width: 100px; min-height: 100px; max-width: 300px; max-height: 300px;
2. }
```

解释：这一组主要是应对可能动态产生元素尺寸变大变小的问题，从而限制它最大和最小的值。

//auto 自适应

```

1. div { width: auto; height: auto;
2. }
```

解释：auto 是默认值，width 在 auto 下是 100% 的值；height 在 auto 下是自适应。

//百分比方式

```

1. #a { background: silver; width: 200px; height: 200px;
2. } #b { background: gray; width: 80%; height: 80%;
3. }
```

```

1. <div id="a">
2.     <div id="b"> 我是 html5 </div>
3. </div>
```

解释：百分比就是相对于父元素长度来衡定的。

二. 元素内边距

CSS 盒模型中可以设置元素内部边缘填充空白的大小，我们称为内边距。样式表如下：

属性	值	说明	CSS 版本
padding-top	长度值或百分比	设置顶部内边距	1
padding-bottom	长度值或百分比	设置底部内边距	1
padding-left	长度值或百分比	设置左边内边距	1
padding-right	长度值或百分比	设置右边内边距	1
padding	1 ~ 4 个长度值或百分比	简写属性	1

//设置四个内边距

```

1. div { padding-top: 10px; padding-bottom: 10px; padding-left: 10px; padding-right: 10px;
2. }
```

//简写形式，分别为上 10px、右 10px、下 10px、左 10px

```

1. div { padding: 10px 10px 10px 10px;
2. }
```

//简写形式，分别为上 10px，左右 50px，下 200px

```

1. div { padding: 10px 50px 200px;
2. }
```

//简写形式，分别是上下 10px，左右 20px

```
1. div { padding: 10px 20px;
2. }
```

//简写形式：上下左右均 10px

```
1. div { padding: 10px;
2. }
```

三. 元素外边距

CSS 盒模型中可以设置元素外部边缘填充空白的大小，我们称为外边距。样式表如下：

属性	值	说明	CSS 版本
margin-top	长度值或百分比	设置顶部内边距	1
margin-bottom	长度值或百分比	设置底部内边距	1
margin-left	长度值或百分比	设置左边内边距	1
margin-right	长度值或百分比	设置右边内边距	1
margin	1 ~ 4 长度值或百分比	简写属性	1

//设置四个内边距

```
1. div { margin-top: 10px; margin-bottom: 10px; margin-left: 10px; margin-right: 10px;
2. }
```

//简写形式，分别为上 10px、右 10px、下 10px、左 10px

```
1. div { margin: 10px 10px 10px 10px;
2. }
```

//简写形式，分别为上 10px，左右 50px，下 200px

```
1. div { margin: 10px 50px 200px;
2. }
```

//简写形式，分别是上下 10px，左边 20px

```
1. div { margin: 10px 20px;
2. }
```

//简写形式：上下左右均 10px

```
1. div { margin: 10px;
2. }
```


四. 处理溢出

当设置了元素固定尺寸且内容过大时，就会出现溢出的问题。溢出主要朝两个方向：右侧和底部。我们可以通过 `overflow` 系列样式来控制它。



溢出处理主要有四种处理值：



//设置溢出为 `auto` 值

```
1. div { overflow-x: auto;  
2. }
```

原文： <https://wizardforce1.gitbooks.io/liyanhui-tutorials/content/20.html>

第 16 章 CSS 盒模型[下]

第 16 章 CSS 盒模型[下]

学习要点：

- 1.元素可见性
- 2.元素盒类型
- 3.元素的浮动

主讲教师：李炎恢

本章主要探讨 HTML5 中 CSS 盒模型，学习怎样了解元素的外观配置以及文档的整体布局。

一．元素可见性

使用 `visibility` 属性可以实现元素的可见性，这种样式一般可以配合 `JavaScript` 来实现效果。样式表如下：

属性	值	说明	CSS 版本
<code>visible</code>	默认值，元素在页面上可见。	2	
<code>visibility</code>	<code>hidden</code>	元素不可见，但会占据空间。	2
<code>collapse</code>	元素不可见，隐藏表格的行与列，如果不是表格，则和 <code>hidden</code> 一样。	2	

//设置元素隐藏，但占位

```
1. div { visibility: hidden;
2. }
```

//隐藏表格的行或列，但不占位，Chrome 和 Opera 不支持

```
1. table tr:first-child { visibility: collapse;
2. }
```

二．元素盒类型

CSS 盒模型中的 `display` 属性，可以更改元素本身盒类型。那么元素有哪些盒类型呢？主要有：1.块级元素（区块）；2 行内元素（内联）；3 行内-块级元素（内联块）；4.隐藏元素。

1.**块级元素**

所谓块级元素，就是能够设置元素尺寸、隔离其他元素功能的元素。比如：`<div>`、`<p>`等文档元素。

2. **行内元素**

所谓行内元素，不能够设置元素尺寸，它只能自适应内容、无法隔离其他元素，其它元素会紧跟其后。比如：
、等文本元素。

3. **行内-块元素**

所谓行内-块元素，可以设置元素尺寸，但无法隔离其他元素的元素。比如。

属性	值	说明	CSS 版本
block	盒子为块级元素	1	1
display	inline	盒子为行内元素	
inline-block	盒子为行内-块元素	2	
none	盒子不可见，不占位	1	

//将行内元素转成块级元素

```
1. span { background: silver; width: 200px; height: 200px; display: block;
2. }
```

//将块级元素转换成行内元素

```
1. div { background: silver; width: 200px; height: 200px; display: inline;
2. }
```

//将块级元素转化成行内-块元素

```
1. div { background: silver; width: 200px; height: 200px; display: inline-block;
2. }
```

//将元素隐藏且不占位

```
1. div { display: none;
2. }
```

display 属性还有非常多的值，有些后面部分讲解，而有些支持度不好或者尚不支持，从而省略。有兴趣的，可以参考 CSS3 手册。

三. 元素的浮动

CSS 盒模型有一种叫浮动盒，就是通过 float 属性建立盒子的浮动方向，样式表如下：

属性	值	说明	CSS 版本
float	left	浮动元素靠左	1
	right	浮动元素靠右	1
	none	禁用浮动	1

//实现联排效果

```
1. #a { background: gray; float: left;
2. } #b { background: maroon; float: left;
3. } #c { background: navy; float: left;
4. }
```

//实现元素右浮动

```
1. #c { background: navy; float: right;
2. }
```

//取消元素的浮动

```
1. #c { background: navy; float: none;
2. }
```

刚才的浮动有一个问题：当一个元素盒子被浮动后，下面的元素会自动堆叠处理，导致元素不可见或部分不可见。我们可以使用 `clear` 属性来处理。

属性	值	说明	CSS 版本
clear	none	允许两边均可浮动	1
left	左边界不得浮动	1	
right	右边界不得浮动	1	
both	两边都不得浮动	1	

//两边均不可浮动

```
1. #c { background: navy; clear: both;
2. }
```

原文: <https://wizardforce1.gitbooks.io/liyanhui-tutorials/content/21.html>

第 17 章 CSS 边框与背景[上]

第 17 章 CSS 边框与背景[上]

学习要点：

- 1. 声明边框
- 2. 边框样式
- 3. 圆角边框

主讲教师：李炎恢

本章主要探讨 HTML5 中 CSS 边框和背景，通过边框和背景的样式设置，给元素增加更丰富的外观。

一．声明边框

边框的声明有三个属性设置，样式表如下：

属性	值	说明	CSS 版本
border-width	长度值	设置边框的宽度，可选	1
border-style	样式名称	设置边框的样式，必选	1
border-color	颜色值	设置边框的颜色，可选	1

这三个属性值，只有 border-style 是必须声明，才可以出现边框。而其他两个属性会出现默认值。

//最简单的边框，边框长度默认 3px，边框颜色为黑色

```
1. div { border-style: solid;
2. }
```

//配置完整的边框

```
1. div { border-style: solid; border-width: 2px; border-color: red;
2. }
```

如果元素长和高均为 200px 时，四个边框均为 2 时，元素的长高总尺寸均为 202px。

二．边框样式

边框的样式主要有三种，分别是边框长度取值、边框的颜色和边框的线条类型。颜色是通用的颜色代码，和所有其他颜色取值一下。而长度和线条类型，边框有自己独到的部分。

边框宽度取值表如下：

--	--

值	说明
长度值	CSS 长度值：比如 px、em 等
百分数	直接设置百分数：1、2、3 等
thin	使用长度名称的预设宽度。这三个值的具体意义由浏览器来定义，从小到大依次增大。
medium	
thick	

一般来说，边框为了更加精准，还要计算元素盒子的总尺寸，使用长度值的比较多。而定义边框线条的样式如下样式表：

值	说明
none	没有边框
dashed	破折线边框
dotted	圆点线边框
double	双线边框
groove	槽线边框
inset	使元素内容具有内嵌效果的边框
outset	使元素内容具有外凸效果的边框
ridge	脊线边框
solid	实线边框

//solid 实线使用频率最高

```
1. div { border-style: solid; border-width: 10px; border-color: red;
2. }
```

如果想对四条边中某一条边单独进行设置，可以使用如下样式表：

属性	说明	CSS 版本
border-top-width	定义顶端	1
border-top-style		
border-top-color		
border-middle-width	定义底部	1
border-middle-style		
border-middle-color		
border-left-width	定义左侧	1
border-left-style		
border-left-color		
border-right-width	定义右边	1
border-right-style		

border-right-color

//只设置顶端

```
1. div { border-top-style: solid; border-top-width: 10px; border-top-color: red;
2. }
```

如果四条变都一致，那么没必要分写成三句样式，直接通过简写即可：

属性	值	说明	CSS 版本
border	<宽度> <样式> <颜色>	设置四条边的边框	1
border-top	只设置上边框		
border-middle	只设置下边框		
border-left	只设置左边框		
border-right	只设置右边框		

//简写形式四条边设置

```
1. div { border: 10px solid red;
2. }
```

三．圆角边框

CSS3 提供了一个非常实用的圆角边框的设置。使用 border-radius 属性，就可以达到这种效果，样式表如下：

属性	值	说明	CSS 版本
border-radius	长度值或百分数	四条边角	3
border-top-left-radius	长度值或百分数	左上边角	3
border-top-right-radius	长度值或百分数	右上边角	3
border-middle-left-radius	长度值或百分数	左下边角	3
border-middle-right-radius	长度值或百分数	右下边角	3

//设置圆角矩形

```
1. div { border: 10px solid red; border-radius: 10px;
2. }
```

//四条边分别设置

```
1. div { border: 10px solid red; border-radius: 10px 20px 30px 40px;
2. }
```

原文：<https://wizardforcel.gitbooks.io/liyanhui-tutorials/content/22.html>

第 17 章 CSS 边框与背景[下]

第 17 章 CSS 边框与背景[下]

学习要点：

1. 设置背景

主讲教师：李炎恢

本章主要探讨 HTML5 中 CSS 边框和背景，通过边框和背景的样式设置，给元素增加更丰富的外观。

一．设置背景

盒模型的尺寸可以通过两种方式实现可见性，一种就是之前的边框，还有一种就是背景。 CSS 背景设置的样式表如下：

属性	说明	CSS 版本
background-color	背景的颜色	1
background-image	背景的图片	1/3
background-repeat	背景图片的样式	1/3
background-size	背景图像的尺寸	3
background-position	背景图像的位置	1
background-attachment	背景图片的滚动	1/3
background-clip	背景图片的裁剪	3
background-origin	背景图片起始点	3
background	背景图片简写方式	1

1.background-color

值	说明	CSS 版本
颜色	设置背景颜色为指定色	1
transparent	设置背景颜色为透明色	1

```
1. div { background-color: silver;
2. }
```

解释：设置元素的背景颜色。属性值是颜色值。

```
1. div b { background-color: transparent;
2. }
```


解释：默认值为 `transparent`，为透明的意思。这样`<div>`内部的元素就会继承`<div>`的背景色。一般来说这个属性使用频率很低，原因是不需要改变色彩时就默认，需要改变色彩时又是颜色值。

```
1. body { background-color: silver;
2. }
```

解释：在`<body>`元素下可以设置整个页面的背景色。

2.background-image

值	说明	CSS 版本
<code>none</code>	取消背景图片的设置	1
<code>url</code>	通过 URL 设置背景图片	1

```
1. body { background-image: url(loading.gif);
2. }
```

解释：`url` 里面是图片的路径，设置整个页面以这个图片为背景，如果图片不足以覆盖，则复制扩展。

```
1. div { background-image: none;
2. }
```

解释：如果多个 `div` 批量设置了背景，而其中某一个不需要背景，可以单独设置 `none` 值取消背景。

在 CSS3 中，背景图片还设置了比如线性、放射性等渐变方式。但由于支持度的问题，比如 IE9 尚未支持。我们把这些新特性放到独立的课程中讲解。

3.background-repeat

值	说明	CSS 版本
<code>repeat-x</code>	水平方向平铺图像	1
<code>repeat-y</code>	垂直方向平铺图像	1
<code>repeat</code>	水平和垂直方向同时平铺图像	1
<code>no-repeat</code>	禁止平铺图像	1

```
1. body { background-image: url(loading.gif); background-repeat: no-repeat;
2. }
```

解释：让背景图片只显示一个，不平铺。CSS3 还提供了两个值，由于支持度不佳，这里忽略。

4.background-position

值	说明	CSS 版本
<code>top</code>	将背景图片定位到元素顶部	1
<code>left</code>	将背景图片定位到元素左部	1

right	将背景图片定位到元素右部	1
bottom	将背景图片定位到元素底部	1
center	将背景图片定位到元素中部	1
长度值	使用长度值偏移图片的位置	1
百分数	使用百分数偏移图片的位置	1

```
1. body { background-image: url(loading.gif); background-repeat: no-repeat; background-position: top;
2. }
```

解释：将背景图片置于页面上方，如果想置于左上方则值为：top left。

```
1. body { background-image: url(loading.gif); background-repeat: no-repeat; background-position: 20px 20px;
2. }
```

解释：使用长度值或百分数，第一值表示左边，第二个值表示上边。

5.background-size

值	说明	CSS 版本
auto	默认值，图像以本尺寸显示	3
cover	等比例缩放图像，使图像至少覆盖容器，但有可能超出容器	3
contain	等比例缩放图像，使其宽度、高度中较大者与容器横向或纵向重合	3
长度值	CSS 长度值，比如 px、em	3
百分数	比如：100%	3

```
1. body { background-image: url(loading.gif); background-size: cover;
2. }
```

解释：使用 cover 相当于 100%，全屏铺面一张大图，这个值非常实用。在等比例放大缩小的过程中，可能会有背景超出，当然，这点无伤大雅。

```
1. div { background-image: url(loading.gif); background-size: contain;
2. }
```

解释：使用 contain 表示，尽可能让图片完整的显示在元素内。

```
1. body { background-image: url(loading.gif); background-size: 240px 240px;
2. }
```

解释：长度值的用法，分别表示长和高。

6.background-attachment

值	说明	CSS 版本
---	----	--------

scroll	默认值，背景固定在元素上，不会随着内容一起滚动	1
fixed	背景固定在视窗上，内容滚动时背景不动	1

```
1. body { background-image: url(loading.gif); background-attachment: fixed;
2. }
```

解释：fixed 属性会导致背景产生水印效果，拖动滚动条而背景不动。

7.background-origin

值	说明	CSS 版本
border-box	在元素盒子内部绘制背景	3
padding-box	在内边距盒子内部绘制背景	3
content-box	在内容盒子内部绘制背景	3

```
1. div { width: 400px; height: 300px; border: 10px dashed red; padding: 50px; background-image: url(img.png);
  background-repeat: no-repeat; background-origin: content-box;
2. }
```

解释：设置背景起始位置。

8.background-clip

值	说明	CSS 版本
border-box	在元素盒子内部裁剪背景	3
padding-box	在内边距盒子内部裁剪背景	3
content-box	在内容黑子内部裁剪背景	3

```
1. div { width: 400px; height: 300px; border: 10px dashed red; padding: 50px; background-image: url(img.png);
  background-repeat: no-repeat; background-origin: border-box; background-clip: padding-box;
2. }
```

解释：在内边距盒子内部裁剪背景。

9.background

```
1. div { width: 400px; height: 300px; border: 10px dashed red; padding: 50px; background: silver url(img.png) no-
  repeat scroll left top/100% border-box content-box;
2. }
```

解释：完整的简写顺序如下：

[background-color]

[background-image]

[background-repeat]

[background-attachment]

[background-position]/[background-size]

[background-origin]

[background-clip]

原文: <https://wizardforcel.gitbooks.io/liyanhui-tutorials/content/23.html>

第 18 章 CSS 表格与列表

第 18 章 CSS 表格与列表

学习要点：

- 1. 表格样式
- 2. 列表样式
- 3. 其他功能

主讲教师：李炎恢

本章主要探讨 HTML5 中 CSS 表格和列表，通过表格和列表的样式设置，让表格和列表显示更加多元化。

一．表格样式

表格有五种独有样式，样式表如下：



1.border-collapse



```
1. table { border-collapse: collapse;
2. }
```

解释：单元格相邻的边框被合并。

2.border-spacing

值	说明	CSS 版本
长度值	0 表示间距，其他使用 CSS 长度值	2

```
1. table { border-spacing: 10px;
2. }
```

解释：border-collapse: separate;的状态下才有效。因为要设置间距，不能合并。

3.caption-side

值	说明	CSS 版本
top	默认值，标题在上方	2

bottom	标题在下方	2
--------	-------	---

```
1. table { table-layout: fixed;
2. }
```

解释：内容过长后，不会拉伸整个单元格。

4.empty-cells

值	说明	CSS 版本
show	默认值，显示边框	2
hide	不显示边框	2

```
1. table { empty-cells: hide;
2. }
```

解释：单元格内容为空是隐藏边框。

5.table-layout

值	说明	CSS 版本
auto	默认值，内容过长时，拉伸整个单元格	2
fixed	内容过长时，不拉伸整个单元格	2

```
1. table {    table-layout: fixed;
2.
3. }
```

解释：内容过长后，不会拉伸整个单元格。

二. 列表样式

列表有四种独有样式，样式表如下：



1.list-style-type



```
1. ul { list-style-type: square;
2. }
```

解释：列表前缀的标记类型，这里是 CSS1 版本的。CSS2 版本还包含比如日文、亚美尼亚数字、希腊文等前缀。有兴趣的可以参考 CSS 手册。

2.list-type-position

值	说明	CSS 版本
outside	默认值，标记位于内容框外部	1
inside	标记位于内容框内部	1

```
1. ul { width: 120px; list-style-position: inside;
2. }
```

解释：标记位于内容框的内部。

3.list-type-image

值	说明	CSS 版本
none	不使用图像	1
url	通过 url 使用图像	1

```
1. ul { list-style-image: url(bullet.png);
2. }
```

解释：使用图片作为前缀的标记。

4.list-style

```
1. ul { list-style: lower-alpha inside url(bullet.png);
2. }
```

解释：简写形式。

三．其他功能

在<table>中<td>单元格，我们可以使用 text-align 属性水平对齐，但是垂直对齐就无法操作了。CSS 提供了 vertical-align 属性用于垂直对齐。

值	说明	CSS 版本
baseline	内容对象与基线对齐	1
top	内容对象与顶端对齐	1
middle	内容对象与中部对齐	1
bottom	内容对象与底部对齐	1

```
1. table tr td { vertical-align: bottom;
2. }
```

解释：将单元格内的内容对象实现垂直对齐。

--	--	--

值	说明	CSS 版本
sub	垂直对齐文本的下标	1
super	垂直对齐文本的上标	1

```
1. b { vertical-align: super;
2. }
```

解释：文本上下标设置。

值	说明	CSS 版本
长度值	设置上下的偏移量，可以是负值	1
百分比	同上	1

```
1. div span { vertical-align: -200px;
2. }
```

解释：负值往下，正值往上。如果默认基线在上面，用负数。如果默认基线在下面，用正值。

原文：<https://wizardforce1.gitbooks.io/liyanhui-tutorials/content/24.html>

第 19 章 CSS 其他样式

第 19 章 CSS 其他样式

学习要点：

- 1. 颜色和透明度
- 2. 盒子阴影和轮廓
- 3. 光标样式

主讲教师：李炎恢

本章主要探讨 HTML5 中 CSS 其他剩下几个常用的样式，包括颜色、透明度、盒子的阴影轮廓以及光标的样式。

一．颜色和透明度

颜色我们之前其实已经用的很多了，比如字体颜色、背景颜色、边框颜色。但除了背景颜色和边框颜色讲解过，字体颜色却没有系统的讲解过。设置字体颜色其实也成为文本块的前景色。

属性	值	说明	CSS 版本
color	颜色值	设置文本前景色	1

```
1. p { color: red;
2. }
```

解释：设置文本颜色。

CSS3 提供了一个属性 opacity，可以设置元素的透明度。

属性	值	说明	CSS 版本
opacity	0 ~ 1	设置元素的透明度	3

```
1. p { color: red; opacity: 0.5;
2. }
```

解释：设置元素的透明度。

二．盒子阴影和轮廓

1. box-shadow

CSS3 提供了一个非常实用的效果样式，就是阴影效果。通过 box-shadow 属性来实现，样式表如下：



```
1. div { width: 200px; height: 200px; border: 10px solid silver; box-shadow: 5px 4px 10px 2px gray;
2. }
```

解释：给元素盒子增加阴影效果。

```
1. box-shadow: 5px 4px 10px 2px gray inset;
```

解释：实现内部阴影。

2.outline

CSS3 还提供了轮廓样式，它和边框一样，只不过它可以在边框的外围再加一层。样式表如下：

属性	值	说明	CSS 版本
outline-color	颜色	外围轮廓的颜色	3
outline-offset	长度	轮廓距离元素边框边缘的偏移量	3
outline-style	样式	轮廓样式，和 border-style 一致	3
outline-width	长度	轮廓宽度	3
outline	简写	<颜色> <样式> <宽度>	3

```
1. div { width: 200px; height: 200px; border: 10px solid silver; outline: 10px double red;
2. }
```

解释：在边框的外围再增加一圈轮廓。

三. 光标样式

我们不但可以指定页面上的元素样式，就连光标的样式也可以指定。样式表如下：

属性	值	说明	CSS 版本
cursor	光标样式	auto, default, none, context-menu, help, pointer, progress, wait, cell, crosshair, text, vertical-text, alias, copy, move, no-drop, not-allowed, e-resize, n-resize, ne-resize, nw-resize, s-resize, se-resize, sw-resize, w-resize, ew-resize, ns-resize, nesw-resize, nwse-resize, col-resize, row-resize, all-scroll	1

```
1. div { cursor: move;
2. }
```

解释：设置当前元素的光标为移动光标。

原文： <https://wizardforce1.gitbooks.io/liyanhui-tutorials/content/25.html>

第 20 章 CSS3 前缀和 rem

第 20 章 CSS3 前缀和 rem

学习要点：

1.CSS3 前缀

2.长度单位 rem

主讲教师：李炎恢

本章主要探讨 HTML5 中 CSS 在发展中实行标准化的一些问题，重点探讨 CSS3 中新属性前缀问题和新的单位 rem。

一. **CSS3** 前缀

在 CSS3 的很多新属性推出时，这些属性还处在不太稳定的阶段，随时可能被剔除。而此时的浏览器厂商为了实现这些属性，采用前缀方法。各大厂商前缀列表如下：

浏览器	厂商前缀
Chrome、Safari	-webkit-
Opera	-o-
Firefox	-moz-
Internet Explorer	-ms-

我们之前学习过几个 CSS3 的新属性，比如：box-shadow、border-radius、opacity等。这几个属性我们在前面的使用中，并没有添加所谓的浏览器厂商前缀。那是因为，这些属性已经在主流浏览器或版本成为了标准。具体进化步骤如下：

- 1.属性尚未提出，这个属性所有浏览器不可用；
- 2.属性被提出，但未列入标准，浏览器厂商通过私有前缀来支持该属性；
- 3.属性被列入标准，可以使用前缀或不使用前缀来实现属性特性；
- 4.属性不需要再使用前缀，所有浏览器都稳定支持。

我们就拿 border-radius 举例，它是 CSS3 的标准属性。早期的时候处于实验阶段，尚未列入标准时，需要使用厂商前缀。具体浏览器支持度如下：

属性	浏览器	带前缀版本	不带前缀版本	标准/实验
border-radius	IE	不支持	9.0+	标准
Firefox	3.0 需带-moz-	4.0+		
Safari	3.1 需带-webkit-	5.1+		

Chrome	4.0	5.0+
Opera	不支持	10.5+

如果是手机等移动端一般采用的是 iOS 或安卓系统，那么基本上它的引擎是 webkit，直接参考-webkit-即可。

在 CSS3 手册上，Chrome 支持 border-radius 的版本为 13.0，而部分教材和文章上写到只要 5.0。当然，这里可能表达的含义可能不同。而截至到 2015 年 4 月份最新的 Chrome 版本已经到 41.0 了，所以，不管是 5.0 还是 13.0 都是老古董了，没必要深究。Opera 支持 border-radius 版本为 11.5，而目前的版本是 28.0，也无伤大雅了。

而被列入标准的 box-shadow 和 opacity 基本与 border-radius 前缀版本一致。

//因为目前处在标准阶段，没必要写前缀了

```
1. div { border-radius: 50px;
2. }
```

//实验阶段的写法

```
1. div { -webkit-border-radius: 50px; -moz-border-radius: 50px; border-radius: 50px;
2. }
```

实验阶段的写法有三句，分别解释一下：-webkit-表示 Chrome 浏览器的私有属性前缀、-moz-表示 Firefox 私有属性前缀，如果是处于实验阶段的旧版本浏览器，那么不支持 border-radius，从而通过厂商前缀的方式来支持。如果是新版浏览器，已经是处于标准阶段，那么又有两种说法：1.如果新版浏览器废弃了前缀，那么前缀写法就不支持了，仅支持标准写法；2.如果新版浏览器没有废弃前缀，那么两种写法都可以，但要注意顺序，且属性值要保持一致。

如果同时出现四个前缀+一个标准写法，四个前缀是当实验阶段时让四种引擎的浏览器厂商支持自己的私有属性，一个标准写法表示当这个属性列入标准后，使用新版浏览器运行时直接执行这个标准属性。

//前缀写法写在标准后面，且值不一样，就会出现問題

```
1. div { border-radius: 50px; -webkit-border-radius: 100px;
2. }
```

特别注意：1.IE 的前缀-ms-，和 Opera 的前缀-o-，在 border-radius 中不存在；2.现在的Opera 浏览器也支持-webkit-前缀，-webkit-border-radius 就能支持；3.Safari for Windows 已被苹果公司在 2012 年放弃，遗留版本为 5.1.7。

最后说明：W3C 官方的立场表示实验阶段的属性仅为了测试，未来有可能修改或删除。而大量的开发者也认为在实际项目中不应该使用前缀，而使用一种优雅降级保证一定的用户体验，而通过渐进增减的方式让新版高级浏览器保证最高的效果。

二. 长度单位**rem**

CSS3 引入了一些新的尺寸单位，这里重点推荐一个：rem 或者成为(根 em)。目前主流的现代浏览器都很稳定的支持。它和 em、百分比不同的是，它不是与父元素挂钩，而是相对于根元素<html>的文本大小来计算的，这样能更好

的统一整体页面的风格。

//首先，来一段 HTML

```
1. <h1>标题<em>小标题</em></h1>
2. <p> 我是一个段落，我是一段<code>代码</code>
3. </p>
```

//其次来一段 CSS

```
1. html { font-size: 62.5%;
2. } h1 { font-size: 3em;
3. } p { font-size: 1.4em;
4. }
```

这里做几个解释，我们在之前的 web 设计中大量使用了 px 单位进行布局。因为，早期的固定布局使用 px 较为方便，逐渐养成了这种习惯。而使用 em 单位其实更加灵活，尤其是在修改样式时，只需要修改一下挂钩元素的那个大小即可，无须每个元素一个个修改。

但就算是 em，还是有一定问题。网页默认的字号大小为 16px，然后通过<html>设置62.5%，将网页基准设置为 10px。而<h1>设置为 3em，就是自身大小的 3 倍；<p>设置为1.4em，就是 10px 的 1.4 倍，即 14px。

现在问题来了，<code>里面的文本想设置 11px，怎么办呢？设置 1.1em 吗？不对，因为它挂钩的父元素不是<html>而是<p>变成了 14px 的 1.1 倍了，而想设置 11px，则需要设置 0.786 倍才行。但是，这样的计算量太大了。所以，W3C 推出了直接基于根元素单位：rem。

//直接基于<html>的单位

```
1. code { font-size: 1.1 rem;
2. }
```

浏览器	rem 单位
Opera	11.6+
Firefox	3.6+
Safari	5.0+
Chrome	6.0+
IE	9.0+

原文：<https://wizardforcel.gitbooks.io/liyanhui-tutorials/content/26.html>

第 21 章 CSS3 文本效果

第 21 章 CSS3 文本效果

学习要点：

1. 文本阴影

2. 文本裁剪

3. 文本描边

4. 文本填充

主讲教师：李炎恢

本章主要探讨 HTML5 中 CSS3 中文本效果，其中也包含一些之前讲过的 CSS3 文本属性。

一. 文本阴影

CSS3 提供了 `text-shadow` 文本阴影效果，这个属性在之前讲过，只是没有涉及浏览器支持情况。

text-shadow	Opera	Firefox	Chrome	IE	Safari
9.5+	3.5+	4+	10+	3.1+	

这里有几个注意点：1. `text-shadow` 在 CSS2 的时候出现过，但各大浏览器碍于消耗大量的资源，迟迟未支持，然后在 CSS2.1 中剔除了。现在，CSS3 已经全面支持了；2. 最低支持版本上，不同手册和教材上都不太同，但版本年代久远，无伤大雅。最准确的可以查询这个网站：<http://caniuse.com>。最需要注意的只有 IE10 才支持，IE9 不支持的；3. 目前的浏览器不需要给这个属性加上任何前缀，具体查询前缀版本可以访问刚才提供的地址。

//正值阴影

```
1. text-shadow: 1px 1px 1px red;
```

//负值阴影

```
1. text-shadow: -1px -1px 1px red;
```

//多重阴影叠加

```
1. text-shadow: 0px 0px 0 rgb(188,178,188),
2.           1px 0px 0 rgb(173,163,173),
3.           2px 0px 0 rgb(157,147,157),
4.           3px 0px 0 rgb(142,132,142),
5.           4px 0px 0 rgb(126,116,126),
6.           5px 0px 0 rgb(111,101,111),
7.           6px 0px 0 rgb(95,85,95),
```

```

8.      7px 0px 0 rgb(79,69,79),
9.      8px 0px 7px rgba(0,0,0,0.35),
10.     8px 0px 1px rgba(0,0,0,0.5),
11.     0px 0px 7px rgba(0,0,0,0.2);

```

二. 文本裁剪

CSS3 提供了 `text-overflow` 属性来控制文本的溢出部分，它的作用是对溢出的部分裁剪掉，然后判定是否添加省略号。首先我们先看下样式表的属性，如下：

属性值	说明
<code>clip</code>	默认值，裁剪文本时不添加“...”省略号
<code>ellipsis</code>	裁剪文本时添加“...”省略号

//必须不换行和使用 `overflow` 控制溢出

```

1. p { width: 160px; white-space: nowrap; background: silver; /*text-overflow: clip;*/ text-overflow: ellipsis;
    overflow: hidden;
2. }

```

对于 `text-overflow` 的支持度，是根据它的属性值来判定的，不同的属性值浏览器支持度不同。

属性值	Opera	Firefox	Chrome	IE	Safari
<code>clip</code>	9.63+	2.0+	1.0+	6.0+	3.1+
<code>ellipsis</code>	10.5+	6.0+	1.0+	6.0+	3.1+

//在 Opera 早期版本 10.0 ~ 10.1 中，需要使用带前缀的 `-o-`。

```

1. p { -o-text-overflow: ellipsis; text-overflow: ellipsis;
2. }

```

而在 Opera 主流版本中，引擎已经靠拢 webkit，则不需要 `-o-` 了。目前来说，也不需要兼容 `-o-` 了。

三. 文本描边

CSS3 提供了描边属性，即 `text-stroke`、`text-stroke-width`、`text-stroke-color`。目前只有 webkit 引擎的浏览器支持，并且必须加上 `-webkit-` 前缀才能有效。

属性	Opera	Firefox	Chrome	IE	Safari
<code>text-stroke</code> 系列	15.0+	不支持	4.0+	不支持	3.1+

//实验阶段的产物，了解即可

```

1. p { font-size: 50px; -webkit-text-stroke: 1px red;
2. }

```

//修改描边的颜色和厚度

```

1. p { font-size: 50px; -webkit-text-stroke: 1px red; -webkit-text-stroke-color: orange; -webkit-text-stroke-
    width: 2px;
2. }
```

四. 文本填充

CSS3 提供了一个文本颜色填充功能: `text-fill-color`, 感觉和 `color` 属性很像。其实在配合其他属性才能达到不一样的效果。

属性	Opera	Firefox	Chrome	IE	Safari
<code>text-fill-color</code>	15.0+	不支持	4.0+	不支持	3.1+

//不配合背景样式时, 和 `color` 属性没区别

```

1. p { font-size: 150px; -webkit-text-fill-color: red;
2. }
```

//和 CSS3 背景的新特性搭配产生渐变文字

```

1. p { font-size: 150px; font-family: 黑体; background: -webkit-linear-gradient(top, #eee, #aaa 50%, #333 51%, #000);
    -webkit-background-clip: text; -webkit-text-fill-color: transparent;
2. }
```

原文: <https://wizardforce1.gitbooks.io/liyanhui-tutorials/content/27.html>

第 21 章 CSS3 文本效果

第 21 章 CSS3 文本效果

学习要点：

1. 文本阴影

2. 文本裁剪

3. 文本描边

4. 文本填充

主讲教师：李炎恢

本章主要探讨 HTML5 中 CSS3 中文本效果，其中也包含一些之前讲过的 CSS3 文本属性。

一. 文本阴影

CSS3 提供了 `text-shadow` 文本阴影效果，这个属性在之前讲过，只是没有涉及浏览器支持情况。

text-shadow	Opera	Firefox	Chrome	IE	Safari
9.5+	3.5+	4+	10+	3.1+	

这里有几个注意点：1. `text-shadow` 在 CSS2 的时候出现过，但各大浏览器碍于消耗大量的资源，迟迟未支持，然后在 CSS2.1 中剔除了。现在，CSS3 已经全面支持了；2. 最低支持版本上，不同手册和教材上都不太同，但版本年代久远，无伤大雅。最准确的可以查询这个网站：<http://caniuse.com>。最需要注意的只有 IE10 才支持，IE9 不支持的；3. 目前的浏览器不需要给这个属性加上任何前缀，具体查询前缀版本可以访问刚才提供的地址。

//正值阴影

```
1. text-shadow: 1px 1px 1px red;
```

//负值阴影

```
1. text-shadow: -1px -1px 1px red;
```

//多重阴影叠加

```
1. text-shadow: 0px 0px 0 rgb(188,178,188),
2.           1px 0px 0 rgb(173,163,173),
3.           2px 0px 0 rgb(157,147,157),
4.           3px 0px 0 rgb(142,132,142),
5.           4px 0px 0 rgb(126,116,126),
6.           5px 0px 0 rgb(111,101,111),
7.           6px 0px 0 rgb(95,85,95),
```

```

8.      7px 0px 0 rgb(79,69,79),
9.      8px 0px 7px rgba(0,0,0,0.35),
10.     8px 0px 1px rgba(0,0,0,0.5),
11.     0px 0px 7px rgba(0,0,0,0.2);

```

二. 文本裁剪

CSS3 提供了 `text-overflow` 属性来控制文本的溢出部分，它的作用是对溢出的部分裁剪掉，然后判定是否添加省略号。首先我们先看下样式表的属性，如下：

属性值	说明
clip	默认值，裁剪文本时不添加“...”省略号
ellipsis	裁剪文本时添加“...”省略号

//必须不换行和使用 `overflow` 控制溢出

```

1. p { width: 160px; white-space: nowrap; background: silver; /*text-overflow: clip;*/ text-overflow: ellipsis;
    overflow: hidden;
2. }

```

对于 `text-overflow` 的支持度，是根据它的属性值来判定的，不同的属性值浏览器支持度不同。

属性值	Opera	Firefox	Chrome	IE	Safari
clip	9.63+	2.0+	1.0+	6.0+	3.1+
ellipsis	10.5+	6.0+	1.0+	6.0+	3.1+

//在 Opera 早期版本 10.0 ~ 10.1 中，需要使用带前缀的 `-o-`。

```

1. p { -o-text-overflow: ellipsis; text-overflow: ellipsis;
2. }

```

而在 Opera 主流版本中，引擎已经靠拢 webkit，则不需要 `-o-` 了。目前来说，也不需要兼容 `-o-` 了。

三. 文本描边

CSS3 提供了描边属性，即 `text-stroke`、`text-stroke-width`、`text-stroke-color`。目前只有 webkit 引擎的浏览器支持，并且必须加上 `-webkit-` 前缀才能有效。

属性	Opera	Firefox	Chrome	IE	Safari
text-stroke 系列	15.0+	不支持	4.0+	不支持	3.1+

//实验阶段的产物，了解即可

```

1. p { font-size: 50px; -webkit-text-stroke: 1px red;
2. }

```

//修改描边的颜色和厚度

```
1. p { font-size: 50px; -webkit-text-stroke: 1px red; -webkit-text-stroke-color: orange; -webkit-text-stroke-width: 2px;
2. }
```

四. 文本填充

CSS3 提供了一个文本颜色填充功能: `text-fill-color`, 感觉和 `color` 属性很像。其实在配合其他属性才能达到不一样的效果。

属性	Opera	Firefox	Chrome	IE	Safari
<code>text-fill-color</code>	15.0+	不支持	4.0+	不支持	3.1+

//不配合背景样式时, 和 `color` 属性没区别

```
1. p { font-size: 150px; -webkit-text-fill-color: red;
2. }
```

//和 CSS3 背景的新特性搭配产生渐变文字

```
1. p { font-size: 150px; font-family: 黑体; background: -webkit-linear-gradient(top, #eee, #aaa 50%, #333 51%, #000);
      -webkit-background-clip: text; -webkit-text-fill-color: transparent;
2. }
```

原文: <https://wizardforce1.gitbooks.io/liyanhui-tutorials/content/28.html>

第 23 章 CSS3 边框图片效果

第 23 章 CSS3 边框图片效果

学习要点：

1. 属性初探
2. 属性解释
3. 简写和版本

主讲教师：李炎恢

本章主要探讨 HTML5 中 CSS3 中边框图片背景的效果，通过这个新属性让边框更加的丰富多彩。

一. 属性解释

CSS3 提供了一个新的属性集合，用这几个属性可以嵌入图片形式的边框。这样，边框就可以自定义了。

1. `border-image-source` //引入背景图片地址
2. `border-image-slice` //切割引入背景图片
3. `border-image-width` //边框图片的宽度
4. `border-image-repeat` //边框背景图片的排列方式
5. `border-image-outset` //边框背景向外扩张
6. `border-image` //上面五个属性的简写方式

二. 属性解释

要实现边框背景，我们对图片也有一定的要求，否则效果不能完全体现出来。图片可以通过九宫格的切分来了解它。我们使用 W3C 官网上教学的图片来讲解一下。



如上图所示，九宫格并不一定要求每一个格子大小都相同。当然，如果相同的话，制作边框背景就相对容易一点。比如如下这张图片：



首先，用 Photoshop 软件分析一下这个标准九宫格的总体大小和每个格子的大小。最终得出图片总大小为 81px 正方形，四个角的大小为 27px 的正方形，其余五个角也是 27px。

那么，第一步：先创建一个盒子区域，大小为 400x400 的矩形。然后设置引入边框图像。

//引入边框图像

```
1. border-image-source: url(border.png);
```

单单只有这句话，webkit 引擎下的浏览器会在盒子区块的四个角看到一丁点图像的影子。而其他浏览器什么都看不到。这是由于没有设置边框背景图像的宽度导致的。

//设置边框图像宽度，上右下左，可以设置四个值

```
1. border-image-width: 81px;
```

这里设置的是边框图像的宽度，而不是边框宽度。当你设置边框宽度，你会发现，文本会偏移。而边框图像的宽度不会挤压文本。

//设置边框的宽度

```
1. border-width: 20px;
```

以上设置完毕后，支持边框背景图片的浏览器会在四个角落铺上这张图片的完整形式。这个时候需要通过引入切割属性来配置背景图片的显示方式。

//首先，边框图像宽度设置为 27px 和一个单格宽高一致

```
1. border-image-width: 27px;
```

//设置切割属性的大小

```
1. border-image-slice: 27;
```

这里的 27 不需要设置 px 像素，因为它默认就是像素。设置 27 之后，我们会发现边框的四个角正好是橘红色的四个角。那么你可以逐步放大或逐步放下这个值，来体验一下它的变化。

//从 27 逐步放大到 81，四个角都慢慢缩小，各自显示一个完整的图像

```
1. border-image-slice: 81;
```

//从 27 逐步缩小到 0，发现四个角都慢慢变大，配合 fill 整体显示一个完整图像

```
1. border-image-slice: 0 fill;
```

上面只是单独设置了一个像素表示四个边切割的大小，你也可以设置百分比、浮点值或者分别设置四个变的大小。

//33.5%差不多 27

```
1. border-image-slice: 33.5%;
```

//上下设置 27，左右设置 0

```
1. border-image-slice: 27 0;
```

如果想让边框背景向外扩张，那么可以进行扩张设置。

//向外扩张 20px，也可以是浮点值，比如 2.2

```
1. border-image-outset: 20px;
```

四个角设定好之后，我们要设定四个变的显示排列方式。使用 `border-image-repeat` 属性，有四个值提供使用，分别如下表：

属性	说明
stretch	指定用拉伸方式填充边框背景图。默认值。
repeat	指定用平铺方式来填充边框背景图。当图片碰到边界时，如果超过则被截断。
round	指定用平铺方式来填充边框背景图。图片会根据边框的尺寸动态调整图片的大小，直至正好可以铺满整个边框。
space	指定用平铺方式来填充边框背景图。图片会根据边框的尺寸动态调整图片的之间的间距，直至正好可以铺满整个边框。

//拉伸方式填充，当然，通过上右下左设置四个边均可

```
1. border-image-repeat: stretch;
```

//平铺填充，超过则被截断

```
1. border-image-repeat: repeat;
```

//平铺填充，动态调整图片大小直至铺满

```
1. border-image-repeat: round;
```

//平铺填充，动态调整图片的间距直至铺满

```
1. border-image-repeat: space;
```

//另一个按钮的小例子

```
1. div { width: 400px; height: 40px; border-image-source: url(button.png); border-image-width: 10px; border-
  image-slice: 10 fill; border-image-repeat: stretch;
2. }
```

三. 简写和版本

//border-image 简写格式很简单，具体如下：

```
1. border-image: <' border-image-source '> || <' border-image-slice '> [ /<' border-image-width '> | / <' border-image-width '>? / <' border-image-outset '> ]? || <' border-image-repeat '>
```

//以上是手册上摘录的，转换成实际格式如下：

```
1. border-image: url(border.png) 27/27px round;
```

对于支持的浏览器及版本如下表：

	Opera	Firefox	Chrome	Safari	IE
部分支持需带前缀	11.5~12.1	3.5 ~ 14	4 ~ 14	3.1 ~ 5.1	无
支持需带前缀	无	无	无	无	无
支持不带前缀	15+	15+	15+	6+	11.0+

//兼容加上前缀

```
1. -webkit-border-image: url(border.png) 27/27px round;
2. -moz-border-image: url(border.png) 27/27px round;
3. -o-border-image: url(border.png) 27/27px round;
4. border-image: url(border.png) 27/27px round;
```

原文：<https://wizardforce1.gitbooks.io/liyanhui-tutorials/content/29.html>

第 24 章 CSS3 变形效果[下]

第 24 章 CSS3 变形效果[下]

学习要点：

1.3D 变形简介

2.transform-style

3.perspective

4.3D 变形属性

主讲教师：李炎恢

本章主要探讨 HTML5 中 CSS3 的变形效果，主要接着上节课的 2D 平面变形转换到 3D 立体变形。

一. **3D** 变形简介

之前我们学习了元素的平移、旋转、缩放和倾斜等功能。这些效果只是单纯在二维平面图上的，我们称之为 2D。那么其实 CSS3 也提供了三维立体的一些功能效果，并且目前较新的主流浏览器都比较支持，只不过比 2D 晚一些，对浏览器的版本要求也要高一些。

由于 3D 是立体三维，在 x、y 轴的基础上一般会多出一个 z 轴，深入跃出轴。以下是 3D 变形的属性值表，如下：

属性值	说明
translate3d(x,y,z)	3D 方式平移元素，设置 x、y 和 z 轴
translateZ(z)	设置 3D 方式平移元素的 z 轴
scale3d(x,y,z)	3D 方式缩放一个元素
scaleZ(z)	设置 3D 方式缩放元素的 z 轴
rotate3d(x,y,z,a)	3d 方式旋转元素
rotateX(a)	分别设置 3D 方式的旋转元素的 x、y 和 z 轴
rotateY(a)	
rotateZ(a)	
perspective(长度值)	设置一个透视投影矩阵
matrix3d(多个值)	定义一个矩阵

3D 变形比 2D 变形出来的要晚一些，所以如果需要兼容旧版本浏览器，可以对照这个表。具体如下：

	Opera	Firefox	Chrome	Safari	IE
支持需带前缀	15 ~ 22	10 ~ 15	12 ~ 35	4 ~ 8	无
支持不带前缀	23+	16+	26+	无	10.0+

//兼容版本完整形式

```
1. -webkit-transform: translateZ(200px);
2. -moz-transform: translateZ(200px);
3. -o-transform: translateZ(200px);
4. -ms-transform: translateZ(200px);
5. transform: translateZ(200px);
```

二. **transform-style**

transform-style 属性是指定嵌套元素如何在 3D 空间中呈现。

属性值	说明
flat	默认值，表示所有子元素在 2D 平面呈现。
preserve-3d	表示子元素在 3D 空间中呈现。

//一般设置到当前元素的父元素

```
1. transform-style: preserve-3d;
```

需要再配合后面的功能属性和变形配置，才能看到效果。同样，这个属性也需要加上各种厂商前缀。

三. **perspective**

perspective 是 3D 变形的重要属性，该属性会设置查看者的位置，并将可视内容映射到一个视锥上，继而投放到一个 2D 平面上。

属性值	说明
none	默认值，表示无限的角度来看 3D 物体，但看上去是平的。
长度值	接受一个长度单位大于 0 的值，其单位不能为百分比。值越大，角度出现的越远，就好比人离远一点看物体。值越小，正相反。

//设置查看者的距离位置，一般设置在元素的父元素上

```
1. perspective: 1000px;
```

需要再配合后面的功能属性和变形配置，才能看到效果。同样，这个属性也需要加上各种厂商前缀。

四. **3D 变形属性**

我们运用前面 3D 功能属性 transform-style 和 perspective 来构建 3D 变形效果。

1. **translate3d(x,y,z)**

//需要 3D 位移的 HTML 结构，必须有父元素包含

```
1. <div id="a">
```

```
2.     
3. </div>
```

//CSS 部分，父元素设置 3D 呈现且设置透视距离

```
1. #a { perspective: 1000px; transform-style: preserve-3d;
2. } img {
3.     /*z 轴可以是负值*/ transform: translate3d(300px,100px,240px);
4. }
```

2.translateZ(z)

//可以单独设置 z 轴，z 轴可以是负值

```
1. img { transform: translateZ(240px);
2. }
```

3.scale3d(x,y,z)

//3D 缩放，单独设置无效，需要配合角度

```
1. img { transform: scale3d(1,1,1.5) rotateX(45deg);
2. }
```

4.scaleZ(z)

//单独设置 z 轴，x 和 y 轴默认为 1

```
1. img { transform: scaleZ(1.5) rotateX(45deg);
2. }
```

5.rotate3d(x,y,z,a)

//设置 3D 旋转，a 表示角度，xyz 是 0 或 1 之间的数值

```
1. transform: rotate3d(1,0,0,45deg);
```

6.rotateX(a)**、rotateY(a)、**rotateZ(a)

//单独设置 3D 旋转

```
1. transform: rotateX(45deg);
2. transform: rotateY(45deg);
3. transform: rotateZ(45deg);
4. transform: rotateX(45deg) rotateY(45deg) rotateZ(45deg);
```

最后一个 matrix3d 就不多说了，忽略。

CSS3 还提供了 `perspective-origin` 属性来设置 3D 变形中的源点角度。该属性默认值为 `50% 50%`也就是 `center center`。

属性值	说明
百分数值	指定元素 x 轴或 y 轴的起点
长度值	指定距离
left	指定 x 轴的位置
center	
right	
top	指定 y 轴的位置
center	
bottom	

//源点设置为右上方变形

```
1. perspective-origin: top right;
```

CSS3 还提供了一个在元素中设置透视的值 `perspective(长度值)`，但它还是和在父元素设置有一定不同。因为父元素整个作为透视，而元素自己作为透视，导致不同。

//具体测试看透视的距离

```
1. img { transform: perspective(1000px) rotateY(45deg);  
2. }
```

原文: <https://wizardforce1.gitbooks.io/liyanhui-tutorials/content/30.html>

第 25 章 CSS3 过渡效果

第 25 章 CSS3 过渡效果

学习要点：

- 1.过渡简介
- 2.transition-property
- 3.transition-duration
- 4.transition-timing-function
- 5.transition-delay
- 6.简写和版本

主讲教师：李炎恢

本章主要探讨 HTML5 中 CSS3 的过渡效果，通过这个功能可以不借助 JavaScript 来实现简单的用户交互功能。

一. 过渡简介

过渡效果一般是通过一些简单的 CSS 动作触发平滑过渡功能，比如：`:hover`、`:focus`、`:active`、`:checked` 等。CSS3 提供了 `transition` 属性来实现这个过渡功能，主要属性如下表：

属性	说明
<code>transition-property</code>	指定过渡或动态模拟的 CSS 属性
<code>transition-duration</code>	指定完成过渡所需的时间
<code>transition-timing-function</code>	指定过渡的函数
<code>transition-delay</code>	指定过渡开始出现的延迟时间
<code>transition</code>	简写形式，按照上门四个属性值连写

我们先创建一个没有过渡效果的元素，然后通过 `:hover` 来触发它。在没有任何过渡效果的触发，会立即生硬的执行触发。

//设置元素样式

```
1. div { width: 200px; height: 200px; border: 1px solid green;
2. }
```

//鼠标悬停后背景变黑，文字变白

```
1. div:hover { background-color: black; color: white; margin-left: 50px;
```

```
2. }
```

二. **transition-property**

首先，设置过渡的第一个属性就是指定过渡的属性。同样，你需要指定你要过渡某个元素的两套样式用于用户和页面的交互。那么就使用 `transition-property` 属性，详细属性值如下表：

属性值	说明
none	没有指定任何样式
all	默认值，指定元素所支持 <code>transition-property</code> 属性的样式
指定样式	指定支持 <code>transition-property</code> 的样式

从上门的列表中来看，一般来说，`none` 用于本身有过渡样式从而取消。而 `all`，则是支持所有 `transition-property` 样式，还有一种是指定 `transition-property` 中的某些样式。那么 `transition-property` 支持的样式有哪些？如下表所示：

样式名称	样式类型
background-color	color(颜色)
background-image	only gradients(渐变)
background-position	percentage, length(百分比, 长度值)
border-bottom-color	color
border-bottom-width	length
border-color	color
border-left-color	color
border-left-width	length
border-right-color	color
border-right-width	length
border-spacing	length
border-top-color	color
border-top-width	length
border-width	length
bottom	length, percentage
color	color
crop	rectangle
font-size	length, percentage
font-weight	number
grid-*	various
height	length, percentage
left	length, percentage
letter-spacing	length

line-height	number, length, percentage
margin-bottom	length
margin-left	length
margin-right	length
margin-top	length
max-height	length, percentage
max-width	length, percentage
min-height	length, percentage
min-width	length, percentage
opacity	number
outline-color	color
outline-offset	integer
outline-width	length
padding-bottom	length
padding-left	length
padding-right	length
padding-top	length
right	length, percentage
text-indent	length, percentage
text-shadow	shadow
top	length, percentage
vertical-align	keywords, length, percentage
visibility	visibility
width	length, percentage
word-spacing	length, percentage
z-index	integer
zoom	number

//设置背景和文字颜色采用过渡效果

```
1. transition-property: background-color, color, margin-left;
```

三. **transition-duration**

如果单纯设置过渡的样式，还不能够立刻实现效果。必须加上过渡所需的时间，因为默认情况下过渡时间为 0。

//设置过渡时间为 1 秒钟，如果是半秒钟可以设置为 .5s

```
1. transition-duration: 1s;
```

四. **transition-timing-function**

当过渡效果运行时，比如产生缓动效果。默认情况下的缓动是：元素样式从初始状态过渡到终止状态时速度由快到慢，逐渐变慢，即 `ease`。也是默认值，其他几种缓动方式如下表所示：



//恒定速度

```
1. transition-timing-function: linear;
```

以上五种都是设定好的属性值，我们也可以自定义这个缓动。使用 `cubic-bezier()` 属性值，里面传递四个参数 `p0, p1, p2, p3`，值在 0~1 之间。

//自定义缓动

```
1. transition-timing-function: cubic-bezier(0.25, 0.67, 0.11, 0.55);
```

至于具体这些数值干什么的，怎么才可以精确得到相关的信息，这个要了解计算机图形学中的三次贝塞尔曲线的相关知识，类似与 photoshop 中的曲线调色。这里我们忽略。

还有一种不是平滑过渡，是跳跃式过渡，属性值为：`steps(n, type)`。第一个值是一个数值，表示跳跃几次。第二个值是 `start` 或者 `end`，可选值。表示开始时跳跃，还是结束时跳跃。

//跳跃 10 次至结束

```
1. transition-timing-function: steps(10, end);
```

五. **transition-delay**

这个属性可以设置一个过渡延迟效果，就是效果在设置的延迟时间后再执行。使用 `transition-delay` 属性值。如果有多个样式效果，可以设置多个延迟时间，以空格隔开。

//设置延迟效果

```
1. transition-delay: 0s, 1s, 0s;
```

六. 简写和版本

我可以直接使用 `transition` 来简写，有两种形式的简写。第一种是，每个样式单独声明；第二种是不去考虑样式，即使用 `all` 全部声明。

//单独声明

```
1. transition: background-color 1s ease 0s, color 1s ease 0s, margin-left 1s ease 0s;
```

//如果每个样式都是统一的，直接使用 `all`

```
1. transition: all 1s ease 0s;
```

为了兼容旧版本，需要加上相应的浏览器前缀，版本信息如下表：

	Opera	Firefox	Chrome	Safari	IE
支持需带前缀	15 ~ 22	5 ~ 15	4 ~ 25	3.1 ~ 6	无
支持不带前缀	23+	16+	26+	6.1+	10.0+

//兼容完整版

```
1. -webkit-transition: all 1s ease 0s;  
2. -moz-transition: all 1s ease 0s;  
3. -o-transition: all 1s ease 0s;  
4. -ms-transition: all 1s ease 0s;  
5. transition: all 1s ease 0s;
```

原文：<https://wizardforcel.gitbooks.io/liyanhui-tutorials/content/31.html>

第 26 章 CSS3 动画效果

第 26 章 CSS3 动画效果

学习要点：

- 1. 动画简介
- 2. 属性详解
- 3. 简写和版本

主讲教师：李炎恢

本章主要探讨 HTML5 中 CSS3 的动画效果，可以通过类似 Flash 那样的关键帧模式控制运行。

一．动画简介

CSS3 提供了类似 Flash 关键帧控制的动画效果，通过 animation 属性实现。那么之前的 transition 属性只能通过指定属性的初始状态和结束状态来实现动画效果，有一定的局限性。

animation 实现动画效果主要由两个部分组成：1.通过类似 Flash 动画中的关键帧声明一个动画；2.在 animation 属性中调用关键帧声明的动画。

CSS3 提供的 animation 是一个复合属性，它包含了很多子属性。如下表所示：

属性	说明
animation-name	用来指定一个关键帧动画的名称，这个动画名必须对应一个@keyframes 规则。CSS 加载时会应用 animation-name 指定的动画，从而执行动画。
animation-duration	用来设置动画播放所需的时间
animation-timing-function	用来设置动画的播放方式
animation-delay	用来指定动画的延迟时间
animation-iteration-count	用来指定动画播放的循环次数
animation-direction	用来指定动画的播放方向
animation-play-state	用来控制动画的播放状态
animation-fill-mode	用来设置动画的时间外属性
animation	以上的简写形式

除了这 9 个属性之外，动画效果还有一个重要的属性，就是关键帧属性：`@keyframes`。它的作用是声明一个动画，然后在 `animation` 调用关键帧声明的动画。

//基本格式，“name”可自定义

```
1. @keyframes name {
2.   /*...*/
3. }
```

二. 属性详解

在讲解动画属性之前，先创建一个基本的样式。

//一个 div 元素

```
1. <div>我是 HTML5</div>
```

//设置 CSS

```
1. div { width: 200px; height: 200px; border: 1px solid green;
2. }
```

1. @keyframes

//创建动画的第一步，先声明一个动画关键帧。

```
1. @keyframes myani { 0% {
2.     margin-left:0px;
3.   } 50% { margin-left:100px;
4.   } 100% { margin-left:0px;
5.   } }
```

//或者重复的，可以并列写在一起

```
1. @keyframes myani { 0%, 100% {
2.     margin-left:0px;
3.   } 50% { background-color: black; margin-left:100px;
4.   } }
```

2. animation-name

//调用@keyframes 动画

```
1. animation-name: myani;
```

属性值	说明
none	默认值，没有指定任何动画

INDEX

是由@keyframes 指定创建的动画名称

3.animation-duration

//设置动画播放的时间

```
1. animation-duration: 1s;
```

当然，以上通过关键帧的方式，这里插入了三个关键帧。0%设置了白色和左偏移为 0，和初始状态一致，表明从这个地方开始动画。50%设置了黑色，左偏移 100px。而 100%则是最后一个设置，又回到了白色和左偏移为 0。整个动画就一目了然了。

而对于关键帧的用法，大部分用百分比比较容易控制，当然，还有其他一些控制方法。

//从什么状态过渡到什么状态

```
1. @keyframes myani { from {
2.     margin-left:0px;
3.   } to { margin-left:100px;
4.   } }
```

4.animation-timing-function

//设置缓动

```
1. animation-timing-function: ease-in;
```



5.animation-delay

//设置延迟时间

```
1. animation-delay: 1s;
```

6.animation-iteration-count

//设置循环次数

```
1. animation-iteration-count: infinite;
```

属性值	说明
次数	默认值为 1
infinite	表示无限次循环

7.animation-direction

//设置缓动方向交替

```
1. animation-direction: alternate;
```

属性值	说明
normal	默认值，每次播放向前
alternate	一次向前，一次向后，一次向前，一次向后这样交替

8.animation-play-state

//设置停止播放动画

```
1. animation-play-state: paused;
```

9.animation-fill-mode

//设置结束后不在返回

```
1. animation-fill-mode: forwards;
```

属性值	说明
none	默认值，表示按预期进行和结束
forwards	动画结束后继续应用最后关键帧位置，即不返回
backforwards	动画结束后迅速应用起始关键帧位置，即返回
both	根据情况产生 forwards 或 backforwards 的效果

//both 需要结合，次数和播放方向 animation-iteration-count: 4; animation-direction: alternate;

六. 简写和版本

//简写形式完整版

```
1. animation: myani 1s ease 2 alternate 0s both;
```

为了兼容旧版本，需要加上相应的浏览器前缀，版本信息如下表：

	Opera	Firefox	Chrome	Safari	IE
支持需带前缀	15 ~ 29	5 ~ 15	4 ~ 42	4 ~ 8	无
支持不带前缀	无	16+	43+	无	10.0+

//兼容完整版，Opera 在这个属性上加入 webkit，所以没有

```
1. -o--webkit-animation: myani 1s ease 2 alternate 0s both;  
2. -moz-animation: myani 1s ease 2 alternate 0s both;  
3. -ms-animation: myani 1s ease 2 alternate 0s both;  
4. transition: all 1s ease 0s;
```

//@keyframes 也需要加上前缀

```
1. @-webkit-keyframes myani {...} @-moz-keyframes myani {...} @-o-keyframes myani {...} @-ms-keyframes myani  
   {...} keyframes myani {...}
```

原文: <https://wizardforce1.gitbooks.io/liyanhui-tutorials/content/32.html>

第 27 章 CSS 传统布局[上]

第 27 章 CSS 传统布局[上]

学习要点：

1. 布局模型

2. 表格布局

3. 浮动布局

主讲教师：李炎恢

本章主要探讨 HTML5 中 CSS 早期所使用的传统布局，很多情况下，这些布局方式还是非常有用的。

一．布局模型

在早期没有平板和智能手机等移动设备大行其道的时期，Web 页面的设计主要是面向 PC 端电脑分辨率展开的。这种分辨率比例比较单一，基本上只要满足最低分辨率设计即可。一般来说有 4:3、16:10、16:9 这样的主要分辨率。那么，从这种比例上来看，长度总是大于宽度的。从最低分辨率 1024 * 768 设计即可。为了使浏览器底部不出现滚动条，需要减去适当的宽度，比如减去 28，最终固定长度设置为 996 即可。当然，也有一些网站在近两年讲最低分辨率设置为 1280 减去滚动条宽度，因为大显示器逐步主流。

除了刚才所说的固定长度的布局，还有一种是流体布局，就是布局的长度为百分比，比如 100%。不管你是什么分辨率，它都能全屏显示，当然，局限性也特别大，只适合一些单一页面，复杂的页面，会随着不同浏览器产生各种阅读障碍。

我们创建一个三行两列模型。并采用表格布局和浮动布局，构建固定和流体布局的方式，模型图如下：



二．表格布局

表格布局，就是通过设定固定的单元格，去除表格边框和填充实现的布局。当然这个布局非常不建议使用，只是教学了解。表格应该用它最为语义的地方，就是二维表的数据显示。

1. 固定布局

//HTML 部分

```
1. <table border="0">
2.     <tr>
3.         <td colspan="2" class="header">header</td>
4.     </tr>
5.     <tr>
6.         <td class="aside">aside</td>
7.         <td class="section">section</td>
```

```

8.     </tr>
9.     <tr>
10.         <td colspan="2" class="footer">footer</td>
11.     </tr>
12. </table>

```

//CSS 部分

```

1. body { margin: 0;
2. } table { margin: 0 auto; width: 960px; border-spacing: 0;
3. } .header { height: 120px;
4. } .aside { width: 200px; height: 500px;
5. } .section { width: 760px; height: 500px;
6. } .footer { height: 120px;
7. }

```

2. 流体布局

表格的固定布局改成流体布局非常简单，只需要设置 table 为 100%即可。

//修改 table

```

1. table { width: 100%;
2. }

```

三. 浮动布局

浮动布局主要采用 float 和 clear 两个属性来构建。

1. 固定布局

//HTML 部分

```

1. <header> header </header>
2.
3. <aside> aside </aside>
4.
5. <section> section </section>
6.
7. <footer> footer </footer>

```

//CSS 部分

```

1. body { width: 960px; margin: 0 auto; color: white;
2. } header { height: 120px;
3. } aside { width: 200px; height: 500px; float: left;
4. } section { width: 760px; height: 500px; float: right;
5. } footer { height: 120px; clear: both;
6. }

```

2.**流体布局**

流体布局只要更改 `body` 元素的限定长度为 `auto` 或 `100%`。然后左右两列分别设置 `20%` 和 `80%`即可。

//CSS 部分

```
1. body { width: auto;
2. } aside { width: 20%;
3. } section { width: 80%;
4. }
```

原文: <https://wizardforce1.gitbooks.io/liyanhui-tutorials/content/33.html>

第 27 章 CSS 传统布局[下]

第 27 章 CSS 传统布局[下]

学习要点：

1.定位布局

2.box-sizing

3.resize

主讲教师：李炎恢

本章主要探讨 HTML5 中 CSS 早期所使用的传统布局，很多情况下，这些布局方式还是非常有用的。

一．定位布局

在使用定位布局前，我们先了解一下定位属性的用法。CSS2 提供了 position 属性来实现元素的绝对定位和相对定位。

属性	说明
static	默认值，无定位。
absolute	绝对定位，使用 top、right、bottom、left进行位移。
relative	相对定位，使用 top、right、bottom、left进行位移。
fixed	以窗口参考定位，使用 top、right、bottom、left 进行位移。

//绝对定位，脱离文档流，以窗口文档左上角 0,0 为起点

```
1. header { position: absolute; top: 100px; left: 100px;
2. }
```

所谓脱离文档流的意思，就是本身这个元素在文档流是占位的。如果脱离了，就不占有文档的位置，好像浮在了空中一般，有了层次感。

由于绝对定位脱离了文档流，出现层次概念。那么每个元素到底在那一层，会不会冲突覆盖。这时通过 z-index 属性来判定它们的层次关系。

属性	说明
auto	默认层次
数字	设置层次，数字越大，层次越高

//设置在 100 层上

```
1. header { z-index: 100;
```

```
2. }
```

//以窗口参考定位，脱离文档流，会随着滚动条滚动而滚动

```
1. header { position: fixed; top: 100px; left: 100px;
2. }
```

//相对定位，不脱离文档流，占位偏移

```
1. header { position: relative; top: 100px; left: 100px;
2. }
```

这三种分别都在各自的情况下使用，均比较常用。但还有一种情况，就是：1.既要脱离文档流（这样元素之间不会相互冲突）；2.以父元素，比如 body 或其他父元素为参考点（这样可以实现区域性绝对定位）；3.还必须是绝对定位。

//第一步，将需要设置参考点的父元素设置为相对，且不设置坐标

```
1. body { position: relative;
2. }
```

//第二步，如果父元素设置了参考点，子元素的绝对定位将以它为基准

```
1. header { position: absolute; top: 0px; left: 0px;
2. }
```

1. **固定布局**

//CSS 部分

```
1. body { width: 960px; margin: 0 auto; position: relative;
2. } header { width: 960px; height: 120px; position: absolute; top: 0; left: 0;
3. } aside { width: 200px; height: 500px; position: absolute; top: 120px; left: 0;
4. } section { width: 760px; height: 500px; position: absolute; top: 120px; /*left: 200px;*/ right: 0;
5. } footer { width: 960px; height: 120px; position: absolute; top: 620px;
6. }
```

在上面，基本都用了定位来进行固定布局。但细心的可以发现，其实只有右侧需要实行绝对定位，其他就按照普通的摆放即可。对于设计成流体布局，只要将长度设置成百分比即可。

二. **box-sizing**

在盒模型那个章节，我们了解到元素盒子如果加入了内边距 padding 和边框 border 后，它的总长度会增加。那么如果这个元素用于非常精确的布局时，我们就需要进行计算增减。这其实是比较烦人的操作，尤其是动态设置页面布局的时候。

CSS3 提供了一个属性 box-sizing，这个属性可以定义元素盒子的解析方式，从而可以选择避免掉布局元素盒子增加内边距和边框的长度增减问题。

属性	说明
content-box	默认值, border 和 padding 设置后用于元素的总长度。
border-box	border 和 padding 设置后不用于元素的总长度。

//设置 border-box 让 border 和 padding 不在额外增加元素大小

```
1. aside { width: 200px; height: 500px; background-color: purple; padding: 10px; border: 5px solid red; box-sizing: border-box; float: left;
2. }
```

box-sizing 是 CSS3 推出的, 各个厂商在实现时设置了私有前缀。

	Opera	Firefox	Chrome	Safari	IE
支持需带前缀	无	2 ~ 28	4 ~ 9	3.1 ~ 5	8.0+
支持不带前缀	10.1+	29+	10+	6+	9.0+

//完整形式

```
1. -webkit-box-sizing: border-box;
2. -moz-box-sizing: border-box;
3. -ms-box-sizing: border-box;
4. box-sizing: border-box;
```

三. **resize**

CSS3 提供了一个 resize 属性, 来更改元素尺寸大小。

属性	说明
none	默认值, 不允许用户调整元素大小。
both	用户可以调节元素的宽度和高度。
horizontal	用户可以调节元素的宽度。
vertical	用户可以调节元素的高度。

一般普通元素, 默认值是不允许的。但如果是表单类的 textarea 元素, 默认是允许的。而普通元素需要设置 overflow:auto, 配合 resize 才会出现可拖拽的图形。

//允许修改

```
1. aside { resize: both; overflow: auto;
2. }
```

原文: <https://wizardforce1.gitbooks.io/liyanhui-tutorials/content/34.html>

第 28 章 CSS3 多列布局

第 28 章 CSS3 多列布局

学习要点：

1. 早期多列问题

2. 属性及版本

3. 属性解释

主讲教师：李炎恢

本章主要探讨 HTML5 中 CSS3 提供的多列布局，通过多列布局我们方便的创建流体的多列布局。

一．早期多列问题

我们有时想布局成报纸、杂志那样的多列方式（至少两列，一般三列以上），但早期 CSS 提供的布局方式都有着极大的限制。如果是固体布局，那么使用浮动或定位布局都可以完成。但对于流体的多列，比如三列以上，那只能使用浮动布局进行，而它又有极大的限制。因为它是区块性的，对于动态的内容无法伸缩自适应，基本无能力。

//五段内容，分为三列

```
1. <div>
2.     <p> 我是第一段内容...省略的部分复制大量文本 </p>
3.     <p> 我是第二段内容...省略的部分复制大量文本 </p>
4.     <p> 我是第三段内容...省略的部分复制大量文本 </p>
5.     <p> 我是第四段内容...省略的部分复制大量文本 </p>
6.     <p> 我是第五段内容...省略的部分复制大量文本 </p>
7. </div>
```

//带标题的五段内容，分为三列

```
1. <div>
2.     <h4>第一段标题</h4>
3.     <p> 我是第一段内容...省略的部分复制大量文本 </p>
4.     <h4>第二段标题</h4>
5.     <p> 我是第二段内容...省略的部分复制大量文本 </p>
6.     <h4>第三段标题</h4>
7.     <p> 我是第三段内容...省略的部分复制大量文本 </p>
8.     <h4>第四段标题</h4>
9.     <p> 我是第四段内容...省略的部分复制大量文本 </p>
10.    <h4>第五段标题</h4>
11.    <p> 我是第五段内容...省略的部分复制大量文本 </p>
12. </div>
```

上门两组内容，如果想用浮动和定位实现流体三列，基本不可能。并且实际应用中，需求可能多变，要更改成四列或

者五列呢？所以，CSS3 提供了多列布局属性 `columns` 来实现这个动态变换的功能。

二. 属性及版本

CSS3 提供了 `columns` 多列布局属性等 7 个属性集合，具体如下：

属性	说明
<code>columns</code>	集成 <code>column-width</code> 和 <code>column-count</code> 两个属性
<code>column-width</code>	定义每列列宽度
<code>column-count</code>	定义分分列列数
<code>column-gap</code>	定义列间距
<code>column-rule</code>	定义列边框
<code>column-span</code>	定义多列布局中子元素跨列效果，firefox 不支持
<code>column-fill</code>	控制每列的列高效果，主流浏览器不支持

由于 `column` 属性集尚未纳入标准，大多数浏览器必须使用厂商前缀才能访问，好在主流的浏览器都可以很好的支持了。下面是主流浏览器的支持和前缀情况。

	Opera	Firefox	Chrome	Safari	IE
支持需带前缀	11.5 ~ 29	2 ~ 40	4 ~ 45	3.1 ~ 8	无
支持不带前缀	无	无	无	无	10.0+

通过上面的表格，我们可以了解到，要想让最新的浏览器全部实现效果，都必须使用前缀。

//完整形式

```
1. -webkit-columns: 150px 4;
2. -moz-columns: 150px 4;
3. columns: border-box;
```

三. 属性解释

为了方便演示，我们在 Firefox 火狐浏览器测试，只用 `-moz-` 前缀演示。

1.columns

`columns` 是一个复合属性，包含 `columns-width` 和 `columns-count` 这两种简写。意为同时设置分列列数和分列宽度。//分成四列，每列宽度自适应

```
1. -moz-columns: auto 4;
```

2.column-width

`column-width` 属性，用于设置每列的宽度。

//设置列宽

```
1. -moz-column-width: 200px;
```

这里设置了 200px，有点最小宽度的意思。当浏览器缩放到小于 200 大小时，将变成 1 列显示。而如果是 auto，则一直保持四列。

属性	说明
auto	默认值，自适应。
长度值	设置列宽。

3.column-count

column-count 属性，用于设置多少列。

//设置列数

```
1. -moz-column-count: 4;
```

属性值	说明
auto	默认值，表示就 1 列。
数值	设置列数。

4.column-gap

column-gap 属性，用于设置列间距

//设置列间距

```
1. -moz-column-gap: 100px;
```

属性值	说明
auto	默认值，表示就 1 列。
数值	设置列数。

5.column-rule

column-rule 属性，设置每列中间的分割线//设置列边线

```
1. -moz-column-rule: 2px dashed gray;
```

属性	说明
column-rule	<宽度> <样式> <颜色>的边框简写形式。
column-rule-width	单独设置边框宽度。
column-rule-style	单独设置边框的样式。
column-rule-color	单独设置边框的颜色。

列边线不会影响到布局，它会根据布局的缩放自我调整是否显示。如果我们把页面缩放到只能显示一列时，它自动消失了。

6.column-span

column-span 属性，设置跨列大标题。

//跨列标题，由于火狐尚未支持，固使用 webkit

```
1. -webkit-column-span: all;
```

属性	说明
none	默认值，表示不跨列。
all	表示跨列。

原文: <https://wizardforce1.gitbooks.io/liyanhui-tutorials/content/35.html>

第 29 章 CSS3 弹性伸缩布局[上]

第 29 章 CSS3 弹性伸缩布局[上]

学习要点：

1. 布局简介

2. 旧版本

主讲教师：李炎恢

本章主要探讨 HTML5 中 CSS3 提供的用来实现未来响应式弹性伸缩布局方案，这里做一个初步的了解。

一. 布局简介

CSS3 提供一种崭新的布局方式：Flexbox 布局，即弹性伸缩布局模型(Flexible Box)。用来提供一个更加有效的方式实现响应式布局。但是用于这个布局方式还处于 W3C 的草案阶段，并且它还分为旧版本、新版本以及混合过渡版本三种不同的编码方式。在发展中，可能还有各种改动，浏览器的兼容性还存在问题。所以，本节课作为初步了解即可。

首先，我们来看下旧版本的浏览器兼容情况：

Flexbox 旧版本兼容情况

属性	IE	Firefox	Chrome	Opera	Safari
带前缀	无	4 ~ 25	4 ~ 31	15 ~ 18	5.17+
不带前缀	无	无	无	无	无

以上的数据，我们摘自 CSS3 手册上的。当然，不同的教材和文章的会略有不同。但误差率也就一到两个版本，影响不大。

首先，第一步：先创建一段内容，分成三个区域。

//HTML 部分

```
1. <div>
2.     <p> 第一段内容... </p>
3.     <p> 第二段内容... </p>
4.     <p> 第三段内容... </p>
5. </div>
```

//CSS 部分

```
1. p { width: 150px; border: 1px solid gray; margin: 5px; padding: 5px;
2. } div { display: -moz-box; display: -webkit-box; display: box;
3. }
```


通过以上设置，在除了 IE 浏览器外，布局实现了水平分布。那么下面，我们就重点研究一下这些属性的含义。

二. 旧版本

如果属性和属性值为：display:box，那么就是 2009 年 7 月份设定的工作草案，属于旧版本。它面向的是一些早期浏览器的弹性布局方案。

首先，我们要将几个需要布局模块的父元素设置一下容器属性 display。

属性值	说明
box	将容器盒模型作为块级弹性伸缩盒显示（旧版本）
inline-box	将容器盒模型作为内联级弹性伸缩盒显示（旧版本）

我们知道块级它是占用整行的，类似<div>元素；而内联级不占用整行，类似元素。但是我们设置了整个盒子，他们都不占用，保持一致。

//设置弹性，以火狐为例

```
1. div { display: -moz-box;
2. }
```

1.box-orient**属性**

box-orient 主要实现盒子内部元素的流动方向。

//设置垂直流动

```
1. div { -webkit-box-orient: vertical;
2. }
```

属性值	说明
horizontal	伸缩项目从左到右水平排列
vertical	伸缩项目从上到下垂直排列
inline-axis	伸缩项目沿着内联轴排列显示
block-axis	伸缩项目沿着块轴排练显示

2.box-direction

box-direction 属性主要是设置伸缩容器中的流动顺序。

//设置逆序

```
1. div { -moz-box-direction: reverse;
2. }
```

属性值	说明
-----	----

normal	默认值，正常顺序
reverse	逆序

3.box-pack

box-pack 属性用于伸缩项目的分布方式。

//分布方式已结束位置靠齐

```
1. div { -moz-box-pack: end;
2. }
```

属性值	说明
start	伸缩项目以起始点靠齐
end	伸缩项目以结束点靠齐
center	伸缩项目以中心点靠齐
justify	伸缩项目平局分布，-webkit-支持，-moz-不支持

4.box-align

box-align 属性用来处理伸缩容器的额外空间。//居中对齐，清理上下额外空间

```
1. div { -moz-box-align: center;
2. }
```

属性值	说明
start	伸缩项目以顶部为基准，清理下部额外空间
end	伸缩项目以底部为基准，清理上部额外空间
center	伸缩项目以中部为基准，平均清理上下部额外空间
baseline	伸缩项目以基线为基准，清理额外的空间
stretch	伸缩项目填充整个容器，默认

5.box-flex

box-flex 属性可以使用浮点数分配伸缩项目的比例//设置每个伸缩项目占用的比例

```
1. p:nth-child(1) { -moz-box-flex: 1;
2. } p:nth-child(2) { -moz-box-flex: 2.5;
3. } p:nth-child(3) { -moz-box-flex: 1;
4. }
```

6.box-ordinal-group

box-ordinal-group 属性可以设置伸缩项目的显示位置。

//将第一个位置的元素，跳转到第三个位置

```
1. p:nth-child(1) { -moz-box-ordinal-group: 3;  
2. }
```

原文: <https://wizardforce1.gitbooks.io/liyanhui-tutorials/content/36.html>

第 29 章 CSS3 弹性伸缩布局[中]

第 29 章 CSS3 弹性伸缩布局[中]

学习要点：

1. 混合过度版

主讲教师：李炎恢

本章主要探讨 HTML5 中 CSS3 提供的用来实现未来响应式弹性伸缩布局方案，这里做一个初步的了解。

一. 混合过渡版

混合版本的 Flexbox 模型是 2011 年提出的工作草案，主要是针对 IE10 浏览器实现的伸缩布局效果，其功能和旧版本的属性大同小异。我们还是采用上一节课的文件，然后使用混合过渡代码，实现 IE10 的伸缩布局。

首先，设置伸缩盒的 display 有如下两个属性值：

属性值	说明
flexbox	将容器盒模型作为块级弹性伸缩盒显示（混合版本）
inline-flexbox	将容器盒模型作为内联级弹性伸缩盒显示（混合版本）

//需要 IE 前缀-ms-

```
1. div { display: -ms-flexbox;
2. }
```

1.flex-direction

flex-direction 属性和旧版本 box-orient 属性一样，都是设置伸缩项目的排列方式。

//设置从上往下排列

```
1. div { -ms-flex-direction: column;
2. }
```

属性值	说明
row	设置从左到右排列
row-reverse	设置从右到左排列
column	设置从上到下排列
column-reverse	设置从下到上排列

2.flex-wrap

flex-wrap 属性类似与旧版本中的 box-lines，但是 box-lines 我们没有讲解，原因是没有浏览器支持它。

//设置无法容纳时，自动换行

```
1. div { -ms-flex-wrap: wrap;
2. }
```

属性值	说明
nowrap	默认值，都在一行或一列显示
wrap	伸缩项目无法容纳时，自动换行
wrap-reverse	伸缩项目无法容纳时，自动换行，方向和 wrap 相反

3.flex-flow

flex-flow 属性是集合了排列方向和控制换行的简写形式。

//简写形式

```
1. div { -ms-flex-flow: row wrap;
2. }
```

4.flex-pack

flex-pack 属性和旧版本中的 box-pack 一样，设置伸缩项目的对其方式。

//按照中心点对齐

```
1. div { -ms-flex-pack: center;
2. }
```

属性值	说明
start	伸缩项目以起始点靠齐
end	伸缩项目以结束点靠齐
center	伸缩项目以中心点靠齐
justify	伸缩项目平局分布

5.flex-align

flex-align 属性和旧版本中的 box-align 一样，处理伸缩项目容器的额外空间。

//处理额外空间

```
1. div { -ms-flex-align: center;
2. }
```

属性值	说明
-----	----

start	伸缩项目以顶部为基准，清理下部额外空间
end	伸缩项目以底部为基准，清理上部额外空间
center	伸缩项目以中部为基准，平均清理上下部额外空间
baseline	伸缩项目以基线为基准，清理额外的空间
stretch	伸缩项目填充整个容器，默认

6.flex

flex 属性和旧版本中的 box-flex 类似，用来控制伸缩容器的比例分配。

//设置比例分配

```
1. p:nth-child(1) { -ms-flex: 1;  
2. } p:nth-child(2) { -ms-flex: 3;  
3. } p:nth-child(3) { -ms-flex: 1;  
4. }
```

7.flex-order

flex-order 属性和 box-ordinal-group 属性一样控制伸缩项目出现的顺序。

//设置伸缩项目顺序

```
1. p:nth-child(1) { -ms-flex-order: 2;  
2. } p:nth-child(2) { -ms-flex-order: 3;  
3. } p:nth-child(3) { -ms-flex-order: 1;  
4. }
```

原文: <https://wizardforcel.gitbooks.io/liyanhui-tutorials/content/37.html>

第 29 章 CSS3 弹性伸缩布局[下]

第 29 章 CSS3 弹性伸缩布局[下]

学习要点：

1. 新版本

主讲教师：李炎恢

本章主要探讨 HTML5 中 CSS3 提供的用来实现未来响应式弹性伸缩布局方案，这里做一个初步的了解。

一. 新版本

新版本的 Flexbox 模型是 2012 年 9 月提出的工作草案，这个草案是由 W3C 推出的最新语法。这个版本立志于指定标准，让新式的浏览器全面兼容，在未来浏览器的更新换代中实现了统一。

首先，设置伸缩盒的 display 有如下两个属性值：

属性值	说明
flex	将容器盒模型作为块级弹性伸缩盒显示（新版本）
inline-flex	将容器盒模型作为内联级弹性伸缩盒显示（新版本）

//大部分不需要前缀

```
1. div { display: flex;
2. }
```

属性	IE	Firefox	Chrome	Opera	Safari
带前缀	无	无	21 ~ 28	无	7.0
不带前缀	11+	20+	29+	12.1	无

从这张表可以看出，只有 webkit 引擎的浏览器 Chrome 和 Safari 的版本需要添加 -webkit-，而 Chrome 浏览器 29 版本以后可以省略了。

1.flex-direction

flex-direction 属性和旧版本 box-orient 属性一样，都是设置伸缩项目的排列方式。

//设置从上往下排列

```
1. div { flex-direction: column;
2. }
```

属性值	说明

row	设置从左到右排列
row-reverse	设置从右到左排列
column	设置从上到下排列
column-reverse	设置从下到上排列

2.flex-wrap

flex-wrap 属性类似与旧版本中的 box-lines，但是 box-lines 我们没有讲解，原因是没有浏览器支持它。

//设置无法容纳时，自动换行

```
1. div { -ms-flex-wrap: wrap;
2. }
```

属性值	说明
nowrap	默认值，都在一行或一列显示
wrap	伸缩项目无法容纳时，自动换行
wrap-reverse	伸缩项目无法容纳时，自动换行，方向和 wrap 相反

3.flex-flow

flex-flow 属性是集合了排列方向和控制换行的简写形式。

//简写形式

```
1. div { flex-flow: row wrap;
2. }
```

4.justify-content

justify-content 属性和旧版本中的 box-pack 一样，设置伸缩项目的对其方式。

//按照中心点对齐

```
1. div { justify-content: space-around;
2. }
```

属性值	说明
flex-start	伸缩项目以起始点靠齐
flex-end	伸缩项目以结束点靠齐
center	伸缩项目以中心点靠齐
space-between	伸缩项目平局分布
space-around	同上，但两端保留一半的空间

5.align-items

`align-items` 属性和旧版本中的 `box-align` 一样，处理伸缩项目容器的额外空间。

//处理额外空间

```
1. div { align-items: center;
2. }
```

属性值	说明
<code>flex-start</code>	伸缩项目以顶部为基准，清理下部额外空间
<code>flex-end</code>	伸缩项目以底部为基准，清理上部额外空间
<code>center</code>	伸缩项目以中部为基准，平均清理上下部额外空间
<code>baseline</code>	伸缩项目以基线为基准，清理额外的空间
<code>stretch</code>	伸缩项目填充整个容器，默认

6.align-self

`align-self` 和 `align-items` 一样，都是清理额外空间，但它是单独设置某一个伸缩项目的。所有的值和 `align-items` 一致。

//单独设置清理额外空间

```
1. p:nth-child(2) { align-self: center;
2. }
```

7.flex

`flex` 属性和旧版本中的 `box-flex` 类似，用来控制伸缩容器的比例分配。

//设置比例分配

```
1. p:nth-child(1) { flex: 1;
2. } p:nth-child(2) { flex: 3;
3. } p:nth-child(3) { flex: 1;
4. }
```

8.order

`order` 属性和 `box-ordinal-group` 属性一样控制伸缩项目出现的顺序。

//设置伸缩项目顺序

```
1. p:nth-child(1) { order: 2;
2. } p:nth-child(2) { order: 3;
3. } p:nth-child(3) { order: 1;
4. }
```

原文: <https://wizardforce1.gitbooks.io/liyanhui-tutorials/content/38.html>

第 30 章 使用 Emmet 插件

第 30 章 使用 Emmet 插件

学习要点：

1. 安装方式
2. 自定义！生成
3. 快速生成

主讲教师：李炎恢

本章主要探讨了解一下 Sublime Text3 的一个 HTML5 代码提示插件：Emmet，这个插件比自带原生的要强大许多。

一．安装方式

Emmet 插件安装一般采用两种方式，1.通过命令安装；2.下载离线安装。我这里直接采用的是下载离线安装方式。具体步骤：

- 1.解压下载好的 Emmet 插件包(这里会提供)；
- 2.将 Emmet 和 PyV8 两个文件夹复制到 Sublime Text3 的程序包中；
- 3.左下角会显示自动安装，安装好后，重启 Sublime Text3；4.在编辑器输入英文状态下的“!”，然后 ctrl+e，出现了 HTML5 的代码库，则安装

成功。

注：如果安装失败或出现其他错误，请自行百度选用其他方式安装，或解决安装出错的问题。

二．自定义！生成

我们输入!，然后 ctrl+e，默认情况下会出现如下代码：

//默认代码

```
1. <!doctype html>
2. <html lang="en">
3.     <head>
4.         <meta charset="UTF-8">
5.         <title>Document</title>
6.     </head>
7.     <body>
8.
9.     </body>
10. </html>
```

这里有两个地方和我们之前生成的代码不一样，第一处是：doctype 没有大写；第二处：lang 是 en 的。其实这两处不改也没有太大关系，但有强迫症的看了可能会难受。具体修改方法如下：

1. 首先，进入程序包 packages；
2. 其次，进入 Emmet 文件夹，再进入 emmet 文件夹，找到 snippets.json 文件；
3. 最后，打开这个文件，找到相应处修改即可。

三. 快速生成

Emmet 提供了非常丰富的 HTML 和 CSS 代码的快速生成功能，通过使用快速生成代码，极大的增加了开发速度。只不过，Emmet 提供的生成方式需要二次学习，起初可能还不如手工敲击的快。所以，需要一定时间的学习磨合。

//快速生成 HTML5 代码结构

! + (ctrl + e 或 tab 键) 或 html:5 + tab 键

```
1. <!DOCTYPE html>
2. <html lang="zh-cn">
3.   <head>
4.     <meta charset="UTF-8">
5.     <title>Document</title>
6.   </head>
7.   <body>
8.
9.   </body>
10. </html>
```

所有代码生成，都需要通过 tab 键来生成代码，后面不在赘述。

//快速生成标签代码 a

```
1. <a href=""></a>
```

//快速生成标签相应的属性值a:link、a:mail

```
1. <a href="http://"></a>
2.
3. &lt;a href="mailto:"&gt;&lt;/a&gt;
```

//生成标签内的值a{超链接}

```
1. <a href="">超链接</a>
```

//生成 CSS 链接 link link

```
1. <link rel="stylesheet" href="">
```

//生成表单控件input、input:hidden

```
1. <input type="text">
2. <input type="hidden" name="">
```

//生成带子标签的一组标签ul+、ol+、dl+、table+

```
1. <ul>
2.     <li></li>
3. </ul>
4.
5. <ol>
6.     <li></li>
7. </ol>
8.
9. <dl>
10.    <dt></dt>
11.    <dd></dd>
12. </dl>
13.
14. <table>
15.    <tr>
16.        <td></td>
17.    </tr>
18. </table>
```

//生成嵌套子标签 nav>ul>li

```
1. <nav>
2.     <ul>
3.         <li></li>
4.     </ul>
5. </nav>
```

//生成相邻兄弟标签 div+p+h1

```
1. <div></div>
2.
3. <p></p>
4.
5. <h1></h1>
```

//生成乘积数量的标签 ul>li*5

```
1. <ul>
2.     <li></li>
3.
4.     <li></li>
5.
```

```

6.     <li></li>
7.
8.     <li></li>
9.
10.    <li></li>
11. </ul>

```

//创建具有 ID 的标签 div#header

```
1. <div id="header"></div>
```

//创建具有 class 的标签 div.header、div.header.sidebar

```

1. <div class="header"></div>
2.
3. <div class="header sidebar"></div>

```

以上是 HTML 部分的代码生成功能，下面来看下 CSS 的快速生成功能：

//生成 position: relative pos

输入 pos 即可出现 position:relative 这组 CSS 样式，并且 relative 是选定状态，有助于你更换属性值。

但是我们发现使用 sublime 结合 Emmet 插件的 CSS 提示非常的灵活，不会死板的必须要输入 pos。下面的输入都可以得到相应的值：

po = position: relative

只要输入 po 或者大于 po 字母量的值，都可以得到 position:relative。当然，如果你只是输入 p，那么由于优先级的考虑，出现的是 padding: |。

如果你输入有误，它也会自动纠错，比如下面这个：

pod = position: relative

如果你想一开始得到的是 absolute 这个属性值，那么直接输入：

poa = position: absolute;

当然，上们的标准写法是这样的：

pos:a = position: absolute;

如果想输入背景的属性，直接使用 bg 即可：

bg = background: |;

使用 bg+可以展开背景属性的完整形式：bg+ = background: #fff url() 0 0 no-repeat;

使用 bg:n 可以设置背景属性值为 none：bg:n = background: none;

备注：HTML 和 CSS 其他大部分代码生成方案，可以参考如下网址：

<http://docs.emmet.io/cheat-sheet/>

原文：<https://wizardforce1.gitbooks.io/liyanhui-tutorials/content/39.html>