

提示 一个页面可以加载多个 JavaScript 文件，可以包含多个嵌入的脚本（参见图 19.2.1）。在默认情况下，浏览器会按照脚本在 HTML 中出现的顺序对它们进行加载（如果需要的话）和执行。关于为什么需要尽可能地避免加载多个脚本以及如何简化脚本（参见图 19.1.4），参见补充材料“脚本和性能的最佳实践”。

```
(function(e,c){var d=c.getElementById
→("choices"),b=c.getElementById("other-
→description");d&&b&&(b.disabled=!0,d.
→onclick=function(a){a||(a=e.event);a=
→a.target||a.srcElement;"radio"===
→a.getAttribute("type")&&("other"!==
→a.id?b.disabled=!0:(b.disabled=!1,
→b.focus()))))})(window,document);
```

图 19.1.4 是的，这段代码与图 19.1.3 的代码是一样的，只是被压缩过了。它看起来就像是一只猫在键盘上跑过留下的字符，但实际上浏览器可以理解它们

提示 可以为外部脚本指定任意有效的文件名，只要它们以 .js 结尾。通常，压缩后的脚本以 .min.js 作为扩展名，这样可以很清晰地将它们与常规文件区分开来。两个版本的文件都要保留——在常规文件中更新代码（因为代码更易于阅读），但在网站上使用的则是压缩后的版本（因为对浏览器来说能更快地解析）。不要忘了每次更新常规文件后都要生成一个新的压缩文件，否则，访问者看到的仍是旧的版本。

提示 如果创建了一个简化文件，一定要确保在 HTML 中对脚本的引用是正确的，例如，`<script src="behavior.min.js">`。否则，页面仍会加载常规的文件，访问者不会获得更小的文件带来的任何好处。在不断修改脚本的过程中，也可以让 src 重新指向常规的文件。

提示 不理解 JavaScript 的浏览器（数量很少）或用户禁用了 JavaScript 的浏览器会忽略 JavaScript 文件。因此要确保页面不依赖于 JavaScript 为用户提供对内容的可访问性和基本体验。（高度依赖 JavaScript 的 Web 应用程序通常例外。）

提示 从技术上说，为页面添加 JavaScript 还有第三种方式：内联脚本。内联脚本是直接放在 HTML 中特定元素属性上指定的 JavaScript 片段。在这里提到这种方式是为了告诉你，应该避免使用它们，就像避免使用内联样式表一样。正如内联样式表将 HTML 和 CSS 混合在一起，内联脚本则将 HTML 和 JavaScript 纠缠在一起，这与将它们分开的最佳实践不符。

脚本和性能的最佳实践

关于脚本和页面性能最佳实践的详细讨论超出了本书的范围，不过这里仍会提到几个影响较大的要点。

最重要的一点，理解浏览器如何处理脚本是很有用的。随着页面的加载，在默认情况下，浏览器会按照脚本在 HTML 中出现的顺序依次对每个脚本进行下载（对于外部脚本）、解析和执行。在处理脚本的过程中，浏览器既不会下载该 script 元素后面出现的内容（哪怕是文本），也不会呈现这些内容。这称为阻塞行为（blocking behavior）。

这条规则对嵌入脚本和外部脚本都有效。可以想象，这会影响页面的呈现速率，影响的程度取决于脚本的大小和它执行的动作。

大多数浏览器都会这样做，因为 JavaScript 可能包含其他脚本所依赖的代码、立即生成内容的代码或对页面进行调整的代码。浏览器需要在结束呈现页面之前考虑所有这些问题。

那么如何避免这一点呢？消除 JavaScript 阻塞最简单的方法就是将所有的 script 元素放置在 HTML 结束之前，即 `</body>` 结束标签的前面，参见图 19.1.1。

随便花点儿时间浏览一下其他人的网站，不难发现脚本是在 head 元素中加载的。除了少数必须这样做的情况，通常认为这是一种过时的做法，应当尽可能地避免。（必须这样做的一种情况就是加载第 11 章所述的 HTML5 剃刀。）如果确实需要在 head 中加载脚本，也要将它们放置在所有加载 CSS 文件的 link 元素之后（这也是出于性能的考虑）。

另一种简单的加快脚本加载速率的方法就是将 JavaScript 放在同一个文件中（或尽可能少的一些文件中）并压缩代码。通常，压缩代码会去除换行、注释及额外的空格（以及其他未压缩的代码存在差异的地方）。可以想象一下只在一个很长的行内编写代码，而永远不敲回车键。看看图 19.1.4 你就明白我说的是什么意思了。

可以使用以下工具压缩脚本（“供下载的版本及文档”链接主要是为高级用法提供的）：

❑ Google Closure Compiler

<http://code.google.com/closure/compiler/>（供下载的版本及文档）

<http://closure-compiler.appspot.com>（在线版本）

❑ UglifyJS（使用第二个链接）

<https://github.com/mishoo/UglifyJS2>（供下载的版本及文档）

<http://lisperator.net/uglifyjs/>（在线版本，选择“Open demo”）

❑ YUI Compressor（使用第二个链接）

<http://developer.yahoo.com/yui/compressor/>（供下载的版本及文档）

<http://refresh-sf.com/yui/>（非官方的在线版本）

上述每个工具都能减小文件的大小，而压缩的结果则因脚本而异。记住，有时压缩代码的工具效果过于强烈，可能会损坏你的脚本。因此，如果页面包括压缩脚本，一定要进行测试。通常，浏览器加载一个文件比加载两个文件（或更多的文件）要快一些，即便单个文件的大小比不同文件的总大小还要大一些（除非大得多）。

这是两种常用且强大的方法，但真正说起来，它们也只是一些皮毛的东西。关于脚本加载的方法及其优化，强烈推荐 Steve Souders 所著的 *Even Faster Web Sites*^①（O'Reilly Media，2009）一书，以及他的网站 www.stevesouders.com。不过有言在先，Souders 的某些讨论相对技术一些。

① 中文版《高性能网站建设进阶指南》已由电子工业出版社出版。——译者注

19.2 添加嵌入脚本

嵌入脚本位于 HTML 文档之内，同嵌入样式表很相似。嵌入的脚本包含在 `script` 元素内，如图 19.2.1 所示。嵌入脚本并不是首选的方法（参见 19.1 节），但有时必须这样做。

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="utf-8" />
  <title>Adding an Embedded Script</title>
  <link rel="stylesheet"
    → href="css/global.css" />
</head>
<body>
... 所有HTML内容写在这里 ...

<script>
/*
JavaScript 代码写在这里
*/
</script>
</body>
</html>
```

图 19.2.1 嵌入脚本没有 `src` 属性。相反，代码位于页面之内。如果要嵌入脚本，应该尽可能地在 `</body>` 结束标签之前这样做。虽然也可以在 `head` 中嵌入脚本（如图 19.2.2 所示），但通常出于性能原因而避免这样做

添加嵌入脚本的步骤

- (1) 在 HTML 文档中，输入 `<script>`。
- (2) 输入脚本的内容。
- (3) 输入 `</script>`。

提示 每个脚本都按照它们出现在 HTML 中的顺序依次进行处理，无论是嵌入脚本还是外部脚本（参见 19.1 节）。

提示 尽管 `script` 元素必须有结束标签（`</script>`），但在有 `src` 属性的情况下，不能在该元素的开始标签和结束标签之间嵌入脚本（参见 19.1 节）。也就是说，`<script src="your-functions.js">` 此处输入其他功能 `</script>` 是无效的。对于任何 `script` 元素，要么仅通过 `src` 加载外部脚本，要么嵌入脚本并不含 `src`。

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="utf-8" />
  <title>Loading an External Script</title>
  <!-- 在加载任何JS文件之前加载样式表 -->
  <link rel="stylesheet"
    → href="global.css" />

  <script>
/*
JavaScript代码写在这里
*/
</script>
</head>
<body>
... 所有的HTML内容写在这里 ...
</body>
</html>
```

图 19.2.2 在这个例子中，嵌入脚本位于 `head` 中。它位于 `link` 元素的后面，从而让 CSS 文件更快地加载。关于为什么要尽可能地避免在 `head` 中嵌入脚本，参见 19.1 节的补充材料“脚本和性能的最佳实践”

19.3 JavaScript 事件

在本章的引言中，我提到对 JavaScript 的深入讨论超出了本书的范围。不过，我仍然想让你大体了解一下 JavaScript 事件，从而对 JavaScript 能做什么有一些基本的认识。

可以编写 JavaScript 对特定的、预定义的事件（由访问者或浏览器触发）进行响应。下面的列表只是编写脚本时可用的事件处理程序的一些简单示例。HTML5 引入了大量其他的事件处理程序，其中很多都是与 audio 和 video 元素相关的。有的触屏设备也开始支持特殊的基于触摸操作的事件处理程序。

注意下面的列表中的“mouse”（鼠标）代表所有“指针设备”。例如，onmousedown 会在访问者使用电子笔、真正的鼠标或其他类似的设备时被触发。

- ❑ onblur: 访问者离开先前获得焦点的元素（参见 onfocus）。
- ❑ onchange: 访问者改变元素的值或内容。通常用于表单字段（关于表单，参见第 16 章）。
- ❑ onclick: 访问者点击特定的区域或在元素（如链接）获得焦点时按下回车键。
- ❑ ondblclick: 访问者双击特定的区域。
- ❑ onfocus: 访问者选择、点击或用制表键将焦点移至特定的元素。
- ❑ onkeydown: 在指定的元素上，访问者按下一个键。
- ❑ onkeypress: 在指定的元素上，访问者按下并松开一个键。

- ❑ onkeyup: 在指定的元素上，访问者在输入后松开一个键。
- ❑ onload: 浏览器完成页面的加载，包括所有的外部文件（如图像、样式表、JavaScript 等）。
- ❑ onmousedown: 在指定的元素上，访问者按下鼠标键。
- ❑ onmousemove: 访问者移动鼠标指针。
- ❑ onmouseout: 访问者在鼠标指针停留的特定元素上移开鼠标。
- ❑ onmouseover: 访问者将鼠标指向元素。
- ❑ onmouseup: 访问者在点击元素后松开鼠标键（与 onmousedown 相反）。
- ❑ onreset: 访问者点击表单的重置按钮或在该按钮获得焦点时按下回车键。
- ❑ onselect: 访问者选择元素中的一个或多个字符。
- ❑ onsubmit: 访问者点击表单的提交按钮或在该按钮获得焦点时按下回车键。

HTML5 事件处理程序的完整列表参见 <http://dev.w3.org/html5/spec-author-view/global-attributes.html>。一些触屏设备（如智能手机、平板电脑）包含的与触摸相关的事件处理程序包括 touchstart、touchend、touchmove 等（www.w3.org/TR/touch-events/）。

本章内容

- 验证代码
- 测试页面
- 尝试一些调试技巧
- 检查常见错误：一般问题
- 检查常见错误：HTML
- 检查常见错误：CSS
- 如果图像不显示

你编写好了全新的页面，却发现它们在浏览器中并没有像你预期的那样显示，或者完全不显示，或者在你的默认浏览器中显示得很好，但客户使用其他的浏览器打开时却发现页面显得有些奇怪。

在 HTML、CSS 和众多的浏览器（尤其是旧浏览器）和平台之间，很容易产生各种各样的问题。本章会提示一些常见的错误，并帮助你清除自己造成的错误。

这些调试技术中的一部分看起来相当基础，但网页的问题往往也非常基础。在查找大的问题之前，要确保没有任何小的问题。

无论如何，应该在一个或多个平台的几种浏览器中对网站进行充分的测试，查看各个页面是不是按照预期的方式工作（参见 20.2 节）。

20.1 验证代码

发现 HTML（如图 20.1.1 所示）和 CSS 中错误（如图 20.1.2 所示）的一种重要方法就是使用验证器，这样就不用你自己追踪了。HTML 验证器可以对代码和语言规则进行比较，并将其发现的不一致的情况显示为错误或警告。它还可以提示语法错误，无效的要素、属性和值，以及错误的元素嵌套（如图 20.1.3 所示）。它无法判断内容是不是由最能描述它的元素进行标记的，因此编写语义化的 HTML 还得靠你自己（参见 1.3 节）。

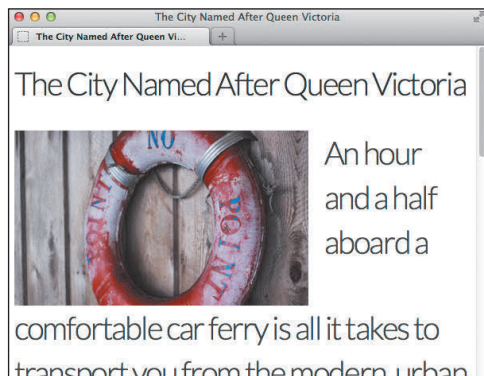


图 20.1.1 将需要检查的 URL 粘贴到 Address（地址）字段。选择 Show Source（显示源代码）选项，这样 HTML 源代码就会出现在验证器找到的错误的下方，有错误的 HTML 片段会突出显示

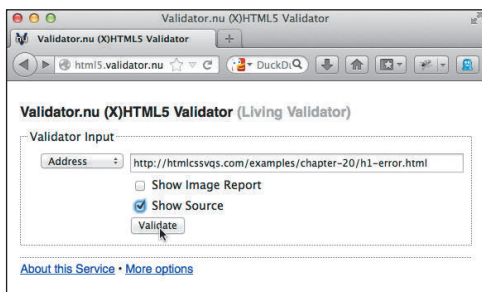


图 20.1.2 肯定哪个地方出了问题，标题下面的文本本不该这么大。我已经检查了 CSS，问题不是出于设置了很大的 font-size 值。问题出在哪儿呢

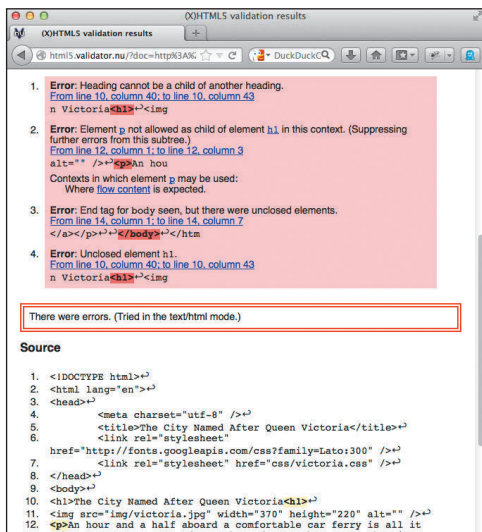


图 20.1.3 在第 10 行发现错误，原因是没有使用 `</h1>` 结束标签，错误地加上了另一个 `<h1>` 开始标签。其他的错误都是由第一个错误引起的，因此只要修复了这个错误，页面就没有任何错误了

将页面放到万维网上之前，不需要确保它们通过验证器的检查，完全没有错误。实际上，大多数网站都有一些错误。而且，W3C 的 CSS 验证器会将用于属性名称的厂商前缀标记为错误，但这并不意味着你应该将这些厂商前缀从样式表中移除（关于厂商前缀，参见第 14 章）。

浏览器可以处理很多类型的错误（同时

忽略一些其他的错误），从而以它们能实现的最佳方式将页面呈现出来。因此，即便页面在验证时有错误，也可能看不出来。不过，有时错误会直接影响页面的显示（如图 20.1.1 所示）或行为。因此，应该使用验证器尽可能地排除代码中的错误。

验证器能找到的错误的示例参见 20.5 节和 20.6 节。

验证代码的步骤

(1) 首先使用 `http://html5.validator.nu`（如图 20.1.2 和图 20.1.3 所示）或 W3C 的 `http://validator.w3.org` 对 HTML 进行检查。更多信息参见头两条提示。

(2) 修复标出来的 HTML 错误，保存修改。需要的话，将文件再次上传到服务器，再重复第 (1) 步。

(3) 可以使用 `http://jigsaw.w3.org/css-validator/` 检查 CSS 错误。类似地，修复你看到的错误，再重新检查页面。

检查 HTML 的一致性

对于 HTML 代码中某些部分的格式，HTML 是相当宽松的。例如，对于 `img` 这样的空元素，使用 `/>` 和 `>` 结尾都是合法的。HTML 验证器不会检查此类一致性问题。如果你希望代码一致，可以使用 HTML Lint（`http://lint.brihten.com/html/`）。通过它可以检查空元素是否闭合，开始和结束标签是否为小写字母，属性是否为小写字母等。

提示 W3C 的验证器（`http://validator.w3.org/`）使用的是 `http://html5.validator.nu/` 提供的验证引擎，因此使用这两种都可以。W3C 的错误消息更易读，但不会突出显示 HTML 源代码的错误部分。

提示 可以通过输入 URL（参见图 20.1.1）、上传 HTML 文件以及将 HTML 粘贴到验证器三种方式对 HTML 进行验证。如果使用上传或复制粘贴的方式，不必将文件上传到服务器就可以进行检查了。

提示 一个 HTML 错误可能导致多个验证器报错结果。例如，缺少一个结束标签会导致多条错误消息（参见图 20.1.3）。如果修复了这个结束标签的问题，所有这些后续错误就都不存在了（如图 20.1.4 所示）。因此，应该按照从上往下的顺序，一次修复少量错误后就立刻再次验证，看看其他的问题是否也已解决。

提示 如果样式表中包含第 11 章介绍的 .clearfix 样式规则，CSS 验证器会显示几个错误。这是由于使用了为旧版本的 Internet Explorer 准备的非标准的 CSS。你可以忽略这些错误，它们不会影响你的页面。

The document is valid HTML5 + ARIA + SVG 1.1 + MathML 2.0
(subject to the utter previewness of this service).

图 20.1.4 通过验证器的检查，HTML 是合法的



图 20.1.5 刷新页面后可以看到页面好看多了

20.2 测试页面

即便代码通过了验证，页面可能仍然不像预期的那样工作，如图 20.2.1 ~ 图 20.2.3 所示，或者可能在一个浏览器中是正常工作的，在另一个浏览器中却有问题。你并不知道用户使用哪种浏览器，因此在不同的浏览器上对页面进行测试是很重要的（参见本节末尾的补充材料“关于浏览器测试的更多信息”）。

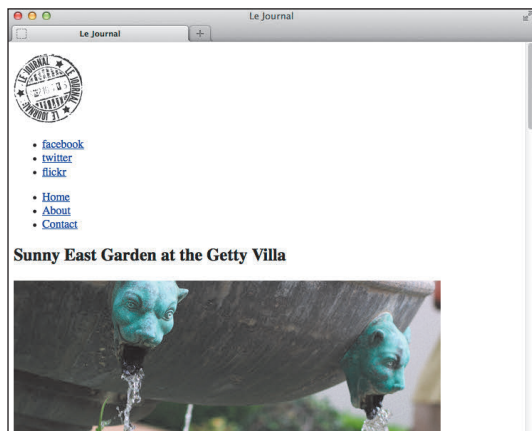


图 20.2.1 这个页面通过了验证，但它看上去并不像预期的那样。问题出在哪儿（参见图 20.6.2）

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="utf-8" />
  <title>Le Journal</title>
  <link rel="stylesheet"
    → href="css/style.css" />
  ...
</head>
<body>
  ...
</body>
</html>
```

图 20.2.2 问题出在指向 CSS 文件的链接上——CSS 文件的名称为 style.css，而这里指向的却是 style.css。浏览器无法找到 CSS，因此对页面的显示是有问题的



图 20.2.3 改正了代码中的文件名后，样式表就被加载进来了。现在页面可以正确显示了

测试 HTML 页面的步骤

(1) 对 HTML 和 CSS 进行验证（参见 20.1 节），作出必要的修改。

(2) 按照 2.7 节的解释打开页面。

(3) 检查整个页面，确保与你希望看到的完全一样。例如：

- ☐ 格式与期望的是否一致
- ☐ 链接的 URL 是否指向了正确的页面或资源（可以通过激活链接并查看结果对 URL 进行测试）
- ☐ 所有的图像都出现了吗？它们的位置和对齐方式是对的吗？
- ☐ 如果你检查的是响应式网页，网页布局是否能适应不同的屏幕尺寸

(4) 在不关闭浏览器中页面的情况下，打开有关的 HTML 或 CSS 文档，作出必要的改动。

(5) 保存修改。

(6) 切换到浏览器，刷新页面并查看所作的改动。

(7) 重复第 (3) 步至第 (6) 步，直到你对网页满意为止。有时候可能需要测试很多次才

能解决问题。如果尝试很多次后仍然失败了，请再次验证代码，确保没有引入新的错误。

(8) 从第 (2) 步开始，在其他的浏览器中执行同样的测试流程，直到满意并认为页面做好了发布准备为止。

(9) 将文件上传至服务器。

(10) 回到浏览器，在地址栏中输入页面的 URL，按下回车键。页面将出现在浏览器中。

(11) 再次对页面进行检查，确保没有任何问题（很容易忘记上传图像或者页面需要的其他文件）。同时，如果访问者可能通过移动设备访问网站的话，别忘了在移动设备上对其进行检查。

测试工作流

一种常见的测试工作流就是在构建页面的过程中定期在几个浏览器中对页面进行检查。当页面完成以后，再在全套浏览器中检查，对代码作必要的修改。完成整个网站之后，可能还需要在浏览器中再次进行测试，确保整个网站没有问题。

我建议首先对网站的本地版本进行测试和改进（参见“测试 HTML 页面的步骤”中的第 (1) ~ (8) 步），这是在文件上传到服务器上（参见第 (9) 步）之前进行的。将其上传到服务器上之后，再对它们做完整的检查，不过针对的是服务器上的文件。无论你对本地文件做了多少次检查，都不要跳过这一步，因为服务器上的网站才是访问者使用的网站（参见第 (10) ~ (11) 步）。

提示 如果浏览器中显示的是 HTML 代码而不是页面，要确保文件使用的是 .html 或 .htm 扩展名（而不是像 .txt 这样的扩展名）。

提示 有时候，页面上的问题并不是你的错，尤其是样式上的问题。在认为问题出在代码上之前，一定要确保浏览器对遇到问题的特性是支持的。附录 A 和附录 B 中有关于 HTML 和 CSS 特性的浏览器支持情况的资源链接。关于浏览器支持信息，Can I Use (<http://caniuse.com>) 和 Quirksmode (www.quirksmode.org/css/) 是两处非常宝贵的资源。

应该在哪些浏览器中进行测试

通常，大多数网站开发人士会在以下浏览器中对网站进行验证。

- ❑ Chrome (www.google.com/chrome) 的最新版本。Chrome 会在你的计算机上自动进行更新，大约每隔六周就会发布一个新的版本。
- ❑ Firefox (www.firefox.com) 的最新版。Firefox 同 Chrome 一样，更新和发布频率也非常快。
- ❑ Internet Explorer 8+，只针对 Windows。（IE8 的市场份额会越来越少，最终可能会从你的验证清单中消失。）可从 www.microsoft.com 下载 IE 的各个版本。
- ❑ Safari (www.apple.com/safari/) 的最新版，有时候是 Safari 5+。Safari 预装在 OS X 中。

至于 Opera，开发人员通常会选择测试它，Opera 在世界上很多地区都有少量的市场份额。Opera 的下载地址为 www.opera.com/。

好消息是，在不考虑 IE8 甚至 IE9 的情况下，这些浏览器对 HTML 和 CSS 特性的支持能力几乎在同一个水平。这意味着你通常很难看到不同浏览器之间的差异，除非你的页面用到了特别新的 HTML5 或 CSS3 特性。因为 IE8 已经很旧了，因此如果你的网站在 IE8 中和在现代浏览器中的样子稍有出入，也是可以接受的。

获取测试用的浏览器

可以使用上面提供的链接下载并安装这些浏览器，但如果你使用的是一台 Mac，该如何测试 IE 呢？如果你使用的是 Windows，如何测试 Safari 以及不同版本的 IE 呢？这里给出了一些方案。

- ❑ 虚拟机 (VM) 是运行在你的计算机上的一个独立的操作系统。微软为不同版本的 Windows 和 IE 提供了虚拟机，因此你可以在 Mac、Windows 或 Linux 机器上对 IE 进行测试。这些虚拟机位于 www.modern.ie/en-US/virtualizationtools#downloads。不能在一台 Windows 机器上运行 Mac 虚拟机以测试 Mac 浏览器。
- ❑ 使用 BrowserStack (www.browserstack.com) 和 Sauce Labs (<http://saucelabs.com>) 可以在很多浏览器和移动设备上免费测试你的页面。

在手机和平板电脑上测试

你可能至少希望在 iOS 和 Android 设备上进行测试。测试页面的移动兼容性是很有挑战的，因为通常很难在移动设备上对网站进行修改。解决的办法包括使用 BrowserStack 和 Sauce Labs（注意，使用模拟器并不一定与实际设备上的效果完全一致）。

- ❑ 使用苹果的 iOS Simulator 测试页面在 iPhone 和 iPad 上的情况。这种方法最大的局限性在于该模拟器只能在 OS X 上运行，在 Windows 上没有可替代的软件。iOS Simulator 是免费的 Xcode 的一部分，下载地址为 <http://developer.apple.com/xcode/>。
- ❑ DeviceAnywhere (www.deviceanywhere.com) 提供了一种在线访问各种不同移动设备进行测试的收费功能。DeviceAnywhere 也有免费版本，不过一次只能用 10 分钟。
- ❑ 使用 Electric Plum 针对 Windows 的 iPhone 和 iPad 模拟器 (www.electricplum.com)。这不是由苹果提供的，因此与苹果的 iOS Simulator 不一样。
- ❑ 使用为其他设备和移动浏览器准备的模拟器。Mobile Boilerplate 维护了一个列表，见 <https://github.com/h5bp/mobile-boilerplate/wiki/Mobile-Emulators-&-Simulators>。
- ❑ 查看 Open Device Lab (<http://opendevicelab.com>)，看看你身边是否有可供免费测试的设备。
- ❑ 如果你有设备，结合使用 Adobe Edge Inspect (<http://html.adobe.com/edge/inspect/>) 可以简化测试和修正问题的过程。

浏览器市场变化很快：当你阅读本书的时候，这些浏览器又会有一些新的版本发布，还会冒出一些新的设备。不过，只要遵循渐进增强原则，你的网站就可以为旧的浏览器提供简单的体验，为现代浏览器提供增强的体验。

20.3 尝试一些调试技巧

你现在应该已经做了一些测试，也发现了一些 bug。下面是用于排查网页错误的一些真实技巧。

- ❑ 首先检查常见的错误。
- ❑ 逐步开展工作。做出小的改动，并在每次改动后进行测试。这样才能在问题出现之后定位到问题的来源。
- ❑ 调试时，从你能确保正确的地方开始。之后再逐一添加可能出问题的部分，每次添加完以后都在浏览器中进行测试，直到找到问题的来源为止。
- ❑ 根据前面的要点，用排除法寻找产生问题的代码片段。例如，可以注释掉一半代码，检查问题是否出于另一半代码，如图 20.3.1 所示。再在有问题的代码中注释掉更小的部分，直到找出问题。（参见 3.16 节和 7.2 节。）

- ❑ 留意输入错误。很多令人费解的错误是由简单的输入错误造成的。例如，在 HTML 中以某种方式拼写类名，却在 CSS 中使用了另一个名称。
- ❑ 在 CSS 中，如果不确定问题是出在属性上还是出在选择器上，可以试着在选择器上添加极其简单的声明，如 `color: red;`、`border: 1px solid red;` 等（如果 red 已经用于网页的设计，也可以选择一种不常见的颜色，如 pink）。如果元素变成红色，那么问题就出在属性上，否则问题就出在选择器上（假定不存在另一个特殊性更强的选择器，也不存在比当前选择器更靠后的选择器）。
- ❑ 使用一个或多个开发工具，直接在浏览器中对 HTML 或 CSS 进行修改测试。或者使用这些工具审查代码，试着对问题进行定位。（参见补充材料“浏览器开发工具”。）

```

...

.excerpt {
    border-top: 1px dotted #ccc;
    margin: 0 .5em 2em 0;
}

.excerpt .title {
    font-size: 1.25em;
    line-height: 1.2;
}

/*
.more,
.excerpt .date {
    text-align: right;
}

.excerpt .date {
    line-height: 1;
    margin: 0 1em 0 0;
    padding: 0;
    position: relative;
    top: -1em;
}
*/

.photo {
    float: left;
    height: 300px;
    width: 400px;
}

...

```

图 20.3.1 我注释掉了这段代码中间的一部分（位于 `/*` 和 `*/` 之间），以查看它是不是出错的原因。注意，很多 HTML 和 CSS 编辑器都包含代码高亮，即自动为元素、选择器等设置不同的颜色。这对调试是有帮助的。例如，如果将 CSS 属性名称打错了，编辑器就不会以预期的颜色显示它，表明它是无效的

浏览器开发工具

浏览器都包含帮助你调试页面或者具有其他功能的工具。你会发现，一种反复用到的特性就是直接修改 CSS 或 HTML 并立即查看它们对页面的影响。这可以让你快速地对修改进行测试，满意之后再将它们包含到代码里去。

下面是各个浏览器中最常用的工具。

- ❑ Chrome: DevTools (<http://developers.google.com/chrome-developer-tools/>)。
- ❑ Firefox: Firefox 有内置工具 (<https://developer.mozilla.org/en-US/docs/Tools>)，但是 Firebug 插件特别流行 (<http://getfirebug.com>)。此外，Web Developer (<http://chrispederick.com/work/web-developer/>) 是稍有区别的一种工具，但也非常好用。在同一个链接的页面上，也可以下载用于 Chrome 的版本。
- ❑ Internet Explorer: F12 Developer Tools ([http://msdn.microsoft.com/en-us/library/hh772704\(v=vs.85\).aspx](http://msdn.microsoft.com/en-us/library/hh772704(v=vs.85).aspx))。
- ❑ Opera: Dragonfly (www.opera.com/dragonfly/)。本书写作之时，这个特性还没有成熟。
- ❑ Safari: Web Inspector (<http://developer.apple.com/technologies/safari/developer-tools.html>)。

网上还有演示如何使用这些工具的文档和视频。在 20.6 节有使用 Chrome DevTools 的示例。

20.4 检查常见错误：一般问题

浏览器之间的差异可能是由一些不明显的浏览器漏洞造成的，也可能是由于使用新技术造成的，但更常见的则是出于一些简单的问题。

每个人在从新手到专家的道路上难免会犯一些低级错误。例如，你有可能认为问题出在代码上，从而花了大量时间进行调试，最后才发现你修改的是一个文件，上传和查看的却是服务器上的另一个文件！

下列建议中的大多数适用于使用网站在服务器上的 URL 对网站进行测试的情形。

查检笼统的常见错误

- ❑ 根据 20.1 节的描述, 对代码进行验证。这是一个很好的起点, 因为这样就能排除代码语法相关的错误了。
- ❑ 确保已上传要测试的文件。
- ❑ 确保上传的文件的位置是正确的。
- ❑ 确保输入的 URL 与要测试的文件是对应的。如果要查看的页面是从另一个页面跳转过来的, 确保链接中的 URL 与页面的文件名和位置是完全匹配的。
- ❑ 确保上传文件之前已经进行了保存(包括最新修改)。
- ❑ 确保上传了所有辅助文件(包括 CSS、图像、音乐、视频等)。
- ❑ 确保 URL 的大小写与文件名的大小写是完全匹配的。(这也是推荐全部使用小写字母的原因, 这样做可以降低输入 URL 时产生错误的可能性——既针对网站开发人员, 也针对访问者。)确保文件名中没有使用空格(应使用

短横线代替空格分隔单词)。

- ❑ 如果在以前的测试中曾禁用某项浏览器功能(如 JavaScript 支持), 确保重新启用这些功能。
- ❑ 确保问题不是出自浏览器本身。对于这一点, 最简单的方法就是换个浏览器对页面再测试一遍。

在接下来的两节里, 我会介绍如何检查 HTML 和 CSS 中的常见错误。

20.5 检查常见错误: HTML

有时, 问题出现在 HTML 中。

检查 HTML 常见错误的方法

- ❑ 很容易出现一两个输入错误, 如图 20.5.1 所示。确保所有的拼写都是正确的, 属性的值都是有效的, 如图 20.5.2 所示。使用 HTML 验证器可以查出这类错误, 从而能迅速地改正(参见 20.1 节)。
- ❑ 留心元素的嵌套。例如, 如果先开始 <p>, 再使用 , 就要确保 结束标签位于最后的 </p> 之前。

还是有问题

如果你阅读完本章, 发现还是有一些问题无法解决, 这里还有一些其他的建议。

- ❑ 当我建议你休息一下的时候, 不要认为我的建议是出于傲慢。有时候, 你能做出的最佳选择就是暂时将问题搁到一边。当你回头再处理时, 答案可能就在眼前。相信我, 我有过这样的经历!
- ❑ 回到页面能正确工作的最近的版本。(为此, 要在建立页面的过程中保存页面的副本, 从而在必要的时候能回到历史版本。)然后再一点一点地增添新的元素, 并随时进行测试。
- ❑ 对于页面链接到的资源(如 CSS、图像、JavaScript 或媒体文件), 直接在浏览器的地址栏中输入资源的 URL, 确保该资源确实是存在的。
- ❑ 还有很多网站用于搜索问题的答案。Stack Overflow (www.stackoverflow.com) 和 SitePoint (www.sitepoint.com/forums/) 便是两个例子。你还可以搜索到一些其他类似的网站。

- ❑ 如果重音字符或特殊符号没有正常显示，要确保文档 head 元素开始后有以下 <meta charset="utf-8" /> 语句（如果不使用 UTF-8，也可以使用其他恰当的字符编码），同时确保文本编辑器是使用与之相同的编码保存 HTML 文件的。如果还有问题，试着使用恰当的字符引用。
- ❑ 确保属性值是用直引号而不是曲引号包围的。如果属性值是用双引号包围的（这也是惯例），则属性值中可以有单引号，如图 20.5.3 所示。如果属性值本身包含双引号，则应使用字符引用，如图 20.5.4 所示。
- ❑ 不要对空元素使用分开的开始标签和结束标签，如图 20.5.5 所示。（从技术上说，即便省略结束标签，或者对空元素添加结束标签，浏览器也能正确地显示，但最好还是严谨一些。）

```

```

图 20.5.1 你能找出问题吗？我将 src 拼错了，还在 width 和 height 的值中使用了单位。如果你没有注意到这些错误，可以使用 HTML 验证器。它们可以标出这类输入错误，从而节省发现这些错误的时间

```

```

图 20.5.2 在正确的版本中，src 的拼写是正确的，width 和 height 的值中也去掉了 px

```

```

图 20.5.3 如果属性值包含单引号，可以照常用双引号包围属性值

```

```

图 20.5.4 如果属性值包含双引号，属性值里的双引用号就应使用字符引用

```
</img>
```

图 20.5.5 不要对空元素（如 img）包含结束标签，HTML 验证器会将这个例子标记为错误

20.6 检查常见错误：CSS

尽管 CSS 的语法相当简单明了，但它也有一些常见的陷阱，特别是如果你习惯了编写 HTML 的话。CSS 验证器可以将如本节讨论的这类错误标示出来，因此，在查看 CSS 以寻找错误之前，要先对样式表进行验证（参见 20.1 节）。

检查 CSS 常见错误的方法

- ❑ 确保使用冒号（:）分隔属性和值，而不是像在 HTML 中那样使用等号，如图 20.6.1 和图 20.6.2 所示。

```
p {
    font-size=1.3em;
}
```

图 20.6.1 哎呀，改变用等号分隔属性和值的习惯很难

```
p {
    font-size: 1.3em;
}
```

图 20.6.2 好多了。属性和值之间应该使用冒号。在冒号之前或之后添加额外的空格不会产生影响，但通常在冒号之后加上一个空格

- ❑ 确保使用分号（；）结束每个属性 – 值对（即声明）。确保没有多余的分号，如图 20.6.3 和图 20.6.4 所示。

```
p {
  font-size: 1.3em font-style: italic;;
  → font-weight: bold;
}
```

图 20.6.3 又错了。每个属性 – 值对之间必须有且只能有一个分号。这里，有一处少写了一个分号，另一处多写了一个分号

```
/* 还是有问题，但更容易发现问题了 */
p {
  font-size: 1.3em
  font-style: italic;;
  font-weight: bold;
}

/* 这是正确的版本 */
p {
  font-size: 1.3em;
  font-style: italic;
  font-weight: bold;
}
```

图 20.6.4 如果每个属性 – 值对都独占一行，错误就更容易发现了，因为这样分号就不会混在一大片属性、值和冒号之中了

- ❑ 不要在数字和单位之间添加空格，如图 20.6.5 和图 20.6.6 所示。

```
p {
  font-size: .8275 em;
}
```

图 20.6.5 又错了。不要在数字和单位之间添加空格

```
p {
  font-size: .8275em;
}
```

图 20.6.6 这次对了。注意冒号和值之间的空格是可选的（但通常包含这个空格）

- ❑ 不要忘了后括号。
- ❑ 确保使用可接受的值。像 `font-style: none;` 这样的声明是无效的，因为该属性的空值为 `normal`。
- ❑ 使用嵌入样式表时，不要忘了 `</style>` 结束标签（无论如何，在多数情况下你都应该避免）。
- ❑ 确保 HTML 文档正确地指向 CSS 文件。
- ❑ 留意 CSS 选择器之间的空格和标点符号。
- ❑ 确保浏览器支持你编写的代码，尤其是最新特性，因为浏览器在 CSS3 成长的过程中也在不断演变。关于浏览器对新特性的支持情况参见 Can I Use (<http://caniuse.com>) 或者 Quirksmode (<http://www.quirksmode.org/css/>)。CSS 验证器无法判断浏览器是否支持某个 CSS 特性，但如果输入的选择器、属性或值是 CSS 中不存在的，验证器仍会给出提示。
- ❑ 使用浏览器开发工具审查浏览器所解析的样式规则，从而快速地查出哪些代码没有按照预期进行解析，或者查看所应用的特殊性规则，如图 20.6.7 所示（参见 20.3 节补充材料“浏览器开发工具”）。

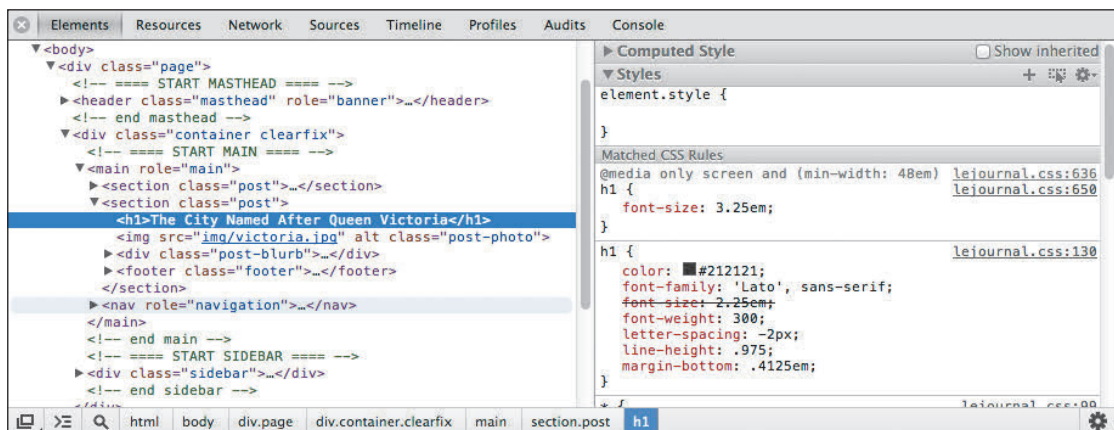


图 20.6.7 这里使用 Chrome 的 DevTools 对代码片段 `<h1>The City Named After Queen Victoria</h1>` 进行了审查。HTML 显示在左侧面板，应用于该元素的 CSS 显示在右侧面板。其他浏览器的工具跟 DevTools 配置相似。font-size 设置上有一条删除线，表示这条声明被另一条规则（显示在媒体查询上面的那条）覆盖了。这样的结果是我在这个例子中想要的效果，但你可以使用这种技术跟踪样式没有按照预期进行应用的原因。还可以编辑 HTML 和 CSS 规则，并直接在浏览器中测试修改。所有的浏览器开发工具都允许这种行为。如果你对结果满意，就可以对 HTML 和 CSS 文件作出真实修改

20.7 如果图像不显示

小红叉、碎片图标、替代文本，或者什么都不显示——这都是图像未能正确加载的标志，如图 20.7.1 和图 20.7.2 所示。

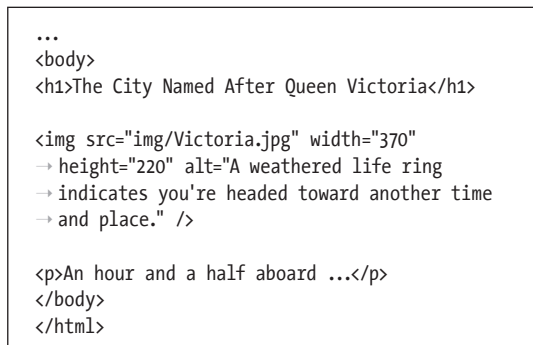


图 20.7.1 图像的文件名为 victoria.jpg，但在 HTML 中，引用的却是 Victoria.jpg（以大写字母 V 开头）。因此，在服务器上检查页面时，图像不会显示



图 20.7.2 在你的计算机上，如果系统不区分文件名的大小写，页面看起来就没有问题。但将页面发布到区分大小写的服务器上以后，便无法找到图像。像 Chrome 这种浏览器就只会显示一个破碎图标

修复缺失的图像

- 首先检查图像在服务器上的文件名是否与 img 元素中引用的名称严格匹配，

包括大小写、扩展名,如图 20.7.1 所示。不要在文件名中包含空格,参见 1.6 节。

- ❑ 确保 `img` 元素的 `src` 属性中的图像 URL 是正确的。一种简单的测试方法就是将图像放到 HTML 页面所在的目录。这样就只需要 `img` 元素中的文件名和扩展名正确就可以了,无需引入路径信息。如果图像显示出来了,问题很可能就出在 `src` 上。不过,将图像放在 HTML 文件所在的目录并不好,因为这样的话,网站很快就会变得无

序。因此,在测试之后,要将图像从 HTML 页面目录移出,并修改指向图像的 `src` 路径。参见 1.7 节。

- ❑ 如果在你的计算机上查看页面时图像是显示的,而将页面上传至服务器之后,图像却没有显示出来,要确保已经将图像上传到了服务器。
- ❑ 把图像保存为 PNG、JPEG 或 GIF 格式了吗? 如果是的话,所有的浏览器都能显示该图像;如果是其他格式,那就很难说了。更多信息参见第 5 章。

本章内容

- 获得域名
- 为网站寻找主机
- 将文件传送至服务器

当你完成自己的杰作并准备好将它展现给公众时，必须将页面传送到万维网上，人们才能看到这些页面。本章介绍了所需的步骤：获取域名，寻找 Web 主机，将域名与主机连接起来，以及将文件上传到主机的服务器上。

如果你希望更改 Web 主机，可以将网站从一个服务器转移到另一个上面。你的域名及整个网站的 URL 可以保持不变。

在发布页面之前和之后，要确保对页面进行彻底的测试。更多细节信息参见第 20 章。

21.1 获得域名

在访问者能够看到你的网站之前，需要为网站关联一个域名，如图 21.1.1 所示。只要你按照本章所示的步骤操作，任何人都可以在浏览器中访问这个域名。

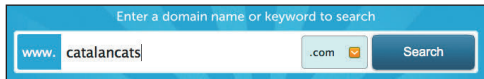


图 21.1.1 只有特定的公司才是可信的域名注册商（这个界面和下面的界面均来自 Namecheap）。你可以使用其中的一个查询你想使用的域名是否可用，也可以通过 Web 主机提供商的网站进行检查

获得域名的步骤

(1) 在浏览器中访问某个域名注册商，查看你想用的域名是否可用，如图 21.1.2 所示。（并参见第一条提示。）如今，要想找一个还未被使用的名字非常不容易，因此你有必要搜索一下你喜欢的名字的变体。

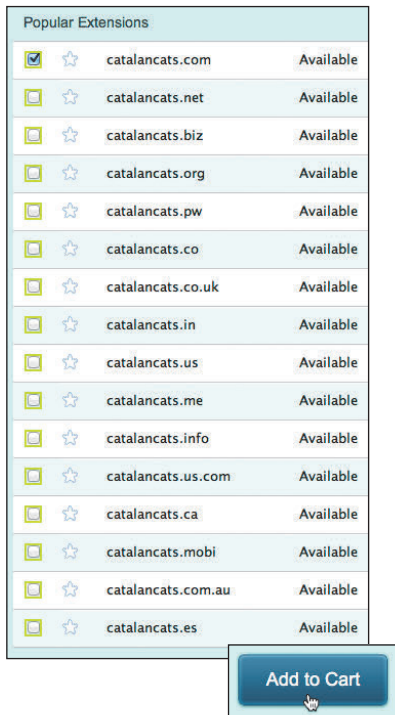


图 21.1.2 如果域名是可用的，就可以通过第三方注册商网站或者 Web 主机提供商进行注册（从图中可以看到，catalancats.com 这个域名是可用的）

(2) 一旦找到一个域名,便可以注册了。不同域名注册商的价格不一样,不过 .com 域名的价格通常大约为每年 10 美元(其他类型的域名的价格可能不一样)。

提示 Namecheap (www.namecheap.com) 和 Hover (www.hover.com) 只是可以注册域名的网站中的两个(没有要求必须使用它们)。通过搜索“domain registrars”(域名注册商)可以找到其他一些类似的网站。此外,很多 Web 主机提供商也提供域名注册服务;反过来,很多域名注册商也提供主机服务。域名注册和主机托管可以使用不同的提供商,很多人都这样做。

提示 要让网站在别人访问 URL 时显示出来,需要先进行一些重要的配置,这些内容参见下一节的补充材料“连接域名和 Web 主机”。

21.2 为网站寻找主机

Web 主机提供商可以提供其服务器上的一部分空间,存放你的网站文件,同时提供其他相关的服务,如创建同域名关联的电子邮件地址(如 yourname@yourdomain.com)。

有很多公司都提供网站托管服务。其中,大部分公司根据其服务的内容按月收取费用。有的公司提供免费的网站托管服务,但要求在你的网站上放置他们的广告(一般不推荐这种方法)。

尽管你可以在互联网上搜索 Web 主机提供商,但还是推荐你向使用某种主机的朋友进行咨询——抑或某位值得信赖的博主提到的他所用的主机的提供商。另外,登记一下网站的页脚和附注栏。

考虑主机提供商时,除了价格之外,还要考虑以下几件事。

- ❑ 他们允许一个主机使用多个域名吗?是否需要为每个网站支付额外的费用?如果你希望自己的网站未来发展得很好,就应该寻找可以使用多个域名的主机。
- ❑ 他们允许网站使用多大的磁盘空间?不要为超出需要的部分付费。话虽如此,通常连最基本的账户也拥有足够的空间。记住,HTML 文件占用的空间非常小,而图像、音频和视频文件则依次占用更大的空间。如果有必要,使用一段时间后,你总是能够升级到拥有更多空间的账户。
- ❑ 他们允许每个月使用多大的数据传输量(带宽)。这个值代表的是发送给访问者的数据(包括 HTML、CSS、图像、媒体文件等)的总大小,而不是允许在服务器存储的空间的大小。因此,如果你预计访问者会从你的网站访问大量的大文件,就需要更大的月传输配额。关于这一点,通常来说,网站刚起步时,最基本的账户就能满足需求,将来可以视情况再升级。
- ❑ 可以使用域名创建多少邮箱(主机提供商通常提供很多个。)
- ❑ 他们有应对大访问量、防止网站崩溃的措施吗?
- ❑ 他们提供哪种技术支持?是通过电话、邮件、还是在线交谈,他们响应请求需要多长时间,他们的网站上有大量的支持信息吗?(在成为他们的客户之前,可以检查这些内容的数量。)
- ❑ 他们多久备份一次服务器上的数据(以防出现问题)?
- ❑ 可以使用哪些服务器端语言和软件包?多数至少会提供 PHP、MySQL,

还有一些会提供 WordPress 或其他高级特性。有些特性需要更昂贵的付费方案。

将 ISP 作为 Web 主机提供商

如果你可以访问互联网,就可能已经通过你的 ISP (Internet Service Provider) 获得了少量的 Web 空间。它可能难以存放整个网站,但放几个页面倒是足够。可以向你的 ISP 询问详细信息。

不过,要记住的是,这类托管空间通常不允许使用单独的域名访问网站,例如,可以使用 `www.someisp.com/your-site/`,但不能使用 `www.yourdomain.com`。因此,如果你希望建设的是专业的网站,就不要使用 ISP 提供的免费空间存放网站。

Web 分析

Web 分析是一份报告,告诉你有多少人访问过你的网站,他们使用了哪些浏览器,哪些页面是最受欢迎的,以及很多其他有用的数据。

你所用的 Web 主机有可能提供这些信息,但如果你希望获得更为丰富的信息,可以使用 Google Analytics 或类似的服务。该工具很容易添加到网站,只要在每个页面中添加一小段代码就可以了。更多信息参见 www.google.com/analytics/。

连接域名和 Web 主机

注册了域名,找到了 Web 主机之后,下面很重要的一步就是将二者结合到一起:你必须将你的域名指向你的 Web 主机,这样,访问者输入网站的 URL 时网站才能加载。

要实现这一步,需要对你的域名配置域名服务器。Web 主机提供商会提供用于这项配置的域名服务器信息。

根据注册域名的方式(参见 21.1 节)的不同,实际的配置是在以下两个地方中的一个进行的。如果域名是通过域名注册商注册的,可以登录到对应的账户,为域名设置域名服务器信息(域名注册商会提供操作说明)。如果域名是通过 Web 主机提供商注册的,可以登录到对应的账户,更新设置。

如果这些内容让你感到有些疑惑,也不必担心。Web 主机提供商和域名注册商会提供操作上的说明,通常,如果需要,他们还能提供手把手的帮助。

需要记住的是,当你修改域名服务器设置时,通常需要 24 至 48 小时(最多 72 小时)的时间,相应的更新才能传播到整个万维网。不过,这种改变不会在所有的地方同时发生。因此,如果你更新了域名服务器信息(并按照 21.3 节的说明上传了网站的文件),你的朋友可能已经可以在他所在的地方正常访问网站了,而你却无法立即看到网站(也可能是相反的情况)。你的网站应该在不久之后就能让所有人都看到。

21.3 将文件传送至服务器

为了让互联网上的人们看到你的页面,你需要将页面上传到 Web 主机服务器。一种简单的方法就是使用 FileZilla (<http://filezilla-project.org>) 这样的 FTP 客户端。FileZilla 是一款自由软件,可以运行在 Windows、Mac OS X 和 Linux 上(关于其他的 FTP 客户端,

参见提示)。很多网页编辑器也包含FTP功能，从而可以在编辑器中发布页面，而不必使用FileZilla这样的程序。

通常，你的Web主机提供商会在你注册托管账户之后通过电子邮件向你发送FTP连接信息。（如果你没有收到，可以联系他们。）一旦获得这些信息，就可以配置服务器连接，并将该配置保存在一个名称下面，如图21.3.1、图21.3.2和图21.3.3所示，从而在以后想要发布文件（或从服务器上下载文件）的时候可以方便地访问服务器。

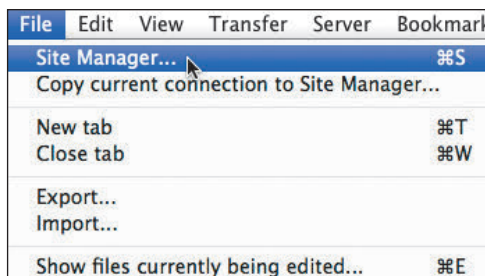


图 21.3.1 如果要在 FileZilla 中输入新的服务器信息，从主窗口中选择 File → Site Manager（文件→站点管理器）。Site Manager 是为每个站点配置 FTP 连接信息的地方

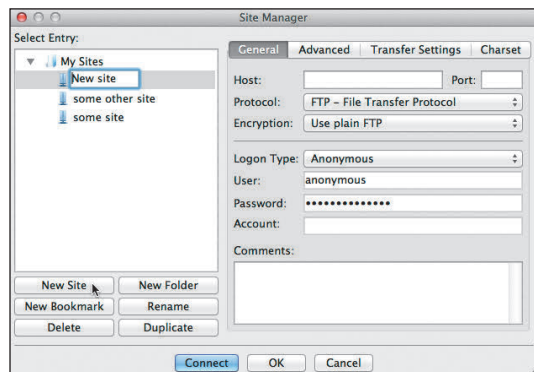


图 21.3.2 在 Site Manager 中点击 New Site（新站点）按钮，My Sites（我的站点）下面就会出现一个临时的名称

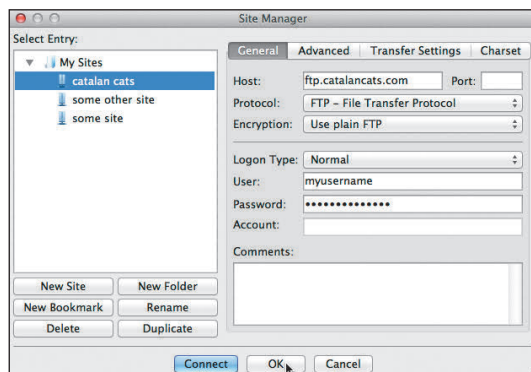


图 21.3.3 使用你选择的名称替换临时名称，再在 General（常规）标签页中配置连接信息。点击 Connect（连接）按钮，会保存这些信息并立即建立与服务器之间的连接。点击 OK（确定）按钮，则仅会保存信息

接下来，连接到服务器（如图21.3.4所示）和传输文件（如图21.3.5和图21.3.6所示）的操作就相当简单了。

注意，FileZilla 在 Mac OS 和 Windows 上的外观有些不同，但其界面是非常相似的（截图来自不同的操作系统）。除非特别指出，以下使用步骤对两个版本来说都是相同的。

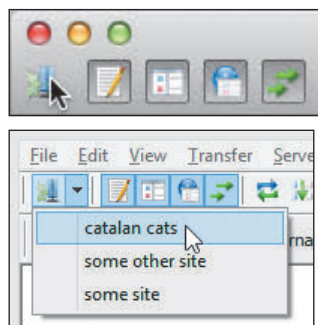


图 21.3.4 当站点的连接信息保存到 Site Manager 之后，就可以在不重复输入的情况下直接连接到 Web 主机的 FTP 服务器。在 Mac 或 Windows 上，通过这里显示的服务器图标回到 Site Manager（也可以通过图 21.3.1 中的菜单）。在 Windows 上，就像这里底部图像显示的，可以激活向下的箭头，再从弹出的下拉菜单中选择站点名称

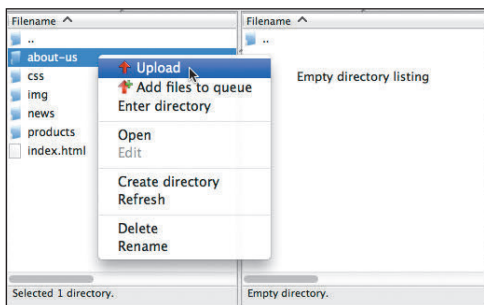


图 21.3.5 在窗口左侧，定位到你的计算机上包含要上传的文件的目录。在窗口右侧，选择服务器的目标目录。再在要上传至服务器的文件或文件夹上点击鼠标右键，选择 Upload（上传）

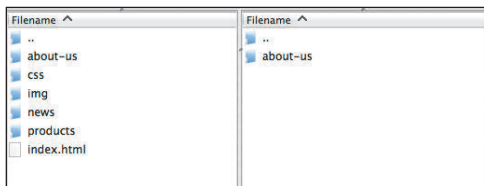


图 21.3.6 新上传的文件夹出现在窗口右侧的框内。对所有要传送至服务器的文件和文件夹执行相同的操作。如果要一次性上传多个文件和文件夹，可以先选中这些文件和文件夹，再点击鼠标右键并选择 Upload

1. 定义新的 FTP 站点的属性

(1) 在 FileZilla 的主菜单中选择 File → Site Manager（如图 21.3.1 所示），或者点击服务器图标（如图 21.3.4 所示），显示 Site Manager 窗口。

(2) 在 Site Manager 窗口中，点击 New Site（新站点）按钮（如图 21.3.2 所示）。

(3) 输入站点名称（替换临时名称）。这个名称不必与域名相同，它只是一个标签。根据你的 Web 主机提供商的说明，填写 General 标签页下各字段的信息。通常，至少需要输入主机 URL，在 Logon Type（登录类型）中选择 Normal（常规），输入用户名和密码（通常在建立账号时创建），如图 21.3.3 所示。

(4) 输完连接信息之后，点击 Connect 按钮保存信息并立即连接到服务器，或者点击 OK 按钮保存信息供以后连接用，如图 21.3.3 所示。

2. 使用 FileZilla 将文件传送到服务器

(1) 打开 FileZilla。

(2) 选择最左端的服务器图标（如图 21.3.4 所示），显示 Site Manager 窗口。再在弹出的下拉菜单中选择你的站点名称，点击 Connect 按钮（Windows 版有捷径，如图 21.3.4 所示）。FileZilla 就会建立与服务器的连接。

(3) 在窗口的右侧，定位到要上传文件的服务器目录。

(4) 在窗口的左侧，定位到你的计算机上存放要上传的文件的目录。

(5) 在左侧框右击目标文件或者文件夹，再在弹出的菜单中选择 Upload（如图 21.3.5 所示），文件就会进行传输，如图 21.3.6 所示（对于视频等大文件，这一过程所需要的时间会长一些）。还可以反过来传输文件（参见第一条提示）。也可以从一端将文件拖放到另一端，而不用右击的办法。

(6) 你对站点的更新现在已经能在线上看到了。通过 www.yourdomain.tld（这里的 *yourdomain.tld* 是你注册的域名；其中，*.tld* 是顶级域名，一般为 *.com*，但也可以注册使用其他扩展名的域名）访问你的站点，确保一切都是正常的。如果需要，编辑你的计算机上的文件，再按照第 (3) 步至第 (5) 步的说明将它们上传至服务器（如果过去了很长时间，则还需执行第 (2) 步）。重复这一步，直到网站完全符合你的预期。

(7) 完成文件传输之后，可以关闭 FileZilla 或在主菜单中选择 Server → Disconnect（服务器→断开连接），如图 21.3.7 所示。

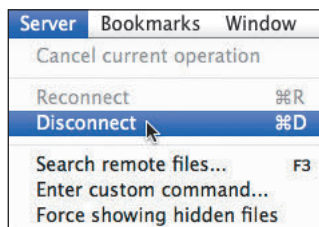


图 21.3.7 完成之后，选择 Server → Disconnect

提示 还可以将文件从网站服务器传至你的计算机。要实现这一步，只需要右键单击右侧框中的文件或文件夹（如图 21.3.5 所示），再在弹出的菜单中选择 Download（下载）。

提示 FileZilla 只是众多 FTP 客户端中的一个。CyberDuck（自由软件，<http://cyberduck.ch>）在 OS X 和 Windows 上均可用，OS X 上的流行软件还有 Transmit（www.panic.com/transmit）和 Fetch（<http://fetchsoftworks.com>）。OS X 还有内置的 FTP 功能（参见 <http://osxdaily.com/2011/02/07/ftp-from-mac/>）。在网上搜索“FTP client”（FTP 客户端）可以找到更多的可以运行在 Windows 和 OS X 上的软件。它们的工作方式是相似的，只是有的软件功能多一些，有的少一些。

提示 传输文件和文件夹时，它们是被复制到目标文件夹的，原来的位置还保留它们的原始版本。

提示 如果目标位置已经存在要传输的文件或文件夹，那么 FTP 程序可能会弹出对话框，确认是否要覆盖（FileZilla 就会这样做）。不过，每个 FTP 客户端都不相同，因此有的软件也可能不经确认而直接覆盖。可以试着传一个测试文件，了解你的 FTP 客户端是如何处理这类情形的。

提示 代码中的相对 URL 在文件夹传至服务器之后仍然有效。

提示 如果访问网站的 URL 时网站无法显示，可能有多种原因。首先，再次检查是否已将文件传至正确的目录。通常，页面应位于 *public_html*、*www* 或其他类似名称的目录下。你的 Web 主机提供商的操作说明中应该指明了正确的位置，如果不确定，可以向他们咨询。如果文件都放在正确的位置而网站依然无法显示，问题就可能出在域名服务器设置上（参见 21.2 节补充材料“连接域名和 Web 主机”）。

提示 如果将某文件的新版本上传至服务器之后却没有看到修改的效果，可以清除浏览器的缓存，并再次检查页面。如果不清楚如何清除缓存，可以搜索浏览器的帮助文档。



本附录收录了几乎所有的 HTML 元素和属性，包括一些本书未作介绍的元素和属性。（大多数情况下，没有介绍它们是因为使用频率很低或者是高级特性。）每个元素都有简单的描述，元素后面还列出了与其相关联的属性。

“参考章节”一列标出了本书介绍对应元素或属性的章节，方便查阅。有时，某个属性的章节可能对应的是对另一个元素的介绍，这意味着对该属性的介绍适用于所有可用的元素。

此外，有一些条目有以下标识：

□ (5)，HTML5 新增的元素或属性；

□ (*)，在 HTML5 之前的 HTML 中就已存在，但 HTML5 对其重新进行了定义。

HTML5 的一些新特性是旧的浏览器（如 IE8）所不支持的。关于最新的浏览器支持情况，参见 <http://caniuse.com>，该网站一直在不断更新中。

随时查阅

CSS 引用和 HTML 参考都可以在本书的配套网站上找到（见 www.htmlcssvqs.com）。

表 A.1 HTML 元素和属性

元素 / 属性	描 述	参考章节
绝大多数 HTML 元素	下列属性适用于绝大多数 HTML 元素	
accesskey	用于为元素添加键盘快捷键	
aria-*	用于关联由 WAI-ARIA 指定的可访问性属性值	16.5
class	用于标识一组元素，以便为它们应用样式	3.14
contenteditable (5)	用于让元素的内容变成可编辑的	
data-*(5)	用于存储页面或应用特有的定制化数据	
dir	用于指定元素的文字方向	4.18
draggable (5)	用于让元素变成可拖拽的	
dropzone (5)	用于将元素标记为可拖拽元素可以放下的区域	
hidden (5)	用于指示元素还不是相关的或不再是相关的	
id	用于标识特定的元素，以便为其添加链接，应用样式或使用 JavaScript 编写脚本	3.14

(续)

元素 / 属性	描 述	参考章节
<code>lang</code>	用于指定元素的书写语言	3.1
<code>role</code>	用于为辅助设备提供关于元素角色的额外信息(由 WAI-ARIA 定义)	3.13
<code>spellcheck</code> (5)	用于指示是否应该对元素的内容进行拼写和语法检查	
<code>style</code>	用于添加本地样式表信息	8.4
<code>tabindex</code>	用于定义访问者使用制表键时在元素之间移动的顺序	6.1
<code>title</code>	用于为元素添加工具提示	3.15
a	用于创建链接和锚	6.1
<code>href</code>	用于指定链接到的页面的 URL 或锚的名称	6.1
<code>hreflang</code> (5)	用于指定所链接的资源的语言	
<code>download</code> (5)	用来指定指向下载资源的链接, 将 <code>download</code> 值设置为文件名	
<code>rel</code>	用于标识链接的性质	6.1
<code>target</code> (*)	用于指定链接应打开的窗口或框架	6.1
<code>type</code>	用于指定资源的 MIME 类型	
abbr (*)	用于解释缩写或首字母缩写的含义	4.8
address	用于为最近的 <code>article</code> 或 <code>body</code> 元素祖先标识联系人信息	4.11
area	用于指定图像映射的坐标	
<code>alt</code>	用于给出关于区域的信息	
<code>coords</code>	用于给出图像映射中区域的坐标	
<code>href</code>	用于指定图像映射中区域链接的目标 URL	
<code>hreflang</code> (5)	用于指定所链接的资源的语言	
<code>download</code> (5)	用于指定指向下载资源的链接。应该将 <code>download</code> 值设置为文件名	
<code>rel</code>	用于标识链接的种类	
<code>shape</code>	用于指定图像映射中区域的形状	
<code>target</code> (*)	用于指定链接应打开的窗口或框架	6.1
article (5)	用于标识页面中的独立成分, 原则上是可独立分发或可再用的	3.8
aside (5)	用于标识页面中的一个区域, 其内容与周围的内容无关	3.10
audio (5)	用于在页面中嵌入音频	17.10
<code>autoplay</code> (5)	用于告诉浏览器在它播放音频文件时立即开始播放	17.11
<code>controls</code> (5)	用于告诉浏览器为音频元素提供控件	17.10
<code>loop</code> (5)	用于告诉音频文件在播放到末尾后不间断地继续从头播放	17.11
<code>muted</code> (5)	用于控制音频输出的默认状态	
<code>preload</code> (5)	用于指定浏览器是否在访问者开始播放音频文件之前开始下载该文件	17.11
<code>src</code> (5)	用于标识要播放的音频文件的 URL	17.10
b (*)	用于标识出于实用的目的提醒读者注意的一块文本, 不传达任何额外的重要性, 也不表示其他的语态和语气	4.3
base	用于指定页面的基准 URL	
<code>href</code>	用于指定用于生成相对 URL 的 URL	

(续)

元素 / 属性	描 述	参考章节
<code>target (*)</code>	用于指定页面上链接的默认目标	6.1
bdi (5)	用于标识独立于周围文本的用做双向文本格式化的一块文本	4.18
<code>dir</code>	用于指定文本方向	4.18
bdo (5)	用于显式地格式化其内容的文本方向	4.18
<code>dir</code>	用于指定文本方向	4.18
blockquote	用于指定页面上的一段引述文本	4.6
<code>cite</code>	用于给出来源的 URL	4.6
body	用于包围页面的主要内容区域	3.1
br	用于创建换行	4.16
button	用于创建按钮	16.15
<code>autofocus</code> (5)	用于指定按钮在页面加载时立即获得焦点	16.5
<code>disabled</code>	用于指示元素在当前状态下是不可用的	16.16
<code>form</code> (5)	用于将元素同另外一个不包含该元素的表单关联起来	
<code>formaction</code> (5)	用于覆盖表单的 <code>action</code> 属性	
<code>formenctype</code> (5)	用于覆盖表单的 <code>enctype</code> 属性	
<code>formmethod</code> (5)	用于覆盖表单的 <code>method</code> 属性	
<code>formnovalidate</code> (5)	用于覆盖表单的 <code>novalidate</code> 属性	16.15
<code>formtarget</code> (5)	用于覆盖表单的 <code>target</code> 属性	
<code>name</code>	用于标识使用按钮发送的数据, 或者用于标识按钮本身 (或许是为了使用某项 JavaScript 功能)	16.15
<code>type</code>	用于在表单元素中使用按钮	16.15
<code>value</code>	用于指定在点击按钮时应该提交的数据	16.15
canvas (5)	提供用于生成依赖于分辨率的位图画布的脚本, 以在线呈现图像	17.14
<code>width, height</code> (5)	用于指定画布的尺寸	
caption	用于创建表格的标题	18.1
cite	用于将文本标记为引述	4.5
code	用于将文本标记为计算机代码	4.13
col	用于把表格中的列组合成非结构化组	
<code>span</code>	用于指定列组中的列数	
colgroup	用于把表格中的列组合成结构化列组	
<code>span</code>	用于指定列组中的列数	
datalist (5)	包含一组选项元素, 这些元素是代表另一表单控件的一套预定义的选项	4.12
dd	用于标记列表中的定义	15.8
del	用于标记删除了的文本	4.12
<code>cite</code>	用于引用对修订进行解释的 URL	4.12
<code>datetime</code>	用于指定修订的时间和日期	4.7
details (5)	用于创建公开的小部件, 访问者可以通过它获取额外的信息或控制	

(续)

元素 / 属性	描 述	参考章节
open (5)	用于指定元素在默认情况下是打开的还是关闭的	
dfn	用于指定列表项目的定义实例	4.9
title	用于提供术语的定义	4.9
div	用于将页面切割为块级区域	3.12
dl	用于创建描述列表	15.8
dt	用于标记要在列表中定义的术语	15.8
em (*)	用于标记要强调的文本	4.3
embed (*)	用于添加多媒体	
src	用于指定多媒体文件的 URL	
type	用于标识多媒体文件的 MIME 类型	
width, height	用于指定嵌入的多媒体播放器的尺寸	
fieldset	用于将一套表单元素组合在一起	16.4
disabled (5)	用于将表单元素组内的所有表单控件设为不可用的	16.16
form (5)	用于将元素同另外一个不包含该元素的表单关联起来	
figcaption (5)	用于为其父元素 figure 的内容提供标题或说明文字	4.4
figure (5)	用于识别在主文档流内被引用, 但在不影响文档流的情况下可以移至他处的内容	4.4
footer (5)	用于识别最近的祖先元素 body 、 section 、 article 或 aside 的页脚	3.11
form	用于指定表单, 表单用于收集要提交的数据	16.2
accept-charset	用于识别要在提交表单时使用的字符编码 (默认为页面的字符集)	
action	用于给出处理表单数据的脚本的 URL	16.2
autocomplete (5)	当该属性设为 off 时, 用于阻止浏览器提供或记住自动完成值 (默认为 on , 即在默认情况下允许自动完成)	16.5
enctype	用于确保文件以正确的编码格式发送至服务器	16.13
method	用于指定数据应如何发送至服务器	16.2
name	用于为表单提供名称, 供以后使用	16.2
novalidate (5)	用于允许表单在不验证的情况下提交	16.2
target (*)	用于识别表单提交的目标窗口或 iframe	6.1
h1, h2, h3, h4, h5, h6	用于创建标题	3.3
head	用于创建 head 部分, 该部分包含关于页面的信息, 包括标题、制作者、关键字、样式表和脚本	3.1
header (5)	用于识别一组介绍性内容或导航帮助	3.5
hr (*)	用于标识段落级的主题变化	
html	用于标识作为 HTML 文档的文本文档	3.1
manifest (5)	用于指定离线时使用的应用程序缓存清单	
i (*)	用于标记用另外一种语态和语气, 或不同于常规方式陈述以表现不同特质的一块文字	4.3
iframe	用于加载嵌套在另一个网页中的网页	
name	用于指定作为目标的 iframe 的名称	

		(续)
元素 / 属性	描 述	参考章节
sandbox (5)	用于出于安全目的, 为 iframe 的内容指定额外的限制	
seamless (5)	用于指定 iframe 是否显示为包含它的页面的一部分	
src	用于指定初始页面的 URL	
srcdoc (5)	用于指定初始页面的 URL	
width, height	用于指定 iframe 的尺寸	
img	用于在页面中插入图像	5.5
alt	用于提供替代文本。当图像无法显示时, 则显示替代文本; 替代文本也是为使用辅助设备的用户提供的	5.6
ismap	用于指示该元素可以提供对服务器端的图像映射 (该元素必须是 a 元素的后代) 的访问	
src	用于指定图像的 URL	5.5
usemap	用于指定应同引用图像一起使用的客户端图像映射	
width, height	用于指定图像的尺寸, 从而让页面的加载更快, 或出于对图像进行缩放的目的	5.7
input	用于创建表单元素	16.1
accept	当输入框类型为 file 时, 用于向浏览器告知需要接受的文件类型	16.1
alt	当输入框类型为 image 时, 用于提供替换文本	16.15
autocomplete (5)	当该属性设为 off 时, 用于阻止浏览器提供或记住自动完成值 (默认为 on, 即在默认情况下允许自动完成)	16.5
autofocus (5)	用于指定输入框在页面加载时立即获得焦点	16.5
checked	用于标记单选按钮或复选框在默认情况下被选中	16.9
dirname (5)	用于标识输入的文本的方向	
disabled	用于指示输入框在当前状态下是不可用的	16.16
form (5)	用于将元素同另外一个不包含该元素的表单关联起来	
formaction (5)	用于覆盖表单的 action 属性	
formenctype (5)	用于覆盖表单的 enctype 属性	
formmethod (5)	用于覆盖表单的 method 属性	
formnovalidate (5)	用于覆盖表单的 novalidate 属性	16.15
formtarget (5)	用于覆盖表单的 target 属性	
list (5)	用于将输入框与数据列表关联起来	16.1
max, min (5)	用于指示输入框元素允许的值的范围	
maxlength	用于指定可输入到输入框元素的字符的最大数量	16.5
multiple (5)	用于指定是否允许用户输入一个以上的值	16.8
name	用于标识元素收集的数据	16.2
pattern (5)	用于提供可对输入框元素的数据进行检查的正则表达式	16.8
placeholder (5)	用于为数据输入提供提示	16.5
readonly	用于防止访问者修改特定的表单元素	16.14
required (5)	用于要求元素在提交表单时不能为空 (当输入框类型为 hidden、image 或按钮类型时不可用)	16.5

(续)

元素 / 属性	描 述	参考章节
size	用于指定文本框或密码框的长度	16.5
src	用于指定活动图像的 URL	16.15
step (5)	用于控制允许输入的值的间隔大小和特殊性	
type	用于指定表单元素的类型为文本框、密码框、单选按钮、复选框、隐藏字段、提交按钮、重置按钮、活动图像、日期 / 时间框、数字框或颜色框；用于从一系列值中进行选择；或用于输入电话号码、电子邮件地址或一组搜索词	16.1
value	用于指定表单元素的默认数据	16.5
width, height	用于指定输入框的尺寸（仅在输入框类型为 image 时可以使用）	16.15
ins	用于标记对文档增加的内容	4.12
cite	用于引用对修订进行解释的 URL	4.12
datetime	用于指定修订的时间和日期	4.7
kbd	用于标记用户输入	4.13
keygen (5)	用于生成公钥—私钥对	
autofocus (5)	用于指定 keygen 元素在页面加载时立即获得焦点	16.5
challenge (5)	用于生成与密钥对伴生的诘问	
disabled (5)	用于指示元素在当前状态下是不可用的	16.16
form (5)	用于将元素同另外一个不包含该元素的表单关联起来	
keytype (5)	用于标识要生成的密钥对类型	
name (5)	用于标识收集的数据	16.2
label	用于为表单元素添加标签	16.6
for	用于指定标签所属的表单元素	16.6
form (5)	用于将元素同另外一个不包含该元素的表单关联起来	
legend	用于为表单元素组添加标签	16.4
li	用于创建列表项目	15.1
value	用于指定列表项目的初始值（当该元素为 ol 的子元素时）	15.4
link	用于指向外部样式表或其他外部资源	8.2
href	用于指定资源的 URL	8.2
hreflang (5)	用于指定所链接的资源的语言	
media	用于定义样式表的目标媒体类型和（或）媒体特性	8.6
rel	用于标识链接种类	8.2
sizes (5)	用于标识引用图标的大小（仅在 rel 属性为 icon 时可以使用）	
title	用于为替代样式表或其他资源添加标签	
type	用于指出资源的 MIME 类型（仅在链接类型不为 text/css 时需要使用）	
main	用于指定页面的主要内容区域	3.7
map (5)	用于创建客户端图像映射	
name	用于对映射命名，从而使其可在以后被引用	
mark (5)	出于引用的目的，对与另一个上下文相关的文本进行突出显示	4.15

(续)

元素 / 属性	描 述	参考章节
menu (*)	用于包含命令列表	
label (5)	用于为菜单添加标签	
type (5)	用于标识所使用的菜单的种类: context、list (默认值) 或 toolbar	
meta	用于关联页面的各种元数据	3.1
charset	用于标识页面本身的字符编码	3.1
content	用于添加关于页面本身的额外信息	
http-equiv	用于创建指向其他页面的自动跳转, 设置默认脚本语言, 声明字符编码	
name	用于标识关于页面的额外信息	
meter (5)	用于表示在已知范围内的量度	4.18
high, low (5)	用于指定量度为 high 或 low	4.18
max, min (5)	用于标识允许指定的值的最大值和最小值	4.18
name (5)	用于标识收集的数据	4.18
optimum (5)	用于标识最优值	4.18
value (5)	用于指定量表的当前值 (必需的属性)	4.18
nav (5)	用于标识页面的一块区域, 该区域包含指向其他页面或页面内不同部分的链接	3.6
noscript	用于提供脚本的替代内容	
object	用于在网页中嵌入对象	
data	用于标识要嵌入的多媒体文件的来源	
form (5)	用于将元素同另外一个不包含该元素的表单关联起来	
name	用于标识对象 (例如, 对其编写脚本)	
type	用于指出对象的 MIME 类型	
typemustmatch	用于指示对象 data 属性中指定的资源的 MIME 类型必须与对象 type 属性中标识的 MIME 类型相同	
usemap	用于指示对象是否拥有相关联的图像映射	
width, height	用于指定对象框的尺寸	
ol	用于创建有序列表	15.1
reversed (5)	用于指定列表是否为反序 (... , 3, 2, 1)	15.1
start (*)	用于指定第一个列表项目的初始值	15.4
type (*)	用于指定每个列表项目开始的数字类型	15.2
optgroup	用于对 select 元素中的 option 元素进行分组, 一组内的 option 元素位于同一个标签下	16.12
disabled	用于指示元素在当前状态下是不可用的	16.16
label	用于为选项组添加标签	16.12
option	用于创建 select 或 datalist 元素中的单独的选项	16.12
disabled	用于指示元素在当前状态下是不可用的	16.16
label	用于指定选项如何出现在菜单中	16.12
selected	用于标记空白表单中默认被选中的菜单选项	16.12

(续)

元素 / 属性	描 述	参考章节
value	用于指定菜单选项的初始值	16.12
output (5)	用于表示计算结果	16.1
for (5)	用于创建计算结果与进入计算过程的值之间的显式关联	
form (5)	用于将元素同另外一个不包含该元素的表单关联起来	
name (5)	用于标识收集的数据	16.2
p	用于创建段落	4.1
param	用于设置对象的属性	
name	用于标识属性的种类	
value	用于设置有名称的属性的值	
pre	用于表示一块预格式化文本	4.14
progress (5)	用于标识任务的完成进度	4.18
max (5)	必须为大于 0 的有效浮点数（如果有的话）	4.18
value (5)	必须为大于或等于 0 的有效浮点数（且小于或等于 max 属性的值，如果有该值的话）	4.18
q	用于引用来自另一来源的短文	4.6
cite	用于给出引用源的 URL	4.6
rp (5)	用于在不支持旁注标记的浏览器中的旁注标记文本周围显示括号	4.18
rt (5)	用于标记旁注标记文本	4.18
ruby (5)	用于允许文本被旁注标记所标记	4.18
s (*)	用于标识不再准确或不再相关的文本	4.12
samp	用于呈现某程序或计算系统的样本输出	4.13
script	用于为页面添加“自动的”脚本	19.1
async (5)	用于影响脚本的加载和执行	
charset	用于指定外部脚本所用的字符集	3.1
defer	用于影响脚本的加载和执行	
src	用于引用外部脚本	19.1
type (*)	用于指定脚本所用的脚本语言（仅在脚本类型不为 text/javascript 时需要使用）	
section (5)	用于识别文档的区块	3.9
select	用于创建可以从一组选项中进行选择的表单控件	16.12
autofocus (5)	用于指定 select 元素在页面加载时立即获得焦点	16.5
disabled	用于指示元素在当前状态下是不可用的	16.16
form (5)	用于将元素同另外一个不包含该元素的表单关联起来	
multiple	用于允许用户在菜单中选择一个以上的选择	16.12
name	用于标识从菜单收集的数据	16.12
required (5)	用于标识用户必须选择一个选项才能提交表单（第一个 option 子元素必须为占位符或空值）	16.5
size	用于指定在初始状态下菜单中可见的项目数（同时用于将菜单显示为列表）	16.12

		(续)
元素 / 属性	描 述	参考章节
small (*)	用于呈现像条文细则等次要注释	4.2
source (5)	用于在 audio 或 video 元素中标识替代媒体资源	17.12 17.7
media (5)	用于标识资源的目标媒体类型	12.4
src (5)	用于标识要播放的音频或视频文件的 URL	17.7
type (5)	用于指出资源的 MIME 类型	17.7
span	用于包围元素中无直接语义含义的内容	4.17
strong (*)	用于标识元素内特别重要的内容	4.3
style	用于在页面中嵌入样式信息	8.3
media	用于标识样式表的用处	12.4
title	为其他样式表标记说明标签	12.4
type (*)	用于标识样式表的 MIME 类型（仅在样式类型不为 text/css 时需要使用）	
sub	用于创建下标	4.10
summary (5)	用于标识 details 父元素内容的摘要、标题或说明文字	
sup	用于创建上标	4.10
svg (5)	用于在页面中嵌入可缩放矢量图形	17.14
table	用于创建表格	18.1
tbody	用于识别表格的主体部分；比之于头部（ thead ）和尾部（ tfoot ）	18.1
td , th	分别用于在表格中创建普通单元格和标题单元格	18.1
colspan	用于让单元格跨越多列	18.2
headers	通过在 headers 值中包含某个 th 的 id 值，从而将这个 th 与一个 td 或者另一个 th 显式关联	
rowspan	用于让单元格跨越多行	18.2
scope	用于指定 th 应用于哪些行、列、行组或列组	18.1
textarea	用于在表单中创建文本块输入区域	16.11
autofocus (5)	用于指定文本区域元素在页面加载时立即获得焦点	16.5
dirname (5)	用于识别输入的文本的方向	
disabled	用于指示元素在当前状态下是不可用的	16.16
form (5)	用于将元素同另外一个不包含该元素的表单关联起来	
maxlength	用于指定可输入到 textarea 元素的字符的最大数量	16.11
name	用于标识从文本块收集的数据	16.11
placeholder (5)	用于为数据输入提供提示	16.5
readonly	用于保护文本区域的内容	16.14
required (5)	用于要求元素在提交表单时不能为空	16.5
rows , cols	分别用于指定文本块的行数和列数	16.11
wrap (5)	用于指定在字段内容提交时使用软换行或硬换行	
tfoot , thead	用于识别表格的尾部和头部	18.1

(续)

元素 / 属性	描 述	参考章节
time (5)	用于指定日期和时间	4.7
datetime (5)	用于为元素的文本所表达的时间或日期提供机器可读版本	4.7
title	用于创建页面的标题（必须使用）	3.2
tr	用于在表格中创建行	18.1
track (5)	用于为 audio 或 video 父元素指定外部计时文本轨道	17.8
default (5)	用于指示默认轨道	
kind (5)	用于识别轨道为 subtitles（字幕）、captions（标题）、descriptions（描述）、chapters（篇章）或 metadata（元数据）	
label (5)	用于为轨道提供用户可读的名称	
src (5)	用于标识轨道数据的 URL	
srclang (5)	用于标识轨道数据的语言	
u (*)	用于显示一段文本，作为虽然明确地呈现却不怎么准确的非文本注解	4.18
ul	用于创建无序列表	15.1
var	用于标记作为变量的文本	4.13
video (5)	用于嵌入视频、电影和有说明文字的音频文件	17.3
autoplay (5)	用于告诉浏览器在它播放视频文件时立即开始播放	17.4
controls (5)	用于告诉浏览器为视频元素提供控件	17.4
loop (5)	用于告诉视频文件在播放到末尾后不间断地继续从头播放	17.5
muted (5)	用于控制音频输出的默认状态	
poster (5)	用于指定占位图像的 URL，该图像在媒体加载时或加载出现问题时显示	17.5
preload (5)	用于指定浏览器是否在访问者开始播放媒体文件之前开始下载该文件	17.6
src (5)	用于标识要播放的视频文件的 URL	17.3
width, height (5)	用于指定视频的尺寸	17.3
wbr (5)	用于识别在没有连字符的单词中可以在必要时进行换行的位置	4.18

本附录包含以下 CSS 引用表格。

- 表 B.1 CSS 属性和值，可用做很多常见或实用的 CSS 属性及其默认值、允许值的快速参考。
- 表 B.2 CSS 选择器和结合符，对 CSS 选择器和结合符的引用，包括在 CSS3 中引入的那些。
- 表 B.3 CSS3 的颜色值，涵盖了在 CSS3 中引入的颜色值（HSL、HSLA 和 RGBA）。
- 表 B.4 媒体查询。

关于哪些浏览器支持哪些 CSS3 属性和值的信息，可参考 www.quirksmode.org/css/contents.html、<http://caniuse.com> 及 <http://findmebyip.com/litmus>。此外，还可以使用像 Modernizr（www.modernizr.com）这样的 JavaScript 库测试浏览器对这些特性的支持情况。

B.1 CSS 属性和值

表 B.1 CSS 属性和值

属性 / 值	描述和注释
background 任 何 background-attachment、background-color、background-image、background-repeat 和（或）background-position 值的组合，或 inherit	用于修改元素的背景颜色和背景图像 初始值取决于单独的属性，不继承的 background-position 可使用百分数 如果要显示多重背景，可使用逗号分隔组合背景值；如果要指定 background-color，应包含在最后一个背景中
background-attachment scroll、fixed 或 inherit	用于决定背景图像是否可以滚动，以及滚动的方式 初始值：scroll；不继承的 如果要显示多重背景，可以为每个背景应用不同的 background-attachment 值（用逗号分隔不同的值）
background-color 颜色值、transparent 或 inherit	用于设置元素的背景颜色 初始值：transparent；不继承的
background-image URL、CSS 渐变、none 或 inherit	用于设置元素的背景图像 初始值：none；不继承的 如果要显示多重背景，可使用逗号分隔图像值

(续)

属性 / 值	描述和注释
background-position 一个或两个百分数、长度（或一个百分数和一个长度），或 top、center、bottom 之一和（或）left、center、right 之一，或使用 inherit	用于设置指定的背景图像的物理位置 初始值：0% 0%；如果仅设置一个百分数，它会用于水平位置，而垂直位置的初始值则被设为 50%；如果仅使用一个关键字，另一个的初始值则为 center；应用于块级和替换元素；不继承的；百分数相对于盒本身的尺寸 如果要显示多重背景，可以为每个背景应用不同的 background-position 值（用逗号分隔不同的值）
background-repeat repeat、repeat-x、repeat-y、no-repeat 或 inherit 之一	用于确定背景图像是否重复及重复方式 初始值：repeat；不继承的 如果要显示多重背景图像，可以为每个背景图像应用不同的 background-repeat 值（用逗号分隔不同的值）
background-size 一个或两个百分数或长度，或 auto，或使用 cover 或 contain	用于指定背景图像的尺寸 初始值：auto；不继承的 如果要显示多重背景图像，可以为每个背景图像应用不同的 background-size 值（用逗号分隔不同的值）
border 任何 border-width、border-style 值和（或）颜色值的组合，或 inherit	用于定义元素四边边框的所有要素 初始值取决于单独的属性；不继承的
border-color 一至四个颜色值、transparent 或 inherit	用于指定元素的一个或多个边的边框的颜色 初始值：元素的 color 属性；不继承的
border-radius	用于为盒创建圆角 初始值：0；不继承的
border-top-right-radius border-bottom-right-radius border-bottom-left-radius border-top-left-radius	用于为盒的一个角设置 border-radius 值 初始值：0；不继承的 注：Firefox 的旧版本使用与此不同的语法创建单独的圆角：-moz-border-radius-topright、-moz-border-radius-bottomright、-moz-border-radius-bottomleft、-moz-border-radius-topleft
border-spacing 一个或两个长度，或 inherit	用于指定表格边框之间的空隙大小 初始值：0；仅可应用于表格元素；继承的
border-style 一至四个以下的值：none、dotted、dashed、solid、double、groove、ridge、inset、outset、inherit	用于为元素的一个或多个边设置边框样式 初始值：none；不继承的
border-top、border-right、border-bottom、border-left 任何用于 border-width、border-style 的单个值和（或）颜色值的组合，或使用 inherit	用于为元素的一个边一次性定义全部三个边框属性 初始值取决于单独的值；不继承的
border-top-color、border-right-color、border-bottom-color、border-left-color 颜色值或 inherit	用于为元素的一个边定义边框颜色 初始值：color 属性的值；不继承的
border-top-style、border-right-style、border-bottom-style、border-left-style none、dotted、dashed、solid、double、groove、ridge、inset、outset 或 inherit	用于为元素的一个边定义边框样式 初始值：none；不继承的

(续)

属性 / 值	描述和注释
border-top-width、border-right-width、border-bottom-width、border-left-width thin、medium、thick 或长度	用于为元素的一个边定义边框宽度 初始值: medium; 不继承的
border-width 一至四个以下的值: thin、medium、thick 或长度	用于为元素的一个或四个边定义边框宽度 初始值: medium; 不继承的
bottom 百分数、长度、auto 或 inherit	用于设置元素相对于其父元素底部边缘的位移大小 初始值: auto; 不继承的; 百分数相对于包含块的高度
box-shadow 可选的 inset, 接着是二至四个长度值, 接着是颜色值	用于为盒添加一个或多个阴影。长度值 (依次) 表示相对于盒右侧的位置 (负数则表示相对于盒左侧的位置)、相对于盒底部的位置 (负数则表示相对于盒顶部的位置)、模糊半径 (不可为负数) 和伸展距离 (负数会让阴影收缩)。每个 box-shadow 值之间用逗号分隔 初始值: none; 继承的
clear none、left、right、both 或 inherit	用于防止元素包围在浮动元素的一边或两边 初始值: none; 只能应用于块级元素; 不继承的
clip auto、rect 或 inherit	用于仅显示元素的一部分 初始值: auto; 只能应用于绝对定位的元素
color 颜色值或 inherit	用于设置元素的文本颜色 初始值: 父元素的颜色; 有的颜色是由浏览器设置的; 继承的
cursor auto、crosshair、default、pointer、progress、move、e-resize、ne-resize、nw-resize、n-resize、se-resize、sw-resize、s-resize、w-resize、text、wait、help、URL 或 inherit 之一	用于设置指针的形状 初始值: auto; 继承的
display inline、block、inline-block、list-item、run-in、compact、table、inline-table、table-row-group、table-header-group、table-footer-group、table-row、table-column-group、table-column、table-cell、table-caption、ruby、ruby-base、ruby-text、ruby-base-group、ruby-text-group、none、inherit 之一	用于确定元素如何显示, 以及是否显示 初始值: 通常为 inline 或 block; 不继承的
float left、right、none、inherit 之一	用于确定元素向父元素的哪一边浮动 初始值: none; 不可应用于定位过的元素 ^① 或生成的内容; 不继承的
font 如果需要, 任何 font-style、font-variant 和 font-weight 值的组合, 接着是必需的 font-size、可选的 line-height 值和必需的 font-family, 或使用 inherit	用于设置文本的字体系列、字体大小 (这二者是必需的) 及可选的字体样式、变体、粗细和行高 初始值取决于单独的属性; 继承的; font-size 和 line-height 可使用百分数; font-size 和 font-family 是必需的, 否则 font 属性是无效的

① 即设为绝对定位、相对定位或固定定位的元素。——译者注

(续)

属性 / 值	描述和注释
font-family 一个或多个由引号包着的字体名称, 接着是可选的表示类属的字体名称, 或使用 inherit	用于为文本选择字体系列 初始值: 取决于浏览器; 继承的
font-size 绝对大小、相对大小、长度、百分数或 inherit	用于设置文本的大小 初始值: medium; 计算的值是继承的; 百分数相对于父元素的字体大小
font-style normal、italic、oblique 或 inherit	用于将文本标记为斜体 初始值: normal; 继承的
font-variant normal、small-caps 或 inherit	用于设置小型大写字母 初始值: normal; 继承的
font-weight normal、bold、bolder、lighter、100、200、300、400、500、600、700、800、900 或 inherit	用于应用、移除、调整粗体格式 初始值: normal; 数字值当做关键字而非整数进行处理 (例如, 不能使用 150); 继承的
height 长度、百分数、auto 或 inherit	用于设置元素的高度 初始值: auto; 可应用于除了非替换行内元素、表格列和列组以外的任何元素; 不继承的
left 长度、百分数、auto 或 inherit	用于设置元素相对于其父元素左侧边缘的位移大小 初始值: auto; 只能用于定位过的元素; 不继承的; 百分数相对于包含块的宽度
letter-spacing normal、长度或 inherit	用于设置字母之间的间隙大小 初始值: normal; 继承的
line-height normal、数字、长度、百分数或 inherit	用于设置文本行之间的距离 初始值: normal; 继承的; 百分数相对于元素自身的字体大小
list-style 任何 list-style-type、list-style-position 和 (或) list-style-image 值的组合, 或使用 inherit	用于设置列表的标识 (常规的或定制的) 及其位置 初始值取决于单独元素的初始值; 只能应用于列表元素; 继承的
list-style-image URL、none 或 inherit	用于为列表指定定制的标识 初始值: none; 只能应用于列表元素; 覆盖 list-style-type; 继承的
list-style-position inside、outside 或 inherit	用于确定列表标识的位置 初始值: outside; 只能应用于列表元素; 继承的
list-style-type disc、circle、square、decimal、lower-roman、upper-roman、lower-alpha、upper-alpha、none 或 inherit	用于设置列表的标识 初始值: disc; 只能应用于列表元素; 如果 list-style-type 是有效的则不使用; 继承的
margin 一至四个以下的值: 长度、百分数、auto 或 inherit	用于设置元素与其父元素和 (或) 同胞元素之间在一个或多个边上的间隔大小 初始值取决于浏览器和 width 值; 不继承的; 百分数相对于包含块的宽度

(续)

属性 / 值	描述和注释
margin-top、margin-right、margin-bottom、margin-left 长度、百分数、auto 或 inherit	用于设置元素与其父元素和（或）同胞元素之间在一个边上的间隔大小 初始值：0；不继承的；百分数相对于包含块的宽度；如果 width、margin-right 和 margin-left 之和大于父元素的包含块，则 margin-right 和 margin-left 的值会被覆盖
max-height、max-width 长度、百分数、none 或 inherit	分别用于设置元素的最大高度和（或）最大宽度 初始值：none；不能用于行内元素或表格元素；不继承的；百分数相对于包含块的高度 / 宽度
min-height、min-width 长度、百分数或 inherit	分别用于设置元素的最小高度和（或）最小宽度 初始值：none；不能用于行内元素或表格元素；不继承的；百分数相对于包含块的高度 / 宽度
opacity 0.0（表示完全透明）至 1.0（表示完全不透明）之间的任何小数	用于让元素半透明或不可见 初始值：1；不继承的
orphans 整数或 inherit	用于指定元素可以单独出现在页面底部的行数 初始值：2；只能用于块级元素；继承的；仅用于打印媒体
overflow visible、hidden、scroll、auto 或 inherit	用于确定当内容超出元素内容区域时额外的内容如何显示 初始值：visible；只能用于块级元素和替换元素；不继承的
padding 一至四个长度或百分数，或使用 inherit	用于指定元素内容区域和边框之间在一个或多个边上的距离 初始值取决于浏览器；不继承的；百分数相对于包含块的宽度
padding-top、padding-right、padding-bottom、padding-left 长度、百分数或 inherit	用于指定元素内容区域和边框之间在一个边上的距离 初始值：0；不继承的；百分数相对于包含块的宽度
page-break-after、page-break-before always、avoid、auto、right、left 或 inherit	用于指定什么时候应出现分页，什么时候不应出现 初始值：auto；只能用于块级元素；不继承的；仅用于打印媒体
page-break-inside avoid、auto 或 inherit	阻止跨页的元素产生分页 初始值：auto；只能用于块级元素；继承的；仅用于打印媒体
position static、relative、absolute、fixed 或 inherit	用于确定元素如何相对于文档流进行定位 初始值：static；不继承的
right 长度、百分数、auto 或 inherit	用于设置元素相对于其父元素右侧边缘的位移大小 初始值：auto；只能用于定位过的元素；不继承的；百分数相对于包含块的宽度
table-layout fixed、auto 或 inherit	用于选择确定单元格宽度的算法 初始值：auto；不继承的
text-align left、right、center、justify、字符串或 inherit	用于指定文本对齐方式 初始值取决于浏览器和书写方向；只能应用于块级元素；继承的
text-decoration 任何 underline、overline、line-through 和 blink 的组合，或 none、inherit	用于修饰文本（大多数为线条） 初始值：none；不继承的

(续)

属性 / 值	描述和注释
text-indent 长度、百分数或 inherit	用于设置段落第一行的缩进量 初始值: 0; 只能应用于块级元素; 继承的; 百分数相对于包含块的宽度
text-overflow clip、ellipsis 或 "string"	用于指定文本不可见时处理溢出的方式 初始值: clip
text-shadow 两个或四个长度值, 接着是颜色值	用于为元素的文本添加一个或多个阴影。长度值 (依次) 表示相对于文本右侧的位置 (负数则表示相对于文本左侧的位置)、相对于文本底部的位置 (负数则表示相对于文本顶部的位置)、模糊半径 (不可为负数) 和伸展距离 (负数会让阴影收缩)。每个 text-shadow 值之间用逗号分隔 初始值: none; 继承的
text-transform capitalize、uppercase、lowercase、none 或 inherit	用于设置元素的文本的大小写 初始值: none; 继承的
transform none 或一系列变形功能 (matrix、translate、translateX、translateY、scale、scaleX、scaleY、rotate、skew、skewX、skewY)	用于对元素进行形状、大小或方向上的变形 初始值: none; 不继承的; 变形功能按照它们所列的顺序进行应用
transform-origin 一个或两个百分数或长度 (或一个百分数和一个长度), 或 top、center、bottom 之一和 (或) left、center、right 之一	用于定义应用于元素的变形的起点 初始值: 50% 50%; 不继承的; 只能应用于块级元素和行内元素; 百分数相对于元素盒的大小
transition 依次定义 transition-property、transition-duration、transition-timing-function 和 transition-delay 的简记法 (用空格分隔)	用于为元素定义变形效果 初始值取决于单独的属性; 可应用于所有的元素, 包括 :before 和 :after 伪元素; 值的顺序对此属性很重要
transition-property none、all 或用逗号分隔的一组 CSS 属性	用于识别在应用了变形的元素上定义的 CSS 属性 初始值: all; 不继承的; 可应用于所有的元素, 包括 :before 和 :after 伪元素
transition-duration 以秒或毫秒为单位的时间值	用于确定完成变形所需的时间 初始值: 0s (0 秒); 不继承的; 可应用于所有的元素, 包括 :before 和 :after 伪元素
transition-timing-function ease、linear、ease-in、ease-out、ease-in-out、cubic-bezier(number, number, number, number)	描述用于变形计算过程的中间值的使用方法 初始值: ease; 可应用于所有的元素, 包括 :before 和 :after 伪元素
transition-delay 以秒或毫秒为单位的时间值	用于定义变形开始的时间 初始值: 0s (0 秒); 不继承的; 可应用于所有的元素, 包括 :before 和 :after 伪元素
top 长度、百分数、auto 或 inherit	用于设置元素相对于其父元素顶部边缘的位移大小 初始值: auto; 只能用于定位过的元素; 不继承的; 百分数相对于包含块的高度

(续)

属性 / 值	描述和注释
vertical-align baseline、sub、super、top、text-top、middle、bottom、text-bottom、百分数、长度或 inherit	用于指定元素在垂直方向上的对齐方式 初始值: baseline; 不能应用于行内元素和表格单元格元素; 不继承的; 百分数相对于元素的 line-height 属性
visibility visible、hidden、collapse 或 inherit	用于在不将元素移出文档流的情况下让元素不可见 初始值: inherit, 事实上是不继承的 (仍存争议)
white-space normal、pre、nowrap、pre-wrap、pre-lined 或 inherit	用于指定如何处理空格 初始值: normal; 只能用于块级元素; 继承的
widows 整数或 inherit	用于指定元素可以单独出现在页面顶部的行数 初始值: 2; 只能用于块级元素; 继承的; 仅用于打印媒体
width 长度、百分数、auto 或 inherit	用于设置元素的宽度 初始值: auto; 不能应用于行内元素、表格行或行组; 不继承的; 百分数相对于包含块的宽度
word-spacing normal、长度或 inherit	用于设置单词之间的距离 初始值: normal; 继承的
z-index auto、整数或 inherit	用于设置元素相对于重叠元素的深度 初始值: auto; 只能应用于定位了的元素; 不继承的

表 B.1 是根据 www.w3.org/TR/CSS21/propidx.html 提供的完整规范制订的, 版权由万维网联盟 (美国麻省理工学院、法国国家计算机科学与控制研究所、日本庆应义塾大学) 所有。保留所有权利。

B.2 CSS 选择器和结合符

表 B.2 CSS 选择器和结合符

模 式	含 义	CSS3	选择器类型
*	任何元素		通用选择器
E	类型为 E 的元素		类型选择器
E[foo]	带 "foo" 属性的 E 元素		属性选择器
E[foo="bar"]	"foo" 属性值恰为 "bar" 的 E 元素 (引号是可选的)		属性选择器
E[foo ~ ="bar"]	"foo" 属性为一组空格分隔的值, 且其中之一恰为 "bar" 的 E 元素 (引号是可选的)		属性选择器
E[foo^="bar"]	"foo" 属性以 "bar" 开头的 E 元素 (引号是可选的)	是	属性选择器
E[foo\$="bar"]	"foo" 属性以 "bar" 结束的 E 元素 (引号是可选的)	是	属性选择器

模 式	含 义	CSS3	选择器类型
E[foo*="bar"]	"foo" 属性值在某处包含 "bar" 的 E 元素（引号是可选的）	是	属性选择器
E[foo "en"]	"foo" 属性为一组连字符分隔的值且（从左边开始）以 "en" 开头的 E 元素（引号是可选的）		属性选择器
E:root	E 元素，文档根元素	是	结构伪类
E:nth-child(n)	E 元素，其父元素的第 n 个子元素	是	结构伪类
E:nth-last-child(n)	E 元素，其父元素的倒数第 n 个子元素	是	结构伪类
E:nth-of-type(n)	E 元素，该类型的第 n 个同胞元素	是	结构伪类
E:nth-last-of-type(n)	E 元素，该类型的倒数第 n 个同胞元素	是	结构伪类
E:first-child	E 元素，其父元素的第一个子元素		结构伪类
E:last-child	E 元素，其父元素的最后一个子元素	是	结构伪类
E:first-of-type	E 元素，该类型的第一个同胞元素	是	结构伪类
E:last-of-type	E 元素，该类型的最后一个同胞元素	是	结构伪类
E:only-child	E 元素，其父元素的唯一子元素	是	结构伪类
E:only-of-type	E 元素，该类型的唯一同胞元素	是	结构伪类
E:empty	没有子元素（含文本结点）的 E 元素	是	结构伪类
E:link, E:visited	作为目标尚未访问过（:link）或已经访问过（:visited）的超链接的 E 元素		链接伪类
E:focus, E:hover, E:active	处于特定用户操作下的 E 元素		用户操作伪类
E:target	作为引用 URI 目标的 E 元素	是	目标伪类
E:lang(fr)	语言为 "fr" 的 E 元素		:lang() 伪类
E:enabled E:disabled	状态为有效的或无效的用户界面 E 元素	是	UI 元素状态伪类
E:checked	选中了的用户界面 E 元素（如单选按钮或复选框）	是	UI 元素状态伪类
E::first-line	E 元素在格式上的第一行		::first-line 伪元素
E::first-letter	E 元素在格式上的第一个字母		::first-letter 伪元素
E::before	E 元素之前的生成内容		::before 伪元素
E::after	E 元素之后的生成内容		::after 伪元素
E.warning	类为 "warning" 的 E 元素		类选择器
E#myid	ID 等于 "myid" 的 E 元素		ID 选择器
E:not(s)	与简单选择器 s（例如 input:not(.warning)）不匹配的 E 元素	是	否定伪类

(续)

模 式	含 义	CSS3	选择器类型
E F	作为 E 元素后代的 F 元素		后代结合符
E > F	作为 E 元素子元素的 F 元素		子元素结合符
E + F	紧接 E 元素后面的 F 元素		相邻同胞元素结合符
E ~ F	位于 E 元素后面的 F 元素	是	通用同胞元素结合符

表 B.2 是根据 www.w3.org/TR/css3-selectors/ 提供的 CSS3 选择器模型制订的，版权由万维网联盟（美国麻省理工学院、法国国家计算机科学与控制研究所、日本庆应义塾大学）所有。保留所有权利。

B.3 CSS3 颜色值

表 B.3 CSS3 颜色值

颜色值	描述和注释
rgb(red-value, green-value, blue-value)	RGB（红、绿、蓝）颜色模式 值可以是 0 到 255 之间的数字或百分数（不能是数字和百分数的组合） rgb(0, 0, 0) 和 rgb(0%, 0%, 0%) 为黑色 rgb(255, 255, 255) 和 rgb(100%, 100%, 100%) 为白色
rgba(red-value, green-value, blue-value, alpha)	RGB 颜色模式，加上 alpha 透明度 颜色值同 RGB 语法相同 第四个参数 alpha 是大于等于 0.0（完全透明）且小于等于 1.0（完全不透明）的小数
hsl(hue-value, saturation-value, lightness-value)	HSL（色相、饱和度、亮度）颜色模式 色相值用颜色环的角度（0 至 360 之间的数字）表示：0 和 360 为红色，120 为绿色，240 为蓝色，位于之间的其他值表示其他颜色 饱和度值用百分数表示：0% 为灰色，100% 为完全饱和的颜色 亮度值用百分数表示：0% 为黑色，100% 为白色，50% 为“正常”
hsla(hue-value, saturation-value, lightness-value, alpha)	HSL 颜色模式，加上 alpha 透明度 颜色值同 HSL 语法相同 第四个参数 alpha 是大于等于 0.0（完全透明）且小于等于 1.0（完全不透明）的小数

B.4 媒体查询

表 B.4 媒体查询

特 性	描述和注释
width min-width max-width 长度	输出设备的目标显示区域的宽度、最小宽度或最大宽度 应用：可视媒体和触觉媒体
height min-height max-height 长度	输出设备的目标显示区域的高度、最小高度或最大高度 应用：可视媒体和触觉媒体

(续)

特 性	描述和注释
device-width min-device-width max-device-width 长度	输出设备的呈现表面的宽度、最小宽度或最大宽度 应用：可视媒体和触觉媒体
device-height min-device-height max-device-height 长度	输出设备的呈现表面的高度、最小高度或最大高度 应用：可视媒体和触觉媒体
orientation portrait 或 landscape	当 height 特性值大于或等于 width 特性值时，方向为 portrait； 否则为 landscape 应用：位图媒体
aspect-ratio min-aspect-ratio max-aspect-ratio 比例（如 4/3 或 16/9）	width 特性值与 height 特性值的比例、最小比例或最大比例 应用：位图媒体
device-aspect-ratio min-device-aspect-ratio max-device-aspect-ratio 比例（如 4/3 或 16/9）	device-width 特性值与 device-height 特性值的比例、最小比例或 最大比例 应用：位图媒体
color min-color max-color 整数	输出设备每种颜色的位数、最小位数或最大位数；如果设备不是 彩色的设备，则值为 0 应用：可视媒体
color-index min-color-index max-color-index 整数	输出设备颜色查询表中项目的数量、最小数量或最大数量；如果 设备不使用颜色查询表，则值为 0 应用：可视媒体
monochrome、 min-monochrome、 max-monochrome 整数	在单色帧缓冲中每像素的位数、最小位数或最大位数；如果设备 不是单色的设备，则值为 0 应用：可视媒体
resolution min-resolution max-resolution 分辨率（如 300dpi 或 118dpcm）	输出设备的分辨率、最小分辨率或最大分辨率（即像素密度）； resolution（不是 min-resolution 或 max-resolution）不会检测使 用非方形像素的设备
scan progressive 或 interlace	电视输出设备的扫描过程 应用：电视媒体
grid 0 或 1	设备是基于栅格的还是基于位图的；如果输出设备是基于栅格的 （如 TTY 终端）则值为 1，否则值为 0；这种媒体查询也可以使用 不带值的形式表示（例如 @media grid） 应用：可视媒体和触觉媒体

关注图灵教育 关注图灵社区

iTuring.cn

在线出版 电子书《码农》杂志 图灵访谈 ……



—— QQ联系我们 ——

读者QQ群: 218139230



—— 微博联系我们 ——

官方账号: @图灵教育 @图灵社区 @图灵新知

市场合作: @图灵袁野 @图灵刘紫凤

写作本版书: @图灵小花 @陈冰_图书出版人

翻译英文书: @李松峰 @朱巍ituring @楼伟珊

翻译日文书或文章: @图灵乐馨

翻译韩文书: @图灵陈曦

电子书合作: @hi_jeanne

图灵访谈/《码农》杂志: @李盼ituring

加入我们: @王子是好人



—— 微信联系我们 ——



图灵教育
turingbooks



图灵访谈
ituring_interview



HTML5与CSS3基础教程

(第8版)

“我们这些有过不少实际经验的设计师往往想当然地认为自己什么都知道，事实并非如此。本书告诉我们，我们知道的很多东西其实都是错的。所有Web设计师都需要看看这本书。”

——Web标准计划创始人Jeffrey Zeldman
对本书第6版的评价

“这本书让我读起来很踏实。”

——豆瓣读者对本书第7版的评价

“自从2006年购买第5版，我就认准这本书了。第8版不论从内容组织还是示例讲解上都有不少惊喜，从中可以看出作者在新版本上耗费了大量心血！”

——亚马逊读者评论

“本书介绍如何设计、组织以及格式化网站，是Web开发方面的经典教程。”

——hostucan.net

本书是风靡全球的HTML和CSS入门教程的最新版，至第6版累积销量已超过100万册，被翻译为十多种语言，长期雄踞亚马逊书店计算机图书排行榜榜首。

第8版秉承作者直观透彻、循序渐进、基础知识与案例实践紧密结合的讲授特色，采用独特的双栏图文并排方式，手把手指导读者从零开始轻松入门。相较第7版，全书2/3以上的内容进行了更新，全面反映了HTML5和CSS3的最新特色，细致阐述了响应式Web设计与移动开发等热点问题。书中主要内容包括：如何创建HTML5页面，如何使用HTML5元素，如何用CSS3为网页添加样式，如何向页面添加JavaScript代码，如何测试做好的页面并将其上传到万维网。另外，本书强调渐进增强这种网站设计方法的重要性，并将其贯穿在全书的具体实践中。

作者专为本书设计了内容丰富的配套网站htmlcssvqs.com，提供海量精彩示例、HTML与CSS元素及属性列表以及其他附加材料，方便读者随时参考与引用。



图灵社区: iTuring.cn

热线: (010)51095186转600

分类建议 计算机/Web开发

人民邮电出版社网址: www.ptpress.com.cn

ISBN 978-7-115-35065-7



ISBN 978-7-115-35065-7

定价: 69.00元

图灵社区

欢迎加入

电子书发售平台

电子出版的时代已经来临，在许多出版界同行还在犹豫彷徨的时候，图灵社区已经采取实际行动拥抱这个出版业巨变。相比纸质书，电子书具有许多明显的优势。它不仅发布快，更新容易，而且尽可能采用了彩色图片（即使有的书纸质版是黑白印刷的）。读者还可以方便地进行搜索、剪贴、复制和打印。

图灵社区进一步把传统出版流程与电子出版业务紧密结合，目前已实现作译者网上交稿、编辑网上审稿、按章发布的电子出版模式。这种新的出版模式，我们称之为“敏捷出版”，它可以让读者以较快的速度了解到国外最新技术图书的内容，弥补以往翻译版技术书“出版即过时”的缺憾。同时，敏捷出版使得作、译、编、读的交流更为方便，可以提前消灭书稿中的错误，最大程度地保证图书出版的质量。

开放出版平台

图灵社区向读者开放在线写作功能，协助你实现自出版的梦想。你可以联合二三好友共同创作一部技术参考书，以免费或收费的形式提供给读者，这极大地降低了出版的门槛。成熟的书稿，有机会入选出版计划，同时出版纸质书。

图灵社区引进出版的外文图书，都将在立项后马上在社区公布。如果有意翻译哪本图书，欢迎来社区申请。只要通过试译的考验，即可签约成为图灵的译者。当然，要想成功地完成一本书的翻译工作，是需要有坚强的毅力的。

读者交流平台

在图灵社区，读者可以十分方便地写文章、提交勘误、发表评论，以各种方式与作译者、编辑人员和其他读者进行交流互动。提交勘误还能够获赠社区银子。欢迎大家积极参与社区开展的访谈、审读、评选等多种活动，赢取银子，可以换书哦！