

- 1 <!-- --> 用于注释
- 2 元素=开始标记+内容+结束标记
- 3 简记写法
- 4 css : Cascading Style Sheets
- 5 html中路径使用 /
- 6 p87 <q>短引用的作用
- 7 <blockquote>用于长引用 单独显示
- 8
 换行
 和
没有区别
- 9 列表
-

...

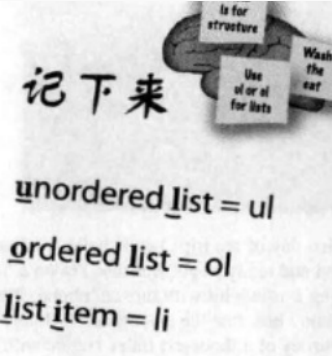
...

...

- 10 和 只能包含
- 同时中可以嵌套 和
- 11 标签告诉浏览器把其中的文本表示为强调的内容。对于所有浏览器来说，这意味着要把这段文字用斜体来显示。
- 12 内联元素与块元素
- <http://www.cnblogs.com/2050/archive/2010/08/24/1806893.html>
- 13 baidu.com是域名， www.baidu.com是网站
- 14 查询已注册的域名 <http://www.internic.net/regist.html>
- 15 href连接到本页目标或他页目标是按照 元素属性的 id 来辨别位置的
- href="http://...."#id来连接目标
- 元素的属性顺序无关
- 16 title属性作为提示，提示连接
- 17 css是使用/**/注释
- 18 特定级别全部相同时 css会选择最后一个匹配的

8 9章尚未看

div中样式以#id作为表示



问：为什么CSS规则前面有一个“body”？这是什么意思？

答：CSS中的“body”表示“{”和“}”之间的所有CSS要应用于HTML <body>元素中的内容。所以，将字体设置为 sans-serif 时，就是说页面主体中的默认字体是 sans-serif。

- HTML和CSS是我们用来创建网页的语言。
- Web服务器存储并提供由HTML和CSS创建的网页。浏览器获取页面，并根据HTML和CSS显示网页的内容。
- HTML是超文本标记语言（HyperText Markup Language）的缩写，用来建立网页的结构。
- CSS是层叠样式表（Cascading Style Sheets）的缩写，用来控制HTML的表现。
- 通过HTML，我们利用标记来标示内容提供结构。我们把匹配标记以及它们包围的内容称为元素。
- 元素由3部分组成：一个开始标记、内容和一个结束标记。不过有些元素（比如- 开始标记可以有属性。我们已经见过一个属性：type。
- 结束标记在左尖括号后面、标记名前面有一个“/”，以明确这是结束标记。
- 所有页面都要有一个<html>元素，其中要有一个<head>元素和一个<body>元素。
- 网页的信息放在<head>元素里。
- <body>元素里的内容就是你将浏览器里看到的东西。
- 大多数空白符（制表符、回车、空格）都会被浏览器忽略，不过可以利用空白符让你的HTML（对你）更有可读性。
- 可以在<style>元素中写CSS规则，为HTML网页增加CSS。<style>元素总要放在<head>元素里。
- 可以使用CSS在HTML中指定元素的特性。

安全第一



属性的写法都是一样的：首先是属性名，后面是一个等于号，然后用双引号括起来的属性值。

你可能看到Web上有一些不严谨的HTML没有加双引号，不过你可不能偷懒。如果不严谨，可能导致你以后会遇到很多问题（本书后面就会看到）。

这样做（最佳实践）

`Great Movies`

Diagram illustrating the correct HTML attribute syntax: `Great Movies`. The components are labeled: `href` is the attribute name (属性名), `=` is the equals sign (等于号), `"` is the opening double quote (双引号), `top10.html` is the attribute value (属性值), and `"` is the closing double quote (双引号).

不要这样做

`Great Movies`

Diagram illustrating the incorrect HTML attribute syntax: `Great Movies`. The attribute value `top10.html` is noted as missing the surrounding double quotes (属性值周围没有双引号).

问：我见过很多页面上还可以单击图像而不只是文本。能用<a>元素来实现吗？

答：可以，如果把一个元素放在<a>标记之间，这个图像就会像文本一样可单击。我们不打算花好几章来深入讨论图像，不过要知道，图像确实也可以作为链接。

问：那么，为什么不在各个其他文件夹中创建一个图像文件夹呢，比如在“about”和“beverages”下分别创建一个图像文件夹？

答：同样地，当然也可以这么做。我们希望某些图像能在多个页面上重用，另外，有些浏览器对路径中的字符个数会做出限制。规范中建议当心超过的一个文件夹里集中维护。如果你的255个字符，但现代浏览器能支持更大网站需要在不同部分使用大量图像，的长度。不过，如果你有一个大型网站，则还是要注意这一点。

问：这么说任何东西只要放在<a>标记之间就会成为可单击的，是吗？比如说，一个段落？

答：确实可以把一个<p>元素放在<a>元素中来链接整个段落。我们大多会在<a>元素中使用文本和图像（或者二者同时使用），不过，如果你需要链接一个<p>或<h1>元素，则完全可以。哪些标记可以嵌在其他标记里，这完全是另一个话题了，不过不用担心，很快就会谈到这个内容。

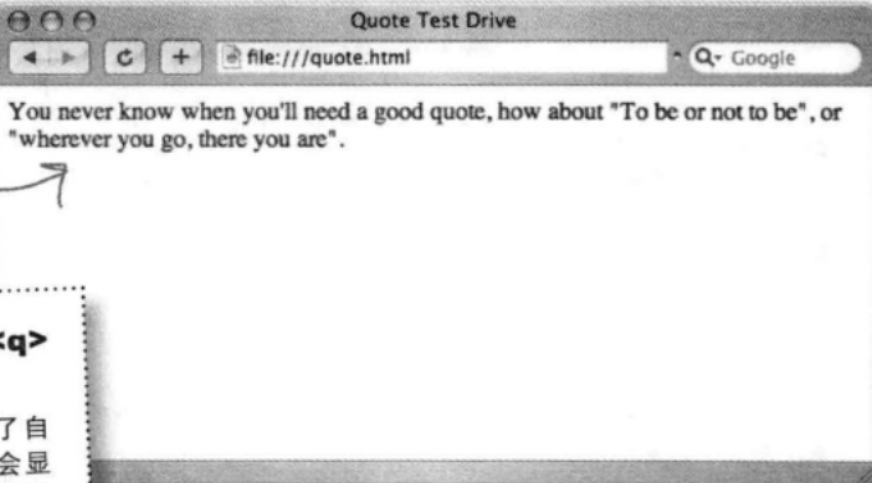
想从一个页面链接到另一个页面时，要使用<a>元素。

<a>元素的href属性指定了链接的目标文件。

<a>元素的内容是链接的标签。这个标签就是你在网页上看到的链接文本。默认地，这个标签会有下划线，指示这是可以单击的。

文字或图像都可以用作链接的标签。

这些引用在浏览器中会这样显示。注意浏览器会负责加上双引号。



Watch it!

并不是所有浏览器都会在`<q>`元素的内容两边加双引号。

这很糟糕，因为如果你增加了自己的双引号，有些浏览器就会显示两组双引号。建议你在不同的浏览器中对`<q>`做个测试，看看会得到什么结果。

问：你说我们可以用CSS对这些元素指定样式，这么说来，如果我希望让`<q>`元素中的文本用灰色斜体显示，可以用CSS来做到。不过，难道不能用``元素把引用置为斜体吗？

答：嗯，也可以，不过这么做不太好，因为你这样使用``元素只是为了实现显示效果，而不是因为你写了要强调的文字。如果你引用某人的话，其中确实强调了某个词，或者如果你想强调引用中的某个重点，那么完全可以在引用中使用``元素。不过，不要只是为了显示斜体而使用``元素。利用CSS，有很多更好、更容易的方法来实现你希望的元素显示效果。

的确，它没有内容。

`
`元素是一个没有任何内容的元素。为什么？因为它只是一个换行，没有其他内容。所以，如果一个元素设计为没有任何实际内容，我们会使用简写来表示这个元素，最后就会像`
`一样。毕竟，如果没有这个简写，每次需要一个换行时都需要写为`
</br>`，这有意义吗？

`
`并不是唯一没有实际内容的元素，还有很多类似这样的元素，我们把这些元素叫做void元素。实际上，之前我们已经见过另一个void元素，那就是``元素。后面几章还会详细讨论``元素。

要记住，之所以用简写不是因为懒惰，更多的是为了提高效率。这样表示void元素会更高效（输入的字符更少，相应的页面中字符数也更少）。实际上，读过HTML之后，你会发现这样看起来也更轻松。

原先它们称为“空元素”，显然不太贴切，所以后来又改名为void元素。就个人而言，我们还是喜欢空元素这个名字。

答:不行。世界上有两类元素：正常元素和void元素，正常元素比如<p>、<h1>和<a>，void元素如
和。不能在这两个阵营之间来回切换。例如，如果你只输入，这只是一个开始标记，没有内容或结束标记（这是不对的）。如果写为，这就是一个空元素，这样是可以的，不过放在页面上没有什么用处！

问:这么说，HTML有有序列表和无序列表。还有其他类型的列表吗？

答:实际上，确实还有另外一类列表：定义列表。定义列表如下所示：

```
<dl>
  <dt>Burma Shave Signs</dt>
  <dd>Road signs common in the U.S. in
the 1920s and 1930s advertising shaving
products.</dd>
  <dt>Route 66</dt>
  <dd>Most famous road in the U.S. highway
system.
</dd>
</dl>
```

列表中每一项都有一个定义术语<dt>和一个定义描述<dd>。

可以输入这个列表，试一试。

你说的没错，这确实会有问题。

因为浏览器要用<和>来开始和结束标记，如果在HTML内容中使用这两个字符，就会有问题。不过HTML提供了一种简便的方法，可以使用一种称为字符实体（character entity）的简单缩写来指定这样一些特殊字符。它是这样工作的：对于被认为“特殊”的字符，或者你可能希望在Web页面中使用某个字符但在你的编辑器里无法输入（比如版权符号），就可以查找相应的缩写在HTML中直接输入。例如，>字符的缩写是 >，<字符的缩写是<。

所以，假如你希望在页面中输入“The <html> element rocks.”。通过使用字符实体，可以这样输入：

The <html> element rocks.

还有一个重要的特殊字符需要知道，这就是&字符（与字符）。如果希望HTML内容中出现一个&，则可以使用字符实体&，而不要直接使用字符&。

那么版权符号呢（版权符号是©right;）？还有所有其他符号和外文字符呢？可以在下面这个URL找到一些常用的字符实体：

http://www.w3schools.com/tags/ref_entities.asp

或者，如果需要更详尽的字符实体清单，则可以访问这个URL：

<http://www.unicode.org/charts/>

答：是这样的，对于小型的网站，通常就是这么做。在向服务器上传文件之前，可以用你的计算机测试你做的修改，确保一切都能正常工作。对于比较大的网站，通常会创建一个测试网站和一个真实网站，这样在转向真实网站之前，可以先在测试网站上预览修改的结果。

如果你使用类似Dreamweaver或Coda等工具，则这些工具允许你在你自己的计算机上测试所做的修改，当你保存文件时，文件会自动传送到网站。

<http://www.starbuzzcoffee.com/index.html>

URL的第一部分指出用来获取资源的协议。

第二部分是网站名。现在你对这一部分应该很了解了。

第3部分是从根文件夹到资源的绝对路径。

确实如此。如果服务器接收到这样一个末尾没有“/”的请求，而且这个目录确实存在，服务器就会帮你加上末尾的斜线。所以如果服务器接收到以下请求：

```
http://www.starbuzzcoffee.com
```

它会把这个请求改为

```
http://www.starbuzzcoffee.com/
```

这会让服务器查找一个默认文件，最后会返回这个文件，就好像你之前输入了以下地址：

```
http://www.starbuzzcoffee.com/index.html
```

问：我们学过可以在[<a>](#)元素的href属性中放入相对路径。既然它们不是绝对路径，服务器是怎么找到的呢？

答：嗨，这个问题问得好。单击一个相对链接时，在后台浏览器会根据这个相对路径和所单击页面的路径创建一个绝对路径。所以，所有Web服务器

为链接增加标题以便访问

对于我们要单击的链接，如果能得到更多信息就好了。这对那些有视力障碍使用屏幕阅读器的人来说尤其重要，因为他们通常不希望把整个URL读出来：（“h” “t” “t” “p” “:” “slash” “slash” “w” “w” “w” “dot”），而链接的标签通常只能提供有限的描述，如“Caffeine Buzz”。

[<a>](#)元素有一个title属性，就是用来提供链接信息的。有些人可能被这个属性名搞糊涂了，因为<head>中还有一个名为<title>的元素。之所以名字相同，这是因为它们本来就是相关的，通常建议title属性值与所链接的Web页面的<title>元素值相同。不过这不是一个严格的要求，在title属性中提供你自己的更具体的描述可能更有意义。

可以像这样为[<a>](#)增加title属性：

```
Read the <a href="http://wickedlysmart.com/buzz"
title="Read all about caffeine on the Buzz">Caffeine Buzz</a>
```



title元素有一个值，这是所要链接的页面的文本描述。

你已经知道如何使用相对链接或URL链接到页面。不论是相对链接还是URL，要链接到页面中的一个特定目标，只需在链接最后加一个#，再加上目标标识符。所以，如果你想从任何Starbuzz Coffee Web页面链接到“chai”目标，可以这样写[<a>](#)元素链接：

```
<a href="index.html#chai">See Chai Tea</a>
```

问:为什么需要为Buzz URL增加“/index.html”来创建指向目标标题的链接呢？为什么不能直接写为：
`http://wickedlysmart.com/buzz#Coffee`？

答:不行，这不一定能正常工作，因为浏览器会帮你在URL末尾加斜线，这有可能取代id引用。不过，你可以这样写：`http://wickedlysmart.com/buzz/#Coffee`，类似于使用“index.html”创建的链接，这会得到同样的结果。如果你不知道默认文件是否是“index.html”，这就会很方便。

使用target打开新窗口

要在新窗口中打开一个页面，需要告诉浏览器你要打开的窗口名。如果没有告诉浏览器要使用哪个特定的窗口，浏览器就会在同一个窗口中打开这个页面。可以为<a>元素增加一个target属性，告诉浏览器使用一个不同的窗口。target属性值会告诉浏览器页面的“目标窗口”。如果使用“_blank”作为目标，浏览器就总是打开一个新窗口显示页面。下面来仔细看一下：

```
<a target="_blank" href="http://wickedlysmart.com/buzz"
title="Read all about caffeine on the Buzz">Caffeine Buzz</a>
```

target属性告诉浏览器在哪里打开href属性中链接指示的Web页面。如果没有target，浏览器就会在同一个窗口中打开这个链接。如果目标为“_blank”，浏览器就会在一个新窗口中打开链接。

there are no
Dumb Questions

问:如果多个<a>元素都有target呢？如果已经打开了一个“_blank”新窗口，那么会在已经打开的这个窗口中打开页面？还是会在另一个新的“_blank”窗口中打开？

答:如果所有<a>元素中的target都指定为“_blank”，那么每个链接都会在一个新的空窗口中打开。不过，这个问题问得很好，因为由此引出一个重点：不一定要把target指定为“_blank”。如果指定另一个名字，如“coffee”，那么有相同目标名“coffee”的所有链接都会在同一窗口中打开。这是因为，为target指定一个特定的名字时，如“coffee”，实际上就是在对显示链接页面的新窗口命名。“_blank”则是一种特殊情况，告诉浏览器总是使用一个新窗口。


```

```

alt属性需要指定描述这个图像的一些文本。

如果图像未能显示，就会使用这个文本来取代它。这有些像你通过电话为某个人读一个web页面，读到图像时，你会读出alt文本来代替图像。

问：不能直接用元素的width属性和height属性来调整图像的大小吗？

答：可以使用width属性和height属性来调整一个图像的大小，不过这不是一个好主意。为什么？因为如果这样做，则仍然会下载完整的大图像。让浏览器来完成调整图像大小的工作，就像你打开了浏览器上的自动调整大小选项一样（如果浏览器支持这个特性）。width属性和height属性实际上是帮助浏览器确定要为此图像预留多大的空间。如果使用这两个属性，则它们应当与图像的实际宽度和高度一致。

换句话说，我们并没有“捣鬼”。因为显示为内联元素，所以显示元素时不会在元素前后插入换行。因此，如果HTML中有多个图像，浏览器窗口足够宽时，浏览器会把它们并排摆放。

那些更大的照片之所以没有并排摆放，是因为浏览器没有足够的空间来并排显示它们。实际上，它只能从上到下显示各个大图像。浏览器总是在块元素前后在垂直方向上显示间距，如果再看看前面的截屏图，则你会看到那些图像尽管上下摆放，但是相互之间没有间距。这也说明了是一个内联元素。

让验证工具（和很多浏览器）接受<meta>标记……

<meta>标记放在<head>元素中（应该记得，<head>包含有关页面的信息）。把这个<meta>标记行增加到你的HTML中。下面首先在“lounge.html”文件中增加这个标记：

```
<!doctype html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Head First Lounge</title>
  </head>
```

这里是<meta>标记。我们把它增加到<head>元素中，放在<title>元素上面。

这一行要增加到<head>元素中所有其他元素的上面。

有一种技术，可以只为<h1>指定第二个规则

不需要分解“h1, h2”规则，我们只需要只为h1再增加另一个规则，为它增加边框样式。

```
h1, h2 {  
  font-family: sans-serif;  
  color: gray;  
}
```

第一个规则还是一样的。我们将使用一个合并规则指定<h1>和<h2>的font-family和color。

```
h1 {  
  border-bottom: 1px solid black;  
}
```

不过，现在要增加第二个规则，只为<h1>增加另一个属性：border-bottom属性。

```
p {  
  color: maroon;  
}
```

注意不要复制 <style>和</style>标记，因为“lounge.css”文件只包含CSS，不能包含HTML。

从“lounge.html”链接到外部样式表

现在我们需要告诉浏览器，它要利用外部样式表为这个页面增加样式。可以利用HTML <link>元素来做到。在你的HTML中使用<link>元素，如下所示：

```
<!doctype html>  
<html>  
  <head>  
    <meta charset="utf-8">  
    <title>Head First Lounge</title>  
    <link type="text/css" rel="stylesheet" href="lounge.css">  
    <style>  
    </style>  
  </head>
```

这是链接到外部样式表的HTML。

不再需要<style>元素，将它删除。

<p>元素中的元素从<p>继承了font-family样式

就像你可能会从你的父母那里继承蓝眼睛和金黄色的头发，同样的，元素也可以从它们的父元素继承样式。在这里，<a>和元素就从<p>元素继承了font-family样式，<p>元素就是它们的父元素。改变段落样式也会改变段落中元素的样式，这是有道理的，不是吗？毕竟，如果不是这样，你就只能事必躬亲，为整个网站中每一个段落中的每一个内联元素增加CSS规则……这太可怕了。

不是所有元素都继承。只有元素，如font-family。

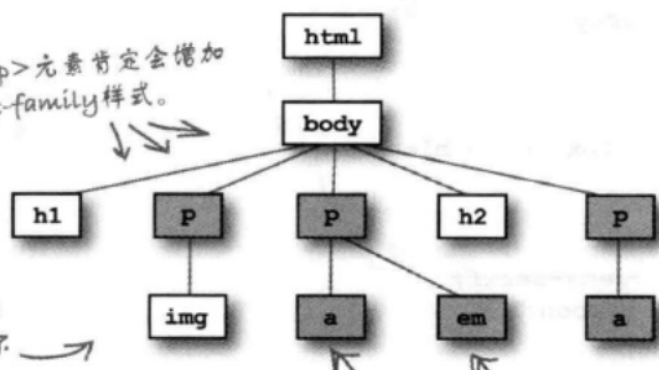
更不用说还容易出错，很麻烦，另外还浪费时间。

下面给出HTML树，看看继承到底是怎样工作的：

如果设置所有<p>元素的font-family，就会影响到下面阴影显示的所有元素。

当然，<p>元素肯定会增加这个font-family样式。

元素是段落的一个子元素，不过，它没有任何文本，所以它不受影响。



两个段落中的<a>和元素从其父元素（<p>元素）继承family。

创建一个类选择器

要在CSS中创建一个类，并选择这个类中的一个元素，可以编写一个类选择器，如下：

再用一个“.”指定一个类。最后是类名。

先选择这个类中的元素，在这里就是元素p。

```
p.greentea { color: green; }
```

就是这个规则……使greentea类中一个段落中的所有文本为绿色。

选择器p.greentea会选择greentea类中的所有段落。

不用，有一种更好的办法。如果希望greentea类中的所有元素都有同一种样式，可以这样写规则：

```
.greentea { color: green; }
```

如果省略所有元素名，只有一个点，后面是类名，那么这个规则会应用到这个类的所有成员。

如果多个选择器选择一个元素呢？

哈，之前就是这种情况，段落属于所有3个类：

```
<p class="greentea raspberry blueberry">
```

这里有多选择器与这个元素匹配，它们都同样定义了color属性。这就是我们所说的冲突。哪个规则会胜出呢？嗯，如果一个规则比其他规则更特定，它就会胜出。不过，“更特定”是什么意思？后面有一章还会来讨论这个内容，告诉你如何确定一个选择器的特定程度，不过现在先来看一些规则，使你对规则的特定程度有所认识：

```
p { color: black; }
```

这个规则会选择所有原来的段落元素。

```
.greentea { color: green; }
```

这个规则选择greentea类的所有成员。这个规则更特定一些。

```
p.greentea { color: green; }
```

这个规则只选择greentea类中的段落，所以这比前一个规则更特定。

```
p.raspberry { color: blue; }
```

这些规则也只是选择一个特定类中的段落。所以它们与p.greentea规则特定程度相同。

```
p.blueberry { color: purple; }
```

```
#elixirs {  
  border-width: thin;  
  border-style: solid;  
  border-color: #007e7e;  
  width: 200px;  
}
```

width属性允许你指定元素内容区的宽度。在这里，我们将内容宽度指定为200像素。

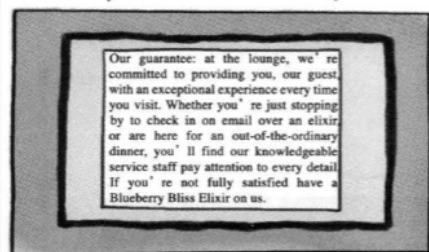
在elixirs <div>上设置这个样式。所以elixirs <div>中的内容将是200像素宽，浏览器的布局规则会起作用，使放在<div>中的所有元素都在这个宽度范围内。

假设在一个CSS规则中使用width属性将内容区宽度设置为300像素。

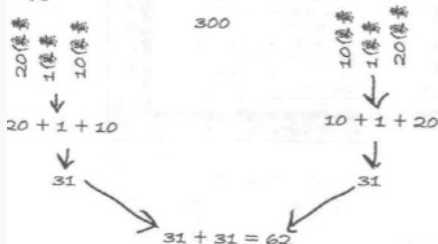
假设将外边距设置为20像素，内边距设置为10像素，另外有1像素宽的边框。这个元素盒子的宽度是多少？嗯，就是内容区的宽度，加上左和右内边距、左和右外边距以及左和右边框的宽度。下面来看如何计算……

(1) 内容区为300像素。

300像素



(2) 确定外边距、内边距和边框的宽度。



(3) 看来共占62像素，所以把它与内容区宽度300像素相加，可以得到300 + 62 = 362像素，这就是整个盒子的宽度。

要记住，你要将所有这些样式应用到elixirs <div>，使它们只影响<div>及其中包含的元素，而不是整个页面。

```
#elixirs {  
  border-width: thin;  
  border-style: solid;  
  border-color: #007e7e;  
  width: 200px;
```

```
padding-right: 20px;  
padding-bottom: 20px;  
padding-left: 20px;
```

```
margin-left: 20px;
```

```
text-align: center;
```

```
background-image: url(images/cocktail.gif);  
background-repeat: repeat-x;
```

<div>的默认内边距为0像素，所以我们要增加一些内边距，为内容提供一点空间。注意，是边没有增加内边距，因为那里空间已经足够大了，（这要归功于<h2>标题的默认外边距（返回去查看上一个测试，你会看到<h2>上面有很大空间）。不过在右边、下边和左边确实需要加内边距。

我们在左边增加了一些外边距，使elixirs相对于页面其他内容稍有些缩进。稍后会看到这很有用……

使用块元素的text-align来对齐其中包含的文本。这里我们打算让文本居中。

最后指定在背景中使用的一个图像，在这里就是鸡尾酒图像。我们将background-repeat属性设置为repeat-x，这使得图像只在水平方向上平铺。

问得好……这看起来有点不对劲，是不是？不过事实上，text-align会对块元素中的所有内联内容对齐。所以，我们对<div>块元素设置了这个属性后，它的所有内联内容都会居中。只是要记住，尽管这个属性的名字是text-align，但实际上它适用于对任何类型的内联元素对齐。还要记住一点：text-align属性只能在块元素上设置。如果直接在内联元素（如）上使用，则不起作用。

父元素和子元素名之间有一个空格。

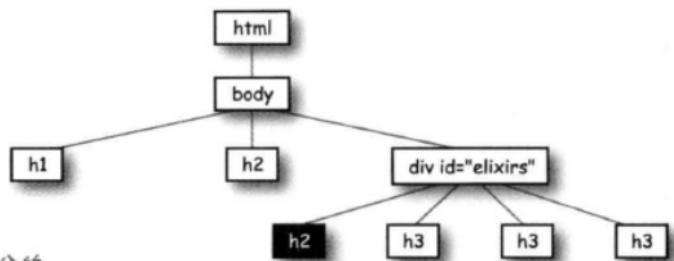
这是父元素。

这是它的子元素。

```
div h2 {  
  color: black;  
}
```

这个规则指出要选择作为一个<div>子元素的所有<h2>。

规则其余部分的写法与以往一样。



这是这个规则在休闲室页面上选择的元素。