

介绍



级联样式表（Cascading Style Sheets，CSS）是一种为网站添加布局效果的出色工具。它可以省去你大量时间，令你可以采用一种全新的方式来设计网站。CSS 是每个网页设计人员所必须掌握的。

本 CSS 教程能够让你在数小时内入门 CSS。它很容易理解，而且将向你介绍所有高级技巧。

学习 CSS 是很有趣的。你在学习本教程的过程中，记得留有充分的时间来实践你在每一课中学到的知识。

使用CSS**要求你有基本的HTML经验**。如果你不熟悉HTML，那么请在开始学习CSS之前先学习[我们提供的HTML教程](#)。

我需要什么软件？

在学习本教程时，请不要使用类似 FrontPage、DreamWeaver 或 Word 这样的软件。这些高级软件对你学习 CSS 没有帮助。相反，它们会限制你，并显著减慢你的学习进度。

你所需要的只是一个免费的简易文本编辑器。

例如，微软视窗操作系统（Microsoft Windows）自带的记事本（Notepad）程序。你可以依次点击“开始菜单→程序→附件”找到该程序。如果你用的是其他操作系统，那么也有类似的简易文本编辑器可用，比如 Linux 上的 Pico 或 Mac（苹果机）上的 Simple Text。

采用简易文本编辑器来学习编写 HTML 和 CSS 是非常好的，因为它不会影响或修改你所输入的代码。这样，代码的对错完全取决于你自己，而不是软件。

你可以使用任何浏览器。但我们建议你浏览器升级至最新版本。

你所需要的全部软件就是一个浏览器和一个简易文本编辑器。

我们现在开始学习 CSS 吧！

第 1 课：CSS 是什么？

也许你曾听说过 CSS，但并不真正清楚 CSS 到底是什么。在这一课，你将学到更多 CSS 的知识，并了解 CSS 可以做什么。

CSS 是 Cascading Style Sheets（级联样式表）的缩写。

可以用 CSS 做什么？

CSS 是一种样式表语言，用于为 HTML 文档定义布局。例如，CSS 涉及字体、颜色、边距、高度、宽度、背景图像、高级定位等方面。拭目以待吧！

HTML 可以用于为网站添加布局效果，但有可能被误用。而 CSS 则提供了更多选择，而且更为精确、完善。现在所有浏览器都支持 CSS。

经过下面若干课的学习之后，你将能够制作自己的 CSS 样式表，为自己的网站增添花样了。

CSS 跟 HTML 的区别在哪里？

HTML 用于**结构化**内容；CSS 用于**格式化**结构化的内容。

嗯，这听上去有点技术性并令人疑惑。不过没关系，我们继续学习。过会儿你就明白了。

在 Tim Berners-Lee 发明万维网（World Wide Web）的那个年代，HTML 语言是唯一用于给文本添加结构的语言。作者可以通过声明“这是一个标题”（利用 h1 标签）或“这是一个段落”（利用 p 标签）的方式来标记文本。

随着 Web 逐渐流行起来，网页设计者们开始寻求为网页增添布局的可能性。为此，当时的浏览器厂商们（网景公司和微软公司）发明了一些新的 HTML 标签（比如等），以引入了新的布局——而非新的结构。

这也导致了原本用于进行文本的结构化的标签（比如<table>等）越来越多地被误用于进行页面的布局。许多新的布局标签（比如<blink>）只有一种浏览器支持。“您需要使用某某浏览器来浏览本页面”成为当时常见于许多网站的声明。

CSS 的发明正是为了改善这一状况，它为 Web 设计师们提供了完善的、所有浏览器都支持的布局能力。而且，文档的表现样式与内容的分离，也令网站维护容易了许多。

采用 CSS 有哪些好处？

CSS 是 Web 设计界的一次革命。CSS 的具体优点包括：

- 通过单个样式表控制多个文档的布局；
- 更精确的布局控制；
- 为不同的媒体类型（屏幕、打印等）采取不同的布局；
- 无数高级、先进的技巧。

● 第 2 课：CSS 的工作原理

- 在这一课，你将学习如何制作自己的第一个样式表。你将了解基本的 CSS 模型，以及在 HTML 文档里使用 CSS 所必需的代码。
- 级联样式表（CSS）里用到的许多 CSS 属性都与 HTML 属性相似，所以，假如你熟悉采用 HTML 进行布局的话，那么这里的许多代码你都不会感到陌生。我们先来看一个具体的例子。

● 基本的 CSS 语法

- 比方说，我们要用红色作为网页的背景色；
- 用 HTML 的话，我们可以这样：

-
- `<body bgcolor="#FF0000">`
-

- 用 CSS 的话，我们可以这样获得同样的效果：

-
- `body {background-color: #FF0000;}`

-

- 你会注意到，HTML 和 CSS 的代码颇有几分相似。上例也向你展示了基本的 CSS 模型：

```
selector {property: value;}
```

↑ ↑ ↙

选择器：
表明花括号
中的属性设
置将应用于
哪些HTML元素
例如“body”

属性：
例如用于设置
背景色的属性
“background-color”
等等。

值：
比如说
background-color
属性的值可以是
“#FF0000”
代表红色）。

-
- 但是把 CSS 代码放在哪里呢？这正是我们下面要讲的。

- **为一个 HTML 文档应用 CSS**

- 为 HTML 文档应用 CSS，有三种方法可供选择。下面对这三种方法进行了概括。我们建议你对第三种方法（即外部样式表）予以关注。

- **方法 1：行内样式表（style 属性）**

- 为 HTML 应用 CSS 的一种方法是使用 HTML 属性 style。我们在上例的基础之上，通过行内样式表将页面背景设为红色：

```
<html>

<head>

<title>例子</title>

</head>

<body style="background-color: #FF0000;">

<p>这个页面是红色的</p>

</body>

</html>
```

- **方法 2：内部样式表（style 元素）**

- 为 HTML 应用 CSS 的另一种方法是采用 HTML 元素 style。比如像这样：

```
<html>
```

- `<head>`
- `<title>例子</title>`
- `<style type="text/css">`
- `body {background-color: #FF0000;}`
- `</style>`
- `</head>`
- `<body>`
- `<p>这个页面是红色的</p>`
- `</body>`
- `</html>`

- **方法 3：外部样式表（引用一个样式表文件）**

- 我们推荐采用这种引用外部样式表的方法。在本教程之后的例子中，我们将全部采用该方法。
- 外部样式表就是一个扩展名为 **css** 的文本文件。跟其他文件一样，你可以把样式表文件放在 Web 服务器上或者本地硬盘上。
- 例如，比方说你的样式表文件名为 **style.css**，它通常被存放于名为 **style** 的目录中。就像下面这样：



- 现在的问题是：如何在一个 HTML 文档里引用一个外部样式表文件（style.css）呢？答案是：在 HTML 文档里创建一个指向外部样式表文件的链接（link）即可，就像下面这样：

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

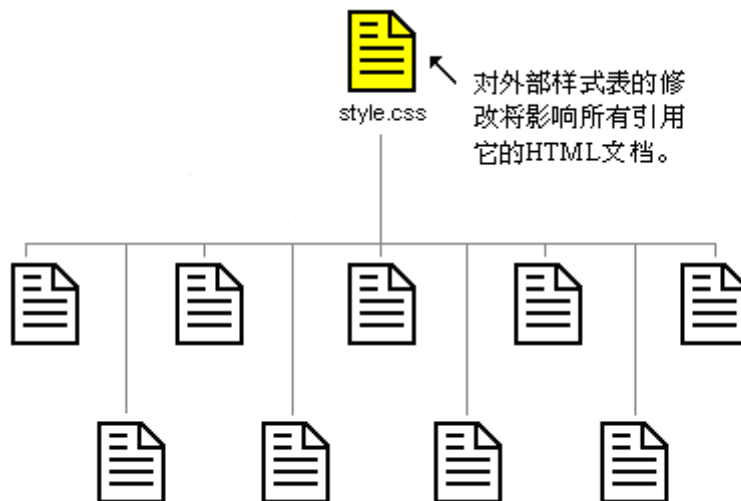
- 注意要在 href 属性里给出样式表文件的地址。

- 这行代码必须被插入 HTML 代码的头部（header），即放在标签<head>和标签</head>之间。就像这样：

- <html>
- <head>
- <title>我的文档</title>
- <link rel="stylesheet" type="text/css"
href="style/style.css" />
- </head>
- <body>
- ...

- 这个链接告诉浏览器：在显示该 HTML 文件时，应使用给出的 CSS 文件进行布局。

这种方法的优越之处在于：多个 HTML 文档可以同时引用一个样式表。换句话说，可以用一个 CSS 文件来控制多个 HTML 文档的布局。



引用同一个样式表的所有HTML文档

-
- 这一方法可以令你省去许多工作。例如，假设你要修改某网站的所有网页（比方说有 100 个网页）的背景颜色，采用外部样式表可以避免你手工一一修改这 100 个 HTML 文档的工作。采用外部样式表，这样的修改只需几秒钟即可搞定——修改外部样式表文件里的代码即可。
- 让我们来实践刚刚所学到的知识。

自己试试看

- 打开记事本（或其他文本编辑器），创建两个文件——一个 HTML 文件，一个 CSS 文件——它们的内容如下：

- default.htm

- `<html>`
- `<head>`
- `<title>我的文档</title>`
- `<link rel="stylesheet" type="text/css"`
`href="style.css" />`
- `</head>`
- `<body>`
- `<h1>我的第一个样式表</h1>`
- `</body>`
- `</html>`

- `style.css`

- `body {`
- `background-color: #FF0000;`
- `}`

- 然后，把这两个文件放在同一目录下。记得在保存文件时使用正确的扩展名（分别为“**htm**”和“**css**”）。
- 用浏览器打开 **default.htm**，你所看到的页面应该具有红色背景。恭喜！你已经完成了自己的第一个样式表！

第 3 课：颜色与背景

本课，你将学习如何在网站上应用颜色与背景。我们还会介绍用于定位和控制背景图像的高级方法。本课将对下列 CSS 属性进行讲解：

- [color](#)
- [background-color](#)

- [background-image](#)
- [background-repeat](#)
- [background-attachment](#)
- [background-position](#)
- [background](#)

前景色: 'color'属性

CSS 属性 `color` 用于指定元素的前景色。

例如，假设你要让页面中的所有标题（headline）都显示为深红色，而这些标题采用的都是 `h1` 元素，那么可以用下面的代码来实现把 `h1` 元素的前景色设为红色。

```
h1 {  
  
    color: #ff0000;  
  
}
```

- [显示示例](#)

颜色值可以用十六进制表示（比如上例中的 `#ff0000`），也可以用颜色名称（比如 `red`）或 RGB 值（比如 `rgb(255,0,0)`）表示。

'background-color'属性

CSS 属性 `background-color` 用于指定元素的背景色。

因为 `body` 元素包含了 HTML 文档的所有内容，所以，如果要改变整个页面的背景色的话，那么为 `body` 元素应用 `background-color` 属性就行了。

你也可以为其他包含标题或文本的元素单独应用背景色。在下例中，我们为 `body` 和 `h1` 元素分别应用了不同的背景色。

```
body {  
  
    background-color: #FFCC66;  
  
}
```

```
h1 {
```

```
color: #990000;

background-color: #FC9804;

}
```

- [显示示例](#)

注意：我们为 h1 元素应用了两个 CSS 属性，它们之间以分号（“;”）分隔。

背景图像[background-image]

CSS 属性 background-image 用于设置背景图像。

在下面的示例中，我们使用了一张蝴蝶的图像作为背景。你可以将该图片下载下来（方法为：鼠标右击该图片，然后选择“图片另存为”），以便在自己的计算机上使用。当然，你也可以选用其他你觉得满意的图片。



如果要把这个蝴蝶的图片作为网页的背景图像，只要在 body 元素上应用 background-image 属性、然后给出蝴蝶图片的存放位置就行了。

```
body {

    background-color: #FFCC66;

    background-image: url("butterfly.gif");

}
```



```
h1 {  
  
    color: #990000;  
  
    background-color: #FC9804;  
  
}
```

- [显示示例](#)

注意我们指定图片存放位置的方式：`url("butterfly.gif")`。这表明图片文件和样式表存放在同一目录下。你也可以引用存放在其他目录的图片，只需给出存放路径即可（比如 `url("../images/butterfly.gif")`）；此外，你甚至可以通过给出图片的地址来引用因特网（Internet）上的图片（比如 `url("http://www.html.net/butterfly.gif")`）。

平铺背景图像[background-repeat]

你有没有发现在上例中那个蝴蝶图片在横向和纵向都被平铺了？CSS 属性 `background-repeat` 就是用于控制平铺的。

下表概括了 `background-repeat` 的四种不同取值。

值	描述	示例
<code>background-repeat: repeat-x</code>	图像横向平铺	显示示例
<code>background-repeat: repeat-y</code>	图像纵向平铺	显示示例
<code>background-repeat: repeat</code>	图像横向和纵向都平铺	显示示例
<code>background-repeat: no-repeat</code>	图像不平铺	显示示例

例如，为了避免平铺背景图像，代码应该这样：

```
body {  
  
    background-color: #FFCC66;
```

```
background-image: url("butterfly.gif");

background-repeat: no-repeat;
}

h1 {

    color: #990000;

    background-color: #FC9804;
}
```

- [显示示例](#)

固定背景图像[background-attachment]

CSS 属性 background-attachment 用于指定背景图像是固定在屏幕上的、还是随着它所在的元素而滚动的。

一个固定的背景图像不会随着用户滚动页面而发生滚动（它是固定在屏幕上的），而一个非固定的背景图像会随着页面的滚动而滚动。

下表概括了 background-attachment 的两种不同取值。你可以点击示例察看二者的区别。

值	描述	示例
background-attachment:scroll	图像会跟随页面滚动——非固定的	显示示例
background-attachment:fixed	图像是固定在屏幕上的	显示示例

例如，下面的代码将背景图像固定在屏幕上。

```
body {

    background-color: #FFCC66;

    background-image: url("butterfly.gif");
```

```
background-repeat: no-repeat;

background-attachment: fixed;

}

h1 {

    color: #990000;

    background-color: #FC9804;

}
```

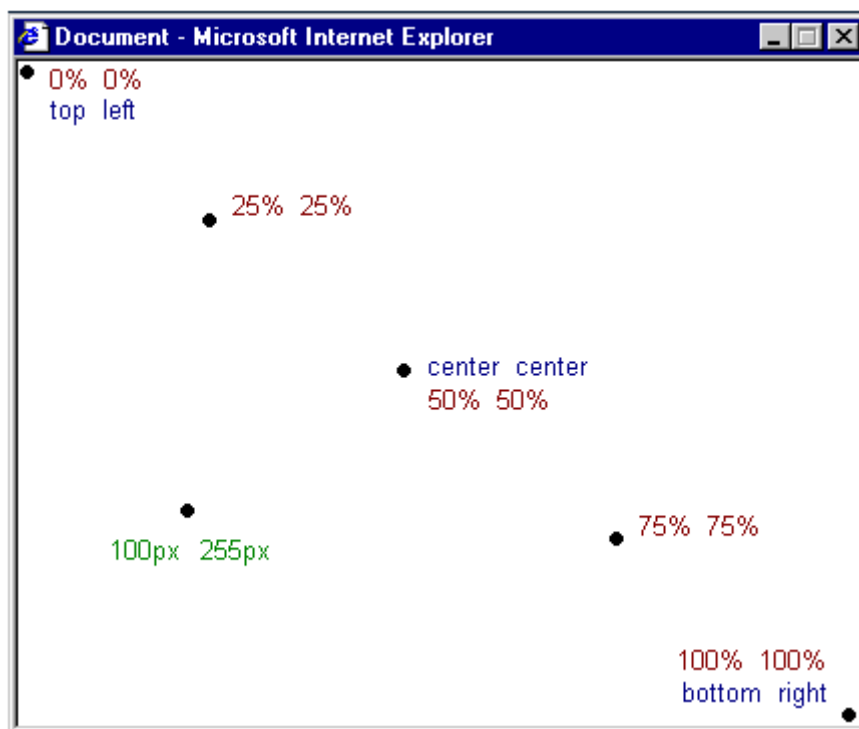
- [显示示例](#)

放置背景图像[background-position]

缺省地，背景图像将被放在屏幕的左上角。但是，你可以通过 CSS 属性 background-position 来修改这一缺省设置，将背景图像摆放在屏幕上你觉得满意的地方。

设置 background-position 属性的值有多种方式。不过，它们都是坐标的格式。举例来说，值“100px 200px”表示背景图像将被放置在位于距浏览器窗口左边 100 像素、顶部 200 像素处。

坐标可以是以百分比或固定单位（比如像素、厘米等）作为单位的值，也可以是“top”、“bottom”、“center”、“left”和“right”这些值。下图对此进行了解释：



下表给出了一些例子。

值	描述	示例
<code>background-position: 2cm 2cm</code>	图像被放置在页面内距左边 2 厘米、顶部 2 厘米的地方	显示示例
<code>background-position: 50% 25%</code>	图像被放置在页面内水平居中、离顶部四分之一处	显示示例
<code>background-position: top right</code>	图像被放置在页面的右上角	显示示例

在下例中，背景图像被放置在页面的右下角：

```
body {
    background-color: #FFCC66;
    background-image: url("butterfly.gif");
```

```
background-repeat: no-repeat;

background-attachment: fixed;

background-position: right bottom;
}

h1 {

    color: #990000;

    background-color: #FC9804;
}
```

- [显示示例](#)

缩写[background]

CSS 属性 background 是上述所有与背景有关的属性的缩写用法。

使用 background 属性可以减少属性的数目，因此令样式表更简短易读。

比如说下面五行代码：

```
background-color: #FFCC66;

background-image: url("butterfly.gif");

background-repeat: no-repeat;

background-attachment: fixed;

background-position: right bottom;
```

如果使用 background 属性的话，实现同样的效果只需一行代码即可搞定：

```
background: #FFCC66 url("butterfly.gif") no-repeat fixed
right bottom;
```

各个值应按下列次序来写：

```
[background-color] | [background-image] | [background-repeat] | [background-attachment] | [background-position]
```

如果省略某个属性不写出来，那么将自动为它取缺省值。比如，如果去掉 background-attachment 和 background-position 的话：

```
background: #FFCC66 url("butterfly.gif") no-repeat;
```

这两个未指定值的属性将被设置为缺省值：scroll 和 top left。

第 4 课：字体

这一课，你将学习字体以及如何用 CSS 来设置字体。我们还会考虑如何解决“网站所选的字体仅当访问者的 PC 上安装有该字体时才会被显示”这一难题。本课将对下列 CSS 属性进行讲解：

- [font-family](#)
- [font-style](#)
- [font-variant](#)
- [font-weight](#)
- [font-size](#)
- [font](#)

字体族[font-family]

CSS 属性 font-family 的作用是设置一组按优先级排序的字体列表，如果该列表中的第一个字体未在访问者计算机上安装，那么就尝试列表中的下一个字体，依此类推，直到列表中的某个字体是已安装的。

有两种类型的名称可用于分类字体：字体族名称（family-name）和族类名称（generic family）。下面来解释这两个术语。

字体族名称（family-name）

字体族名称（就是我们通常所说的“字体”）的例子包括“Arial”、“Times New Roman”、“宋体”、“黑体”等等。

族类（generic family）

一个族类是一组具有统一外观的字体族。sans-serif 就是一例，它代表一组没有“脚”的字体。

下面我们通过三个族类的例子来进行解释：

Times New Roman
Garamond
Georgia

这三个字体族属于 **serif** 族类。
它们的共同特点是，
笔画两端有“脚”。

Trebuchet
Arial
Verdana

这三个字体族属于 **sans-serif** 族类。
它们的共同特点是，
笔画两端没有“脚”。

Courier
Courier New
Andale Mono

这三个字体族属于 **monospace** 族类。
它们的共同特点是，
所有字符的宽度都一样。

你在给出字体列表时，自然应把首选字体放在前面、把候选字体放在后面。建议你在列表的最后给出一个族类（generic family），这样，当没有一个指定字体可用时，页面至少可以采用一个相同族类的字体来显示。

下面是一个按优先级排列的字体列表的例子：

```
h1 {font-family: arial, verdana, sans-serif;}  
h2 {font-family: "Times New Roman", serif;}
```

● [显示示例](#)

h1 标题将采用 Arial 字体显示。如果访问者的计算机未安装 Arial，那么就使用 Verdana 字体。假如 Verdana 字体也没安装的话，那么将采用一个属于 **sans-serif** 族类的字体来显示这个 h1 标题。

注意我们为“Times New Roman”采用的写法：因为其中包含空格，所以我们用引号将它括起来。

字体样式[font-style]

CSS 属性 font-style 定义所选字体的显示样式：**normal**（正常）、**italic**（斜体）或 **oblique**（倾斜）。在下例中，所有 h2 标题都将显示为斜体。

```
h1 {font-family: arial, verdana, sans-serif;}

h2 {font-family: "Times New Roman", serif; font-style:
italic;}
```

- [显示示例](#)

字体变化[font-variant]

CSS 属性 `font-variant` 的值可以是：**normal**（正常）或 **small-caps**（小体大写字母）。**small-caps** 字体是一种以小尺寸显示的大写字母来代替小写字母的字体。不太明白？我们来看几个例子：

Sans Book SC	Sans Bold SC	Serif Book SC	Serif Bold SC
ABCABC	ABCABC	ABCABC	ABCABC

如果 `font-variant` 属性被设置为 **small-caps**，而没有可用的支持小体大写字母的字体，那么浏览器多半会将文字显示为正常尺寸（而不是小尺寸）的大写字母。

```
h1 {font-variant: small-caps;}

h2 {font-variant: normal;}
```

- [显示示例](#)

字体浓淡[font-weight]

CSS 属性 `font-weight` 指定字体显示的浓淡程度。其值可以是 **normal**（正常）或 **bold**（加粗）。有些浏览器甚至支持采用 100 到 900 之间的数字（以百为单位）来衡量字体的浓淡。

```
p {font-family: arial, verdana, sans-serif;}

td {font-family: arial, verdana, sans-serif; font-weight:
bold;}
```


- [显示示例](#)

字体大小[font-size]

字体的大小用 CSS 属性 `font-size` 来设置。

字体大小可通过多种不同单位（比如像素或百分比等）来设置。在本教程中，我们将关注于最常用和最合适的单位。比如：

```
h1 {font-size: 30px;}  
h2 {font-size: 12pt;}  
h3 {font-size: 120%;}  
p {font-size: 1em;}
```

- [显示示例](#)

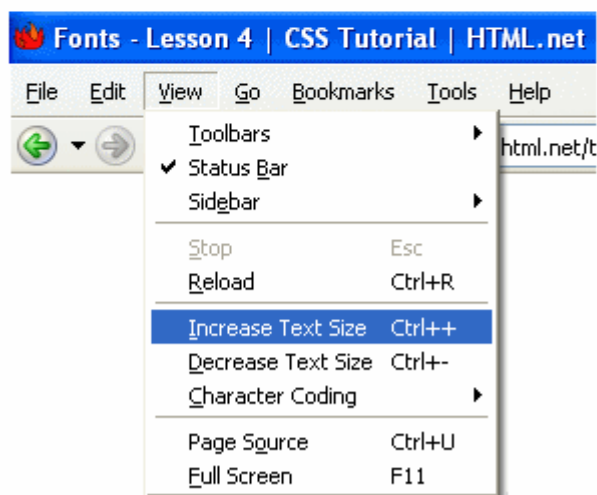
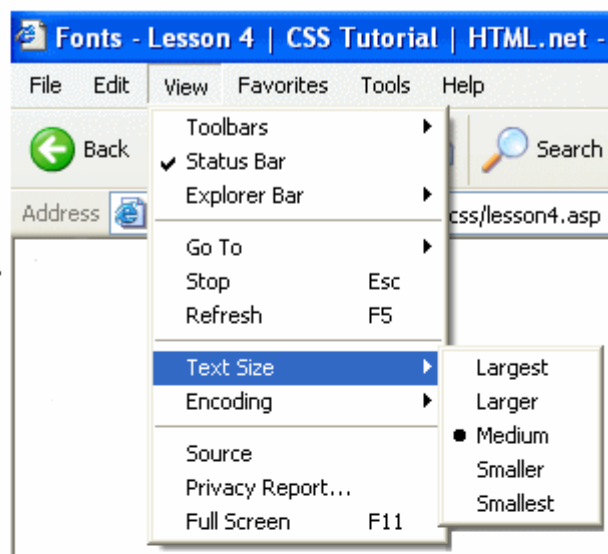
上面四种单位有着本质的区别。**'px'和'pt'**将字体设置为固定大小，而**'%'和'em'**允许页面浏览者自行调整字体的显示尺寸。有些页面浏览者可能是残疾者、年长者、视力不佳者，或者他所使用的电脑显示屏显示质量差。**为了令你的网站对所有人都具有良好的可用性（accessibility），你应采用像'%'或'em'这种允许用户调节字体显示大小的单位。**

下面你能看到我们展示如何在 Mozilla Firefox 和 Internet Explorer 里调整字体大小。自己试试看！这个功能很不错吧？

In most browsers you can adjust the text size

Internet Explorer →

Mozilla Firebird



缩写[font]

CSS 属性 font 是上述各有关字体的 CSS 属性的缩写用法。

比如说下面四行应用于 p 元素的代码：

```
p {
```

```
    font-style: italic;
```

```
    font-weight: bold;
```

```
font-size: 30px;

font-family: arial, sans-serif;

}
```

如果用 **font** 属性的话，上述四行代码可简化为：

```
p {

    font: italic bold 30px arial, sans-serif;

}
```

font 属性的值应按以下次序书写：

font-style | font-variant | font-weight | font-size | font-family

第 5 课：文本

文本的显示格式与样式对于网页设计师来说是一个重要问题。这一课将向你介绍 CSS 在文本布局方面令人激动的特性。本课将对下列 CSS 属性进行讲解：

- [text-indent](#)
- [text-align](#)
- [text-decoration](#)
- [letter-spacing](#)
- [text-transform](#)

文本缩进[text-indent]

CSS 属性 text-indent 用于为段落设置首行缩进，以令其具有美观的格式。在下例中，我们为采用 p 元素的段落应用了 **30 像素**的首行缩进。

```
p {

    text-indent: 30px;

}
```

- [显示示例](#)

文本对齐[text-align]

CSS 属性 `text-align` 与 HTML 属性 `align` 的功能相同。该属性的值可以是：**left**（左对齐）、**right**（右对齐）或者 **center**（居中）。除了上面三种选择以外，你还可以将该属性的值设为 **justify**（两端对齐），即伸缩行中的文字以左右靠齐。报刊杂志经常采用这种布局。

在下例中，标题（th）中的文字被设置为右对齐，而表中数据（td）被设置为居中。正常的文本段落被设置为两端对齐。

```
th {  
  
    text-align: right;  
  
}  
  
td {  
  
    text-align: center;  
  
}  
  
p {  
  
    text-align: justify;  
  
}
```

- [显示示例](#)

文本装饰[text-decoration]

CSS 属性 `text-decoration` 令我们可以为文本增添不同的“装饰”或“效果”。例如，你可以为文本增添下划线、删除线、上划线等等。在接下来的例子中，我们为 h1 标题增添了下划线，为 h2 标题增添了上划线，为 h3 标题增添了删除线。

```
h1 {  
  
    text-decoration: underline;
```

```
}

h2 {

    text-decoration: overline;

}

h3 {

    text-decoration: line-through;

}
```

- [显示示例](#)

字符间距[letter-spacing]

CSS 属性 letter-spacing 用于设置文本的水平字间距。我们可以把期望的字间距宽度作为这个属性的值。例如，假如你希望 p 元素里的文本段落的字间距为 **3 个像素**，而 h1 标题的字间距为 **6 个像素**，代码可以这样写：

```
h1 {

    letter-spacing: 6px;

}

p {

    letter-spacing: 3px;

}
```

- [显示示例](#)

文本转换[text-transform]

CSS 属性 `text-transform` 用于控制文本的大小写。无论字母本来的大小写，你可以通过该属性令它**首字母大写**（`capitalize`）、**全部大写**（`uppercase`）或者**全部小写**（`lowercase`）。

比如，单词“headline”在展现给网页浏览者时，可以是“HEADLINE”或者“Headline”。`text-transform` 属性有四个可选值：

`capitalize`

将每个单词的首字母转换为大写。例如：“john doe”将被转换为“John Doe”。

`uppercase`

所有字母都转换为大写。例如：“john doe”将被转换为“JOHN DOE”。

`lowercase`

所有字母都转换为小写。例如：“JOHN DOE”将被转换为“john doe”。

`none`

不作任何转换——文本怎么写的就怎么显示。

来举个例子，我们将使用一个姓名列表。所有姓名都用 ``（列表项）标签来标记。我们对姓名采用首字母大写的方式，而对标题采用全部大写的方式。

查看过该例的 HTML 代码后你会发现，其实在 HTML 代码里我们写的姓名和标题全部都是小写。

```
h1 {  
  
    text-transform: uppercase;  
  
}  
  
li {  
  
    text-transform: capitalize;  
  
}
```

第 6 课：链接

你在前面几课学到的属性也可以应用到链接上（比如修改颜色、字体、添加下划线等）。但不同的是，CSS 允许你根据链接是未访问的、已访问的、活动的、是否有鼠标悬停等分别定义不同的属性。这样，我们便可为网站增添奇特而有用的效果。你需要通过伪类（`pseudo-class`）来控制这些效果。

伪类是什么？

伪类（`pseudo-class`）令你可以在为 HTML 元素定义 CSS 属性时将条件和事件考虑在内。

我们来看一个例子。正如你所知道的，在 HTML 里，链接是通过 `a` 元素来定义的。因此，在 CSS 里，我们可以将 `a` 作为一个选择器（`selector`）：

```
a {  
  
    color: blue;  
  
}
```

一个链接可以有不同的状态。例如，它可以是已访问过的，也可以是未访问过的。你可以通过伪类分别为访问过的链接和未访问过的链接设置不同的样式。

```
a:link {  
  
    color: blue;  
  
}
```

```
a:visited {  
  
    color: red;  
  
}
```

为未访问过的链接和已访问过的链接分别使用伪类 `a:link` 和 `a:visited`。活动的链接对应的伪类为 `a:active`，有鼠标悬停的链接对应的伪类为 `a:hover`。

我们将逐个解释这四个伪类，并给出示例。

伪类: link

伪类 `:link` 用于浏览器从未访问过的链接。

在下面的示例代码中，我们将未访问过的链接设为浅蓝色。

```
a:link {  
  
    color: #6699CC;  
  
}
```

- [显示示例](#)

伪类: visited

伪类:visited 用于浏览者已访问过的链接。比如，下面的代码将已访问过的链接设为深紫色：

```
a:visited {  
  
    color: #660099;  
  
}
```

- [显示示例](#)

伪类: active

伪类:active 用于活动的链接（即获得当前焦点的链接）。

下例将活动的链接设为具有黄色背景：

```
a:active {  
  
    background-color: #FFFF00;  
  
}
```

- [显示示例](#)

伪类: hover

伪类:hover 用于有鼠标悬停的链接。

这能制造出有趣的效果。例如，如果你要当鼠标光标移到链接上时将链接显示为橙色斜体，那么 CSS 可以这样写：

```
a:hover {
```



```
color: orange;

font-style: italic;

}
```

- [显示示例](#)

例 1：有鼠标悬停的链接的效果

为链接设置悬停效果十分流行。所以，我们将多看几个:hover 伪类的例子。

例 1a：字符间距

我们在[第 5 课](#)学过，可以用letter-spacing属性来调整字符间距。现在为了取得特殊效果，我们将它应用到链接上：

```
a:hover {

    letter-spacing: 10px;

    font-weight:bold;

    color:red;

}
```

- [显示示例](#)

例 1b：大写和小写

同样在[第 5 课](#)我们学过，可以通过text-transform属性来转换字母的大小写。这也可用于为链接制造效果：

```
a:hover {

    text-transform: uppercase;

    font-weight:bold;

    color:blue;

}
```

```
background-color:yellow;

}
```

- [显示示例](#)

通过上面两个例子你会发现，我们可以通过属性的组合创造出无数种效果。你可以创建自己的效果——试试吧！

例 2：去掉链接的下划线

如何去掉链接的下划线是一个常见的问题。

关于是否去掉链接的下划线，你必须仔细考虑清楚，因为这有可能严重降低网站的易用性（usability）。人们已经习惯于蓝色有下划线的链接了，他们看到它就知道那是可以点击的。甚至连我母亲都知道这一点！如果你去掉链接的下划线或修改链接的字体颜色的话，这有可能会令用户感到困惑、并因此不能充分享用你网站上的内容。

尽管如此，去掉链接的下划线是很容易的。你肯定记得我们在[第 5 课](#)学过text-decoration属性，它可用于决定是否给文本添加下划线。要去掉下划线，只要把text-decoration属性的值设为none就行了。

```
a {

    text-decoration:none;

}
```

除此以外，text-decoration 属性也可以与其它属性一起应用在伪类上。

```
a:link {

    color: blue;

    text-decoration:none;

}
```

```
a:visited {
```

```
        color: purple;

        text-decoration:none;
    }

a:active {

    background-color: yellow;

    text-decoration:none;
}

a:hover {

    color:red;

    text-decoration:none;
}
```

第 7 课：元素的分类与标识（class 和 id）

有时，你希望对特定元素或者特定一组元素应用特殊的样式。在这一课，我们将深入学习如何利用 class 和 id 来为所选元素指定属性。

如何实现为网站上许多标题中的某一个单独应用颜色？如何实现把网站上的链接分为不同的类，并对各类链接分别应用不同的样式？这只是本课将解决的诸多问题中的最具代表性的两个问题。

用 class 对元素进行分类

比方说，我们有两个由链接组成的列表，它们分别是用于制造白葡萄酒和红葡萄酒的葡萄。其 HTML 代码如下：

```
<p>制造白葡萄酒的葡萄：</p>

<ul>

<li><a href="ri.htm">雷司令（Riesling）</a></li>

<li><a href="ch.htm">夏敦埃（Chardonnay）</a></li>
```

```
<li><a href="pb.htm">白比诺 (Pinot Blanc) </a></li>
</ul>

<p>制造红葡萄酒的葡萄: </p>

<ul>

<li><a href="cs.htm">卡百内索维农 (Cabernet Sauvignon)
</a></li>

<li><a href="me.htm">墨尔乐 (Merlot) </a></li>

<li><a href="pn.htm">黑比诺 (Pinot Noir) </a></li>

</ul>
```

- [显示示例](#)

现在，我们希望白葡萄酒的链接全部显示为黄色，红葡萄酒的链接全部显示为红色，其余的链接显示为缺省的兰色。

为了实现这一要求，我们将链接分为两类。对链接的分类是通过为链接设置 HTML 属性 class 实现的。

参加如下代码：

```
<p>制造白葡萄酒的葡萄: </p>

<ul>

<li><a href="ri.htm" class="whitewine">雷司令 (Riesling)
</a></li>

<li><a href="ch.htm" class="whitewine">夏敦埃 (Chardonnay)
</a></li>
```

```
<li><a href="pb.htm" class="whitewine">白比诺 (Pinot Blanc)
</a></li>
```

```
</ul>
```

```
<p>制造红葡萄酒的葡萄： </p>
```

```
<ul>
```

```
<li><a href="cs.htm" class="redwine">卡百内索维农 (Cabernet
Sauvignon) </a></li>
```

```
<li><a href="me.htm" class="redwine">墨尔乐 (Merlot)
</a></li>
```

```
<li><a href="pn.htm" class="redwine">黑比诺 (Pinot Noir)
</a></li>
```

```
</ul>
```

然后，我们就可以为白葡萄酒和红葡萄酒的链接分别应用不同的风格了。

```
a {
    color: blue;
}
```

```
a. whitewine {
    color: #FFBB00;
}
```

```
a.redwine {  
  
    color: #800000;  
  
}
```

- [显示示例](#)

如上例所示，你可以通过在样式表里利用 **classname** 来为属于某一类的元素定义 CSS 属性。

利用 id 标识元素

除了可以对元素进行分类以外，你还可以标识单个元素。这是通过 HTML 属性 id 实现的。

HTML 属性 id 的特别之处在于，在同一 HTML 文档中不能有两个具有相同 id 值的元素。文档中的每个 id 值都必须是唯一的。在其他情况下，你应该使用 class 属性。下面，我们来看一个使用 id 属性的例子：

```
<h1>第 1 章</h1>
```

```
...
```

```
<h2>第 1.1 节</h2>
```

```
...
```

```
<h2>第 1.2 节</h2>
```

```
...
```

```
<h1>第 2 章</h1>
```

```
...
```

```
<h2>第 2.1 节</h2>
```

```
...
```

```
<h2>第 2.1.1 小节</h2>
```

...

上例是一个文章的各章节的标题。我们自然可以为其中每一章节指定一个 id（如下）：

```
<h1 id="c1">第 1 章</h1>

...

<h2 id="c1-1">第 1.1 节</h2>

...

<h2 id="c1-2">第 1.2 节</h2>

...

<h1 id="c2">第 2 章</h1>

...

<h2 id="c2-1">第 2.1 节</h2>

...

<h3 id="c2-1-2">第 2.1.1 节</h3>

...
```

假如我们要求第 1.2 节显示为红色，那么 CSS 可以这样写：

```
#c1-2 {

    color: red;

}
```

第 8 课：组织元素（span 和 div）

span 和 div 元素用于组织和结构化文档，并经常联合 class 和 id 属性一起使用。

在这一课中，我们将进一步探究 `span` 和 `div` 的用法，因为这两个 HTML 元素对于 CSS 是很重要的。

- 用 `span` 组织元素
- 用 `div` 组织元素

用 `span` 组织元素

`span` 元素可以说是一种中性元素，因为它不对文档本身添加任何东西。但是与 CSS 结合使用的话，`span` 可以对文档中的部分文本增添视觉效果。

让我们用一句本杰明·弗兰克林（Benjamin Franklin）的名言来举个例子：

<p>早睡早起
使人健康、富裕又聪颖。</p>

假设我们想用红色来强调弗兰克林先生所认为的“不要在睡眠中度过一天”的好处，我们可以用 `` 标签来标记上述每一点好处。然后，我们将这几个 `span` 设置为相同的 `class`。这样，我们稍后就可以在样式表里针对这个 `class` 定义特定的样式。

<p>早睡早起
使人健康、
富裕
和聪颖。</p>

相应的 CSS 代码如下：

```
span.benefit {  
    color:red;  
}
```

- [显示示例](#)

当然，你也可以采用 id 来为 span 元素添加样式。不过正如我们在上一课所学的，如果采用 id 的话，你必须为这三个 span 元素各自分别指定一个唯一的 id。

用 div 组织元素

如前面例子所示，span 的使用局限在一个块元素内，而 div 可以被用来组织一个或多个块元素。

除去这点区别，div 和 span 在组织元素方面相差无几。让我们来看一个例子。我们将历届美国总统按其所属的政营分别组织为两个列表：

```
<div id="democrats">
<ul>
<li>富兰克林·D·罗斯福</li>
<li>哈利·S·杜鲁门</li>
<li>约翰·F·肯尼迪</li>
<li>林登·B·约翰逊</li>
<li>吉米·卡特</li>
<li>比尔·克林顿</li>
</ul>
</div>
```

```
<div id="republicans">
<ul>
<li>德怀特·D·艾森豪威尔</li>
<li>理查德·尼克松</li>
<li>杰拉尔德·福特</li>
<li>罗纳德·里根</li>
<li>乔治·布什</li>
```

```
<li>乔治 • W • 布什</li>

</ul>

</div>
```

在这里，我们可以采用跟上例同样的方法来处理样式表：

```
#democrats {

    background:blue;

}

#republicans {

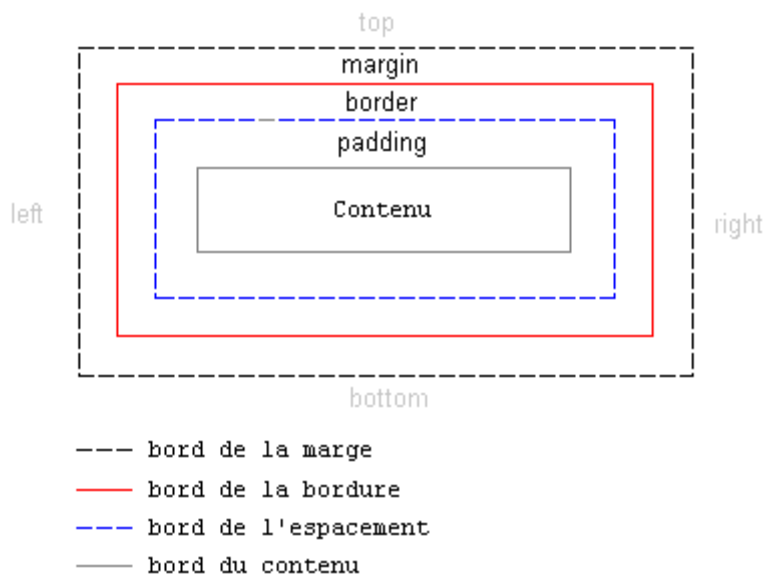
    background:red;

}
```

第 9 课：盒状模型

CSS 中的盒状模型（box model）用于描述一个为 HTML 元素形成的矩形盒子。盒状模型还涉及为各个元素调整外边距（margin）、边框（border）、内边距（padding）和内容的具体操作。下图显示了盒状模型的结构：

CSS 中的盒状模型



上面的图示看上去可能感觉有点理论化，好吧，让我们试着用一个实例来解释盒状模型。在我们的例子中，有一个标题和一些文本。该例的 HTML 代码如下（摘自世界人权宣言）：

```
<h1>Article 1:</h1>
```

```
<p>All human beings are born free
```

```
and equal in dignity and rights.
```

```
They are endowed with reason and conscience
```

```
and should act towards one another in a
```

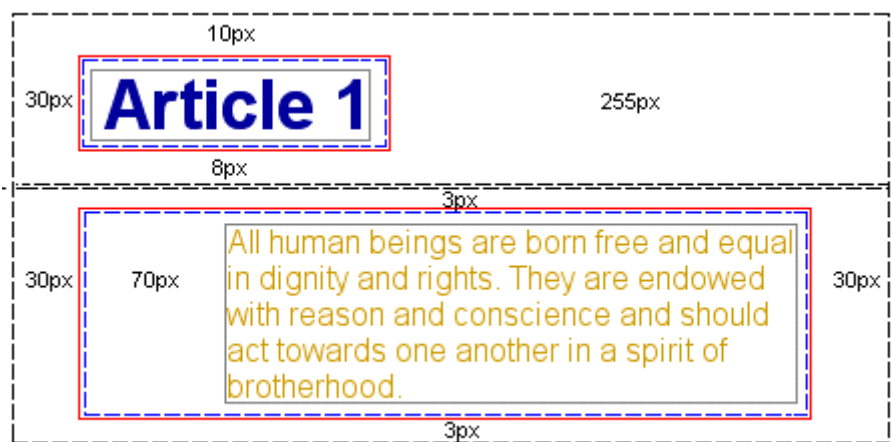
```
spirit of brotherhood</p>
```

通过添加一些颜色及字体信息，该例可以有如下显示效果：

Article 1

All human beings are born free and equal in dignity and rights. They are endowed with reason and conscience and should act towards one another in a spirit of brotherhood.

这个例子包含了两个元素：h1 和 p。这两个元素的盒状模型如下图所示：



尽管上图显得有点复杂，不过它展示了每个 HTML 元素是如何被盒子所围绕的。我们可以通过 CSS 来调整这些盒子。

第 10 课：外边距和内边距

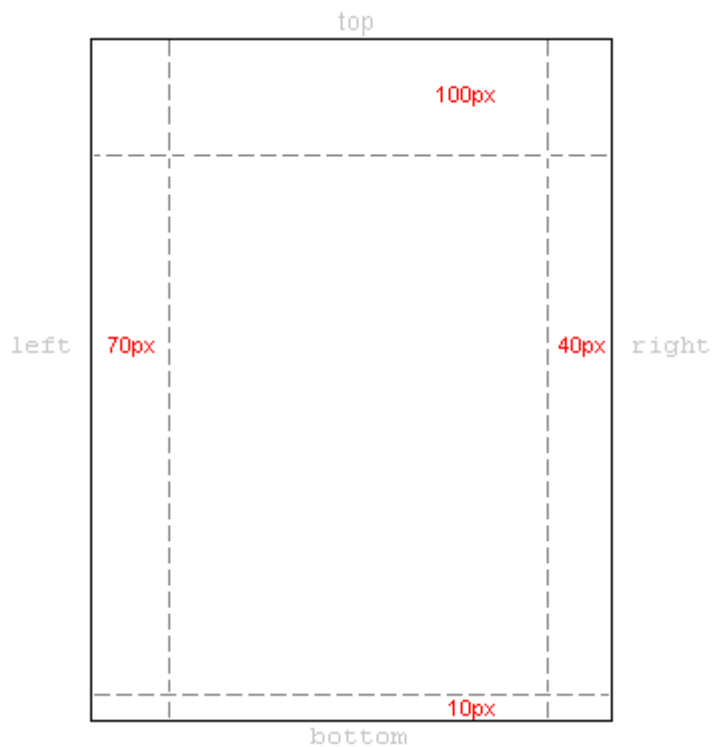
上一课，你学习了盒状模型。在这一课，我们将了解如何通过设置 margin（外边距）和 padding（内边距）这两个 CSS 属性来改变元素的显示。

- [为元素设置外边距](#)
- [为元素设置内边距](#)

为元素设置外边距

一个元素有上（top）、下（bottom）、左（left）、右（right）四个边。外边距（margin）表示从一个元素的边到相邻元素(或者文档边界)之间的距离。可以参考[第 9 课](#)的图示。

在下面这个例子中，我们将了解如何为文档本身（即 body 元素）定义外边距。下图显示了我们对外边距的要求。



满足上述要求的 CSS 代码如下：

```
body {  
  
    margin-top:100px;  
  
    margin-right:40px;  
  
    margin-bottom:10px;  
  
    margin-left:70px;  
  
}
```

或者你也可以采用一种较优雅的缩写形式：

```
body {  
  
    margin: 100px 40px 10px 70px;
```

```
}
```

- [显示示例](#)

几乎所有元素都可以采用跟上面一样的方法来设置外边距。例如，我们可以为所有用<p>标记的文本段落定义外边距：

```
body {  
  
    margin: 100px 40px 10px 70px;  
  
}  
  
p {  
  
    margin: 5px 50px 5px 50px;  
  
}
```

- [显示示例](#)

为元素设置内边距

内边距（padding）也可以被理解成“填充物”。这样理解是合理的，因为内边距并不影响元素间的距离，它只定义元素的内容与元素边框之间的距离。

下面我们通过一个简单的例子来说明内边距的用法。在这个例子中，所有标题都具有背景色：

```
h1 {  
  
    background: yellow;  
  
}  
  
h2 {
```

```
background: orange;

}
```

- [显示示例](#)

通过为标题设置内边距，你可以控制在标题文本周围填充多少空白：

```
h1 {

    background: yellow;

    padding: 20px 20px 20px 80px;

}

h2 {

    background: orange;

    padding-left: 120px;

}
```

- [显示示例](#)

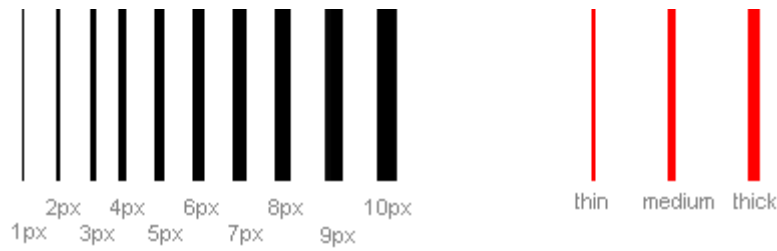
第 11 课：边框

边框（border）可以有多种用途，比如作为装饰元素或者作为划分两物的分界线。在设置边框方面，CSS 为你提供了无尽选择。

- [border-width](#)
- [border-color](#)
- [border-style](#)
- [一些示例](#)
- [缩写 \[border\]](#)

边框宽度[border-width]

边框宽度由 CSS 属性 border-width 定义，其值可以是“thin”（薄）、“medium”（普通）或“thick”（厚）等，也可以是像素值。如下图所示：



边框颜色[border-color]

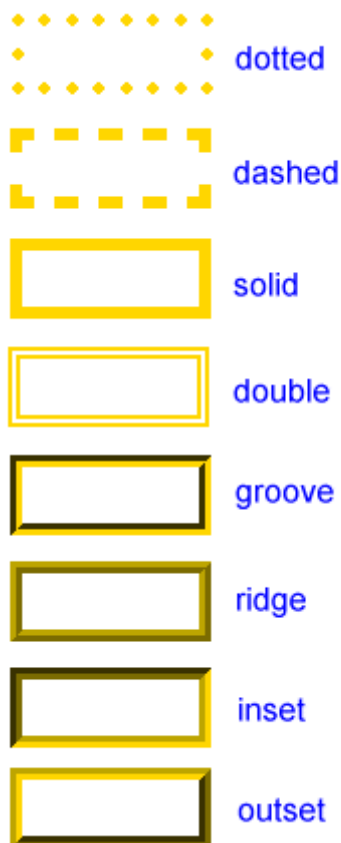


CSS 属性 `border-color` 用于定义边框的颜色。其值就是正常的颜色值，例如：“#123456”、“`rgb(123,123,123)`”、“yellow”等。

边框样式[border-style]

边框样式有多种可供选择。下图显示了 8 种不同样式的边框在 Internet Explorer 5.5 里的实际显示效果。在这个例子里，我们为这 8 种边框都选择了“金色 (gold)”作为边框颜色、“厚(thick)”作为边框宽度。当然，这只是个例子，你可以为边框设置别的颜色和厚度。

如果你不想有任何边框，可以为它取值为“none”或者“hidden”。



一些示例

我们可以将上面三个有关边框的 CSS 属性组合起来使用，从而制造出多种多样的变化。来举个例子，我们要为一个文档中的 `h1`、`h2`、`u1` 和 `p` 等元素分别定义不同的边框。

尽管其显示效果也许并不那么美观，但是它很好地向你展示了多种变化的可能：


```
h1 {  
    border-width: thick;  
    border-style: dotted;  
    border-color: gold;  
}
```

```
h2 {  
    border-width: 20px;  
    border-style: outset;  
    border-color: red;  
}
```

```
p {  
    border-width: 1px;  
    border-style: dashed;  
    border-color: blue;  
}
```

```
ul {  
    border-width: thin;  
    border-style: solid;
```

```
border-color: orange;

}
```

- [显示示例](#)

我们也可以为上边框、下边框、右边框、左边框分别指定特定的 CSS 属性。具体做法如下例所示：

```
h1 {

    border-top-width: thick;

    border-top-style: solid;
    border-top-color: red;


    border-bottom-width: thick;
    border-bottom-style: solid;
    border-bottom-color: blue;


    border-right-width: thick;
    border-right-style: solid;
    border-right-color: green;


    border-left-width: thick;
    border-left-style: solid;
    border-left-color: orange;

}
```

```
}
```

- [显示示例](#)

缩写[border]

跟许多其他属性一样，你也可以将有关边框的 CSS 属性缩写为一个 **border** 属性。让我们看一个例子：

```
p {  
  
    border-width: 1px;  
  
    border-style: solid;  
  
    border-color: blue;  
  
}
```

可被缩写为：

```
p {  
  
    border: 1px solid blue;  
  
}
```

第 12 课：高度和宽度

到目前为止，我们还没有在元素尺寸方面给予关注。在这一课，我们将看到为元素定义高度和宽度是十分简单的。

- [设定宽度\[width\]](#)
- [设定高度\[height\]](#)

设定宽度[width]

你可以通过 **width** 属性来设定一个元素的宽度，即在水平方向上的尺寸。

下面是一个简单的例子，它为我们提供了一个可以容纳文本的盒子：



```
div.box {  
  
    width: 200px;  
  
    border: 1px solid black;  
  
    background: orange;  
  
}
```

- [显示示例](#)

设定高度[height]

注意上一个例子，盒子里内容的长短决定了盒子的高度。你可以通过 height 属性来设定一个元素的高度。比方说，我们要把上面那个例子中的盒子高度设定为 500 像素：

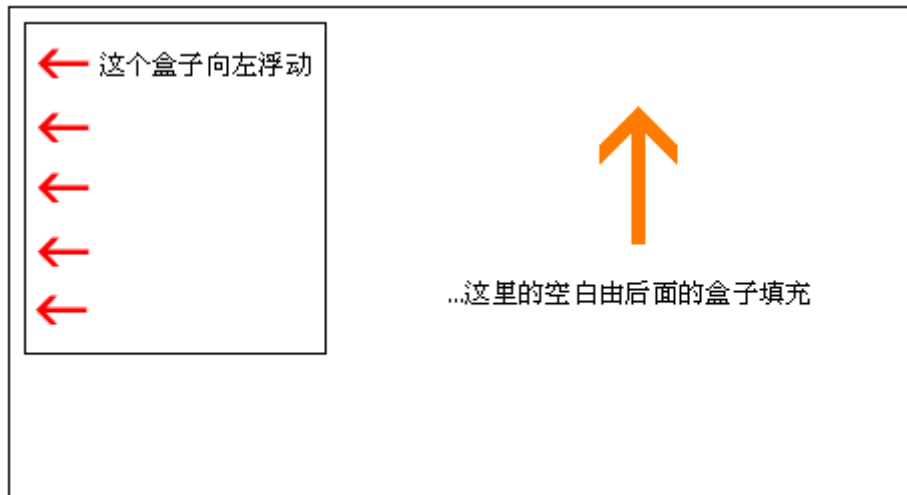
```
div.box {  
  
    height: 500px;  
  
    width: 200px;  
  
    border: 1px solid black;  
  
    background: orange;  
  
}
```

- [显示示例](#)

第 13 课：浮动元素（float）

我们可以通过CSS属性float令元素向左或向右浮动。也就是说，令盒子及其中的内容浮动到文档（或者是上层盒子）的右边或者左边（参见[第 9 课](#)关于盒状模型的描述）。

下图阐明了其原理：



举个例子，假如我们想让一张图片被一段文字围绕着，那么其显示效果将如下所示：



如何实现这个效果？

上例的 HTML 代码如下所示：

```
<div id="picture">  
      
</div>  
  
<p>causas naturales et antecedentes,
```

```
idciro etiam nostrarum voluntatum...</p>
```

要实现图片向左浮动、而且被文字环绕的效果，你只需先设定图片所在盒子的宽度，然后再把 CSS 属性 float 设为 left 即可：

```
#picture {  
  
    float:left;  
  
    width: 100px;  
  
}
```

- [显示示例](#)

另一个例子：列

浮动也可以用于实现在文档中分列。要创建多个列，你需要在 HTML 代码里用 div 来结构化想要的各个列：

```
<div id="column1">  
  
    <p>Haec disserens qua de re agatur  
  
    et in quo causa consistat non videt...</p>  
  
</div>  
  
<div id="column2">  
  
    <p>causas naturales et antecedentes,  
  
    idciro etiam nostrarum voluntatum...</p>  
  
</div>
```

```
<div id="column3">

    <p>nam nihil esset in nostra

    potestate si res ita se haberet...</p>

</div>
```

下面，我们把各列的宽度设定为“33%”，并通过定义 float 属性令各列向左浮动：

```
#column1 {

    float:left;

    width: 33%;

}

#column2 {

    float:left;

    width: 33%;

}

#column3 {

    float:left;

    width: 33%;

}
```

float 属性的值可以是 **left**、**right** 或者 **none**。

clear 属性

CSS 属性 clear 用于控制浮动元素的后继元素的行为。

缺省地，后继元素将向上移动，以填补由于前面元素的浮动而空出的可用空间。在前面的例子中，文本自动上移到了比尔盖茨的图片旁。

clear 属性的值可以是 **left**、**right**、**both** 或 **none**。原则是这样的：如果一个盒子的 clear 属性被设为“both”，那么该盒子的上边距将始终处于前面的浮动盒子（如果存在的话）的下边距之下。

```
<div id="picture">
    
</div>

<h1>Bill Gates</h1>

<p class="floatstop">causas naturales et antecedentes,
idcirco etiam nostrarum voluntatum...</p>
```

要避免文本上移到图片旁，我们可以在 CSS 中添加以下代码：

```
#picture {
    float:left;
    width: 160px;
}

.floatstop {
```



```
clear:both;
```

```
}
```

第 14 课：元素的定位

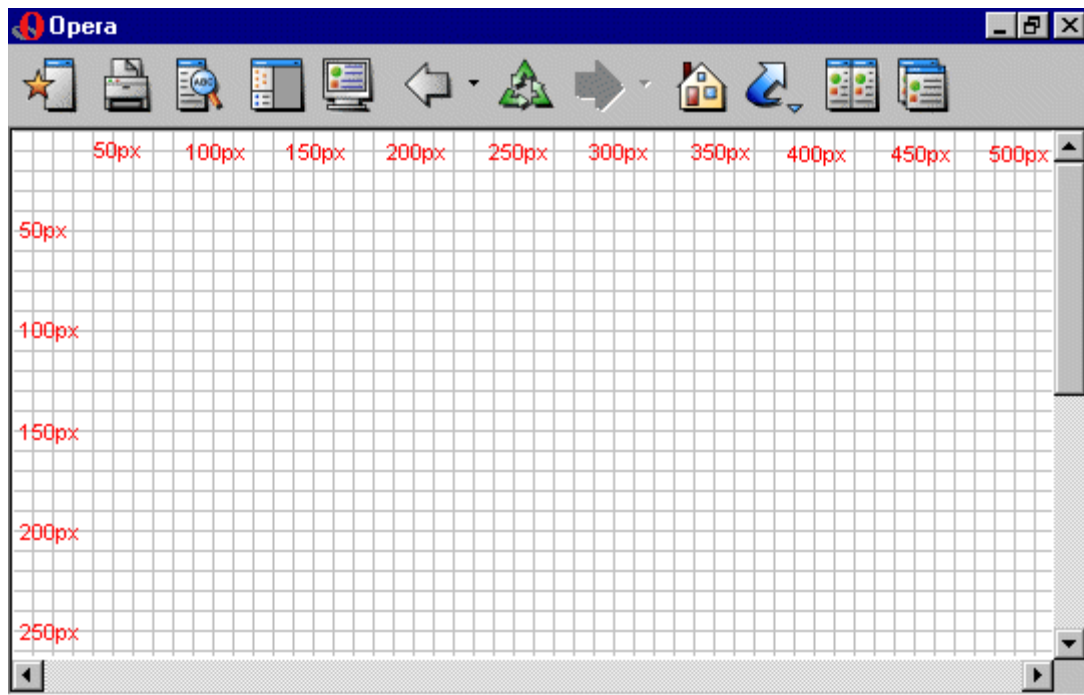
CSS定位令你可以将一个元素精确地放在页面上你所指定的地方。联合使用定位与浮动（参见[第 13 课](#)），你将能够创建多种高级而精确的布局。

本课我们将讨论以下内容：

- [CSS定位的原理](#)
- [绝对定位](#)
- [相对定位](#)

CSS 定位的原理

把浏览器窗口想象成一个坐标系：



CSS 定位的原理是：你可以将任何盒子（box）放置在坐标系统的任何位置上。

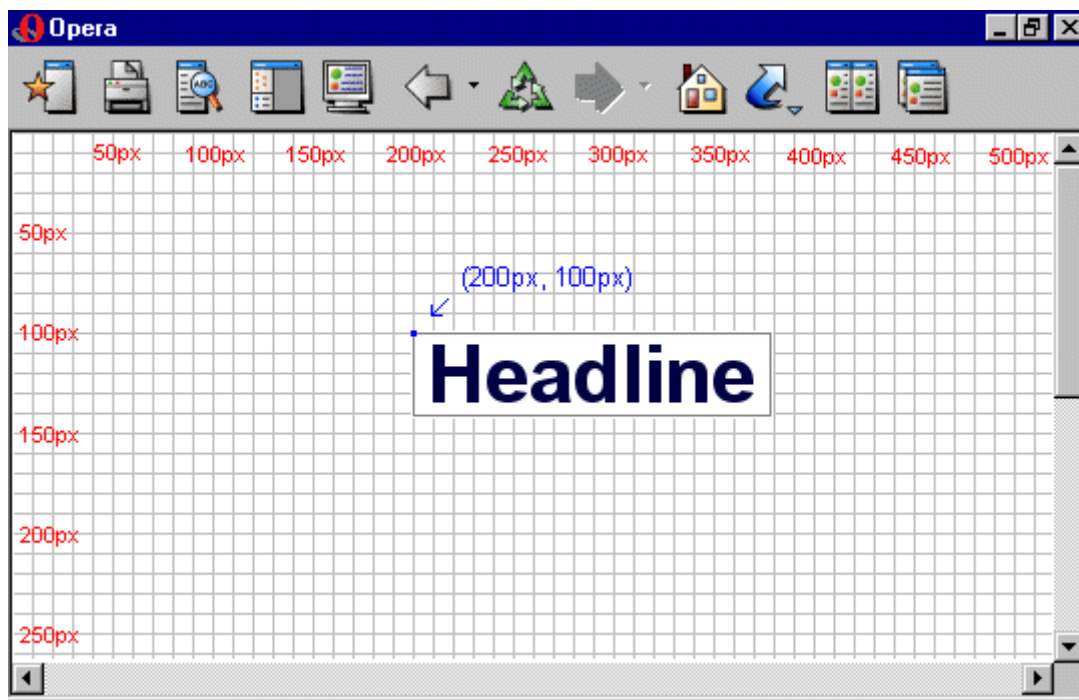
假设我们要放置一个标题。通过使用盒状模型（参见[第 9 课](#)），标题将显示如下：

Headline

如果我们要把这个标题放置在距文档顶部 100 像素、左边 200 像素的地方，我们可以在 CSS 中输入以下代码：

```
h1 {  
  
    position: absolute;  
  
    top: 100px;  
  
    left: 200px;  
  
}
```

得到的显示效果如下：



正如你所看到的，采用 CSS 定位技术来放置元素是非常精确的。相对于使用表格、透明图像或其他方法而言，CSS 定位要简单得多。

绝对定位

一个采用绝对定位的元素不获得任何空间。这意味着：该元素在被定位后不会留下空位。

要对元素进行绝对定位，应将 `position` 属性的值设为 **absolute**。接着，你可以通过属性 **left**、**right**、**top** 和 **bottom** 来设定将盒子放置在哪里。

举个绝对定位的例子，假如我们要在文档的四个角落各放置一个盒子：

```
#box1 {  
  
    position: absolute;  
  
    top: 50px;  
  
    left: 50px;  
  
}
```

```
#box2 {  
  
    position: absolute;  
  
    top: 50px;  
  
    right: 50px;  
  
}
```

```
#box3 {  
  
    position: absolute;  
  
    bottom: 50px;  
  
    right: 50px;  
  
}
```

```
#box4 {  
  
    position: absolute;  
  
    bottom: 50px;  
  
    left: 50px;
```

```
}
```

- [显示示例](#)

相对定位

要对元素进行相对定位，应将 `position` 属性的值设为 **relative**。绝对定位与相对定位的区别在于计算位置的方式。

采用相对定位的元素，其位置是**相对于它在文档中的原始位置**计算而来的。这意味着，相对定位是通过将元素从原来的位置向右、向左、向上或向下移动来定位的。采用相对定位的元素会获得相应的空间。

举个相对定位的例子，我们可以相对于三张图片在页面上的原始位置来对它们进行相对定位。注意这些图片将在文档中各自的原始位置处留下空位。

```
#dog1 {  
  
    position:relative;  
  
    left: 350px;  
  
    bottom: 150px;  
}
```

```
#dog2 {  
  
    position:relative;  
  
    left: 150px;  
  
    bottom: 500px;  
}
```

```
#dog3 {  
  
    position:relative;  
  
    left: 50px;
```

```
bottom: 700px;

}
```

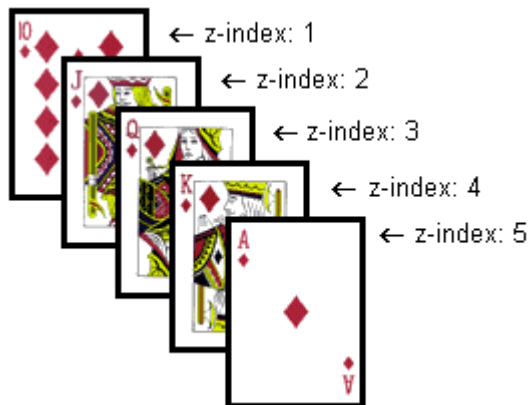
- [显示范例](#)

第 15 课：用 z-index 进行层次堆叠

CSS 可以处理高度、宽度、深度三个维度。在前面的课程中，我们已经了解了前两个维度。在本课中，我们将学习如何令不同元素具有层次。简言之，就是关于元素堆叠的次序问题。

为此，你可以为每个元素指定一个数字（z-index）。其原理是：数字较大的元素将叠加在数字较小的元素之上。

比方说，我们正在打扑克，并且拿了一手同花大顺。我们可以通过为各张牌设定一个 z-index 的方式来表示这手牌：



在这个例子中，我们采用了 1-5 五个连续的数字来表示堆叠次序，但是你也可以用五个不同的其他数字来取得同样的效果。这里的要点在于：用数字的大小次序反映希望的堆叠次序。

扑克牌这个例子的代码可以这样写：

```
#ten_of_diamonds {

    position: absolute;

    left: 100px;

    top: 100px;

    z-index: 1;
```

```
}
```

```
#jack_of_diamonds {  
    position: absolute;  
    left: 115px;  
    top: 115px;  
    z-index: 2;  
}
```

```
#queen_of_diamonds {  
    position: absolute;  
    left: 130px;  
    top: 130px;  
    z-index: 3;  
}
```

```
#king_of_diamonds {  
    position: absolute;  
    left: 145px;  
    top: 145px;  
    z-index: 4;  
}
```

```
#ace_of_diamonds {  
  
    position: absolute;  
  
    left: 160px;  
  
    top: 160px;  
  
    z-index: 5;  
  
}
```

- [显示示例](#)

第 16 课: Web 标准与验证

W3C即[万维网联盟 \(World Wide Web Consortium\)](#)，是一个制订相关Web标准（如HTML、CSS和XML等）的非盈利组织。微软、Mozilla基金会以及许多其他的公司与组织都是W3C的成员，它们共同协商确定Web标准的未来发展。

假如你曾有过 Web 开发经验的话，你也许知道，同一网页在不同浏览器上的显示效果会存在着天壤之别。要设计一个能在 Mozilla、Internet Explorer、Opera 及其他现有浏览器上都能良好显示的网页，是件十分费时和令人头痛的事。

制订标准的目的是为了在关于“如何使用 Web 技术”这个问题上达成统一意见。这意味着，Web 开发者只要遵循标准就能确保他所设计的网页能在不同浏览器上均有较良好的显示效果。**因此，我们建议你支持 W3C 的工作，并对你的 CSS 进行验证，以确保符合标准。**

CSS 验证器

为了便于验证是否符合[CSS标准](#)，W3C开发了一个称为[验证器](#)的程序。它可以读取你的样式表，并验证你的样式表是否符合CSS标准，如果不符合的话，它会列出错误并给出警告信息。

为了方便进行样式表的验证，你可以直接在网页上进行验证。你只要在下面的输入框里提供自己的样式单的 URL，并点击“验证样式表”即可。如果有错误，网页上将给出有关错误信息。

验证样式表

如果验证器没有发现任何错误，那么网页上将显示下面的图片。你可以将这个图片放在你的网页里，表明你所用的 CSS 是符合标准的。



你可以通过下面这个链接进入W3C CSS验证器页面: <http://jigsaw.w3.org/css-validator/>