# 语法

CSS 规则由两个主要的部分构成：选择器，以及一条或多条声明:

选择器通常是您需要改变样式的 HTML 元素。

每条声明由一个属性和一个值组成。

属性（property）是您希望设置的样式属性（style attribute）。每个属性有一个值。属性和值被冒号分开。

CSS声明总是以分号(;)结束，声明组以大括号({})括起来:

p {color:red;text-align:center;}

## CSS 颜色值的写法

在描述颜色的时候，除了使用英文单词 red，我们还可以使用十六进制的颜色值 #ff0000：

p { color: #ff0000; }

为了节约字节，我们可以使用 CSS 的缩写形式：

p { color: #f00; }

我们还可以通过两种方法使用 RGB 值：

p { color: rgb(255,0,0); } p { color: rgb(100%,0%,0%); }

**提示：**当使用 RGB 百分比时，即使当值为 0 时也要写百分比符号。但是在其他的情况下就不需要这么做了。比如说，当尺寸为 0 像素时，0 之后不需要使用 px 单位。

## CSS 注释

注释是用来解释你的代码，并且可以随意编辑它，浏览器会忽略它。

CSS注释以 "/\*" 开始, 以 "\*/" 结束

# CSS Id 和 Class选择器

如果你要在HTML元素中设置CSS样式，你需要在元素中设置"id" 和 "class"选择器。

## id 选择器

id 选择器可以为标有特定 id 的 HTML 元素指定特定的样式。

HTML元素以id属性来设置id选择器,CSS 中 id 选择器以 "#" 来定义。

#para1{text-align:center;color:red;}

Remark ID属性不要以数字开头，数字开头的ID在 Mozilla/Firefox 浏览器中不起作用。

Remark ID属性只能在每个 HTML 文档中出现一次。

## class 选择器

class 选择器用于描述一组元素的样式，class 选择器有别于id选择器，class可以在多个元素中使用。

class 选择器在HTML中以class属性表示, 在 CSS 中，类选择器以一个点"."号显示

.center {text-align:center;}

 类名的第一个字符不能使用数字！它无法在 Mozilla 或 Firefox 中起作用。



# CSS 创建

插入样式表的方法有三种:

* 外部样式表
* 内部样式表
* 内联样式

## 外部样式表

当样式需要应用于很多页面时，外部样式表将是理想的选择。在使用外部样式表的情况下，你可以通过改变一个文件来改变整个站点的外观。每个页面使用 标签链接到样式表。 标签在（文档的）头部

<head> <link rel="stylesheet" type="text/css" href="mystyle.css"> </head>

浏览器会从文件 mystyle.css 中读到样式声明，并根据它来格式文档。

外部样式表可以在任何文本编辑器中进行编辑。文件不能包含任何的 html 标签。样式表应该以 .css 扩展名进行保存。下面是一个样式表文件的例子：

hr {color:sienna;}   
p {margin-left:20px;}   
body {background-image:url(/images/back40.gif);}

Remark不要在属性值与单位之间留有空格。假如你使用 "margin-left: 20 px" 而不是 "margin-left: 20px" ，它仅在 IE 6 中有效，但是在 Mozilla/Firefox 或 Netscape 中却无法正常工作。

## 内部样式表

当单个文档需要特殊的样式时，就应该使用内部样式表。你可以使用 <style> 标签在文档头部定义内部样式表，就像这样:

<**head**>

<**style**>

**hr** {**color**:sienna;}

**p** {**margin-left**:20px;}

**body** {**background-image**:url("images/back40.gif");}

</**style**>

</**head**>

## 内联样式

由于要将表现和内容混杂在一起，内联样式会损失掉样式表的许多优势。请慎用这种方法，例如当样式仅需要在一个元素上应用一次时。

要使用内联样式，你需要在相关的标签内使用样式（style）属性。Style 属性可以包含任何 CSS 属性。本例展示如何改变段落的颜色和左外边距：

<**p** **style**="**color**:sienna;**margin-left**:20px">这是一个段落。</**p**>

## 多重样式

如果某些属性在不同的样式表中被同样的选择器定义，那么属性值将从更具体的样式表中被继承过来。

例如，外部样式表拥有针对 h3 选择器的三个属性：

h3   
{   
color:red;   
text-align:left;   
font-size:8pt;   
}

而内部样式表拥有针对 h3 选择器的两个属性：

h3   
{   
text-align:right;   
font-size:20pt;   
}

假如拥有内部样式表的这个页面同时与外部样式表链接，那么 h3 得到的样式是：

color:red;

text-align:right;   
font-size:20pt;

即颜色属性将被继承于外部样式表，而文字排列（text-alignment）和字体尺寸（font-size）会被内部样式表中的规则取代。

## 多重样式将层叠为一个

样式表允许以多种方式规定样式信息。样式可以规定在单个的 HTML 元素中，在 HTML 页的头元素中，或在一个外部的 CSS 文件中。甚至可以在同一个 HTML 文档内部引用多个外部样式表。

### 层叠次序

当同一个 HTML 元素被不止一个样式定义时，会使用哪个样式呢？

一般而言，所有的样式会根据下面的规则层叠于一个新的虚拟样式表中，其中数字 4 拥有最高的优先权。

1. 浏览器缺省设置
2. 外部样式表
3. 内部样式表（位于 head 标签内部）
4. 内联样式（在 HTML 元素内部）

因此，内联样式（在 HTML 元素内部）拥有最高的优先权，这意味着它将优先于以下的样式声明： 标签中的样式声明，外部样式表中的样式声明，或者浏览器中的样式声明（缺省值）。

Remark**提示:**如果你使用了外部文件的样式在内部样式中也定义了该样式，则内部样式表会取代外部文件的样式。

## 多重样式优先级深入理解

优先级用于浏览器是通过判断哪些属性值与元素最相关以决定并应用到该元素上的。

优先级仅由选择器组成的匹配规则决定的。

优先级就是分配给指定的CSS声明的一个权重，它由匹配的选择器中的每一种选择器类型的数值决定。

### 多重样式优先级顺序

下列是一份优先级逐级增加的选择器列表，其中数字 7 拥有最高的优先权：

1. 通用选择器（\*）
2. 元素(类型)选择器
3. 类选择器
4. 属性选择器
5. 伪类
6. ID 选择器
7. 内联样式

### !important 规则例外

当 !important 规则被应用在一个样式声明中时，该样式声明会覆盖CSS中任何其他的声明，无论它处在声明列表中的哪里。尽管如此，!important规则还是与优先级毫无关系。使用 !important 不是一个好习惯，因为它改变了你样式表本来的级联规则，从而使其难以调试。

一些经验法则：

* Always 要优化考虑使用样式规则的优先级来解决问题而不是 !important
* Only 只在需要覆盖全站或外部 css（例如引用的 ExtJs 或者 YUI ）的特定页面中使用 !important
* Never 永远不要在全站范围的 css 上使用 !important
* Never 永远不要在你的插件中使用 !important

### 权重计算:



以下是对于上图的解释：

* 内联样式表的权值最高 1000
* ID 选择器的权值为 100
* Class 类选择器的权值为 10
* HTML 标签选择器的权值为 1

### CSS 优先级法则：

* A 选择器都有一个权值，权值越大越优先；
* B 当权值相等时，后出现的样式表设置要优于先出现的样式表设置；
* C 创作者的规则高于浏览者：即网页编写者设置的CSS 样式的优先权高于浏览器所设置的样式；
* D 继承的CSS 样式不如后来指定的CSS 样式；
* E 在同一组属性设置中标有"!important"规则的优先级最大；

# CSS Backgrounds(背景)

## 背景颜色

background-color 属性定义了元素的背景颜色。

页面的背景颜色使用在body的选择器中:

body {background-color:#b0c4de;}

CSS中，颜色值通常以以下方式定义:

* 十六进制 - 如："#ff0000"
* RGB - 如："rgb(255,0,0)"
* 颜色名称 - 如："red"

**提示：**你可以为所有元素设置背景色，包括 body 一直到 em 和 a 等行内元素。

**提示：**background-color 不能继承，其默认值是 transparent。如果一个元素没有指定背景色，那么背景就是透明的，这样其父元素的背景才可见。

## 背景图像

background-image 属性描述了元素的背景图像.

默认情况下，背景图像进行平铺重复显示，以覆盖整个元素实体.

页面背景图片设置实例:

body {background-image:url('paper.gif');}

## 背景图像 - 水平或垂直平铺

如果需要在HTML页面上对背景图像进行平铺，可以使用 [background-repeat](https://www.w3cschool.cn/cssref/pr-background-repeat.html) 属性。

默认情况下 background-image 属性会在页面的水平或者垂直方向平铺。

body   
{   
background-image:url('gradient2.png');   
background-repeat:repeat-x;   
}

## 背景图像- 设置定位与不平铺

Remark让背景图像不影响文本的排版

如果你不想让图像平铺，你可以使用 background-repeat 属性:

body   
{   
background-image:url('img\_tree.png');   
background-repeat:no-repeat;   
}

可以利用 background-position 属性改变图像在背景中的位置

body   
{   
background-image:url('img\_tree.png');   
background-repeat:no-repeat;   
background-position:right top;   
}

**提示：**为 background-position 属性提供值有很多方法。首先，可以使用一些关键字：top、bottom、left、right 和 center；其次，可以使用长度值，如 100px 或 5cm；最后也可以使用百分数值。不同类型的值对于背景图像的放置稍有差异。

### 关键字

图像放置关键字最容易理解的作用就像其名字的意义。例如，top left 使图像放置在元素内边距区的左上角。

只要保证不超过两个关键字：一个对应水平方向，另一个对应垂直方向，那么你可以设置位置关键字以任何顺序出现。

如果只有一个关键字，则会默认另一个关键字为 center。

下面是等价的位置关键字：

|  |  |
| --- | --- |
| **单一关键字** | **等价的关键字** |
| center | center center |
| top | top center 或 center top |
| bottom | bottom center 或 center bottom |
| right | right center 或 center right |
| left | left center 或 center left |

### 百分数值

百分数值的表现方式更为复杂。假设你希望用百分数值将图像在其元素中居中，你可以按照下面的代码进行设置：

body

{   
background-image:url('img\_tree.png');   
background-repeat:no-repeat;   
  background-position:50% 50%;   
}

这会导致图像适当放置，其中心与其元素的中心对齐。

因此，如果你想把一个图像放在水平方向 2/3、垂直方向 1/3 处，可以这样声明：

body   
{   
background-image:url('img\_tree.png');   
background-repeat:no-repeat;   
  background-position:66% 33%;   
}

### 长度值

长度值解释的是元素内边距区左上角的偏移，偏移点是图像的左上角。

比如，如果设置值为 50px 100px，图像的左上角将在元素内边距区左上角向右 50 像素、向下 100 像素的位置上：

body   
{   
background-image:url('img\_tree.png');   
background-repeat:no-repeat;   
  background-position:50px 100px;   
}

注意，这一点与百分数值不同，因为偏移只是从一个左上角到另一个左上角。也就是说，图像的左上角与 background-position 声明中的指定的点对齐。

## 背景- 简写属性

在以上实例中我们可以看到页面的背景颜色通过了很多的属性来控制。

为了简化这些属性的代码，我们可以将这些属性合并在同一个属性中.

背景颜色的简写属性为 "background":

body {background:#ffffff url('img\_tree.png') no-repeat right top;}

当使用简写属性时，属性值的顺序为：:

* background-color
* background-image
* background-repeat
* background-attachment
* background-position

以上属性无需全部使用，你可以按照页面的实际需要使用.

## CSS 背景属性

|  |  |
| --- | --- |
| **Property** | **描述** |
| [background](https://www.w3cschool.cn/cssref/css3-pr-background.html) | 简写属性，作用是将背景属性设置在一个声明中。 |
| [background-attachment](https://www.w3cschool.cn/cssref/pr-background-attachment.html) | 背景图像是否固定或者随着页面的其余部分滚动。 |
| [background-color](https://www.w3cschool.cn/cssref/pr-background-color.html) | 设置元素的背景颜色。 |
| [background-image](https://www.w3cschool.cn/cssref/pr-background-image.html) | 把图像设置为背景。 |
| [background-position](https://www.w3cschool.cn/cssref/pr-background-position.html) | 设置背景图像的起始位置。 |
| [background-repeat](https://www.w3cschool.cn/cssref/pr-background-repeat.html) | 设置背景图像是否及如何重复。 |

# CSS Text(文本)

通过CSS的Text属性，你可以改变页面中文本的颜色、字符间距、对齐文本、装饰文本、对文本进行缩进等等，你可以观察下述的一段简单的应用了CSS文本格式的段落。

## Text Color

颜色属性被用来设置文字的颜色。

颜色是通过CSS最经常的指定：

* 十六进制值 - 如"＃FF0000"
* 一个RGB值 - "RGB（255,0,0）"
* 颜色的名称 - 如"红"

body {color:blue;}   
h1 {color:#00ff00;}   
h2 {color:rgb(255,0,0);}

对于W3C标准的CSS：如果你定义了颜色属性，你还必须定义背景色属性。



## 文本的对齐方式

文本排列属性是用来设置文本的水平对齐方式。

文本可居中或对齐到左或右,两端对齐.

当text-align设置为"justify"，每一行被展开为宽度相等，左，右外边距是对齐（如杂志和报纸）。

h1 {text-align:center;}   
p.date {text-align:right;}   
p.main {text-align:justify;}

**提示：**如果想把一个行内元素的第一行“缩进”，可以用左内边距或外边距创造这种效果。

## 文本修饰

text-decoration 属性用来设置或删除文本的装饰。

从设计的角度看 text-decoration属性主要是用来删除链接的下划线：

a {text-decoration:none;}

h1 {text-decoration:overline;}   
h2 {text-decoration:line-through;}   
h3 {text-decoration:underline;}

我们不建议强调指出不是链接的文本，因为这常常混淆用户



## 文本转换

文本转换属性是用来指定在一个文本中的大写和小写字母。

可用于所有字句变成大写或小写字母，或每个单词的首字母大写。

p.uppercase {text-transform:uppercase;}   
p.lowercase {text-transform:lowercase;}   
p.capitalize {text-transform:capitalize;}

## 文本缩进

文本缩进属性是用来指定文本的第一行的缩进。

CSS 提供了 text-indent 属性，该属性可以方便地实现文本缩进。

通过使用 text-indent 属性，所有元素的第一行都可以缩进一个给定的长度。

p {text-indent:50px;}

## 文本间隔

word-spacing 属性可以改变字（单词）之间的标准间隔。其默认值 normal 与设置值为 0 是一样的。

指定段字之间的空间，应该是30像素：

p   
{   
word-spacing:30px;   
}

## 指定字符之间的空间

这个例子演示了如何增加或减少字符之间的空间。

h1 {letter-spacing:2px;}

h2 {letter-spacing:-3px;}

## 指定行与行之间的空间

这个例子演示了如何指定在一个段落中行之间的空间

p.small {line-height:70%;}

p.big {line-height:200%;}

## 设置元素的文本方向

这个例子演示了如何改变元素的文本方向。

div.ex1 {direction:rtl;}

## 增加单词之间的空白空间

这个例子演示了如何增加一个段落中的单词之间的空白空间。

p

{

word-spacing:30px;

}

## 在元素内禁用文字环绕

这个例子演示了如何禁用一个元素内的文字环绕。

p

{

white-space:nowrap;

}

## 垂直对齐图像

这个例子演示了如何设置文本的垂直对齐图像。

img.top {vertical-align:text-top;}

img.bottom {vertical-align:text-bottom;}

## 添加文本阴影

这个例子演示了如何设置文本阴影。

h1 {text-shadow:2px 2px #FF0000;}

## 所有CSS文本属性。

|  |  |
| --- | --- |
| **属性** | **描述** |
| [color](https://www.w3cschool.cn/cssref/pr-text-color.html) | 设置文本颜色 |
| [direction](https://www.w3cschool.cn/cssref/pr-text-direction.html) | 设置文本方向。 |
| [letter-spacing](https://www.w3cschool.cn/cssref/pr-text-letter-spacing.html) | 设置字符间距 |
| [line-height](https://www.w3cschool.cn/cssref/pr-dim-line-height.html) | 设置行高 |
| [text-align](https://www.w3cschool.cn/cssref/pr-text-text-align.html) | 对齐元素中的文本 |
| [text-decoration](https://www.w3cschool.cn/cssref/pr-text-text-decoration.html) | 向文本添加修饰 |
| [text-indent](https://www.w3cschool.cn/cssref/pr-text-text-indent.html) | 缩进元素中文本的首行 |
| [text-shadow](https://www.w3cschool.cn/cssref/css3-pr-text-shadow.html) | 设置文本阴影 |
| [text-transform](https://www.w3cschool.cn/cssref/pr-text-text-transform.html) | 控制元素中的字母 |
| [unicode-bidi](https://www.w3cschool.cn/cssref/pr-text-unicode-bidi.html) | 设置或返回文本是否被重写 |
| [vertical-align](https://www.w3cschool.cn/cssref/pr-pos-vertical-align.html) | 设置元素的垂直对齐 |
| [white-space](https://www.w3cschool.cn/cssref/pr-text-white-space.html) | 设置元素中空白的处理方式 |
| [word-spacing](https://www.w3cschool.cn/cssref/pr-text-word-spacing.html) | 设置字间距 |

# CSS Fonts(字体)

CSS字体属性定义字体，加粗，大小，文字样式。

## CSS字型

在CSS中，有两种类型的字体系列名称：

* **通用字体系列** - 拥有相似外观的字体系统组合（如 "Serif" 或 "Monospace"）
* **特定字体系列** - 一个特定的字体系列（如 "Times" 或 "Courier"）

|  |  |  |
| --- | --- | --- |
| **Generic family** | **字体系列** | **说明** |
| Serif | Times New Roman  Georgia | Serif字体中字符在行的末端拥有额外的装饰 |
| Sans-serif | Arial  Verdana | "Sans"是指无 - 这些字体在末端没有额外的装饰 |
| Monospace | Courier New  Lucida Console | 所有的等宽字符具有相同的宽度 |

Remark 除了各种特定的字体系列外，CSS 定义了 5 种通用字体系列：

* Serif 字体
* Sans-serif 字体
* Monospace 字体
* Cursive 字体
* Fantasy 字体

## 字体系列

font-family 属性设置文本的字体系列。

font-family 属性应该设置几个字体名称作为一种"后备"机制，如果浏览器不支持第一种字体，他将尝试下一种字体。

**注意**: 如果字体系列的名称超过一个字，它必须用引号，如Font Family："宋体"。

多个字体系列是用一个逗号分隔指明：

p{font-family:"Times New Roman", Times, serif;}

## 字体样式

主要是用于指定斜体文字的字体样式属性。

这个属性有三个值：

* 正常 - 正常显示文本
* 斜体 - 以斜体字显示的文字
* 倾斜的文字 - 文字向一边倾斜（和斜体非常类似，但不太支持）

p.normal {font-style:normal;}   
p.italic {font-style:italic;}   
p.oblique {font-style:oblique;}

**italic 和 oblique 的区别**

* 斜体（italic）是一种简单的字体风格，对每个字母的结构有一些小改动，来反映变化的外观。
* 倾斜（oblique）文本是正常竖直文本的一个倾斜版本。

通常情况下，italic 和 oblique 文本在 web 浏览器中看上去完全一样。

## 字体大小

font-size 属性设置文本的大小。

能否管理文字的大小，在网页设计中是非常重要的。但是，你不能通过调整字体大小使段落看上去像标题，或者使标题看上去像段落。

请务必使用正确的HTML标签，就<h1> - <h6>表示标题和<p>表示段落：

字体大小的值可以是绝对或相对的大小。

绝对大小：

* 设置一个指定大小的文本
* 不允许用户在所有浏览器中改变文本大小
* 确定了输出的物理尺寸时绝对大小很有用

相对大小：

* 相对于周围的元素来设置大小
* 允许用户在浏览器中改变文字大小

Remark如果你不指定一个字体的大小，默认大小和普通文本段落一样，是16像素（16px=1em）。

### 设置字体大小像素

设置文字的大小与像素，让您完全控制文字大小：

h1 {font-size:40px;}   
h2 {font-size:30px;}   
p {font-size:14px;}

虽然可以通过浏览器的缩放工具调整文本大小，但是，这种调整是整个页面，而不仅仅是文本

### 用em来设置字体大小

为了避免Internet Explorer 中无法调整文本的问题，许多开发者使用 em 单位代替像素。

em的尺寸单位由W3C建议。

1em和当前字体大小相等。在浏览器中默认的文字大小是16px。

因此，1em的默认大小是16px。可以通过下面这个公式将像素转换为em：px/16=em

（注：16 等于父元素的默认字体大小，假设父元素的 font-size 为 30px，那么公式需改为：*pixels*/30=*em*）

h1 {font-size:2.5em;} /\* 40px/16=2.5em \*/   
h2 {font-size:1.875em;} /\* 30px/16=1.875em \*/   
p {font-size:0.875em;} /\* 14px/16=0.875em \*/

在上面的例子，em的文字大小是与前面的例子中像素一样。不过，如果使用 em 单位，则可以在所有浏览器中调整文本大小。

不幸的是，仍然是IE浏览器的问题。调整文本的大小时，会比正常的尺寸更大或更小。

### 使用百分比和EM组合

在所有浏览器的解决方案中，设置 <body>元素的默认字体大小的是百分比：

body {font-size:100%;}   
h1 {font-size:2.5em;}   
h2 {font-size:1.875em;}   
p {font-size:0.875em;}

## 设置字体加粗

这个例子演示了如何设置字体的加粗。

p.normal {font-weight:normal;}

p.light {font-weight:lighter;}

p.thick {font-weight:bold;}

p.thicker {font-weight:900;}

## 可以设置字体的转变

这个例子演示了如何设置字体的转变。

p.normal {font-variant:normal;}

p.small {font-variant:small-caps;}

## 在一个声明中的所有字体属性

本例演示如何使用简写属性将字体属性设置在一个声明之内。

p.ex1

{

font:15px arial,sans-serif;

}

p.ex2

{

font:italic bold 12px/30px Georgia,serif;

}

## 设置字体风格

本例演示如何设置字体风格。

p.normal {font-style:normal;}

p.italic {font-style:italic;}

p.oblique {font-style:oblique;}

## 所有CSS字体属性

|  |  |
| --- | --- |
| **Property** | **描述** |
| [font](https://www.w3cschool.cn/cssref/pr-font-font.html) | 在一个声明中设置所有的字体属性 |
| [font-family](https://www.w3cschool.cn/cssref/pr-font-font-family.html) | 指定文本的字体系列 |
| [font-size](https://www.w3cschool.cn/cssref/pr-font-font-size.html) | 指定文本的字体大小 |
| [font-style](https://www.w3cschool.cn/cssref/pr-font-font-style.html) | 指定文本的字体样式 |
| [font-variant](https://www.w3cschool.cn/cssref/pr-font-font-variant.html) | 以小型大写字体或者正常字体显示文本。 |
| [font-weight](https://www.w3cschool.cn/cssref/pr-font-weight.html) | 指定字体的粗细。 |

# CSS 链接(link)

不同的链接可以有不同的样式。

## 链接样式

链接的样式，可以用任何CSS属性（如颜色，字体，背景等）。

特别的链接，可以有不同的样式，这取决于他们是什么状态。

这四个链接状态是：

* a:link - 正常，未访问过的链接
* a:visited - 用户已访问过的链接
* a:hover - 当用户鼠标放在链接上时
* a:active - 链接被点击的那一刻

当设置为若干链路状态的样式，也有一些顺序规则：

a:hover 必须跟在 a:link 和 a:visited后面

a:active 必须跟在 a:hover后面

## 文本修饰

text-decoration 属性主要用于删除链接中的下划线：

a:link {text-decoration:none;}   
a:visited {text-decoration:none;}   
a:hover {text-decoration:underline;}   
a:active {text-decoration:underline;}

## 背景颜色

背景颜色属性指定链接背景色：

a:link {background-color:#B2FF99;}   
a:visited {background-color:#FFFF85;}   
a:hover {background-color:#FF704D;}   
a:active {background-color:#FF704D;}

## 添加不同样式的超链接

这个例子演示了如何为超链接添加其他样式。

a.one:link {color:#ff0000;}

a.one:visited {color:#0000ff;}

a.one:hover {color:#ffcc00;}

a.two:link {color:#ff0000;}

a.two:visited {color:#0000ff;}

a.two:hover {font-size:150%;}

a.three:link {color:#ff0000;}

a.three:visited {color:#0000ff;}

a.three:hover {background:#66ff66;}

a.four:link {color:#ff0000;}

a.four:visited {color:#0000ff;}

a.four:hover {font-family:monospace;}

a.five:link {color:#ff0000;text-decoration:none;}

a.five:visited {color:#0000ff;text-decoration:none;}

a.five:hover {text-decoration:underline;}

## 高级 - 创建链接框

这个例子演示了一个更高级的例子，我们结合若干CSS属性显示为方框。

a:link,a:visited

{

display:block;

font-weight:bold;

color:#FFFFFF;

background-color:#98bf21;

width:120px;

text-align:center;

padding:4px;

text-decoration:none;

}

a:hover,a:active

{

background-color:#7A991A;

}

# CSS 列表样式(ul)

CSS列表属性作用如下：

* 设置不同的列表项标记为有序列表
* 设置不同的列表项标记为无序列表
* 设置列表项标记为图像

## 列表

在HTML中，有两种类型的HTML列表：

无序列表 - 列表项标记用特殊图形（如小黑点、小方框等）

有序列表 - 列表项的标记有数字或字母

使用CSS，可以列出进一步的样式，并可用图像作列表项标记。

## 不同的列表项标记

list-style-type属性指定列表项标记的类型是：

ul.a {list-style-type: circle;}  
ul.b {list-style-type: square;}  
ol.c {list-style-type: upper-roman;}  
ol.d {list-style-type: lower-alpha;}

下列是对list-style-type属性的常见属性值的描述：

* none：不使用项目符号
* disc：实心圆
* circle：空心圆
* square：实心方块
* demical：阿拉伯数字
* lower-alpha：小写英文字母
* upper-alpha：大写英文字母
* lower-roman：小写罗马数字
* upper-roman：大写罗马数字

## 作为列表项标记的图像

要指定列表项标记的图像，使用列表样式图像属性：

ul  
{  
list-style-image: url('sqpurple.gif');  
}

**提示：**利用 list-style-position 可以确定标志出现在列表项内容之外还是内容内部。

如果你想在所有的浏览器放置同样的形象标志，就应使用浏览器兼容性解决方案

## 浏览器兼容性解决方案

同样在所有的浏览器，下面的例子会显示的图像标记：

ul  
{  
list-style-type: none;  
padding: 0px;  
margin: 0px;  
}  
ul li  
{  
background-image: url(sqpurple.gif);  
background-repeat: no-repeat;  
background-position: 0px 5px;   
padding-left: 14px;   
}

例子解释：

* ul:
  + 设置列表样式类型为没有列表项标记
  + 设置填充和边距0px（浏览器兼容性）
* ul中所有li:
  + 设置图像的URL，并设置它只显示一次（无重复）
  + 您需要的定位图像位置（左0px和上下5px）
  + 用padding-left属性把文本置于列表中

## 列表 -缩写属性

在单个属性中可以指定所有的列表属性。这就是所谓的缩写属性。

为列表使用缩写属性，列表样式属性设置如下：

ul  
{  
list-style: square url("sqpurple.gif");  
}

如果使用缩写属性值的顺序是：

* list-style-type
* list-style-position (有关说明，请参见下面的CSS属性表)
* list-style-image

如果上述值丢失一个，其余仍在指定的顺序，就没关系。

## 所有不同的列表项标记

ul.a {list-style-type:circle;}

ul.b {list-style-type:disc;}

ul.c {list-style-type:square;}

ol.d {list-style-type:armenian;}

ol.e {list-style-type:cjk-ideographic;}

ol.f {list-style-type:decimal;}

ol.g {list-style-type:decimal-leading-zero;}

ol.h {list-style-type:georgian;}

ol.i {list-style-type:hebrew;}

ol.j {list-style-type:hiragana;}

ol.k {list-style-type:hiragana-iroha;}

ol.l {list-style-type:katakana;}

ol.m {list-style-type:katagana-iroha;}

ol.n {list-style-type:lower-alpha;}

ol.o {list-style-type:lower-greek;}

ol.p {list-style-type:lower-latin;}

ol.q {list-style-type:lower-roman;}

ol.r {list-style-type:upper-alpha;}

ol.s {list-style-type:upper-latin;}

ol.t {list-style-type:upper-roman;}

ol.u {list-style-type:none;}

ol.v {list-style-type:inherit;}

## 所有的CSS列表属性

|  |  |
| --- | --- |
| **属性** | **描述** |
| [list-style](https://www.w3cschool.cn/cssref/pr-list-style.html) | 简写属性。用于把所有用于列表的属性设置于一个声明中 |
| [list-style-image](https://www.w3cschool.cn/cssref/pr-list-style-image.html) | 将图象设置为列表项标志。 |
| [list-style-position](https://www.w3cschool.cn/cssref/pr-list-style-position.html) | 设置列表中列表项标志的位置。 |
| [list-style-type](https://www.w3cschool.cn/cssref/pr-list-style-type.html) | 设置列表项标志的类型。 |

# CSS Table(表格)

## 表格边框

指定CSS表格边框，使用border属性。

table, th, td   
{   
border: 1px solid black;   
}

请注意，在上面的例子中的表格有双边框。这是因为表和th/ td元素有独立的边界。

为了显示一个表的单个边框，使用 border-collapse属性。

## 折叠边框

border-collapse 属性设置表格的边框是否被折叠成一个单一的边框或隔开：

table   
{   
border-collapse:collapse;   
}   
table,th, td   
{   
border: 1px solid black;   
}

## 表格宽度和高度

Width和height属性定义表格的宽度和高度。

下面的例子是设置100％的宽度，50像素的th元素的高度的表格：

table   
{   
width:100%;   
}   
th   
{   
height:50px;   
}

## 表格文字对齐

表格中的文本对齐和垂直对齐属性。

text-align属性设置水平对齐方式，像左，右，或中心：

td   
{   
text-align:right;   
}

垂直对齐属性设置垂直对齐，比如顶部，底部或中间：

td   
{   
height:50px;   
vertical-align:bottom;   
}

## 表格填充

如果在表的内容中控制空格之间的边框，应使用td和th元素的填充属性：

td   
{   
padding:15px;   
}

## 表格颜色

下面的例子指定边框的颜色，和th元素的文本和背景颜色：

table, td, th   
{   
border:1px solid green;   
}   
th   
{   
background-color:green;   
color:white;   
}

## 制作一个个性表格

#customers

{

font-family:"Trebuchet MS", Arial, Helvetica, sans-serif;

width:100%;

border-collapse:collapse;

}

#customers td, #customers th

{

font-size:1em;

border:1px solid #98bf21;

padding:3px 7px 2px 7px;

}

#customers th

{

font-size:1.1em;

text-align:left;

padding-top:5px;

padding-bottom:4px;

background-color:#A7C942;

color:#ffffff;

}

#customers tr.alt td

{

color:#000000;

background-color:#EAF2D3;

}

## 设置表格标题的位置

caption {caption-side:bottom;}

## 指定表格的宽度和高度

table,td,th

{

border:1px solid black;

}

table

{

width:100%;

}

th

{

height:50px;

}

# CSS 盒子模型

所有HTML元素可以看作盒子，在CSS中，"box model"这一术语是用来设计和布局时使用。

CSS盒模型本质上是一个盒子，封装周围的HTML元素，它包括：边距，边框，填充，和实际内容。

盒模型允许我们在其它元素和周围元素边框之间的空间放置元素。

下面的图片说明了盒子模型(Box Model)：



不同部分的说明：

* **Margin（外边距）** - 清除边框区域。Margin没有背景颜色，它是完全透明
* **Border（边框）** - 边框周围的填充和内容。边框是受到盒子的背景颜色影响
* **Padding（内边距）** - 清除内容周围的区域。会受到框中填充的背景颜色影响
* **Content（内容）** - 盒子的内容，显示文本和图像

为了在所有浏览器中的元素的宽度和高度设置正确的话，你需要知道的盒模型是如何工作的。

提示：在盒模型中，外边距可以是负值，而且在很多情况下都要使用负值的外边距。

## 元素的宽度和高度

**重要:** 当您指定一个CSS元素的宽度和高度属性时，你只是设置内容区域的宽度和高度。要知道，完全大小的元素，你还必须添加填充，边框和边距。

下面的例子中的元素的总宽度为300px：

width:250px;   
padding:10px;   
border:5px solid gray;   
margin:10px;

最终元素的总宽度计算公式是这样的：

总元素的宽度=宽度+左填充+右填充+左边框+右边框+左边距+右边距

元素的总高度最终计算公式是这样的：

总元素的高度=高度+顶部填充+底部填充+上边框+下边框+上边距+下边距

## 浏览器的兼容性问题

一旦为页面设置了恰当的 DTD，大多数浏览器都会按照上面的图示来呈现内容。然而 IE 5 和 6 的呈现却是不正确的。根据 W3C 的规范，元素内容占据的空间是由 width 属性设置的，而内容周围的 padding 和 border 值是另外计算的。不幸的是，IE5.X 和 6 在怪异模式中使用自己的非标准模型。这些浏览器的 width 属性不是内容的宽度，而是内容、内边距和边框的宽度的总和。

虽然有方法解决这个问题。但是目前最好的解决方案是回避这个问题。也就是，不要给元素添加具有指定宽度的内边距，而是尝试将内边距或外边距添加到元素的父元素和子元素。

IE8 及更早IE版本不支持 填充的宽度和边框的宽度属性设。

解决IE8及更早版本不兼容问题可以在HTML页面声明 <!DOCTYPE html>即可。