

以下内容摘自  <http://www.w3school.com.cn/css/index.asp>

CSS 背景属性 (Background)

属性	描述	CSS
<u>background</u>	在一个声明中设置所有的背景属性。	1
<u>background-attachment</u>	设置背景图像是否固定或者随着页面的其余部分滚动。	1
<u>background-color</u>	设置元素的背景颜色。	1
<u>background-image</u>	设置元素的背景图像。	1
<u>background-position</u>	设置背景图像的开始位置。	1
<u>background-repeat</u>	设置是否及如何重复背景图像。	1

CSS background 属性

定义和用法

`background` 简写属性在一个声明中设置所有的背景属性。

可以按顺序设置如下属性：

- `background-color`
- `background-image`
- `background-repeat`
- `background-attachment`
- `background-position`

如果不设置其中的某个值，也不会出问题，比如 `background:#ff0000 url('smiley.gif');` 也是允许的。

通常建议使用这个属性，而不是分别使用单个属性，因为这个属性在较老的浏览器中能够得到更好的支持，而且需要键入的字母也更少。

默认值:	<i>not specified</i>
继承性:	no
版本:	CSS1
JavaScript 语法:	<code>object.style.background="white url(paper.gif) repeat-y"</code>

实例

如何在一个声明中设置所有背景属性：

```
body
```

```
{  
background: #00FF00 url(bgimage.gif) no-repeat fixed top;  
}
```

浏览器支持

所有浏览器都支持 `background` 属性。

任何的版本的 Internet Explorer (包括 IE8) 都不支持属性值 "inherit"。

可能的值

值	描述
<code>background-color</code>	规定要使用的背景颜色。参阅: background-color 中可能的值。
<code>background-image</code>	规定要使用的背景图像。参阅: background-image 中可能的值。
<code>background-repeat</code>	规定如何重复背景图像。参阅: background-repeat 中可能的值。
<code>background-attachment</code>	规定背景图像是否固定或者随着页面的其余部分滚动。参阅: background-attachment 中可能的值。
<code>background-position</code>	规定背景图像的位置。参阅: background-position 中可能的值。
<code>inherit</code>	规定应该从父元素继承 <code>background</code> 属性的设置。

CSS `background-color` 属性

定义和用法

`background-color` 属性设置元素的背景颜色。

元素背景的范围

`background-color` 属性为元素设置一种纯色。这种颜色会填充元素的内容、内边距和边框区域，扩展到元素边框的外边界（但不包括外边距）。如果边框有透明部分（如虚线边框），会透过这些透明部分显示出背景色。

`transparent` 值

尽管在大多数情况下，没有必要使用 `transparent`。不过如果您不希望某元素拥有背景色，同时又不希望用户对浏览器的颜色设置影响到您的设计，那么设置 `transparent` 值还是有必要的。

默认值:	transparent
继承性:	no
版本:	CSS1
JavaScript 语法:	<code>object.style.backgroundColor="#00FF00"</code>

实例

```
body
{
    background-color:yellow;
}

h1
{
    background-color:#00ff00;
}

p
{
    background-color:rgb(255,0,255);
}
```

TIY

浏览器支持

所有浏览器都支持 `background-color` 属性。

注释: 任何版本的 Internet Explorer (包括 IE8) 都不支持属性值 "inherit"。

可能的值

值	描述
<code>color_name</code>	规定颜色值为颜色名称的背景颜色 (比如 <code>red</code>)。
<code>hex_number</code>	规定颜色值为十六进制值的背景颜色 (比如 <code>#ff0000</code>)。
<code>rgb_number</code>	规定颜色值为 <code>rgb</code> 代码的背景颜色 (比如 <code>rgb(255,0,0)</code>)。
<code>transparent</code>	默认。背景颜色为透明。

inherit	规定应该从父元素继承 background-color 属性的设置。
---------	------------------------------------

CSS background-image 属性

定义和用法

background-image 属性为元素设置背景图像。

元素的背景占据了元素的全部尺寸，包括内边距和边框，但不包括外边距。

默认地，背景图像位于元素的左上角，并在水平和垂直方向上重复。

提示：请设置一种可用的背景颜色，这样的话，假如背景图像不可用，页面也可获得良好的视觉效果。

详细说明

background-image 属性会在元素的背景中设置一个图像。

根据 [background-repeat](#) 属性的值，图像可以无限平铺、沿着某个轴（x 轴或 y 轴）平铺，或者根本不平铺。

初始背景图像（原图像）根据 [background-position](#) 属性的值放置。

默认值:	none
继承性:	no
版本:	CSS1
JavaScript 语法:	<code>object.style.backgroundImage="url(stars.gif)"</code>

实例

```
body
{
    background-image: url(bgimage.gif);
    background-color: #000000;
}
```

[TIY](#)

浏览器支持

所有浏览器都支持 `background-image` 属性。

注释: 任何版本的 Internet Explorer (包括 IE8) 都不支持属性值 "inherit"。

可能的值

值	描述
<code>url('URL')</code>	指向图像的路径。
<code>none</code>	默认值。不显示背景图像。
<code>inherit</code>	规定应该从父元素继承 <code>background-image</code> 属性的设置。

CSS `background-repeat` 属性

定义和用法

`background-repeat` 属性设置是否及如何重复背景图像。

默认地，背景图像在水平和垂直方向上重复。

详细说明

`background-repeat` 属性定义了图像的平铺模式。

从原图像开始重复，原图像由 [background-image](#) 定义，并根据 [background-position](#) 的值放置。

默认值:	<code>repeat</code>
继承性:	<code>no</code>
版本:	CSS1
JavaScript 语法:	<code>object.style.backgroundPosition="center"</code>

实例

```
body
{
    background-image: url(stars.gif);
    background-repeat: repeat-y;
}
```

[TIY](#)

浏览器支持

所有浏览器都支持 `background-repeat` 属性。

注释: 任何版本的 Internet Explorer (包括 IE8) 都不支持属性值 "inherit"。

提示和注释

提示: 背景图像的位置是根据 `background-position` 属性设置的。如果未规定 `background-position` 属性, 图像会被放置在元素的左上角。

可能的值

值	描述
<code>repeat</code>	默认。背景图像将在垂直方向和水平方向重复。
<code>repeat-x</code>	背景图像将在水平方向重复。
<code>repeat-y</code>	背景图像将在垂直方向重复。
<code>no-repeat</code>	背景图像将仅显示一次。
<code>inherit</code>	规定应该从父元素继承 <code>background-repeat</code> 属性的设置。

CSS `background-attachment` 属性

定义和用法

`background-attachment` 属性设置背景图像是否固定或者随着页面的其余部分滚动。

默认值:	<code>scroll</code>
继承性:	<code>no</code>
版本:	CSS1
JavaScript 语法:	<code>object.style.backgroundAttachment="fixed"</code>

实例

如何设置固定的背景图像:

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

浏览器支持

所有浏览器都支持 `background-attachment` 属性。

注释: 任何版本的 Internet Explorer (包括 IE8) 都不支持属性值 "inherit"。

可能的值

值	描述
<code>scroll</code>	默认值。背景图像会随着页面其余部分的滚动而移动。
<code>fixed</code>	当页面的其余部分滚动时，背景图像不会移动。
<code>inherit</code>	规定应该从父元素继承 <code>background-attachment</code> 属性的设置。

CSS `background-position` 属性

定义和用法

`background-position` 属性设置背景图像的起始位置。

这个属性设置背景原图像 (由 [background-image](#) 定义) 的位置，背景图像如果要重复，将从这一点开始。

提示: 您需要把 `background-attachment` 属性设置为 "fixed"，才能保证该属性在 Firefox 和 Opera 中正常工作。

默认值:	0% 0%
继承性:	no
版本:	CSS1
JavaScript 语法:	<code>object.style.backgroundPosition="center"</code>

实例

如何定位背景图像：

```
body
{
background-image:url('bgimage.gif');
background-repeat:no-repeat;
```

```
background-attachment:fixed;  
background-position:center;  
}
```

浏览器支持

所有浏览器都支持 `background-position` 属性。

注释: 任何版本的 Internet Explorer (包括 IE8) 都不支持属性值 "inherit"。

可能的值

值	描述
top left top center top right center left center center center right bottom left bottom center bottom right	如果您仅规定了一个关键词, 那么第二个值将是"center"。 默认值: 0% 0%。
x% y%	第一个值是水平位置, 第二个值是垂直位置。 左上角是 0% 0%。右下角是 100% 100%。 如果您仅规定了一个值, 另一个值将是 50%。
xpos ypos	第一个值是水平位置, 第二个值是垂直位置。 左上角是 0 0。单位是像素 (0px 0px) 或任何其他的 CSS 单位。 如果您仅规定了一个值, 另一个值将是50%。 您可以混合使用 % 和 position 值。

CSS 边框属性 (Border 和 Outline)

属性	描述	CSS
border	在一个声明中设置所有的边框属性。	1

border-bottom	在一个声明中设置所有的下边框属性。	1
border-bottom-color	设置下边框的颜色。	2
border-bottom-style	设置下边框的样式。	2
border-bottom-width	设置下边框的宽度。	1
border-color	设置四条边框的颜色。	1
border-left	在一个声明中设置所有的左边框属性。	1
border-left-color	设置左边框的颜色。	2
border-left-style	设置左边框的样式。	2
border-left-width	设置左边框的宽度。	1
border-right	在一个声明中设置所有的右边框属性。	1
border-right-color	设置右边框的颜色。	2
border-right-style	设置右边框的样式。	2
border-right-width	设置右边框的宽度。	1
border-style	设置四条边框的样式。	1
border-top	在一个声明中设置所有的上边框属性。	1
border-top-color	设置上边框的颜色。	2
border-top-style	设置上边框的样式。	2
border-top-width	设置上边框的宽度。	1
border-width	设置四条边框的宽度。	1
outline	在一个声明中设置所有的轮廓属性。	2
outline-color	设置轮廓的颜色。	2
outline-style	设置轮廓的样式。	2
outline-width	设置轮廓的宽度。	2

CSS border 属性

定义和用法

border 简写属性在一个声明设置所有的边框属性。

可以按顺序设置如下属性：

- border-width

- `border-style`
- `border-color`

如果不设置其中的某个值，也不会出问题，比如 `border:solid #ff0000;` 也是允许的。

默认值:	<i>not specified</i>
继承性:	no
版本:	CSS1
JavaScript 语法:	<code>object.style.border="3px solid blue"</code>

实例

设置 4 个边框的样式：

```
p
{
  border:5px solid red;
}
```

浏览器支持

所有浏览器都支持 `border` 属性。

注释: 任何的版本的 Internet Explorer (包括 IE8) 都不支持属性值 "inherit"。

可能的值

值	描述
<code>border-width</code>	规定边框的宽度。参阅： border-width 中可能的值。
<code>border-style</code>	规定边框的样式。参阅： border-style 中可能的值。
<code>border-color</code>	规定边框的颜色。参阅： border-color 中可能的值。
<code>inherit</code>	规定应该从父元素继承 <code>border</code> 属性的设置。

CSS border-width 属性

定义和用法

`border-width` 简写属性为元素的所有边框设置宽度，或者单独地为各边边框设置宽度。

只有当边框样式不是 `none` 时才起作用。如果边框样式是 `none`，边框宽度实际上会重置为 0。不允许指

定负长度值。

例子 1

```
border-width:thin medium thick 10px;
```

- 上边框是细边框
- 右边框是中等边框
- 下边框是粗边框
- 左边框是 10px 宽的边框

例子 2

```
border-width:thin medium thick;
```

- 上边框是 10px
- 右边框和左边框是中等边框
- 下边框是粗边框

例子 3

```
border-width:thin medium;
```

- 上边框和下边框是细边框
- 右边框和左边框是中等边框

例子 4

```
border-width:thin;
```

- 所有 4 个边框都是细边框

默认值:	medium
继承性:	no
版本:	CSS1
JavaScript 语法:	<code>object.style.borderWidth="thin thick"</code>

实例

设置四个边框的宽度：

```
p
```

```
{  
border-style:solid;  
border-width:15px;  
}
```

浏览器支持

所有浏览器都支持 `border-width` 属性。

注释: 任何的版本的 Internet Explorer (包括 IE8) 都不支持属性值 "inherit"。

可能的值

值	描述
<code>thin</code>	定义细的边框。
<code>medium</code>	默认。定义中等的边框。
<code>thick</code>	定义粗的边框。
<code>length</code>	允许您自定义边框的宽度。
<code>inherit</code>	规定应该从父元素继承边框宽度。

CSS `border-style` 属性

定义和用法

`border-style` 属性用于设置元素所有边框的样式，或者单独地为各边设置边框样式。

只有当这个值不是 `none` 时边框才可能出现。

例子 1

```
border-style:dotted solid double dashed;
```

- 上边框是点状
- 右边框是实线
- 下边框是双线
- 左边框是虚线

例子 2

```
border-style:dotted solid double;
```

- 上边框是点状
- 右边框和左边框是实线
- 下边框是双线

例子 3

```
border-style:dotted solid;
```

- 上边框和下边框是点状
- 右边框和左边框是实线

例子 4

```
border-style:dotted;
```

- 所有 4 个边框都是点状

默认值:	<i>not specified</i>
继承性:	no
版本:	CSS1
JavaScript 语法:	<code>object.style.borderWidth="dotted double"</code>

实例

设置 4 个边框的样式:

```
p  
{  
border-style:solid;  
}
```

浏览器支持

所有浏览器都支持 **border-style** 属性。

注释: 任何的版本的 Internet Explorer (包括 IE8) 都不支持属性值 "inherit"。

可能的值

值	描述
none	定义无边框。
hidden	与 "none" 相同。不过应用于表时除外，对于表， <code>hidden</code> 用于解决边框冲突。
dotted	定义点状边框。在大多数浏览器中呈现为实线。
dashed	定义虚线。在大多数浏览器中呈现为实线。
solid	定义实线。
double	定义双线。双线的宽度等于 <code>border-width</code> 的值。
groove	定义 3D 凹槽边框。其效果取决于 <code>border-color</code> 的值。
ridge	定义 3D 垒状边框。其效果取决于 <code>border-color</code> 的值。
inset	定义 3D inset 边框。其效果取决于 <code>border-color</code> 的值。
outset	定义 3D outset 边框。其效果取决于 <code>border-color</code> 的值。
inherit	规定应该从父元素继承边框样式。

描述

最不可预测的边框样式是 `double`。它定义为两条线的宽度再加上这两条线之间的空间等于 `border-width` 值。不过，CSS 规范并没有说其中一条线是否比另一条粗或者两条线是否应该是一样的粗，也没有指出线之间的空间是否应当比线粗。所有这些都有用户代理决定，创作人员对这个决定没有任何影响。

CSS `border-color` 属性

定义和用法

`border-color` 属性设置四条边框的颜色。此属性可设置 1 到 4 种颜色。

`border-color` 属性是一个简写属性，可设置一个元素的所有边框中可见部分的颜色，或者为 4 个边分别设置不同的颜色。请看下面的例子：

例子 1

```
border-color:red green blue pink;
```

- 上边框是红色
- 右边框是绿色

- 下边框是蓝色
- 左边框是粉色

例子 2

```
border-color:red green blue;
```

- 上边框是红色
- 右边框和左边框是绿色
- 下边框是蓝色

例子 3

```
border-color:dotted red green;
```

- 上边框和下边框是红色
- 右边框和左边框是绿色

例子 4

```
border-color:red;
```

- 所有 4 个边框都是红色

要记住，边框的样式不能为 `none` 或 `hidden`，否则边框不会出现。

注释：请始终把 `border-style` 属性声明到 `border-color` 属性之前。元素必须在您改变其颜色之前获得边框。

默认值:	<i>not specified</i>
继承性:	no
版本:	CSS1
JavaScript 语法:	<code>object.style.borderColor="#FF0000 blue"</code>

实例

设置 4 个边框的颜色：

```
p  
{  
border-style:solid;
```

```
border-color:#ff0000 #0000ff;  
}
```

TIY

浏览器支持

所有浏览器都支持 `border-color` 属性。

注释: Internet Explorer 6 (以及更早的版本) 不支持属性值 "transparent"。

注释: 任何的版本的 Internet Explorer (包括 IE8) 都不支持属性值 "inherit"。

可能的值

值	描述
<code>color_name</code>	规定颜色值为颜色名称的边框颜色 (比如 <code>red</code>)。
<code>hex_number</code>	规定颜色值为十六进制值的边框颜色 (比如 <code>#ff0000</code>)。
<code>rgb_number</code>	规定颜色值为 <code>rgb</code> 代码的边框颜色 (比如 <code>rgb(255,0,0)</code>)。
<code>transparent</code>	默认值。边框颜色为透明。
<code>inherit</code>	规定应该从父元素继承边框颜色。

CSS 文本属性 (Text)

属性	描述	CSS
<u>color</u>	设置文本的颜色。	1
<u>direction</u>	规定文本的方向 / 书写方向。	2
<u>letter-spacing</u>	设置字符间距。	1
<u>line-height</u>	设置行高。	1
<u>text-align</u>	规定文本的水平对齐方式。	1
<u>text-decoration</u>	规定添加到文本的装饰效果。	1
<u>text-indent</u>	规定文本块首行的缩进。	1
<u>text-shadow</u>	规定添加到文本的阴影效果。	2
<u>text-transform</u>	控制文本的大小写。	1

unicode-bidi	设置文本方向。	2
white-space	规定如何处理元素中的空白。	1
word-spacing	设置单词间距。	1

CSS 列表属性(list)

属性	描述
list-style	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	将图象设置为列表项标志。
list-style-position	设置列表中列表项标志的位置。
list-style-type	设置列表项标志的类型。
marker-offset	

CSS list-style 属性

定义和用法

`list-style` 简写属性在一个声明中设置所有的列表属性。

说明

该属性是一个简写属性，涵盖了所有其他列表样式属性。由于它应用到所有 `display` 为 `list-item` 的元素，所以在普通的 HTML 和 XHTML 中只能用于 `li` 元素，不过实际上它可以应用到任何元素，并由 `list-item` 元素继承。

可以按顺序设置如下属性：

- `list-style-type`
- `list-style-position`
- `list-style-image`

可以不设置其中的某个值，比如 "`list-style:circle inside;`" 也是允许的。未设置的属性会使用其默认值。

默认值:	disc outside none
继承性:	yes
版本:	CSS1
JavaScript 语法:	

浏览器支持

所有浏览器都支持 `list-style` 属性。

注释: 任何的版本的 Internet Explorer (包括 IE8) 都不支持属性值 "inherit"。

可能的值

值	描述
<code>list-style-type</code>	设置列表项标记的类型。参阅: list-style-type 中可能的值。
<code>list-style-position</code>	设置在何处放置列表项标记。参阅: list-style-position 中可能的值。
<code>list-style-image</code>	使用图像来替换列表项的标记。参阅: list-style-image 中可能的值。
<code>inherit</code>	规定应该从父元素继承 <code>list-style</code> 属性的值。

CSS `list-style-image` 属性

定义和用法

`list-style-image` 属性使用图像来替换列表项的标记。

说明

这个属性指定作为一个有序或无序列表项标志的图像。图像相对于列表项内容的放置位置通常使用 `list-style-position` 属性控制。

注释: 请始终规定一个 "`list-style-type`" 属性以防图像不可用。

默认值:	none
继承性:	yes
版本:	CSS1
JavaScript 语法:	<code>object.style.listStyleImage="url('/images/blueball.gif')"</code>

实例

把图像设置为列表中的项目标记:

```
ul
{
    list-style-image:url("/i/arrow.gif");
    list-style-type:square;
```

```
}
```

TIY

浏览器支持

所有浏览器都支持 `list-style-image` 属性。

注释: 任何的版本的 Internet Explorer (包括 IE8) 都不支持属性值 "inherit"。

可能的值

值	描述
<code>URL</code>	图像的路径。
<code>none</code>	默认。无图形被显示。
<code>inherit</code>	规定应该从父元素继承 <code>list-style-image</code> 属性的值。

CSS `list-style-position` 属性

CSS 参考手册

定义和用法

`list-style-position` 属性设置在何处放置列表项标记。

说明

该属性用于声明 **列表标志相对于列表项内容的位置**。外部 (`outside`) 标志会放在离列表项边框边界一定距离处，不过这距离在 CSS 中未定义。内部 (`inside`) 标志处理为好像它们是插入在列表项内容最前面的行内元素一样。

默认值:	<code>outside</code>
继承性:	<code>yes</code>
版本:	CSS1
JavaScript 语法:	<code>object.style.listStylePosition="inside"</code>

实例

规定列表中列表项目标记的位置：

```
ul
{
    list-style-position:inside;
}
```

TIY

浏览器支持

所有浏览器都支持 `list-style-position` 属性。

注释: 任何的版本的 Internet Explorer (包括 IE8) 都不支持属性值 "inherit"。

可能的值

值	描述
<code>inside</code>	列表项目标记放置在文本以内，且环绕文本根据标记对齐。
<code>outside</code>	默认值。保持标记位于文本的左侧。列表项目标记放置在文本以外，且环绕文本不根据标记对齐。
<code>inherit</code>	规定应该从父元素继承 <code>list-style-position</code> 属性的值。

CSS `list-style-type` 属性

CSS 参考手册

定义和用法

`list-style-type` 属性设置列表项标记的类型。

默认值:	disc
继承性:	yes
版本:	CSS1
JavaScript 语法:	<code>object.style.listStyleType="square"</code>

实例

设置不同的列表样式:

```
ul.circle {list-style-type:circle;}  
ul.square {list-style-type:square;}  
ol.upper-roman {list-style-type:upper-roman;}  
ol.lower-alpha {list-style-type:lower-alpha;}
```

[TIY](#) (在页面底部可以找到更多实例)

浏览器支持

所有浏览器都支持 `list-style-type` 属性。

注释: 任何的版本的 Internet Explorer (包括 IE8) 都不支持属性值 "decimal-leading-zero"、"lower-greek"、"lower-latin"、"upper-latin"、"armenian"、"georgian" 或 "inherit"。

可能的值

none	无标记。
disc	默认。标记是实心圆。
circle	标记是空心圆。
square	标记是实心方块。
decimal	标记是数字。
decimal-leading-zero	0开头的数字标记。(01, 02, 03, 等。)
lower-roman	小写罗马数字(i, ii, iii, iv, v, 等。)
upper-roman	大写罗马数字(I, II, III, IV, V, 等。)
lower-alpha	小写英文字母 The marker is lower-alpha (a, b, c, d, e, 等。)
upper-alpha	大写英文字母 The marker is upper-alpha (A, B, C, D, E, 等。)
lower-greek	小写希腊字母(alpha, beta, gamma, 等。)
lower-latin	小写拉丁字母(a, b, c, d, e, 等。)
upper-latin	大写拉丁字母(A, B, C, D, E, 等。)
hebrew	传统的希伯来编号方式
armenian	传统的亚美尼亚编号方式
georgian	传统的乔治亚编号方式(an, ban, gan, 等。)
cjk-ideographic	简单的表意数字

hiragana	标记是: a, i, u, e, o, ka, ki, 等。(日文片假名)
katakana	标记是: A, I, U, E, O, KA, KI, 等。(日文片假名)
hiragana-iroha	标记是: i, ro, ha, ni, ho, he, to, 等。(日文片假名)
katakana-iroha	标记是: I, RO, HA, NI, HO, HE, TO, 等。(日文片假名)

CSS 表格属性

CSS Table 属性

属性	描述
border-collapse	设置是否把表格边框合并为单一的边框。
border-spacing	设置分隔单元格边框的距离。
caption-side	设置表格标题的位置。
empty-cells	设置是否显示表格中的空单元格。
table-layout	设置显示单元、行和列的算法。

CSS border-collapse 属性

[CSS 参考手册](#)

定义和用法

`border-collapse` 属性设置表格的边框是否被合并为一个单一的边框，还是象在标准的 `HTML` 中那样分开显示。

默认值:	separate
继承性:	yes
版本:	CSS2
JavaScript 语法:	<code>object.style.borderCollapse="collapse"</code>

实例

为表格设置合并边框模型：

```
table
```

```
{
```

```
border-collapse:collapse;  
}
```

TIY

浏览器支持

所有主流浏览器都支持 `border-collapse` 属性。

注释: 任何的版本的 Internet Explorer (包括 IE8) 都不支持属性值 "inherit"。

注释: 如果没有规定 !DOCTYPE, 则 `border-collapse` 可能产生意想不到的结果。

可能的值

值	描述
<code>separate</code>	默认值。边框会被分开。不会忽略 <code>border-spacing</code> 和 <code>empty-cells</code> 属性。
<code>collapse</code>	如果可能，边框会合并为一个单一的边框。会忽略 <code>border-spacing</code> 和 <code>empty-cells</code> 属性。
<code>inherit</code>	规定应该从父元素继承 <code>border-collapse</code> 属性的值。

CSS `border-spacing` 属性

CSS 参考手册

定义和用法

`border-spacing` 属性设置相邻单元格的边框间的距离 (仅用于“边框分离”模式)。

注释: 某些版本的 IE 浏览器不支持此属性。

说明

该属性指定分隔边框模型中单元格边界之间的距离。在指定的两个长度值中，第一个是水平间隔，第二个是垂直间隔。除非 `border-collapse` 被设置为 `separate`，否则将忽略这个属性。尽管这个属性只应用于表，不过它可以由表中的所有元素继承。

默认值:	<i>not specified</i>
继承性:	yes
版本:	CSS2

JavaScript 语法: `object.style.borderSpacing="15px"`

实例

为表格设置 `border-spacing`:

```
table
{
    border-collapse:separate;
    border-spacing:10px 50px;
}
```

TIY

浏览器支持

所有主流浏览器都支持 `border-spacing` 属性。

注释: 如果已规定 `!DOCTYPE`, 那么 Internet Explorer 8 (以及更高版本) 支持 `border-spacing` 属性。

可能的值

值	描述
<code>length length</code>	规定相邻单元的边框之间的距离。使用 <code>px</code> 、 <code>cm</code> 等单位。不允许使用负值。 如果定义一个 <code>length</code> 参数, 那么定义的是水平和垂直间距。 如果定义两个 <code>length</code> 参数, 那么第一个设置水平间距, 而第二个设置垂直间距。
<code>inherit</code>	规定应该从父元素继承 <code>border-spacing</code> 属性的值。

CSS `caption-side` 属性

CSS 参考手册

定义和用法

`caption-side` 属性设置表格标题的位置。

说明

该属性指定了表标题相对于表框的放置位置。表标题显示为好像它是表之前（或之后）的一个块级元素。

默认值:	top
继承性:	yes
版本:	CSS2
JavaScript 语法:	<code>object.style.captionSide="bottom"</code>

实例

规定表格标题的放置方式：

```
caption
{
  caption-side:bottom;
}
```

TIY

浏览器支持

所有浏览器都支持 `caption-side` 属性。

注释: 如果已规定 !DOCTYPE, 那么 Internet Explorer 8 (以及更高版本) 支持 `caption-side` 属性。

可能的值

值	描述
top	默认值。把表格标题定位在表格之上。
bottom	把表格标题定位在表格之下。
inherit	规定应该从父元素继承 <code>caption-side</code> 属性的值。

CSS empty-cells 属性

[CSS 参考手册](#)

定义和用法

`empty-cells` 属性设置是否显示表格中的空单元格 (仅用于“分离边框”模式)。

注释: 某些版本的 IE 浏览器不支持此属性。

说明

该属性定义了不包含任何内容的表单元格如何表示。如果显示，就会绘制出单元格的边框和背景。除非 `border-collapse` 设置为 `separate`，否则将忽略这个属性。

默认值:	show
继承性:	yes
版本:	CSS2
JavaScript 语法:	<code>object.style.emptyCells="hide"</code>

实例

隐藏表格中空单元格上的边框和背景：

```
table
{
    border-collapse:separate;
    empty-cells:hide;
}
```

TIY

浏览器支持

所有浏览器都支持 `empty-cells` 属性。

注释: 如果已规定 `!DOCTYPE`，那么 Internet Explorer 8（以及更高版本）支持 `empty-cells` 属性。

可能的值

值	描述
hide	默认。不在空单元格周围绘制边框。
show	在空单元格周围绘制边框。
inherit	规定应该从父元素继承 <code>empty-cells</code> 属性的值。

CSS table-layout 属性

CSS 参考手册

定义和用法

tableLayout 属性用来显示表格单元格、行、列的算法规则。

固定表格布局:

固定表格布局与自动表格布局相比，允许浏览器更快地对表格进行布局。

在固定表格布局中，水平布局仅取决于表格宽度、列宽度、表格边框宽度、单元格间距，而与单元格的内容无关。

通过使用固定表格布局，用户代理在接收到第一行后就可以显示表格。

自动表格布局:

在自动表格布局中，列的宽度是由列单元格中没有折行的最宽的内容设定的。

此算法有时会较慢，这是由于它需要在确定最终的布局之前访问表格中所有的内容。

说明

该属性指定了完成表布局时所用的布局算法。固定布局算法比较快，但是不太灵活，而自动算法比较慢，

不过更能反映传统的 **HTML 表**。

默认值:	auto
继承性:	yes
版本:	CSS2
JavaScript 语法:	<code>object.style.tableLayout="fixed"</code>

实例

设置表格布局算法：

```
table
{
  table-layout: fixed;
}
```

TIY

浏览器支持

所有浏览器都支持 `table-layout` 属性。

注释: 任何的版本的 Internet Explorer (包括 IE8) 都不支持属性值 "inherit"。

可能的值

值	描述
<code>automatic</code>	默认。列宽度由单元格内容设定。
<code>fixed</code>	列宽由表格宽度和列宽度设定。
<code>inherit</code>	规定应该从父元素继承 <code>table-layout</code> 属性的值。

CSS 字体属性

属性	描述
<code>font</code>	简写属性。作用是把所有针对字体的属性设置在一个声明中。
<code>font-family</code>	设置字体系列。
<code>font-size</code>	设置字体的尺寸。
<code>font-size-adjust</code>	当首选字体不可用时，对替换字体进行智能缩放。(CSS2.1 已删除该属性。)
<code>font-stretch</code>	对字体进行水平拉伸。(CSS2.1 已删除该属性。)
<code>font-style</code>	设置字体风格。
<code>font-variant</code>	以小型大写字体或者正常字体显示文本。
<code>font-weight</code>	设置字体的粗细。

CSS font 属性

定义和用法

`font` 简写属性在一个声明中设置所有字体属性。

注释: 此属性也有第六个值: "line-height", 可设置行间距。

说明

这个简写属性用于一次设置元素字体的两个或更多方面。使用 `icon` 等关键字可以适当地设置元素的字体，使之与用户计算机环境中的某个方面一致。注意，如果没有使用这些关键词，至少要指定字体大小和字体

系列。

可以按顺序设置如下属性：

- `font-style`
- `font-variant`
- `font-weight`
- `font-size/line-height`
- `font-family`

可以不设置其中的某个值，比如 `font:100% verdana;` 也是允许的。未设置的属性会使用其默认值。

默认值:	<i>not specified</i>
继承性:	yes
版本:	CSS1
JavaScript 语法:	<code>object.style.font="italic small-caps bold 12px arial,sans-serif"</code>

实例

在一个声明中设置所有字体属性：

```
p.ex1
{
    font:italic arial,sans-serif;
}

p.ex2
{
    font:italic bold 12px/20px arial,sans-serif;
}
```

TIY

浏览器支持

所有浏览器都支持 `font` 属性。

注释：任何的版本的 Internet Explorer（包括 IE8）都不支持属性值 "inherit"。

可能的值

值	描述
<code>font-style</code>	规定字体样式。参阅: font-style 中可能的值。
<code>font-variant</code>	规定字体异体。参阅: font-variant 中可能的值。
<code>font-weight</code>	规定字体粗细。参阅: font-weight 中可能的值。
<code>font-size/line-height</code>	规定字体尺寸和行高。参阅: font-size 和 line-height 中可能的值。
<code>font-family</code>	规定字体系列。参阅: font-family 中可能的值。
<code>caption</code>	定义被标题控件（比如按钮、下拉列表等）使用的字体。
<code>icon</code>	定义被图标标记使用的字体。
<code>menu</code>	定义被下拉列表使用的字体。
<code>message-box</code>	定义被对话框使用的字体。
<code>small-caption</code>	<code>caption</code> 字体的小型版本。
<code>status-bar</code>	定义被窗口状态栏使用的字体。

CSS font-family 属性

[CSS 参考手册](#)

定义和用法

`font-family` 规定元素的字体系列。

`font-family` 可以把多个字体名称作为一个“回退”系统来保存。如果浏览器不支持第一个字体，则会尝试下一个。也就是说，`font-family` 属性的值是用于某个元素的字体族名称或/及类族名称的一个优先表。浏览器会使用它可识别的第一个值。

有两种类型的字体系列名称：

- 指定的系列名称：具体字体的名称，比如: "times"、"courier"、"arial"。
- 通常字体系列名称：比如: "serif"、"sans-serif"、"cursive"、"fantasy"、"monospace"

提示： 使用逗号分割每个值，并始终提供一个类族名称作为最后的选择。

注意： 使用某种特定的字体系列（`Geneva`）完全取决于用户机器上该字体系列是否可用；这个属性没有指示任何字体下载。因此，强烈推荐使用一个通用字体系列名作为后路。

默认值:	<i>not specified</i>
------	----------------------

继承性:	yes
版本:	CSS1
JavaScript 语法:	<code>object.style.fontFamily="arial,sans-serif"</code>

实例

为段落设置字体：

```
p  
{  
    font-family:"Times New Roman",Georgia,Serif;  
}
```

TIY

浏览器支持

所有主流浏览器都支持 `font-family` 属性。

注释: 任何的版本的 Internet Explorer (包括 IE8) 都不支持属性值 "inherit"。

可能的值

值	描述
<ul style="list-style-type: none"><i>family-name</i><i>generic-family</i>	用于某个元素的字体族名称或/及类族名称的一个优先表。 默认值：取决于浏览器。
<code>inherit</code>	规定应该从父元素继承字体系列。

CSS font-size 属性

CSS 参考手册

定义和用法

`font-size` 属性可设置字体的尺寸。

说明

该属性设置元素的字体大小。注意，实际上它设置的是字体中字符框的高度；实际的字符字形可能比这些框高或矮（通常会矮）。

各关键字对应的字体必须比一个最小关键字相应字体要高，并且要小于下一个最大关键字对应的字体。

默认值:	medium
继承性:	yes
版本:	CSS1
JavaScript 语法:	<code>object.style.fontSize="larger"</code>

实例

设置不同的 HTML 元素的尺寸：

```
h1 {font-size:250%;}  
h2 {font-size:200%;}  
p {font-size:100%}
```

TIY

浏览器支持

所有主流浏览器都支持 `font-size` 属性。

注释: 任何的版本的 Internet Explorer (包括 IE8) 都不支持属性值 "inherit"。

可能的值

值	描述
<code>xx-small</code> <code>x-small</code> <code>small</code> <code>medium</code> <code>large</code> <code>x-large</code> <code>xx-large</code>	把字体的尺寸设置为不同的尺寸，从 <code>xx-small</code> 到 <code>xx-large</code> 。 默认值: <code>medium</code> 。
<code>smaller</code>	把 <code>font-size</code> 设置为比父元素更小的尺寸。
<code>larger</code>	把 <code>font-size</code> 设置为比父元素更大的尺寸。
<code>length</code>	把 <code>font-size</code> 设置为一个固定的值。
<code>%</code>	把 <code>font-size</code> 设置为基于父元素的一个百分比值。
<code>inherit</code>	规定应该从父元素继承字体尺寸。

CSS font-size-adjust 属性

定义和用法

`font-size-adjust` 属性为某个元素规定一个 `aspect` 值，这样就可以保持首选字体的 `x-height`。

说明

字体的小写字母 "x" 的高度与 "font-size" 高度之间的比率被称为一个字体的 `aspect` 值。当字体拥有高的 `aspect` 值时，那么当此字体被设置为很小的尺寸时会更易阅读。举例：`Verdana` 的 `aspect` 值是 0.58(意味着当字体尺寸为 100px 时，它的 `x-height` 是 58px)。`Times New Roman` 的 `aspect` 值是 0.46。这就意味着 `Verdana` 在小尺寸时比 `Times New Roman` 更易阅读。

默认值:	none
继承性:	yes
版本:	CSS2
JavaScript 语法:	<code>object.style.fontSizeAdjust="0.70"</code>

实例

设置不同的 HTML 元素的 `font-size-adjust` 属性：

```
h1
{
    font-size-adjust:0.58;
}

p
{
    font-size-adjust:0.60;
}
```

[TIY](#)

浏览器支持

Internet Explorer 不支持 `font-size-adjust` 属性。

可能的值

值	描述
none	默认。如果此字体不可用，则不保持此字体的 x-height。
number	<p>定义字体的 aspect 值比率。</p> <p>可使用的公式:</p> <p>首选字体的字体尺寸 * (font-size-adjust 值 / 可用字体的 aspect 值) = 可应用到可用字体的字体尺寸</p> <p>举例:</p> <p>如果 14px 的 Verdana (aspect 值是 0.58) 不可用，但是某个可用的字体的 aspect 值是 0.46，那么替代字体的尺寸将是 $14 * (0.58/0.46) = 17.65\text{px}$。</p>

CSS font-stretch 属性

定义和用法

font-stretch 属性可对当前的 font-family 进行伸缩变形。

默认值:	normal
继承性:	yes
版本:	CSS2
JavaScript 语法:	<code>object.style.fontStretch="ultra-expanded"</code>

实例

设置 HTML 元素的 font-stretch 属性:

```
h1
{
    font-stretch:ultra-condensed;
}
```

浏览器支持

所有主流浏览器都不支持 font-stretch 属性。

可能的值

值	描述
normal	默认值。把缩放比例设置为标准。
wider	把伸展比例设置为更进一步的伸展值
narrower	把收缩比例设置为更进一步的收缩值
ultra-condensed extra-condensed condensed semi-condensed semi-expanded expanded extra-expanded ultra-expanded	设置 font-family 的缩放比例。 "ultra-condensed" 是最宽的值，而 "ultra-expanded" 是最窄的值。

CSS font-style 属性

[CSS 参考手册](#)

定义和用法

font-style 属性定义字体的风格。

说明

该属性设置使用斜体、倾斜或正常字体。斜体字体通常定义为字体系列中的一个单独的字体。理论上讲，用户代理可以根据正常字体计算一个斜体字体。

默认值:	normal
继承性:	yes
版本:	CSS1
JavaScript 语法:	<code>object.style.fontStyle="italic"</code>

实例

为三个段落设置不同的字体样式：

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

TIY

浏览器支持

所有主流浏览器都支持 `font-style` 属性。

注释: 任何的版本的 Internet Explorer (包括 IE8) 都不支持属性值 "inherit"。

可能的值

值	描述
<code>normal</code>	默认值。浏览器显示一个标准的字体样式。
<code>italic</code>	浏览器会显示一个斜体的字体样式。
<code>oblique</code>	浏览器会显示一个倾斜的字体样式。
<code>inherit</code>	规定应该从父元素继承字体样式。

CSS font-variant 属性

CSS 参考手册

定义和用法

`font-variant` 属性设置小型大写字母的字体显示文本，这意味着所有的小写字母均会被转换为大写，但是所有使用小型大写字体的字母与其余文本相比，其字体尺寸更小。

说明

该属性主要用于定义小型大写字母文本。理论上，用户代理可以根据正常字体计算出小型大写字母字体。

默认值:	<code>normal</code>
继承性:	<code>yes</code>
版本:	CSS1
JavaScript 语法:	<code>object.style.fontVariant="small-caps"</code>

实例

把段落设置为小型大写字母字体：

```
p.small
```

```
{
```

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

TIY

浏览器支持

所有主流浏览器都支持 `font-variant` 属性。

注释: 任何的版本的 Internet Explorer (包括 IE8) 都不支持属性值 "inherit"。

可能的值

值	描述
normal	默认值。浏览器会显示一个标准的字体。
small-caps	浏览器会显示小型大写字母的字体。
inherit	规定应该从父元素继承 <code>font-variant</code> 属性的值。

CSS `font-weight` 属性

CSS 参考手册

定义和用法

`font-weight` 属性设置文本的粗细。

说明

该属性用于设置显示元素的文本中所用的字体加粗。数字值 400 相当于关键字 `normal`, 700 等价于 `bold`。每个数字值对应的字体加粗必须至少与下一个最小数字一样细，而且至少与下一个最大数字一样粗。

默认值:	normal
继承性:	yes
版本:	CSS1
JavaScript 语法:	<code>object.style.fontWeight="900"</code>

实例

设置三个段落的字体的粗细:

```
p.normal {font-weight:normal;}  
p.thick {font-weight:bold;}  
p.thicker {font-weight:900;}
```

TIY

浏览器支持

所有主流浏览器都支持 `font-weight` 属性。

注释: 任何的版本的 Internet Explorer (包括 IE8) 都不支持属性值 "inherit"。

可能的值

值	描述
normal	默认值。定义标准的字符。
bold	定义粗体字符。
bolder	定义更粗的字符。
lighter	定义更细的字符。
100 200 300 400 500 600 700 800 900	定义由粗到细的字符。400 等同于 <code>normal</code> , 而 700 等同于 <code>bold</code> 。

CSS 边框属性

"CSS" 列中的数字指示哪个 CSS 版本定义了该属性。

属性	描述	CSS
outline	在一个声明中设置所有的轮廓属性。	2
outline-color	设置轮廓的颜色。	2

outline-style	设置轮廓的样式。	2
outline-width	设置轮廓的宽度。	2

CSS outline 属性

[CSS 参考手册](#)

定义和用法

outline（轮廓）是绘制于元素周围的一条线，位于边框边缘的外围，可起到突出元素的作用。

注释：轮廓线不会占据空间，也不一定是矩形。

outline 简写属性在一个声明中设置所有的轮廓属性。

可以按顺序设置如下属性：

- outline-color
- outline-style
- outline-width

如果不设置其中的某个值，也不会出问题，比如 `outline:solid #ff0000;` 也是允许的。

默认值:	invert none medium
继承性:	no
版本:	CSS2
JavaScript 语法:	<code>object.style.outline="#0000FF dotted thin"</code>

实例

设置 4 个边框的样式：

```
p  
{  
  outline:#00FF00 dotted thick;  
}
```

TIY

浏览器支持

除了 Internet Explorer，所有浏览器都支持 `outline` 属性。

可能的值

值	描述
<code>outline-color</code>	规定边框的颜色。参阅： outline-color 中可能的值。
<code>outline-style</code>	规定边框的样式。参阅： outline-style 中可能的值。
<code>outline-width</code>	规定边框的宽度。参阅： outline-width 中可能的值。
<code>inherit</code>	规定应该从父元素继承 <code>outline</code> 属性的设置。

CSS outline-color 属性

CSS 参考手册

定义和用法

`outline`（轮廓）是绘制于元素周围的一条线，位于边框边缘的外围，可起到突出元素的作用。`outline` 属性可设置元素周围的轮廓线。

注释：请始终在 `outline-color` 属性之前声明 `outline-style` 属性。元素只有获得轮廓以后才能改变其轮廓的颜色。

注释：轮廓线不会占据空间，也不一定是矩形。

`outline-color` 属性设置一个元素整个轮廓中可见部分的颜色。要记住，轮廓的样式不能是 `none`，否则轮廓不会出现。

默认值:	invert
继承性:	no
版本:	CSS2
JavaScript 语法:	<code>object.style.outlineColor="#0000FF"</code>

实例

设置点状轮廓的颜色：

```
p
```

```
{  
outline-style:dotted;  
outline-color:#00ff00;  
}
```

TIY

浏览器支持

所有主流浏览器都支持 `outline-color` 属性。

只有在规定了 `!DOCTYPE` 时，Internet Explorer 才支持 `outline-color` 属性。

可能的值

值	描述
<code>color_name</code>	规定颜色值为颜色名称的轮廓颜色（比如 <code>red</code> ）。
<code>hex_number</code>	规定颜色值为十六进制值的轮廓颜色（比如 <code>#ff0000</code> ）。
<code>rgb_number</code>	规定颜色值为 <code>rgb</code> 代码的轮廓颜色（比如 <code>rgb(255,0,0)</code> ）。
<code>invert</code>	默认。执行颜色反转（逆向的颜色）。可使轮廓在不同的背景颜色中都是可见。
<code>inherit</code>	规定应该从父元素继承轮廓颜色的设置。

CSS `outline-style` 属性

CSS 参考手册

定义和用法

`outline-style` 属性用于设置元素的整个轮廓的样式。样式不能是 `none`，否则轮廓不会出现。

`outline`（轮廓）是绘制于元素周围的一条线，位于边框边缘的外围，可起到突出元素的作用。`outline` 属性设置元素周围的轮廓线。

注释：请始终在 `outline-color` 属性之前声明 `outline-style` 属性。元素只有获得轮廓以后才能改变其轮廓的颜色。

注释：轮廓线不会占据空间，也不一定是矩形。

默认值:	none
继承性:	no
版本:	CSS2
JavaScript 语法:	<code>object.style.outlineStyle="dotted"</code>

实例

设置轮廓的样式：

```
p  
{  
    outline-style:dotted;  
}
```

TIY

浏览器支持

所有主流浏览器都支持 `outline-style` 属性。

只有在规定了 `!DOCTYPE` 时，Internet Explorer 才支持 `outline-style` 属性。

可能的值

值	描述
<code>none</code>	默认。定义无轮廓。
<code>dotted</code>	定义点状的轮廓。
<code>dashed</code>	定义虚线轮廓。
<code>solid</code>	定义实线轮廓。
<code>double</code>	定义双线轮廓。双线的宽度等同于 <code>outline-width</code> 的值。
<code>groove</code>	定义 3D 凹槽轮廓。此效果取决于 <code>outline-color</code> 值。
<code>ridge</code>	定义 3D 凸槽轮廓。此效果取决于 <code>outline-color</code> 值。
<code>inset</code>	定义 3D 凹边轮廓。此效果取决于 <code>outline-color</code> 值。
<code>outset</code>	定义 3D 凸边轮廓。此效果取决于 <code>outline-color</code> 值。
<code>inherit</code>	规定应该从父元素继承轮廓样式的设置。

CSS outline-width 属性

[CSS 参考手册](#)

定义和用法

`outline-width` 属性设置元素整个轮廓的宽度，只有当轮廓样式不是 `none` 时，这个宽度才会起作用。如果样式为 `none`，宽度实际上会重置为 0。不允许设置负长度值。

`outline`（轮廓）是绘制于元素周围的一条线，位于边框边缘的外围，可起到突出元素的作用。`outline` 属性设置元素周围的轮廓线。

注释：请始终在 `outline-width` 属性之前声明 `outline-style` 属性。元素只有获得轮廓以后才能改变其轮廓的颜色。

注释：轮廓线不会占据空间，也不一定是矩形。

默认值:	medium
继承性:	no
版本:	CSS2
JavaScript 语法:	<code>object.style.outlineWidth="thin"</code>

实例

设置点状轮廓的颜色：

```
p  
{  
    outline-style:dotted;  
    outline-width:5px;  
}
```

[TIP](#)

浏览器支持

所有主流浏览器都支持 `outline-width` 属性。

只有在规定了 !DOCTYPE 时，Internet Explorer 才支持 `outline-width` 属性。

可能的值

值	描述
<code>thin</code>	规定细轮廓。
<code>medium</code>	默认。规定中等的轮廓。
<code>thick</code>	规定粗的轮廓。
<code>length</code>	允许您规定轮廓粗细的值。
<code>inherit</code>	规定应该从父元素继承轮廓宽度的设置。