# CSS创建

## 外部样式表

当样式需要应用于很多页面时，使用<link> 标签在（文档的）头部

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

## 内部样式表

当单个文档需要特殊的样式时，使用<style标签>在（文档的）头部

<head>

<style>

hr {color:sienna;}

p {margin-left:20px;}

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

</style>

</head>

## 内联样式表

当样式仅需要在一个元素上应用一次时，使用style属性：

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

## 多重样式

一般情况下，优先级如下：

(内联样式）Inline style > （内部样式）Internal style sheet >（外部样式）External style sheet > 浏览器默认样式，并且不重复的属性继承。

# CSS注释

/\*…\*/

# CSS选择器

## 元素选择器

h3

{

text-align:center;

}

## id选择器

HTML元素以id属性来设置id选择器（ ID属性不要以数字开头，有些浏览器不起作用）：

<p id="para1">Hello World!</p>

CSS 中 id 选择器以 "#" 来定义：

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

## 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（同样类名的第一个字符不能使用数字，有些浏览器不起作用）：

<h1 class="center">标题居中</h1>

<p class="center">段落居中。</p>

在 CSS 中，类选择器以一个点"."号显示：

所有拥有 center 类的 HTML 元素均为居中

.center

{

text-align:center;

}

所有的 p 元素使用 class="center" 让该元素的文本居中

p.center

{

text-align:center;

}

为所有 class="center" 元素内部的 p 元素指定一个样式

.center p

{

text-align:center;

}

## 属性选择器

* 下面的例子是把包含标题（title）的所有元素变为蓝色：

[title]

{

color:blue;

}

* 下面的实例改变了标题title='runoob'元素的边框样式:

[title=runoob]

{

border:5px solid green;

}

## 分组选择器

为了尽量减少代码，你可以使用分组选择器，每个选择器用逗号分隔。

h1,h2,p

{

color:green;

}

# CSS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:

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"

body {background-color:#b0c4de;}

## 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。

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

## 背景图像-水平或垂直平铺

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

只在水平/垂直方向平铺 (repeat-x/repeat-y)

body

{

background-image:url('gradient2.png');

background-repeat:repeat-x;

}

## 背景图像-设置定位与不平铺

Body

{

background-image:url('img\_tree.png');

background-repeat:no-repeat;

background-position:right top;

}

## 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":当使用简写属性时，属性值的顺序为：

1. background-color
2. background-image
3. background-repeat
4. background-attachment
5. background-position

# CSS文本

## 文本颜色

body {color:red;}

h1 {color:#00ff00;}

h2 {color:rgb(255,0,0);}

## 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。

h1 {text-align:center;}

p.date {text-align:right;}

p.main {text-align:justify;}

## 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：

/\*删除下划线\*/

a {text-decoration:none;}

/\*上方加横线\*/

h1 {text-decoration:overline;}

/\*中间加横线\*/

h2 {text-decoration:line-through;}

/\*加下划线\*/

h3 {text-decoration:underline;}

## 文本大小写转换

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。

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。

/\*全大写\*/

p.uppercase {text-transform:uppercase;}

/\*全小写\*/

p.lowercase {text-transform:lowercase;}

/\*首字母大写\*/

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。

p {text-indent:50px;}

# CSS字体

## CSS字体

设置字体有以下规则：

* 使用中文字体，或带空格的英文字体名，需要用“引号”将该字体名扩起来，单个单词的字体名称则无需加上引号，例：

p{font-family:"宋体";}

* “font-family”可以有多个值，多个值设置的作用是，当浏览用户本地计算机没有找到设置的第一个值所表示的字体时，会去找第二个，以此类推，若找到最后一个值都没有找到匹配的字体，浏览器则会从通用的字体类型中找到与你相似的，例：

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

* 考虑到网页最终显示效果在不同设备上的一致性，我们通常使用的中文字体只有：“宋体”、“微软雅黑”、“黑体”、“楷体”、“幼圆”等，而目前以“微软雅黑”为最常用设计字体。
* 常用的英文字体有：“Arial”、“Helvetica”、“Tahoma”、“Verdana”、“Lucida Grande”、“Times New Roman”、“Georgia”等。
* 常用的组合方式：

<https://www.runoob.com/cssref/css-websafe-fonts.html>

## 字体样式

font-style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{font-style:normal;}

p{font-style:italic;}

p{font-style:oblique;}

## 字体大小

### 用px来设置字体大小

正确配合HTML标签使用，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

h1 {font-size:40px;}

h2 {font-size:30px;}

p {font-size:14px;}

#### 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

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）

### 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。

1em = 16px，例：

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 \*/

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

推荐使用这种方式：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;}

这样在所有浏览器中，可以显示相同的文本大小，并允许所有浏览器缩放文本的大小。

## 设置字体加粗

HTML<b>标签：<b>加粗文本</b>

font-weight属性：

p.normal {font-weight:normal;}

p.light {font-weight:lighter;}

p.thick {font-weight:bold;}

p.thicker {font-weight:900;}

## 所有CSS字体属性

| 属性 | 描述 | 实例 |
| --- | --- | --- |
| [font](https://www.runoob.com/cssref/pr-font-font.html) | 在一个声明中设置所有的字体属性 | p.ex1{font:15px arial,sans-serif;}  p.ex2{font:italic bold 12px/30px Georgia,serif;} |
| [font-family](https://www.runoob.com/cssref/pr-font-font-family.html) | 指定文本的字体系列 | p{font-family:"Times New Roman", Times, serif;} |
| [font-size](https://www.runoob.com/cssref/pr-font-font-size.html) | 指定文本的字体大小 | body {font-size:100%;}  h1 {font-size:2.5em;}  h2 {font-size:1.875em;}  p {font-size:0.875em;} |
| [font-style](https://www.runoob.com/cssref/pr-font-font-style.html) | 指定文本的字体样式 | p{font-style:normal;}  p{font-style:italic;}  p{font-style:oblique;} |
| [font-variant](https://www.runoob.com/cssref/pr-font-font-variant.html) | 以小型大写字体或者正常字体显示文本 | p{font-variant:normal;}  p{font-variant:small-caps;} |
| [font-weight](https://www.runoob.com/cssref/pr-font-weight.html) | 指定字体的粗细 | p{font-weight:normal;}  p{font-weight:lighter;}  p{font-weight:bold;}  p{font-weight:900;} |

# CSS链接

## 链接样式

默认情况下，HTML链接<a>将以以下形式出现在浏览器中：

* 一个未访问过的链接显示为蓝色字体并带有下划线。
* 访问过的链接显示为紫色并带有下划线。
* 点击链接时，链接显示为红色并带有下划线。

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

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

例：

a:link {color:#000000;} /\* 未访问链接\*/

a:visited {color:#00FF00;} /\* 已访问链接 \*/

a:hover {color:#FF00FF;} /\* 鼠标移动到链接上 \*/

a:active {color:#0000FF;} /\* 鼠标点击时 \*/

## 文本修饰

text-decoration 属性主要用于删除链接中的下划线，参见5.3：

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,参见4.1：

a:link {background-color:#B2FF99;}

a:visited {background-color:#FFFF85;}

a:hover {background-color:#FF704D;}

a:active {background-color:#FF704D;}

## 合并设置

a:link { color:#FF00FF; text-decoration:none; background-color:#B2FF99;}

# CSS列表

在HTML中，有两种类型的列表，相关列表标签无序列表<ul>、有序列表<ol>、列表项目<li>：

* 无序列表<ul> - 列表项标记用特殊图形（如小黑点、小方框等）
* 有序列表<ol> - 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，CSS列表属性作用如下：

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

## 不同的列表项标记

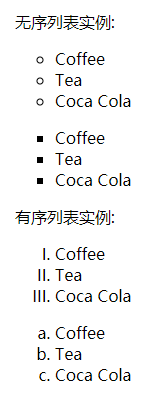
list-style-type属性指定列表项标记的类型是：

ul.a {list-style-type: circle;} /\*项标记为圆\*/

ul.b {list-style-type: square;} /\*项标记为方块\*/

ol.c {list-style-type: upper-roman;} /\*项标记为罗马数字\*/

ol.d {list-style-type: lower-alpha;} /\*项标记为阿拉伯字母\*/



所有的列表项标记样式：

<https://www.runoob.com/try/try.php?filename=trycss_list-style-type_all>

## 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'); }

## 浏览器兼容性解决方案

上面的例子在所有浏览器中显示并不相同，IE和Opera显示图像标记比火狐，Chrome和Safari更高一点点。

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，过程如下：

ul

{

list-style-type: none;

padding: 0px;

margin: 0px;

}

/\*选择器没有’,’说明<li>包含于<ul>\*/

ul li

{

background-image: url(sqpurple.gif);

background-repeat: no-repeat;

background-position: 0px 5px;

padding-left: 14px;

}

解释：

ul:

* 设置列表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（HTML列表项）:

* 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-简写属性

在单个属性[list-style](https://www.runoob.com/cssref/pr-list-style.html)中可以指定所有的列表属性。这就是所谓的简写属性：

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

顺序规则：

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

## 所有的CSS列表属性

| 属性 | 描述 | 详情 |
| --- | --- | --- |
| [list-style](https://www.runoob.com/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 | <https://www.runoob.com/cssref/pr-list-style.html> |
| [list-style-image](https://www.runoob.com/cssref/pr-list-style-image.html) | 将图象设置为列表项标志。 | <https://www.runoob.com/cssref/pr-list-style-image.html> |
| [list-style-position](https://www.runoob.com/cssref/pr-list-style-position.html) | 设置列表中列表项标志的位置。 | <https://www.runoob.com/cssref/pr-list-style-position.html> |
| [list-style-type](https://www.runoob.com/cssref/pr-list-style-type.html) | 设置列表项标志的类型。 | <https://www.runoob.com/cssref/pr-list-style-type.html> |

# CSS表格

HTML表格相关标签<table>表、<tr>行、<td>单元格、<th>表头包含于行<tr>、<caption>标题包含于<table>表，使用 CSS 可以使 HTML 表格更美观。

## 表格边框

指定CSS表格边框，使用border属性。

下面的例子指定了一个表格的Th和TD元素的黑色边框：

/\*注意选择器三个标签有份号，表示同时设置这三个标签的border

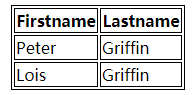
表格有双边框。这是因为表和th/ td元素有独立的边界。\*/

table, th, td

{

border: 1px solid black;

}



## 折叠边框(border-collapse)

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;

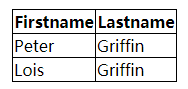
}

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，向左，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（padding）

单元格空白填充，应使用td和th元素的填充属性，使用table会很难看：

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;

}

## [设置表格标题的位置](https://www.runoob.com/try/try.php?filename=trycss_table_caption-side)

/\*标题在底部\*/

caption {caption-side:bottom;}

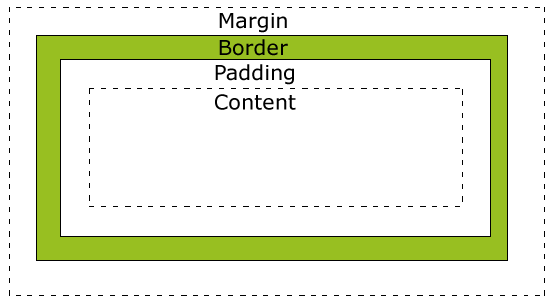
# 盒子模型（Box Model）

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) - 清除边框外的区域，外边距是透明的。
* Border(边框) - 围绕在内边距和内容外的边框。
* Padding(内边距) - 清除内容周围的区域，内边距是透明的。
* Content(内容) - 盒子的内容，显示文本和图像。

## 元素的宽度和高度

当指定一个CSS元素的宽度(width)和高度(height)属性时，你只是设置内容区域(Content)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。

* 最终元素的总宽度计算公式是这样的：

总元素的宽度=宽度(width)+左填充(Padding)+右填充(Padding)+左边框(Border)+右边框(Border)+左边距(Margin)+右边距(Margin)

* 元素的总高度最终计算公式是这样的：

总元素的高度=高度(width)+顶部填充(Padding)+底部填充(Padding)+上边框左边框(Border)+下边框左边框(Border)+上边距(Margin)+下边距(Margin)

## 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版本不支持设置填充的宽度和边框的宽度属性。

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。

# CSS边框（Border）

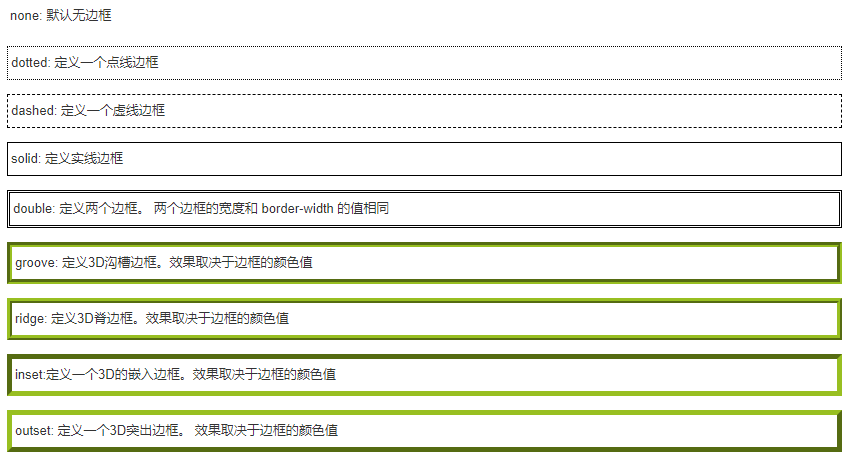
## 边框属性

| 属性 | 描述 |
| --- | --- |
| [border](https://www.runoob.com/cssref/pr-border.html) | 简写属性，用于把针对四个边的属性设置在一个声明。 |
| [border-style](https://www.runoob.com/cssref/pr-border-style.html) | 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。 |
| [border-width](https://www.runoob.com/cssref/pr-border-width.html) | 简写属性，用于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。 |
| [border-color](https://www.runoob.com/cssref/pr-border-color.html) | 简写属性，设置元素的所有边框中可见部分的颜色，或为 4 个边分别设置颜色。 |
| [border-bottom](https://www.runoob.com/cssref/pr-border-bottom.html) | 简写属性，用于把下边框的所有属性设置到一个声明中。 |
| [border-bottom-color](https://www.runoob.com/cssref/pr-border-bottom-color.html) | 设置元素的下边框的颜色。 |
| [border-bottom-style](https://www.runoob.com/cssref/pr-border-bottom-style.html) | 设置元素的下边框的样式。 |
| [border-bottom-width](https://www.runoob.com/cssref/pr-border-bottom-width.html) | 设置元素的下边框的宽度。 |
| [border-left](https://www.runoob.com/cssref/pr-border-left.html) | 简写属性，用于把左边框的所有属性设置到一个声明中。 |
| [border-left-color](https://www.runoob.com/cssref/pr-border-left-color.html) | 设置元素的左边框的颜色。 |
| [border-left-style](https://www.runoob.com/cssref/pr-border-left-style.html) | 设置元素的左边框的样式。 |
| [border-left-width](https://www.runoob.com/cssref/pr-border-left-width.html) | 设置元素的左边框的宽度。 |
| [border-right](https://www.runoob.com/cssref/pr-border-right.html) | 简写属性，用于把右边框的所有属性设置到一个声明中。 |
| [border-right-color](https://www.runoob.com/cssref/pr-border-right-color.html) | 设置元素的右边框的颜色。 |
| [border-right-style](https://www.runoob.com/cssref/pr-border-right-style.html) | 设置元素的右边框的样式。 |
| [border-right-width](https://www.runoob.com/cssref/pr-border-right-width.html) | 设置元素的右边框的宽度。 |
| [border-top](https://www.runoob.com/cssref/pr-border-top.html) | 简写属性，用于把上边框的所有属性设置到一个声明中。 |
| [border-top-color](https://www.runoob.com/cssref/pr-border-top-color.html) | 设置元素的上边框的颜色。 |
| [border-top-style](https://www.runoob.com/cssref/pr-border-top-style.html) | 设置元素的上边框的样式。 |
| [border-top-width](https://www.runoob.com/cssref/pr-border-top-width.html) | 设置元素的上边框的宽度。 |

## 边框样式（border-style）

边框样式属性指定要显示什么样的边界。

border-style 值:



## 边框宽度（border-width）

您可以通过 border-width 属性为边框指定宽度。

为边框指定宽度有两种方法：可以指定长度值，比如 2px 或 0.1em(单位为 px, pt, cm, em 等)，或者使用 3 个关键字之一，它们分别是 thick（厚） 、medium（默认值） 和 thin（薄）。

注意：CSS 没有定义 3 个关键字的具体宽度，所以一个用户可能把 thick 、medium 和 thin 分别设置为等于 5px、3px 和 2px，而另一个用户则分别设置为 3px、2px 和 1px。

## 边框颜色（border-color）

border-color属性用于设置边框的颜色。可以设置的颜色：

name - 指定颜色的名称，如 "red"

RGB - 指定 RGB 值, 如 "rgb(255,0,0)"

Hex - 指定16进制值, 如 "#ff0000"

您还可以设置边框的颜色为"transparent（透明）"。

注意： border-color单独使用是不起作用的，必须得先使用border-style来设置边框样式。

## 边框-单独设置各边某属性

### 单独设置各个边框

p

{

border-top-style:dotted;

border-right-style:solid;

border-bottom-style:dotted;

border-left-style:solid;

}

### 属性值数量自动控制

border-style属性可以有1-4个值：

* border-style:dotted solid double dashed;

上边框是 dotted

右边框是 solid

底边框是 double

左边框是 dashed

* border-style:dotted solid double;

上边框是 dotted

左、右边框是 solid

底边框是 double

* border-style:dotted solid;

上、底边框是 dotted

右、左边框是 solid

* border-style:dotted;

四面边框是 dotted

### 宽度（border-width）/颜色（border-color）

宽度和颜色与样式同理，例：

p

{

border-style:dotted solid;

border-color:red blue;

}

## 边框-简写属性

你可以在"border"属性中设置：

顺序无限定

border-width

border-style (required)

border-color

例：

//全边框

border:5px solid red;

//单边框

border-top:5px solid red;

# CSS尺寸(Dimension)

## 使用百分比

使用百分比计算的是相对于父元素的百分比。

## 元素最大/最小尺寸

当元素设置了最大尺寸，那么其他针对此元素的样式设置只设置最大尺寸的范围。

* 50px：

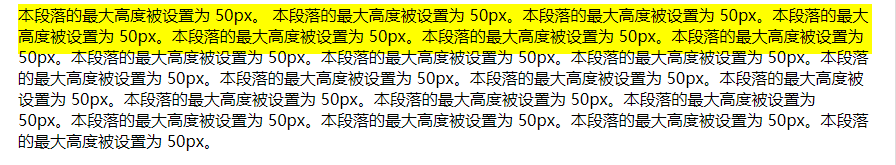
p

{

max-height:50px;

background-color:yellow;

}



* 100 px：

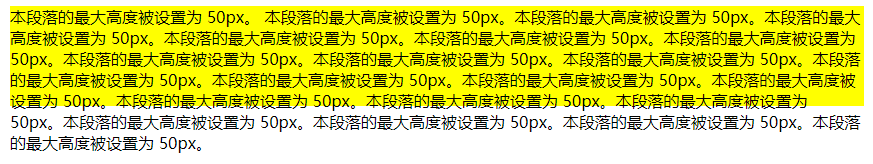
p

{

max-height:100px;

background-color:yellow;

}



* 100000px：

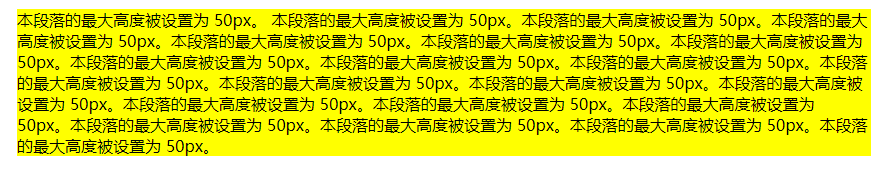
p

{

max-height:100000px;

background-color:yellow;

}



最大宽度的自适应能力比最大高度的强，因为可以加滚动条。

p

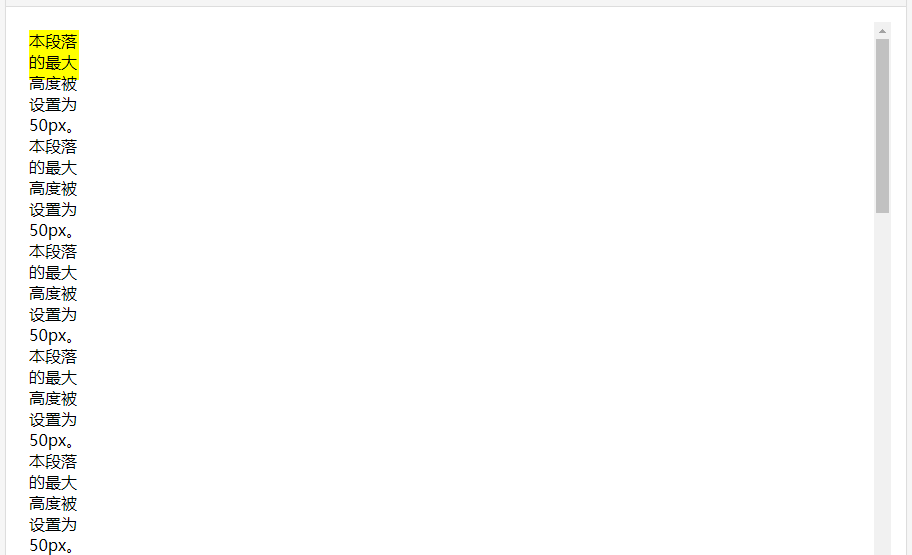
{

max-height:50px;

max-width: 50px;

background-color:yellow;

}



# CSS显示(Display)

## 块与内联元素特性与显示

* 块级元素(block)特性：

总是独占一行，表现为另起一行开始，而且其后的元素也必须另起一行显示;

宽度(width)、高度(height)、内边距(padding)和外边距(margin)都可控制;

* 内联元素(inline)特性：

和相邻的内联元素在同一行;

宽度(width)、高度(height)、内边距的top/bottom(padding-top/padding-bottom)和外边距的top/bottom(margin-top/margin-bottom)都不可改变，就是里面文字或图片的大小;

* 块级元素主要有：

address , blockquote , center , dir , div , dl , fieldset , form , h1 , h2 , h3 , h4 , h5 , h6 , hr , isindex , menu , noframes , noscript , ol , p , pre , table , ul , li

* 内联元素主要有

a , abbr , acronym , b , bdo , big , br , cite , code , dfn , em , font , i , img , input , kbd , label , q , s , samp , select , small , span , strike , strong , sub , sup ,textarea , tt , u , var

* 可变元素(根据上下文关系确定该元素是块元素还是内联元素)：

applet ,button ,del ,iframe , ins ,map ,object , script

* CSS中块级、内联元素的应用：

利用CSS我们可以摆脱上面表格里HTML标签归类的限制，自由地在不同标签/元素上应用我们需要的属性。

主要用的CSS样式有以下三个：

1. display:block -- 显示为块级元素
2. display:inline -- 显示为内联元素
3. display:inline-block -- 显示为内联块元素，表现为同行显示并可修改宽高内外边距等属性

我们常将<ul>元素加上display:inline-block样式，原本垂直的列表就可以水平显示了。

## visibility 属性

visibility: hidden 相当于 display:none，能把元素隐藏起来，但两者的区别在于：

* display:none 元素不再占用空间。
* visibility: hidden 使元素在网页上不可见，但仍占用空间。
* 然而，visibility 还可能取值为 collapse 。

当设置元素 visibility: collapse 后，一般的元素的表现与 visibility: hidden 一样，也即其会占用空间。但如果该元素是与 table 相关的元素，例如 table row、table column、table column group、table column group 等，其表现却跟 display: none 一样，也即其占用的空间会释放。

在不同浏览器下，对 visibility: collapse 的处理方式不同：

* visibility: collapse 的上述特性仅在 Firefox 下起作用。
* 在 IE 即使设置了 visibility: collapse，还是会显示元素。
* 在 Chrome 下，即使会将元素隐藏，但无论是否是与 table 相关的元素，visibility: collapse 与 visibility: hidden 没有什么区别，即仍会占用空间。