CSS 高级语法

* [CSS 基础语法](http://www.w3school.com.cn/css/css_syntax.asp)
* [CSS 派生选择器](http://www.w3school.com.cn/css/css_syntax_descendant_selector.asp)

**选择器的分组**

你可以对选择器进行分组，这样，被分组的选择器就可以分享相同的声明。用逗号将需要分组的选择器分开。在下面的例子中，我们对所有的标题元素进行了分组。所有的标题元素都是绿色的。

h1,h2,h3,h4,h5,h6 {

color: green;

}

**继承及其问题**

根据 CSS，子元素从父元素继承属性。但是它并不总是按此方式工作。看看下面这条规则：

body {

font-family: Verdana, sans-serif;

}

根据上面这条规则，站点的 body 元素将使用 Verdana 字体（假如访问者的系统中存在该字体的话）。

通过 CSS 继承，子元素将继承最高级元素（在本例中是 body）所拥有的属性（这些子元素诸如 p, td, ul, ol, ul, li, dl, dt,和 dd）。不需要另外的规则，所有 body 的子元素都应该显示 Verdana 字体，子元素的子元素也一样。并且在大部分的现代浏览器中，也确实是这样的。

但是在那个浏览器大战的血腥年代里，这种情况就未必会发生，那时候对标准的支持并不是企业的优先选择。比方说，Netscape 4 就不支持继承，它不仅忽略继承，而且也忽略应用于 body 元素的规则。IE/Windows 直到 IE6 还存在相关的问题，在表格内的字体样式会被忽略。我们又该如何是好呢？

**友善地对待Netscape 4**

幸运地是，你可以通过使用我们称为 "Be Kind to Netscape 4" 的冗余法则来处理旧式浏览器无法理解继承的问题。

body {

font-family: Verdana, sans-serif;

}

p, td, ul, ol, li, dl, dt, dd {

font-family: Verdana, sans-serif;

}

4.0 浏览器无法理解继承，不过他们可以理解组选择器。这么做虽然会浪费一些用户的带宽，但是如果需要对 Netscape 4 用户进行支持，就不得不这么做。

**继承是一个诅咒吗？**

如果你不希望 "Verdana, sans-serif" 字体被所有的子元素继承，又该怎么做呢？比方说，你希望段落的字体是 Times。没问题。创建一个针对 p 的特殊规则，这样它就会摆脱父元素的规则：

body {

font-family: Verdana, sans-serif;

}

td, ul, ol, ul, li, dl, dt, dd {

font-family: Verdana, sans-serif;

}

p {

font-family: Times, "Times New Roman", serif;

}

CSS 派生选择器

* [CSS 高级语法](http://www.w3school.com.cn/css/css_syntax_pro.asp)
* [CSS id 选择器](http://www.w3school.com.cn/css/css_syntax_id_selector.asp)

**派生选择器**

**通过依据元素在其位置的上下文关系来定义样式，你可以使标记更加简洁。**

在 CSS1 中，通过这种方式来应用规则的选择器被称为上下文选择器 (contextual selectors)，这是由于它们依赖于上下文关系来应用或者避免某项规则。在 CSS2 中，它们称为派生选择器，但是无论你如何称呼它们，它们的作用都是相同的。

派生选择器允许你根据文档的上下文关系来确定某个标签的样式。通过合理地使用派生选择器，我们可以使 HTML 代码变得更加整洁。

比方说，你希望列表中的 strong 元素变为斜体字，而不是通常的粗体字，可以这样定义一个派生选择器：

li strong {

font-style: italic;

font-weight: normal;

}

请注意标记为 <strong> 的蓝色代码的上下文关系：

<p><strong>我是粗体字，不是斜体字，因为我不在列表当中，所以这个规则对我不起作用</strong></p>

<ol>

<li><strong>我是斜体字。这是因为 strong 元素位于 li 元素内。</strong></li>

<li>我是正常的字体。</li>

</ol>

在上面的例子中，只有 li 元素中的 strong 元素的样式为斜体字，无需为 strong 元素定义特别的 class 或 id，代码更加简洁。

再看看下面的 CSS 规则：

strong {

color: red;

}

h2 {

color: red;

}

h2 strong {

color: blue;

}

下面是它施加影响的 HTML：

<p>The strongly emphasized word in this paragraph is<strong>red</strong>.</p>

<h2>This subhead is also red.</h2>

<h2>The strongly emphasized word in this subhead is<strong>blue</strong>.</h2>

**相关内容**

如果您需要更深入地学习关于派生选择器的知识，请阅读 W3School 的高级教程中的以下内容：

* [CSS 后代选择器](http://www.w3school.com.cn/css/css_selector_descendant.asp)
* [CSS 子元素选择器](http://www.w3school.com.cn/css/css_selector_child.asp)
* [CSS 相邻兄弟选择器](http://www.w3school.com.cn/css/css_selector_adjacent_sibling.asp)

CSS id 选择器

* [CSS 派生选择器](http://www.w3school.com.cn/css/css_syntax_descendant_selector.asp)
* [CSS 类选择器](http://www.w3school.com.cn/css/css_syntax_class_selector.asp)

**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。**

**id 选择器以 "#" 来定义。**

下面的两个 id 选择器，第一个可以定义元素的颜色为红色，第二个定义元素的颜色为绿色：

#red {color:red;}

#green {color:green;}

下面的 HTML 代码中，id 属性为 red 的 p 元素显示为红色，而 id 属性为 green 的 p 元素显示为绿色。

<p id="red">这个段落是红色。</p>

<p id="green">这个段落是绿色。</p>

**注意：**id 属性只能在每个 HTML 文档中出现一次。想知道原因吗，请参阅 [XHTML:网站重构](http://www.w3school.com.cn/xhtml/xhtml_structural_01.asp)。

**id 选择器和派生选择器**

**在现代布局中，id 选择器常常用于建立派生选择器。**

#sidebar p {

font-style: italic;

text-align: right;

margin-top: 0.5em;

}

上面的样式只会应用于出现在 id 是 sidebar 的元素内的段落。这个元素很可能是 div 或者是表格单元，尽管它也可能是一个表格或者其他块级元素。它甚至可以是一个内联元素，比如 <em></em> 或者 <span></span>，不过这样的用法是非法的，因为不可以在内联元素 <span> 中嵌入 <p> （如果你忘记了原因，请参阅 [XHTML:网站重构](http://www.w3school.com.cn/xhtml/xhtml_structural_01.asp)）。

**一个选择器，多种用法**

**即使被标注为 sidebar 的元素只能在文档中出现一次，这个 id 选择器作为派生选择器也可以被使用很多次：**

#sidebar p {

font-style: italic;

text-align: right;

margin-top: 0.5em;

}

#sidebar h2 {

font-size: 1em;

font-weight: normal;

font-style: italic;

margin: 0;

line-height: 1.5;

text-align: right;

}

在这里，与页面中的其他 p 元素明显不同的是，sidebar 内的 p 元素得到了特殊的处理，同时，与页面中其他所有 h2 元素明显不同的是，sidebar 中的 h2 元素也得到了不同的特殊处理。

**单独的选择器**

**id 选择器即使不被用来创建派生选择器，它也可以独立发挥作用：**

#sidebar {

border: 1px dotted #000;

padding: 10px;

}

根据这条规则，id 为 sidebar 的元素将拥有一个像素宽的黑色点状边框，同时其周围会有 10 个像素宽的内边距（padding，内部空白）。老版本的 Windows/IE 浏览器可能会忽略这条规则，除非你特别地定义这个选择器所属的元素：

div#sidebar {

border: 1px dotted #000;

padding: 10px;

}

* [CSS 派生选择器](http://www.w3school.com.cn/css/css_syntax_descendant_selector.asp)
* [CSS 类选择器](http://www.w3school.com.cn/css/css_syntax_class_selector.asp)

CSS 类选择器

* [CSS id 选择器](http://www.w3school.com.cn/css/css_syntax_id_selector.asp)
* [CSS 属性选择器](http://www.w3school.com.cn/css/css_syntax_attribute_selector.asp)

**在 CSS 中，类选择器以一个点号显示：**

.center {text-align: center}

在上面的例子中，所有拥有 center 类的 HTML 元素均为居中。

在下面的 HTML 代码中，h1 和 p 元素都有 center 类。这意味着两者都将遵守 ".center" 选择器中的规则。

<h1 class="center">

This heading will be center-aligned

</h1>

<p class="center">

This paragraph will also be center-aligned.

</p>

**注意：**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。

**和 id 一样，class 也可被用作派生选择器：**

.fancy td {

color: #f60;

background: #666;

}

在上面这个例子中，类名为 fancy 的更大的元素内部的表格单元都会以灰色背景显示橙色文字。（名为 fancy 的更大的元素可能是一个表格或者一个 div）

**元素也可以基于它们的类而被选择：**

td.fancy {

color: #f60;

background: #666;

}

在上面的例子中，类名为 fancy 的表格单元将是带有灰色背景的橙色。

<td class="fancy">

你可以将类 fancy 分配给任何一个表格元素任意多的次数。那些以 fancy 标注的单元格都会是带有灰色背景的橙色。那些没有被分配名为 fancy 的类的单元格不会受这条规则的影响。还有一点值得注意，class 为 fancy 的段落也不会是带有灰色背景的橙色，当然，任何其他被标注为 fancy 的元素也不会受这条规则的影响。这都是由于我们书写这条规则的方式，这个效果被限制于被标注为 fancy 的表格单元（即使用 td 元素来选择 fancy 类）。

CSS 属性选择器

* [CSS 类选择器](http://www.w3school.com.cn/css/css_syntax_class_selector.asp)
* [CSS 创建](http://www.w3school.com.cn/css/css_howto.asp)

**对带有指定属性的 HTML 元素设置样式。**

可以为拥有指定属性的 HTML 元素设置样式，而不仅限于 class 和 id 属性。

**注释：**只有在规定了 !DOCTYPE 时，IE7 和 IE8 才支持属性选择器。在 IE6 及更低的版本中，不支持属性选择。

**属性选择器**

下面的例子为带有 title 属性的所有元素设置样式：

[title]

{

color:red;

}

[亲自试一试](http://www.w3school.com.cn/tiy/t.asp?f=csse_selector_attribute_att)

**属性和值选择器**

下面的例子为 title="W3School" 的所有元素设置样式：

[title=W3School]

{

border:5px solid blue;

}

[亲自试一试](http://www.w3school.com.cn/tiy/t.asp?f=csse_selector_attribute_value)

**属性和值选择器 - 多个值**

下面的例子为包含指定值的 title 属性的所有元素设置样式。适用于由空格分隔的属性值：

[title~=hello] { color:red; }

[亲自试一试](http://www.w3school.com.cn/tiy/t.asp?f=csse_selector_attribute_valuelist_space)

下面的例子为带有包含指定值的 lang 属性的所有元素设置样式。适用于由连字符分隔的属性值：

[lang|=en] { color:red; }

[亲自试一试](http://www.w3school.com.cn/tiy/t.asp?f=csse_selector_attribute_valuelist_hyphen)

**设置表单的样式**

属性选择器在为不带有 class 或 id 的表单设置样式时特别有用：

input[type="text"]

{

width:150px;

display:block;

margin-bottom:10px;

background-color:yellow;

font-family: Verdana, Arial;

}

input[type="button"]

{

width:120px;

margin-left:35px;

display:block;

font-family: Verdana, Arial;

}

[亲自试一试](http://www.w3school.com.cn/tiy/t.asp?f=csse_selector_attribute_form)

**CSS 选择器参考手册**

|  |  |
| --- | --- |
| **选择器** | **描述** |
| [[*attribute*]](http://www.w3school.com.cn/cssref/selector_attribute.asp) | 用于选取带有指定属性的元素。 |
| [[*attribute*=*value*]](http://www.w3school.com.cn/cssref/selector_attribute_value.asp) | 用于选取带有指定属性和值的元素。 |
| [[*attribute*~=*value*]](http://www.w3school.com.cn/cssref/selector_attribute_value_contain.asp) | 用于选取属性值中包含指定词汇的元素。 |
| [[*attribute*|=*value*]](http://www.w3school.com.cn/cssref/selector_attribute_value_start.asp) | 用于选取带有以指定值开头的属性值的元素，该值必须是整个单词。 |
| [[*attribute*^=*value*]](http://www.w3school.com.cn/cssref/selector_attr_begin.asp) | 匹配属性值以指定值开头的每个元素。 |
| [[*attribute*$=*value*]](http://www.w3school.com.cn/cssref/selector_attr_end.asp) | 匹配属性值以指定值结尾的每个元素。 |
| [[*attribute*\*=*value*]](http://www.w3school.com.cn/cssref/selector_attr_contain.asp) | 匹配属性值中包含指定值的每个元素。 |

* [CSS 类选择器](http://www.w3school.com.cn/css/css_syntax_class_selector.asp)
* [CSS 创建](http://www.w3school.com.cn/css/css_howto.asp)

CSS 元素选择器

* [CSS 浮动](http://www.w3school.com.cn/css/css_positioning_floating.asp)
* [CSS 选择器分组](http://www.w3school.com.cn/css/css_selector_grouping.asp)

**CSS 元素选择器**

最常见的 CSS 选择器是元素选择器。换句话说，文档的元素就是最基本的选择器。

如果设置 HTML 的样式，选择器通常将是某个 HTML 元素，比如 p、h1、em、a，甚至可以是 html 本身：

html {color:black;}

h1 {color:blue;}

h2 {color:silver;}

[亲自试一试](http://www.w3school.com.cn/tiy/t.asp?f=csse_selector_type_1)

可以将某个样式从一个元素切换到另一个元素。

假设您决定将上面的段落文本（而不是 h1 元素）设置为灰色。只需要把 h1 选择器改为 p：

html {color:black;}

p {color:gray;}

h2 {color:silver;}

[亲自试一试](http://www.w3school.com.cn/tiy/t.asp?f=csse_selector_type_2)

**类型选择器**

在 W3C 标准中，元素选择器又称为类型选择器（type selector）。

“类型选择器匹配文档语言元素类型的名称。类型选择器匹配文档树中该元素类型的每一个实例。”

下面的规则匹配文档树中所有 h1 元素：

h1 {font-family: sans-serif;}

因此，我们也可以为 XML 文档中的元素设置样式：

**XML 文档：**

<?xml version="1.0" encoding="ISO-8859-1"?>

<?xml-stylesheet type="text/css" href="note.css"?>

<note>

<to>George</to>

<from>John</from>

<heading>Reminder</heading>

<body>Don't forget the meeting!</body>

</note>

**CSS 文档：**

note

{

font-family:Verdana, Arial;

margin-left:30px;

}

to

{

font-size:28px;

display: block;

}

from

{

font-size:28px;

display: block;

}

heading

{

color: red;

font-size:60px;

display: block;

}

body

{

color: blue;

font-size:35px;

display: block;

}

[查看效果](http://www.w3school.com.cn/example/csse/note_css.xml)

通过上面的例子，您可以看到，CSS 元素选择器（类型选择器）可以设置 XML 文档中元素的样式。

如果您需要更多关于为 XML 文档添加样式的知识，请访问 w3school 的 [XML 教程](http://www.w3school.com.cn/xml/index.asp)。

CSS 类选择器详解

* [CSS 选择器分组](http://www.w3school.com.cn/css/css_selector_grouping.asp)
* [CSS ID 选择器详解](http://www.w3school.com.cn/css/css_selector_id.asp)

**类选择器允许以一种独立于文档元素的方式来指定样式。**

**CSS 类选择器**

类选择器允许以一种独立于文档元素的方式来指定样式。

该选择器可以单独使用，也可以与其他元素结合使用。

**提示：**只有适当地标记文档后，才能使用这些选择器，所以使用这两种选择器通常需要先做一些构想和计划。

要应用样式而不考虑具体设计的元素，最常用的方法就是使用类选择器。

**修改 HTML 代码**

在使用类选择器之前，需要修改具体的文档标记，以便类选择器正常工作。

为了将类选择器的样式与元素关联，必须将 class 指定为一个适当的值。请看下面的 HTML 代码：

<h1 class="important">

This heading is very important.

</h1>

<p class="important">

This paragraph is very important.

</p>

在上面的代码中，两个元素的 class 都指定为 important：第一个标题（ h1 元素），第二个段落（p 元素）。

**语法**

然后我们使用以下语法向这些归类的元素应用样式，即类名前有一个点号（.），然后结合通配选择器：

\*.important {color:red;}

如果您只想选择所有类名相同的元素，可以在类选择器中忽略通配选择器，这没有任何不好的影响：

.important {color:red;}

[亲自试一试](http://www.w3school.com.cn/tiy/t.asp?f=csse_selector_class_1)

**结合元素选择器**

类选择器可以结合元素选择器来使用。

例如，您可能希望只有段落显示为红色文本：

p.important {color:red;}

选择器现在会匹配 class 属性包含 important 的所有 p 元素，但是其他任何类型的元素都不匹配，不论是否有此 class 属性。选择器 p.important 解释为：“其 class 属性值为 important 的所有段落”。 因为 h1 元素不是段落，这个规则的选择器与之不匹配，因此 h1 元素不会变成红色文本。

如果你确实希望为 h1 元素指定不同的样式，可以使用选择器 h1.important：

p.important {color:red;}

h1.important {color:blue;}

[亲自试一试](http://www.w3school.com.cn/tiy/t.asp?f=csse_selector_class_2)

**CSS 多类选择器**

在上一节中，我们处理了 class 值中包含一个词的情况。在 HTML 中，一个 class 值中可能包含一个词列表，各个词之间用空格分隔。例如，如果希望将一个特定的元素同时标记为重要（important）和警告（warning），就可以写作：

<p class="important warning">

This paragraph is a very important warning.

</p>

这两个词的顺序无关紧要，写成 warning important 也可以。

我们假设 class 为 important 的所有元素都是粗体，而 class 为 warning 的所有元素为斜体，class 中同时包含 important 和 warning 的所有元素还有一个银色的背景 。就可以写作：

.important {font-weight:bold;}

.warning {font-style:italic;}

.important.warning {background:silver;}

[亲自试一试](http://www.w3school.com.cn/tiy/t.asp?f=csse_selector_class_multiple)

通过把两个类选择器链接在一起，仅可以选择**同时包含这些类名**的元素（类名的顺序不限）。

如果一个多类选择器包含类名列表中没有的一个类名，匹配就会失败。请看下面的规则：

.important.urgent {background:silver;}

不出所料，这个选择器将只匹配 class 属性中包含词 important 和 urgent 的 p 元素。因此，如果一个 p 元素的 class 属性中只有词 important 和 warning，将不能匹配。不过，它能匹配以下元素：

<p class="important urgent warning">

This paragraph is a very important and urgent warning.

</p>

[亲自试一试](http://www.w3school.com.cn/tiy/t.asp?f=csse_selector_class_multiple_2)

**重要事项：**在 IE7 之前的版本中，不同平台的 Internet Explorer 都不能正确地处理多类选择器。

CSS 选择器分组

* [CSS 元素选择器](http://www.w3school.com.cn/css/css_selector_type.asp)
* [CSS 类选择器详解](http://www.w3school.com.cn/css/css_selector_class.asp)

**选择器分组**

假设希望 h2 元素和段落都有灰色。为达到这个目的，最容易的做法是使用以下声明：

h2, p {color:gray;}

将 h2 和 p 选择器放在规则左边，然后用逗号分隔，就定义了一个规则。其右边的样式（color:gray;）将应用到这两个选择器所引用的元素。逗号告诉浏览器，规则中包含两个不同的选择器。如果没有这个逗号，那么规则的含义将完全不同。参见后代选择器。

可以将任意多个选择器分组在一起，对此没有任何限制。

例如，如果您想把很多元素显示为灰色，可以使用类似如下的规则：

body, h2, p, table, th, td, pre, strong, em {color:gray;}

**提示：**通过分组，创作者可以将某些类型的样式“压缩”在一起，这样就可以得到更简洁的样式表。

以下的两组规则能得到同样的结果，不过可以很清楚地看出哪一个写起来更容易：

/\* no grouping \*/

h1 {color:blue;}

h2 {color:blue;}

h3 {color:blue;}

h4 {color:blue;}

h5 {color:blue;}

h6 {color:blue;}

/\* grouping \*/

h1, h2, h3, h4, h5, h6 {color:blue;}

[亲自试一试](http://www.w3school.com.cn/tiy/t.asp?f=csse_grouping_selector)

分组提供了一些有意思的选择。例如，下例中的所有规则分组都是等价的，每个组只是展示了对选择器和声明分组的不同方法：

/\* group 1 \*/

h1 {color:silver; background:white;}

h2 {color:silver; background:gray;}

h3 {color:white; background:gray;}

h4 {color:silver; background:white;}

b {color:gray; background:white;}

/\* group 2 \*/

h1, h2, h4 {color:silver;}

h2, h3 {background:gray;}

h1, h4, b {background:white;}

h3 {color:white;}

b {color:gray;}

/\* group 3 \*/

h1, h4 {color:silver; background:white;}

h2 {color:silver;}

h3 {color:white;}

h2, h3 {background:gray;}

b {color:gray; background:white;}

亲自试一试：

* [group 1](http://www.w3school.com.cn/tiy/t.asp?f=csse_grouping_selector_1)
* [group 2](http://www.w3school.com.cn/tiy/t.asp?f=csse_grouping_selector_2)
* [group 3](http://www.w3school.com.cn/tiy/t.asp?f=csse_grouping_selector_3)

请注意，group 3 中使用了“声明分组”。稍后我们会为您介绍“声明分组”。

**通配符选择器**

CSS2 引入了一种新的简单选择器 - 通配选择器（universal selector），显示为一个星号（\*）。该选择器可以与任何元素匹配，就像是一个通配符。

例如，下面的规则可以使文档中的每个元素都为红色：

\* {color:red;}

[亲自试一试](http://www.w3school.com.cn/tiy/t.asp?f=csse_grouping_selector_universal)

这个声明等价于列出了文档中所有元素的一个分组选择器。利用通配选择器，只需敲一次键（仅一个星号）就能使文档中所有元素的 color 属性值指定为 red。

**声明分组**

我们既可以对选择器进行分组，也可以对声明分组。

假设您希望所有 h1 元素都有红色背景，并使用 28 像素高的 Verdana 字体显示为蓝色文本，可以写以下样式：

h1 {font: 28px Verdana;}

h1 {color: blue;}

h1 {background: red;}

但是上面这种做法的效率并不高。尤其是当我们为一个有多个样式的元素创建这样一个列表时会很麻烦。相反，我们可以将声明分组在一起：

h1 {font: 28px Verdana; color: white; background: black;}

这与前面的 3 行样式表的效果完全相同。

注意，对声明分组，一定要在各个声明的最后使用分号，这很重要。浏览器会忽略样式表中的空白符。只要加了分号，就可以毫无顾忌地采用以下格式建立样式：

h1 {

font: 28px Verdana;

color: blue;

background: red;

}

[亲自试一试](http://www.w3school.com.cn/tiy/t.asp?f=csse_grouping_declaration)

怎么样，上面这种写法的可读性是不是更强。

不过，如果忽略了第二个分号，用户代理就会把这个样式表解释如下：

h1 {

font: 28px Verdana;

color: blue background: red;

}

[亲自试一试](http://www.w3school.com.cn/tiy/t.asp?f=csse_grouping_declaration_error)

因为 background 对 color 来说不是一个合法值，而且由于只能为 color 指定一个关键字，所以用户代理会完全忽略这个 color 声明（包括 background: black 部分）。这样 h1 标题只会显示为蓝色，而没有红色背景，不过更有可能根本得不到蓝色的 h1。相反，这些标题只会显示为默认颜色（通常是黑色），而且根本没有背景色。font: 28px Verdana 声明仍能正常发挥作用，因为它确实正确地以一个分号结尾。

与选择器分组一样，声明分组也是一种便利的方法，可以缩短样式表，使之更清晰，也更易维护。

**提示：**在规则的最后一个声明后也加上分号是一个好习惯。在向规则增加另一个声明时，就不必担心忘记再插入一个分号。

**结合选择器和声明的分组**

我们可以在一个规则中结合选择器分组和声明分组，就可以使用很少的语句定义相对复杂的样式。

下面的规则为所有标题指定了一种复杂的样式：

h1, h2, h3, h4, h5, h6 {

color:gray;

background: white;

padding: 10px;

border: 1px solid black;

font-family: Verdana;

}

[亲自试一试](http://www.w3school.com.cn/tiy/t.asp?f=csse_grouping)

上面这条规则将所有标题的样式定义为带有白色背景的灰色文本，其内边距是 10 像素，并带有 1 像素的实心边框，文本字体是 Verdana。

CSS 子元素选择器

* [CSS 后代选择器](http://www.w3school.com.cn/css/css_selector_descendant.asp)
* [CSS 相邻兄弟选择器](http://www.w3school.com.cn/css/css_selector_adjacent_sibling.asp)

**与后代选择器相比，子元素选择器（Child selectors）只能选择作为某元素子元素的元素。**

**选择子元素**

如果您不希望选择任意的后代元素，而是希望缩小范围，只选择某个元素的子元素，请使用子元素选择器（Child selector）。

例如，如果您希望选择只作为 h1 元素子元素的 strong 元素，可以这样写：

h1 > strong {color:red;}

这个规则会把第一个 h1 下面的两个 strong 元素变为红色，但是第二个 h1 中的 strong 不受影响：

<h1>This is <strong>very</strong> <strong>very</strong> important.</h1>

<h1>This is <em>really <strong>very</strong></em> important.</h1>

[亲自试一试](http://www.w3school.com.cn/tiy/t.asp?f=csse_selector_child)

**语法解释**

您应该已经注意到了，子选择器使用了大于号（子结合符）。

子结合符两边可以有空白符，这是可选的。因此，以下写法都没有问题：

h1 > strong

h1> strong

h1 >strong

h1>strong

如果从右向左读，选择器 h1 > strong 可以解释为“选择作为 h1 元素子元素的所有 strong 元素”。

**结合后代选择器和子选择器**

请看下面这个选择器：

table.company td > p

上面的选择器会选择作为 td 元素子元素的所有 p 元素，这个 td 元素本身从 table 元素继承，该 table 元素有一个包含 company 的 class 属性。

CSS 相邻兄弟选择器

* [CSS 子元素选择器](http://www.w3school.com.cn/css/css_selector_child.asp)
* [CSS 伪类](http://www.w3school.com.cn/css/css_pseudo_classes.asp)

**相邻兄弟选择器（Adjacent sibling selector）可选择紧接在另一元素后的元素，且二者有相同父元素。**

**选择相邻兄弟**

如果需要选择紧接在另一个元素后的元素，而且二者有相同的父元素，可以使用相邻兄弟选择器（Adjacent sibling selector）。

例如，如果要增加紧接在 h1 元素后出现的段落的上边距，可以这样写：

h1 + p {margin-top:50px;}

这个选择器读作：“选择紧接在 h1 元素后出现的段落，h1 和 p 元素拥有共同的父元素”。

[亲自试一试](http://www.w3school.com.cn/tiy/t.asp?f=csse_selector_adjacent_sibling)

**语法解释**

相邻兄弟选择器使用了加号（+），即相邻兄弟结合符（Adjacent sibling combinator）。

**注释：**与子结合符一样，相邻兄弟结合符旁边可以有空白符。

请看下面这个文档树片段：

<div>

<ul>

<li>List item 1</li>

<li>List item 2</li>

<li>List item 3</li>

</ul>

<ol>

<li>List item 1</li>

<li>List item 2</li>

<li>List item 3</li>

</ol>

</div>

在上面的片段中，div 元素中包含两个列表：一个无序列表，一个有序列表，每个列表都包含三个列表项。这两个列表是相邻兄弟，列表项本身也是相邻兄弟。不过，第一个列表中的列表项与第二个列表中的列表项不是相邻兄弟，因为这两组列表项不属于同一父元素（最多只能算堂兄弟）。

请记住，用一个结合符只能选择两个相邻兄弟中的第二个元素。请看下面的选择器：

li + li {font-weight:bold;}

上面这个选择器只会把列表中的第二个和第三个列表项变为粗体。第一个列表项不受影响。

[亲自试一试](http://www.w3school.com.cn/tiy/t.asp?f=csse_selector_adjacent_sibling_2)

**结合其他选择器**

相邻兄弟结合符还可以结合其他结合符：

html > body table + ul {margin-top:20px;}

这个选择器解释为：选择紧接在 table 元素后出现的所有兄弟 ul 元素，该 table 元素包含在一个 body 元素中，body 元素本身是 html 元素的子元素。

CSS 定位 (Positioning)

* [CSS 外边距合并](http://www.w3school.com.cn/css/css_margin_collapsing.asp)
* [CSS 相对定位](http://www.w3school.com.cn/css/css_positioning_relative.asp)

**CSS 定位 (Positioning) 属性允许你对元素进行定位。**

**CSS 定位和浮动**

CSS 为定位和浮动提供了一些属性，利用这些属性，可以建立列式布局，将布局的一部分与另一部分重叠，还可以完成多年来通常需要使用多个表格才能完成的任务。

定位的基本思想很简单，它允许你定义元素框相对于其正常位置应该出现的位置，或者相对于父元素、另一个元素甚至浏览器窗口本身的位置。显然，这个功能非常强大，也很让人吃惊。要知道，用户代理对 CSS2 中定位的支持远胜于对其它方面的支持，对此不应感到奇怪。

另一方面，CSS1 中首次提出了浮动，它以 Netscape 在 Web 发展初期增加的一个功能为基础。浮动不完全是定位，不过，它当然也不是正常流布局。我们会在后面的章节中明确浮动的含义。

**一切皆为框**

div、h1 或 p 元素常常被称为块级元素。这意味着这些元素显示为**一块内容**，即“块框”。与之相反，span 和 strong 等元素称为“行内元素”，这是因为它们的内容显示在行中，即“行内框”。

您可以使用 [display 属性](http://www.w3school.com.cn/cssref/pr_class_display.asp)改变生成的框的类型。这意味着，通过将 display 属性设置为 block，可以让行内元素（比如 <a> 元素）表现得像块级元素一样。还可以通过把 display 设置为 none，让生成的元素根本没有框。这样的话，该框及其所有内容就不再显示，不占用文档中的空间。

但是在一种情况下，即使没有进行显式定义，也会创建块级元素。这种情况发生在把一些文本添加到一个块级元素（比如 div）的开头。即使没有把这些文本定义为段落，它也会被当作段落对待：

<div>

some text

<p>Some more text.</p>

</div>

在这种情况下，这个框称为无名块框，因为它不与专门定义的元素相关联。

块级元素的文本行也会发生类似的情况。假设有一个包含三行文本的段落。每行文本形成一个无名框。无法直接对无名块或行框应用样式，因为没有可以应用样式的地方（注意，行框和行内框是两个概念）。但是，这有助于理解在屏幕上看到的所有东西都形成某种框。

**CSS 定位机制**

CSS 有三种基本的定位机制：普通流、浮动和绝对定位。

除非专门指定，否则所有框都在普通流中定位。也就是说，普通流中的元素的位置由元素在 (X)HTML 中的位置决定。

块级框从上到下一个接一个地排列，框之间的垂直距离是由框的垂直外边距计算出来。

行内框在一行中水平布置。可以使用水平内边距、边框和外边距调整它们的间距。但是，垂直内边距、边框和外边距不影响行内框的高度。由一行形成的水平框称为**行框（Line Box）**，行框的高度总是足以容纳它包含的所有行内框。不过，设置行高可以增加这个框的高度。

在下面的章节，我们会为您详细讲解相对定位、绝对定位和浮动。

**CSS position 属性**

通过使用 [position 属性](http://www.w3school.com.cn/cssref/pr_class_position.asp)，我们可以选择 4 种不同类型的定位，这会影响元素框生成的方式。

position 属性值的含义：

**static**

元素框正常生成。块级元素生成一个矩形框，作为文档流的一部分，行内元素则会创建一个或多个行框，置于其父元素中。

**relative**

元素框偏移某个距离。元素仍保持其未定位前的形状，它原本所占的空间仍保留。

**absolute**

元素框从文档流完全删除，并相对于其包含块定位。包含块可能是文档中的另一个元素或者是初始包含块。元素原先在正常文档流中所占的空间会关闭，就好像元素原来不存在一样。元素定位后生成一个块级框，而不论原来它在正常流中生成何种类型的框。

**fixed**

元素框的表现类似于将 position 设置为 absolute，不过其包含块是视窗本身。

**提示：**相对定位实际上被看作普通流定位模型的一部分，因为元素的位置相对于它在普通流中的位置。

**实例**

[定位：相对定位](http://www.w3school.com.cn/tiy/t.asp?f=csse_position_relative)

本例演示如何相对于一个元素的正常位置来对其定位。

[定位：绝对定位](http://www.w3school.com.cn/tiy/t.asp?f=csse_position_absolute)

本例演示如何使用绝对值来对元素进行定位。

[定位：固定定位](http://www.w3school.com.cn/tiy/t.asp?f=csse_position_fixed)

本例演示如何相对于浏览器窗口来对元素进行定位。

[使用固定值设置图像的上边缘](http://www.w3school.com.cn/tiy/t.asp?f=csse_position_top)

本例演示如何使用固定值设置图像的上边缘。

[使用百分比设置图像的上边缘](http://www.w3school.com.cn/tiy/t.asp?f=csse_position_top_percent)

本例演示如何使用百分比值设置图像的上边缘。

[使用像素值设置图像的底部边缘](http://www.w3school.com.cn/tiy/t.asp?f=csse_position_bottom)

本例演示如何使用像素值设置图像的底部边缘。

[使用百分比设置图像的底部边缘](http://www.w3school.com.cn/tiy/t.asp?f=csse_position_bottom_percent)

本例演示如何使用百分比值设置图像的底部边缘。

[使用固定值设置图像的左边缘](http://www.w3school.com.cn/tiy/t.asp?f=csse_position_left)

本例演示如何使用固定值设置图像的左边缘。

[使用百分比设置图像的左边缘](http://www.w3school.com.cn/tiy/t.asp?f=csse_position_left_percent)

本例演示如何使用百分比值设置图像的左边缘。

[使用固定值设置图像的右边缘](http://www.w3school.com.cn/tiy/t.asp?f=csse_position_right)

本例演示如何使用固定值设置图像的右边缘。

[使用百分比设置图像的右边缘](http://www.w3school.com.cn/tiy/t.asp?f=csse_position_right_percent)

本例演示如何使用百分比值设置图像的右边缘。

[如何使用滚动条来显示元素内溢出的内容](http://www.w3school.com.cn/tiy/t.asp?f=csse_overflow)

本例演示当元素内容太大而超出规定区域时，如何设置溢出属性来规定相应的动作。

[如何隐藏溢出元素中溢出的内容](http://www.w3school.com.cn/tiy/t.asp?f=csse_pos_overflow_hidden)

本例演示在元素中的内容太大以至于无法适应指定的区域时，如何设置 overflow 属性来隐藏其内容。

[如何设置浏览器来自动地处理溢出](http://www.w3school.com.cn/tiy/t.asp?f=csse_pos_overflow_auto)

本例演示如何设置浏览器来自动地处理溢出。

[设置元素的形状](http://www.w3school.com.cn/tiy/t.asp?f=csse_clip)

本例演示如何设置元素的形状。此元素被剪裁到这个形状内，并显示出来。

[垂直排列图象](http://www.w3school.com.cn/tiy/t.asp?f=csse_vertical-align)

本例演示如何在文本中垂直排列图象。

[Z-index](http://www.w3school.com.cn/tiy/t.asp?f=csse_zindex2)

Z-index可被用于将在一个元素放置于另一元素之后。

[Z-index](http://www.w3school.com.cn/tiy/t.asp?f=csse_zindex1)

上面的例子中的元素已经更改了Z-index。

**CSS 定位属性**

CSS 定位属性允许你对元素进行定位。

|  |  |
| --- | --- |
| **属性** | **描述** |
| [position](http://www.w3school.com.cn/cssref/pr_class_position.asp) | 把元素放置到一个静态的、相对的、绝对的、或固定的位置中。 |
| [top](http://www.w3school.com.cn/cssref/pr_pos_top.asp) | 定义了一个定位元素的上外边距边界与其包含块上边界之间的偏移。 |
| [right](http://www.w3school.com.cn/cssref/pr_pos_right.asp) | 定义了定位元素右外边距边界与其包含块右边界之间的偏移。 |
| [bottom](http://www.w3school.com.cn/cssref/pr_pos_bottom.asp) | 定义了定位元素下外边距边界与其包含块下边界之间的偏移。 |
| [left](http://www.w3school.com.cn/cssref/pr_pos_left.asp) | 定义了定位元素左外边距边界与其包含块左边界之间的偏移。 |
| [overflow](http://www.w3school.com.cn/cssref/pr_pos_overflow.asp) | 设置当元素的内容溢出其区域时发生的事情。 |
| [clip](http://www.w3school.com.cn/cssref/pr_pos_clip.asp) | 设置元素的形状。元素被剪入这个形状之中，然后显示出来。 |
| [vertical-align](http://www.w3school.com.cn/cssref/pr_pos_vertical-align.asp) | 设置元素的垂直对齐方式。 |
| [z-index](http://www.w3school.com.cn/cssref/pr_pos_z-index.asp) | 设置元素的堆叠顺序。 |

* [CSS 外边距合并](http://www.w3school.com.cn/css/css_margin_collapsing.asp)
* [CSS 相对定位](http://www.w3school.com.cn/css/css_positioning_relative.asp)

CSS 相对定位

* [CSS 定位概述](http://www.w3school.com.cn/css/css_positioning.asp)
* [CSS 绝对定位](http://www.w3school.com.cn/css/css_positioning_absolute.asp)

**设置为相对定位的元素框会偏移某个距离。元素仍然保持其未定位前的形状，它原本所占的空间仍保留。**

**CSS 相对定位**

相对定位是一个非常容易掌握的概念。如果对一个元素进行相对定位，它将出现在它所在的位置上。然后，可以通过设置垂直或水平位置，让这个元素“相对于”它的起点进行移动。

如果将 top 设置为 20px，那么框将在原位置顶部下面 20 像素的地方。如果 left 设置为 30 像素，那么会在元素左边创建 30 像素的空间，也就是将元素向右移动。

#box\_relative {

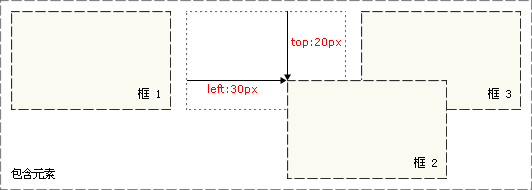
position: relative;

left: 30px;

top: 20px;

}

如下图所示：



注意，在使用相对定位时，无论是否进行移动，元素仍然占据原来的空间。因此，移动元素会导致它覆盖其它框。

**CSS 相对定位实例**

[定位：相对定位](http://www.w3school.com.cn/tiy/t.asp?f=csse_position_relative)

本例演示如何相对于一个元素的正常位置来对其定位。

CSS 绝对定位

* [CSS 相对定位](http://www.w3school.com.cn/css/css_positioning_relative.asp)
* [CSS 浮动](http://www.w3school.com.cn/css/css_positioning_floating.asp)

**设置为绝对定位的元素框从文档流完全删除，并相对于其包含块定位，包含块可能是文档中的另一个元素或者是初始包含块。元素原先在正常文档流中所占的空间会关闭，就好像该元素原来不存在一样。元素定位后生成一个块级框，而不论原来它在正常流中生成何种类型的框。**

**CSS 绝对定位**

绝对定位使元素的位置与文档流无关，因此不占据空间。这一点与相对定位不同，相对定位实际上被看作普通流定位模型的一部分，因为元素的位置相对于它在普通流中的位置。

普通流中其它元素的布局就像绝对定位的元素不存在一样：

#box\_relative {

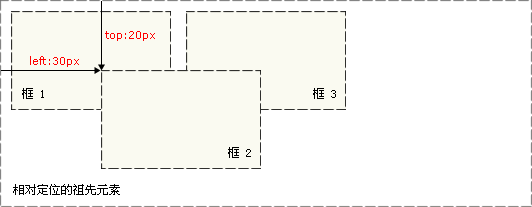
position: absolute;

left: 30px;

top: 20px;

}

如下图所示：



绝对定位的元素的位置相对于**最近的已定位祖先元素**，如果元素没有已定位的祖先元素，那么它的位置相对于**最初的包含块**。

对于定位的主要问题是要记住每种定位的意义。所以，现在让我们复习一下学过的知识吧：相对定位是“相对于”元素在文档中的初始位置，而绝对定位是“相对于”最近的已定位祖先元素，如果不存在已定位的祖先元素，那么“相对于”最初的包含块。

**注释：**根据用户代理的不同，最初的包含块可能是画布或 HTML 元素。

**提示：**因为绝对定位的框与文档流无关，所以它们可以覆盖页面上的其它元素。可以通过设置 [z-index 属性](http://www.w3school.com.cn/cssref/pr_pos_z-index.asp)来控制这些框的堆放次序。

**CSS 绝对定位实例**

[定位：绝对定位](http://www.w3school.com.cn/tiy/t.asp?f=csse_position_absolute)

本例演示如何使用绝对值来对元素进行定位。

CSS 浮动

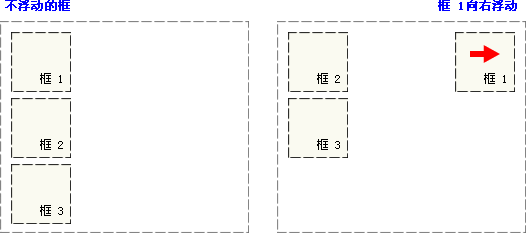
* [CSS 绝对定位](http://www.w3school.com.cn/css/css_positioning_absolute.asp)
* [CSS 元素选择器](http://www.w3school.com.cn/css/css_selector_type.asp)

**浮动的框可以向左或向右移动，直到它的外边缘碰到包含框或另一个浮动框的边框为止。**

**由于浮动框不在文档的普通流中，所以文档的普通流中的块框表现得就像浮动框不存在一样。**

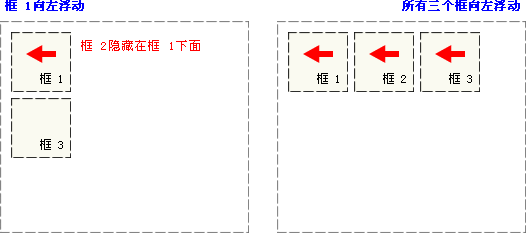
**CSS 浮动**

请看下图，当把框 1 向右浮动时，它脱离文档流并且向右移动，直到它的右边缘碰到包含框的右边缘：

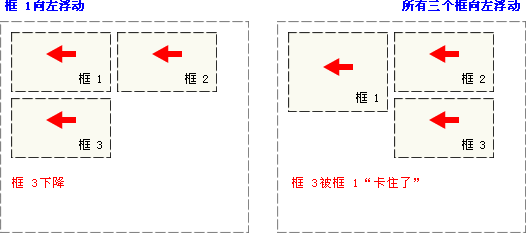


再请看下图，当框 1 向左浮动时，它脱离文档流并且向左移动，直到它的左边缘碰到包含框的左边缘。因为它不再处于文档流中，所以它不占据空间，实际上覆盖住了框 2，使框 2 从视图中消失。

如果把所有三个框都向左移动，那么框 1 向左浮动直到碰到包含框，另外两个框向左浮动直到碰到前一个浮动框。



如下图所示，如果包含框太窄，无法容纳水平排列的三个浮动元素，那么其它浮动块向下移动，直到有足够的空间。如果浮动元素的高度不同，那么当它们向下移动时可能被其它浮动元素“卡住”：



**CSS float 属性**

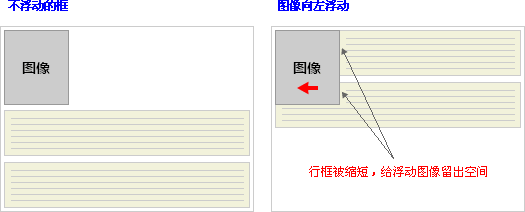
在 CSS 中，我们通过 float 属性实现元素的浮动。

如需更多有关 float 属性的知识，请访问参考手册：[CSS float 属性](http://www.w3school.com.cn/cssref/pr_class_float.asp)。

**行框和清理**

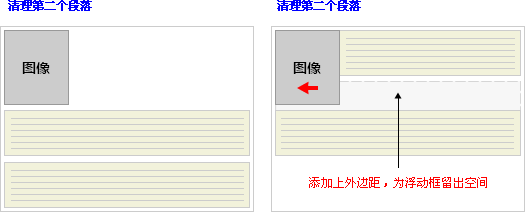
浮动框旁边的行框被缩短，从而给浮动框留出空间，行框围绕浮动框。

因此，创建浮动框可以使文本围绕图像：



要想阻止行框围绕浮动框，需要对该框应用 [clear 属性](http://www.w3school.com.cn/cssref/pr_class_clear.asp)。clear 属性的值可以是 left、right、both 或 none，它表示框的哪些边不应该挨着浮动框。

为了实现这种效果，在被清理的元素的**上外边距**上添加足够的空间，使元素的顶边缘垂直下降到浮动框下面：



这是一个有用的工具，它让周围的元素为浮动元素留出空间。

让我们更详细地看看浮动和清理。假设希望让一个图片浮动到文本块的左边，并且希望这幅图片和文本包含在另一个具有背景颜色和边框的元素中。您可能编写下面的代码：

.news {

background-color: gray;

border: solid 1px black;

}

.news img {

float: left;

}

.news p {

float: right;

}

<div class="news">

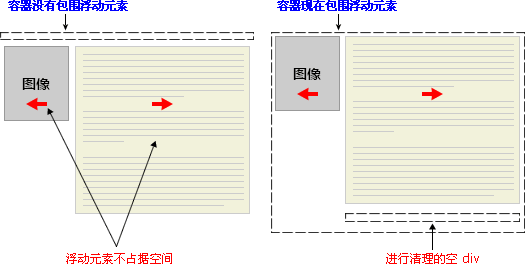
<img src="news-pic.jpg" />

<p>some text</p>

</div>

这种情况下，出现了一个问题。因为浮动元素脱离了文档流，所以包围图片和文本的 div 不占据空间。

如何让包围元素在视觉上包围浮动元素呢？需要在这个元素中的某个地方应用 clear：



不幸的是出现了一个新的问题，由于没有现有的元素可以应用清理，所以我们只能添加一个空元素并且清理它。

.news {

background-color: gray;

border: solid 1px black;

}

.news img {

float: left;

}

.news p {

float: right;

}

.clear {

clear: both;

}

<div class="news">

<img src="news-pic.jpg" />

<p>some text</p>

<div class="clear"></div>

</div>

这样可以实现我们希望的效果，但是需要添加多余的代码。常常有元素可以应用 clear，但是有时候不得不为了进行布局而添加无意义的标记。

不过我们还有另一种办法，那就是对容器 div 进行浮动：

.news {

background-color: gray;

border: solid 1px black;

float: left;

}

.news img {

float: left;

}

.news p {

float: right;

}

<div class="news">

<img src="news-pic.jpg" />

<p>some text</p>

</div>

这样会得到我们希望的效果。不幸的是，下一个元素会受到这个浮动元素的影响。为了解决这个问题，有些人选择对布局中的所有东西进行浮动，然后使用适当的有意义的元素（常常是站点的页脚）对这些浮动进行清理。这有助于减少或消除不必要的标记。

事实上，W3School 站点上的所有页面都采用了这种技术，如果您打开我们使用 CSS 文件，您会看到我们对页脚的 div 进行了清理，而页脚上面的三个 div 都向左浮动。

**CSS clear 属性**

我们刚才详细讨论了 CSS 清理的工作原理和 clear 属性应用方法。如果您希望学习更多有关 clear 属性的知识，请访问参考手册：[CSS clear 属性](http://www.w3school.com.cn/cssref/pr_class_clear.asp)。

**浮动和清理 实例**

[float 属性的简单应用](http://www.w3school.com.cn/tiy/t.asp?f=csse_float)

使图像浮动于一个段落的右侧。

[将带有边框和边界的图像浮动于段落的右侧](http://www.w3school.com.cn/tiy/t.asp?f=csse_float2)

使图像浮动于段落的右侧。向图像添加边框和边界。

[带标题的图像浮动于右侧](http://www.w3school.com.cn/tiy/t.asp?f=csse_float3)

使带有标题的图像浮动于右侧

[使段落的首字母浮动于左侧](http://www.w3school.com.cn/tiy/t.asp?f=csse_float4)

使段落的首字母浮动于左侧，并向这个字母添加样式。

[创建水平菜单](http://www.w3school.com.cn/tiy/t.asp?f=csse_float5)

使用具有一栏超链接的浮动来创建水平菜单。

[创建无表格的首页](http://www.w3school.com.cn/tiy/t.asp?f=csse_float6)

使用浮动来创建拥有页眉、页脚、左侧目录和主体内容的首页。

[清除元素的侧面](http://www.w3school.com.cn/tiy/t.asp?f=csse_class-clear)

本例演示如何使用清除元素侧面的浮动元素。

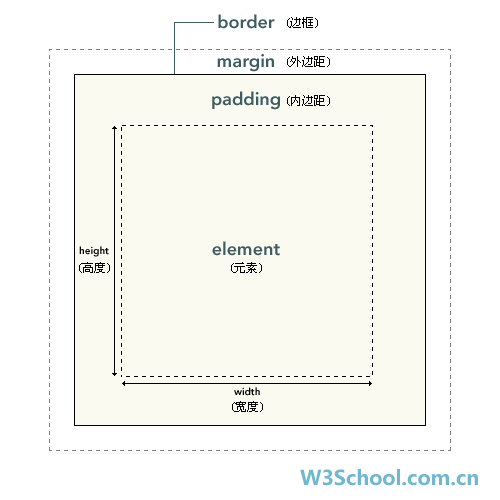
* [CSS 绝对定位](http://www.w3school.com.cn/css/css_positioning_absolute.asp)
* [CSS 元素选择器](http://www.w3school.com.cn/css/css_selector_type.asp)

CSS 框模型概述

* [CSS 轮廓](http://www.w3school.com.cn/css/css_outline.asp)
* [CSS 内边距](http://www.w3school.com.cn/css/css_padding.asp)

**CSS 框模型 (Box Model) 规定了元素框处理元素内容、**[**内边距**](http://www.w3school.com.cn/css/css_padding.asp)**、**[**边框**](http://www.w3school.com.cn/css/css_border.asp) **和** [**外边距**](http://www.w3school.com.cn/css/css_margin.asp) **的方式。**

**CSS 框模型概述**



元素框的最内部分是实际的内容，直接包围内容的是内边距。内边距呈现了元素的背景。内边距的边缘是边框。边框以外是外边距，外边距默认是透明的，因此不会遮挡其后的任何元素。

**提示：**背景应用于由内容和内边距、边框组成的区域。

内边距、边框和外边距都是可选的，默认值是零。但是，许多元素将由用户代理样式表设置外边距和内边距。可以通过将元素的 margin 和 padding 设置为零来覆盖这些浏览器样式。这可以分别进行，也可以使用通用选择器对所有元素进行设置：

\* {

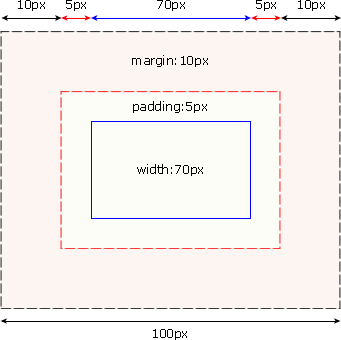
margin: 0;

padding: 0;

}

在 CSS 中，width 和 height 指的是内容区域的宽度和高度。增加内边距、边框和外边距不会影响内容区域的尺寸，但是会增加元素框的总尺寸。

假设框的每个边上有 10 个像素的外边距和 5 个像素的内边距。如果希望这个元素框达到 100 个像素，就需要将内容的宽度设置为 70 像素，请看下图：



#box {

width: 70px;

margin: 10px;

padding: 5px;

}

**提示：**内边距、边框和外边距可以应用于一个元素的所有边，也可以应用于单独的边。

**提示：**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。

**浏览器兼容性**

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。

**术语翻译**

* element : 元素。
* padding : 内边距，也有资料将其翻译为填充。
* border : 边框。
* margin : 外边距，也有资料将其翻译为空白或空白边。

在 w3school，我们把 padding 和 margin 统一地称为内边距和外边距。边框内的空白是内边距，边框外的空白是外边距，很容易记吧：）

CSS 内边距

* [CSS 框模型概述](http://www.w3school.com.cn/css/css_boxmodel.asp)
* [CSS 边框](http://www.w3school.com.cn/css/css_border.asp)

**元素的内边距在边框和内容区之间。控制该区域最简单的属性是 padding 属性。**

**CSS padding 属性定义元素边框与元素内容之间的空白区域。**

**CSS padding 属性**

CSS padding 属性定义元素的内边距。padding 属性接受长度值或百分比值，但不允许使用负值。

例如，如果您希望所有 h1 元素的各边都有 10 像素的内边距，只需要这样：

h1 {padding: 10px;}

您还可以按照上、右、下、左的顺序分别设置各边的内边距，各边均可以使用不同的单位或百分比值：

h1 {padding: 10px 0.25em 2ex 20%;}

**单边内边距属性**

也通过使用下面四个单独的属性，分别设置上、右、下、左内边距：

* [padding-top](http://www.w3school.com.cn/cssref/pr_padding-top.asp)
* [padding-right](http://www.w3school.com.cn/cssref/pr_padding-right.asp)
* [padding-bottom](http://www.w3school.com.cn/cssref/pr_padding-bottom.asp)
* [padding-left](http://www.w3school.com.cn/cssref/pr_padding-left.asp)

您也许已经想到了，下面的规则实现的效果与上面的简写规则是完全相同的：

h1 {

padding-top: 10px;

padding-right: 0.25em;

padding-bottom: 2ex;

padding-left: 20%;

}

**内边距的百分比数值**

前面提到过，可以为元素的内边距设置百分数值。百分数值是相对于其父元素的 width 计算的，这一点与外边距一样。所以，如果父元素的 width 改变，它们也会改变。

下面这条规则把段落的内边距设置为父元素 width 的 10%：

p {padding: 10%;}

例如：如果一个段落的父元素是 div 元素，那么它的内边距要根据 div 的 width 计算。

<div style="width: 200px;">

<p>This paragragh is contained within a DIV that has a width of 200 pixels.</p>

</div>

**注意：**上下内边距与左右内边距一致；即上下内边距的百分数会相对于父元素宽度设置，而不是相对于高度。

**CSS 内边距实例：**

[所有内边距属性在一个声明中](http://www.w3school.com.cn/tiy/t.asp?f=csse_padding)

本例演示使用简写属性将所有的内边距属性设置于一个声明中，可以有一到四个值。

[设置下内边距 1](http://www.w3school.com.cn/tiy/t.asp?f=csse_padding-bottom)

本例演示如何使用厘米值来设置单元格的下内边距。

[设置下内边距 2](http://www.w3school.com.cn/tiy/t.asp?f=csse_padding-bottom_percent)

本例演示如何使用百分比值来设置单元格的下内边距。

[设置左内边距 1](http://www.w3school.com.cn/tiy/t.asp?f=csse_padding-left)

本例演示如何使用厘米值来设置单元格的左内边距。

[设置左内边距 2](http://www.w3school.com.cn/tiy/t.asp?f=csse_padding-left_percent)

本例演示如何使用百分比值来设置单元格的左内边距。

[设置右内边距 1](http://www.w3school.com.cn/tiy/t.asp?f=csse_padding-right)

本例演示如何使用厘米值来设置单元格的右内边距。

[设置右内边距 2](http://www.w3school.com.cn/tiy/t.asp?f=csse_padding-right_percent)

本例演示如何使用百分比值来设置单元格的右内边距。

[设置上内边距 1](http://www.w3school.com.cn/tiy/t.asp?f=csse_padding-top)

本例演示如何使用厘米值来设置单元格的上内边距。

[设置上内边距 2](http://www.w3school.com.cn/tiy/t.asp?f=csse_padding-top_percent)

本例演示如何使用百分比值来设置单元格的上内边距。

**CSS 内边距属性**

|  |  |
| --- | --- |
| **属性** | **描述** |
| [padding](http://www.w3school.com.cn/cssref/pr_padding.asp) | 简写属性。作用是在一个声明中设置元素的所内边距属性。 |
| [padding-bottom](http://www.w3school.com.cn/cssref/pr_padding-bottom.asp) | 设置元素的下内边距。 |
| [padding-left](http://www.w3school.com.cn/cssref/pr_padding-left.asp) | 设置元素的左内边距。 |
| [padding-right](http://www.w3school.com.cn/cssref/pr_padding-right.asp) | 设置元素的右内边距。 |
| [padding-top](http://www.w3school.com.cn/cssref/pr_padding-top.asp) | 设置元素的上内边距。 |

CSS 边框

* [CSS 内边距](http://www.w3school.com.cn/css/css_padding.asp)
* [CSS 外边距](http://www.w3school.com.cn/css/css_margin.asp)

**元素的边框 (border) 是围绕元素内容和内边距的一条或多条线。**

**CSS border 属性允许你规定元素边框的样式、宽度和颜色。**

**CSS 边框**

在 HTML 中，我们使用表格来创建文本周围的边框，但是通过使用 CSS 边框属性，我们可以创建出效果出色的边框，并且可以应用于任何元素。

元素外边距内就是元素的的边框 (border)。元素的边框就是围绕元素内容和内边据的一条或多条线。

每个边框有 3 个方面：宽度、样式，以及颜色。在下面的篇幅，我们会为您详细讲解这三个方面。

**边框与背景**

CSS 规范指出，边框绘制在“元素的背景之上”。这很重要，因为有些边框是“间断的”（例如，点线边框或虚线框），元素的背景应当出现在边框的可见部分之间。

CSS2 指出背景只延伸到内边距，而不是边框。后来 CSS2.1 进行了更正：元素的背景是内容、内边距和边框区的背景。大多数浏览器都遵循 CSS2.1 定义，不过一些较老的浏览器可能会有不同的表现。

**边框的样式**

样式是边框最重要的一个方面，这不是因为样式控制着边框的显示（当然，样式确实控制着边框的显示），而是因为如果没有样式，将根本没有边框。

CSS 的 [border-style 属性](http://www.w3school.com.cn/cssref/pr_border-style.asp)定义了 10 个不同的非 inherit 样式，包括 none。

例如，您可以为把一幅图片的边框定义为 outset，使之看上去像是“凸起按钮”：

a:link img {border-style: outset;}

**定义多种样式**

您可以为一个边框定义多个样式，例如：

p.aside {border-style: solid dotted dashed double;}

上面这条规则为类名为 aside 的段落定义了四种边框样式：实线上边框、点线右边框、虚线下边框和一个双线左边框。

我们又看到了这里的值采用了 top-right-bottom-left 的顺序，讨论用多个值设置不同内边距时也见过这个顺序。

**定义单边样式**

如果您希望为元素框的某一个边设置边框样式，而不是设置所有 4 个边的边框样式，可以使用下面的单边边框样式属性：

* [border-top-style](http://www.w3school.com.cn/cssref/pr_border-top_style.asp)
* [border-right-style](http://www.w3school.com.cn/cssref/pr_border-right_style.asp)
* [border-bottom-style](http://www.w3school.com.cn/cssref/pr_border-bottom_style.asp)
* [border-left-style](http://www.w3school.com.cn/cssref/pr_border-left_style.asp)

因此这两种方法是等价的：

p {border-style: solid solid solid none;}

p {border-style: solid; border-left-style: none;}

**注意：**如果要使用第二种方法，必须把单边属性放在简写属性之后。因为如果把单边属性放在 border-style 之前，简写属性的值就会覆盖单边值 none。

**边框的宽度**

您可以通过 [border-width 属性](http://www.w3school.com.cn/cssref/pr_border-width.asp)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；或者使用 3 个关键字之一，它们分别是 thin 、medium（默认值） 和 thick。

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

所以，我们可以这样设置边框的宽度：

p {border-style: solid; border-width: 5px;}

或者：

p {border-style: solid; border-width: thick;}

**定义单边宽度**

您可以按照 top-right-bottom-left 的顺序设置元素的各边边框：

p {border-style: solid; border-width: 15px 5px 15px 5px;}

上面的例子也可以简写为（这样写法称为**值复制**）：

p {border-style: solid; border-width: 15px 5px;}

您也可以通过下列属性分别设置边框各边的宽度：

* [border-top-width](http://www.w3school.com.cn/cssref/pr_border-top_width.asp)
* [border-right-width](http://www.w3school.com.cn/cssref/pr_border-right_width.asp)
* [border-bottom-width](http://www.w3school.com.cn/cssref/pr_border-bottom_width.asp)
* [border-left-width](http://www.w3school.com.cn/cssref/pr_border-left_width.asp)

因此，下面的规则与上面的例子是等价的：

p {

border-style: solid;

border-top-width: 15px;

border-right-width: 5px;

border-bottom-width: 15px;

border-left-width: 5px;

}

**没有边框**

在前面的例子中，您已经看到，如果希望显示某种边框，就必须设置边框样式，比如 solid 或 outset。

那么如果把 border-style 设置为 none 会出现什么情况：

p {border-style: none; border-width: 50px;}

尽管边框的宽度是 50px，但是边框样式设置为 none。在这种情况下，不仅边框的样式没有了，其宽度也会变成 0。边框消失了，为什么呢？

这是因为如果边框样式为 none，即边框根本不存在，那么边框就不可能有宽度，因此边框宽度自动设置为 0，而不论您原先定义的是什么。

记住这一点非常重要。事实上，忘记声明边框样式是一个常犯的错误。根据以下规则，所有 h1 元素都不会有任何边框，更不用说 20 像素宽了：

h1 {border-width: 20px;}

由于 border-style 的默认值是 none，如果没有声明样式，就相当于 border-style: none。**因此，如果您希望边框出现，就必须声明一个边框样式。**

**边框的颜色**

设置边框颜色非常简单。CSS 使用一个简单的 [border-color 属性](http://www.w3school.com.cn/cssref/pr_border-color.asp)，它一次可以接受最多 4 个颜色值。

可以使用任何类型的颜色值，例如可以是命名颜色，也可以是十六进制和 RGB 值：

p {

border-style: solid;

border-color: blue rgb(25%,35%,45%) #909090 red;

}

如果颜色值小于 4 个，值复制就会起作用。例如下面的规则声明了段落的上下边框是蓝色，左右边框是红色：

p {

border-style: solid;

border-color: blue red;

}

**注释：**默认的边框颜色是元素本身的前景色。如果没有为边框声明颜色，它将与元素的文本颜色相同。另一方面，如果元素没有任何文本，假设它是一个表格，其中只包含图像，那么该表的边框颜色就是其父元素的文本颜色（因为 color 可以继承）。这个父元素很可能是 body、div 或另一个 table。

**定义单边颜色**

还有一些单边边框颜色属性。它们的原理与单边样式和宽度属性相同：

* [border-top-color](http://www.w3school.com.cn/cssref/pr_border-top_color.asp)
* [border-right-color](http://www.w3school.com.cn/cssref/pr_border-right_color.asp)
* [border-bottom-color](http://www.w3school.com.cn/cssref/pr_border-bottom_color.asp)
* [border-left-color](http://www.w3school.com.cn/cssref/pr_border-left_color.asp)

要为 h1 元素指定实线黑色边框，而右边框为实线红色，可以这样指定：

h1 {

border-style: solid;

border-color: black;

border-right-color: red;

}

**透明边框**

我们刚才讲过，如果边框没有样式，就没有宽度。不过有些情况下您可能希望创建一个不可见的边框。

CSS2 引入了边框颜色值 transparent。这个值用于创建有宽度的不可见边框。请看下面的例子：

<a href="#">AAA</a>

<a href="#">BBB</a>

<a href="#">CCC</a>

我们为上面的链接定义了如下样式：

a:link, a:visited {

border-style: solid;

border-width: 5px;

border-color: transparent;

}

a:hover {border-color: gray;}

如需查看以上样式的效果，请点击：[TIY](http://www.w3school.com.cn/tiy/t.asp?f=csse_border_color_transparent)。

从某种意义上说，利用 transparent，使用边框就像是额外的内边距一样；此外还有一个好处，就是能在你需要的时候使其可见。这种透明边框相当于内边距，因为元素的背景会延伸到边框区域（如果有可见背景的话）。

**重要事项：**在 IE7 之前，IE/WIN 没有提供对 transparent 的支持。在以前的版本，IE 会根据元素的 color 值来设置边框颜色。

**CSS 边框实例：**

[所有边框属性在一个声明之中](http://www.w3school.com.cn/tiy/t.asp?f=csse_border)

本例演示用简写属性来将所有四个边框属性设置于同一声明中。

[设置四边框样式](http://www.w3school.com.cn/tiy/t.asp?f=csse_border-style)

本例演示如何设置四边框样式。

[设置每一边的不同边框](http://www.w3school.com.cn/tiy/t.asp?f=csse_border-style2)

本例演示如何在元素的各边设置不同的边框。

[所有边框宽度属性在一个声明之中](http://www.w3school.com.cn/tiy/t.asp?f=csse_border-width)

本例演示用简写属性来将所有边框宽度属性设置于同一声明中。

[设置四个边框的颜色](http://www.w3school.com.cn/tiy/t.asp?f=csse_border-color)

本例演示如何设置四个边框的颜色。可以设置一到四个颜色。

[所有下边框属性在一个声明中](http://www.w3school.com.cn/tiy/t.asp?f=csse_border-bottom)

本例演示用简写属性来将所有下边框属性设置在同一声明中。

[设置下边框的颜色](http://www.w3school.com.cn/tiy/t.asp?f=csse_border-bottom-color)

本例演示如何设置下边框的颜色。

[设置下边框的样式](http://www.w3school.com.cn/tiy/t.asp?f=csse_border-bottom-style)

本例演示如何设置下边框的样式。

[设置下边框的宽度](http://www.w3school.com.cn/tiy/t.asp?f=csse_border-bottom-width)

本例演示如何设置下边框的宽度。

[所有左边框属性在一个声明之中](http://www.w3school.com.cn/tiy/t.asp?f=csse_border-left)

所有左边框属性在一个声明之中

[设置左边框的颜色](http://www.w3school.com.cn/tiy/t.asp?f=csse_border-left-color)

本例演示如何设置左边框的颜色。

[设置左边框的样式](http://www.w3school.com.cn/tiy/t.asp?f=csse_border-left-style)

本例演示如何设置左边框的样式。

[设置左边框的宽度](http://www.w3school.com.cn/tiy/t.asp?f=csse_border-left-width)

本例演示如何设置左边框的宽度。

[所有右边框属性在一个声明之中](http://www.w3school.com.cn/tiy/t.asp?f=csse_border-right)

本例演示一个简写属性，用于把所有右边框属性设置在一条声明中。

[设置右边框的颜色](http://www.w3school.com.cn/tiy/t.asp?f=csse_border-right-color)

本例演示如何设置右边框的颜色。

[设置右边框的样式](http://www.w3school.com.cn/tiy/t.asp?f=csse_border-right-style)

本例演示如何设置右边框的样式。

[设置右边框的宽度](http://www.w3school.com.cn/tiy/t.asp?f=csse_border-right-width)

本例演示如何设置右边框的宽度。

[所有上边框属性在一个声明之中](http://www.w3school.com.cn/tiy/t.asp?f=csse_border-top)

本例演示用简写属性来将所有上边框属性设置于同一声明之中。

[设置上边框的颜色](http://www.w3school.com.cn/tiy/t.asp?f=csse_border-top-color)

本例演示如何设置上边框的颜色。

[设置上边框的样式](http://www.w3school.com.cn/tiy/t.asp?f=csse_border-top-style)

本例演示如何设置上边框的样式。

[设置上边框的宽度](http://www.w3school.com.cn/tiy/t.asp?f=csse_border-top-width)

本例演示如何设置上边框的宽度。

**CSS 边框属性**

|  |  |
| --- | --- |
| **属性** | **描述** |
| [border](http://www.w3school.com.cn/cssref/pr_border.asp) | 简写属性，用于把针对四个边的属性设置在一个声明。 |
| [border-style](http://www.w3school.com.cn/cssref/pr_border-style.asp) | 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](http://www.w3school.com.cn/cssref/pr_border-width.asp) | 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](http://www.w3school.com.cn/cssref/pr_border-color.asp) | 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](http://www.w3school.com.cn/cssref/pr_border-bottom.asp) | 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](http://www.w3school.com.cn/cssref/pr_border-bottom_color.asp) | 设置元素的下边框的颜色。 |
| [border-bottom-style](http://www.w3school.com.cn/cssref/pr_border-bottom_style.asp) | 设置元素的下边框的样式。 |
| [border-bottom-width](http://www.w3school.com.cn/cssref/pr_border-bottom_width.asp) | 设置元素的下边框的宽度。 |
| [border-left](http://www.w3school.com.cn/cssref/pr_border-left.asp) | 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](http://www.w3school.com.cn/cssref/pr_border-left_color.asp) | 设置元素的左边框的颜色。 |
| [border-left-style](http://www.w3school.com.cn/cssref/pr_border-left_style.asp) | 设置元素的左边框的样式。 |
| [border-left-width](http://www.w3school.com.cn/cssref/pr_border-left_width.asp) | 设置元素的左边框的宽度。 |
| [border-right](http://www.w3school.com.cn/cssref/pr_border-right.asp) | 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](http://www.w3school.com.cn/cssref/pr_border-right_color.asp) | 设置元素的右边框的颜色。 |
| [border-right-style](http://www.w3school.com.cn/cssref/pr_border-right_style.asp) | 设置元素的右边框的样式。 |
| [border-right-width](http://www.w3school.com.cn/cssref/pr_border-right_width.asp) | 设置元素的右边框的宽度。 |
| [border-top](http://www.w3school.com.cn/cssref/pr_border-top.asp) | 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](http://www.w3school.com.cn/cssref/pr_border-top_color.asp) | 设置元素的上边框的颜色。 |
| [border-top-style](http://www.w3school.com.cn/cssref/pr_border-top_style.asp) | 设置元素的上边框的样式。 |
| [border-top-width](http://www.w3school.com.cn/cssref/pr_border-top_width.asp) | 设置元素的上边框的宽度。 |

* [CSS 内边距](http://www.w3school.com.cn/css/css_padding.asp)
* [CSS 外边距](http://www.w3school.com.cn/css/css_margin.asp)

CSS 外边距

* [CSS 边框](http://www.w3school.com.cn/css/css_border.asp)
* [CSS 外边距合并](http://www.w3school.com.cn/css/css_margin_collapsing.asp)

**围绕在元素边框的空白区域是外边距。设置外边距会在元素外创建额外的“空白”。**

**设置外边距的最简单的方法就是使用 margin 属性，这个属性接受任何长度单位、百分数值甚至负值。**

**CSS margin 属性**

设置外边距的最简单的方法就是使用 [margin 属性](http://www.w3school.com.cn/cssref/pr_margin.asp)。

margin 属性接受任何长度单位，可以是像素、英寸、毫米或 em。

margin 可以设置为 auto。更常见的做法是为外边距设置长度值。下面的声明在 h1 元素的各个边上设置了 1/4 英寸宽的空白：

h1 {margin : 0.25in;}

下面的例子为 h1 元素的四个边分别定义了不同的外边距，所使用的长度单位是像素 (px)：

h1 {margin : 10px 0px 15px 5px;}

与内边距的设置相同，这些值的顺序是从上外边距 (top) 开始围着元素顺时针旋转的：

margin: top right bottom left

另外，还可以为 margin 设置一个百分比数值：

p {margin : 10%;}

百分数是相对于父元素的 width 计算的。上面这个例子为 p 元素设置的外边距是其父元素的 width 的 10%。

margin 的默认值是 0，所以如果没有为 margin 声明一个值，就不会出现外边距。但是，在实际中，浏览器对许多元素已经提供了预定的样式，外边距也不例外。例如，在支持 CSS 的浏览器中，外边距会在每个段落元素的上面和下面生成“空行”。因此，如果没有为 p 元素声明外边距，浏览器可能会自己应用一个外边距。当然，只要你特别作了声明，就会覆盖默认样式。

**值复制**

还记得吗？我们曾经在前两节中提到过值复制。下面我们为您讲解如何使用值复制。

有时，我们会输入一些重复的值：

p {margin: 0.5em 1em 0.5em 1em;}

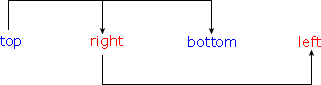
通过值复制，您可以不必重复地键入这对数字。上面的规则与下面的规则是等价的：

p {margin: 0.5em 1em;}

这两个值可以取代前面 4 个值。这是如何做到的呢？CSS 定义了一些规则，允许为外边距指定少于 4 个值。规则如下：

* 如果缺少左外边距的值，则使用右外边距的值。
* 如果缺少下外边距的值，则使用上外边距的值。
* 如果缺少右外边距的值，则使用上外边距的值。

下图提供了更直观的方法来了解这一点：



换句话说，如果为外边距指定了 3 个值，则第 4 个值（即左外边距）会从第 2 个值（右外边距）复制得到。如果给定了两个值，第 4 个值会从第 2 个值复制得到，第 3 个值（下外边距）会从第 1 个值（上外边距）复制得到。最后一个情况，如果只给定一个值，那么其他 3 个外边距都由这个值（上外边距）复制得到。

利用这个简单的机制，您只需指定必要的值，而不必全部都应用 4 个值，例如：

h1 {margin: 0.25em 1em 0.5em;} /\* 等价于 0.25em 1em 0.5em 1em \*/

h2 {margin: 0.5em 1em;} /\* 等价于 0.5em 1em 0.5em 1em \*/

p {margin: 1px;} /\* 等价于 1px 1px 1px 1px \*/

这种办法有一个小缺点，您最后肯定会遇到这个问题。假设希望把 p 元素的上外边距和左外边距设置为 20 像素，下外边距和右外边距设置为 30 像素。在这种情况下，必须写作：

p {margin: 20px 30px 30px 20px;}

这样才能得到您想要的结果。遗憾的是，在这种情况下，所需值的个数没有办法更少了。

再来看另外一个例子。如果希望除了左外边距以外所有其他外边距都是 auto（左外边距是 20px）：

p {margin: auto auto auto 20px;}

同样的，这样才能得到你想要的效果。问题在于，键入这些 auto 有些麻烦。如果您只是希望控制元素单边上的外边距，请使用单边外边距属性。

**单边外边距属性**

您可以使用单边外边距属性为元素单边上的外边距设置值。假设您希望把 p 元素的左外边距设置为 20px。不必使用 margin（需要键入很多 auto），而是可以采用以下方法：

p {margin-left: 20px;}

您可以使用下列任何一个属性来只设置相应上的外边距，而不会直接影响所有其他外边距：

* [margin-top](http://www.w3school.com.cn/cssref/pr_margin-top.asp)
* [margin-right](http://www.w3school.com.cn/cssref/pr_margin-right.asp)
* [margin-bottom](http://www.w3school.com.cn/cssref/pr_margin-bottom.asp)
* [margin-left](http://www.w3school.com.cn/cssref/pr_margin-left.asp)

一个规则中可以使用多个这种单边属性，例如：

h2 {

margin-top: 20px;

margin-right: 30px;

margin-bottom: 30px;

margin-left: 20px;

}

当然，对于这种情况，使用 margin 可能更容易一些：

p {margin: 20px 30px 30px 20px;}

不论使用单边属性还是使用 margin，得到的结果都一样。一般来说，如果希望为多个边设置外边距，使用 margin 会更容易一些。不过，从文档显示的角度看，实际上使用哪种方法都不重要，所以应该选择对自己来说更容易的一种方法。

**提示和注释**

**提示：**Netscape 和 IE 对 body 标签定义的默认边距（margin）值是 8px。而 Opera 不是这样。相反地，Opera 将内部填充（padding）的默认值定义为 8px，因此如果希望对整个网站的边缘部分进行调整，并将之正确显示于 Opera 中，那么必须对 body 的 padding 进行自定义。

**CSS 外边距实例：**

[设置文本的左外边距](http://www.w3school.com.cn/tiy/t.asp?f=csse_margin-left)

本例演示如何设置文本的左外边距。

[设置文本的右外边距](http://www.w3school.com.cn/tiy/t.asp?f=csse_margin-right)

本例演示如何设置文本的右外边距。

[设置文本的上外边距](http://www.w3school.com.cn/tiy/t.asp?f=csse_margin-top)

本例演示如何设置文本的上外边距。

[设置文本的下外边距](http://www.w3school.com.cn/tiy/t.asp?f=csse_margin-bottom)

本例演示如何设置文本的下外边距。

[所有的外边距属性在一个声明中。](http://www.w3school.com.cn/tiy/t.asp?f=csse_margin)

本例演示如何将所有的外边距属性设置于一个声明中。

**CSS 外边距属性**

|  |  |
| --- | --- |
| **属性** | **描述** |
| [margin](http://www.w3school.com.cn/cssref/pr_margin.asp) | 简写属性。在一个声明中设置所有外边距属性。 |
| [margin-bottom](http://www.w3school.com.cn/cssref/pr_margin-bottom.asp) | 设置元素的下外边距。 |
| [margin-left](http://www.w3school.com.cn/cssref/pr_margin-left.asp) | 设置元素的左外边距。 |
| [margin-right](http://www.w3school.com.cn/cssref/pr_margin-right.asp) | 设置元素的右外边距。 |
| [margin-top](http://www.w3school.com.cn/cssref/pr_margin-top.asp) | 设置元素的上外边距。 |

* [CSS 边框](http://www.w3school.com.cn/css/css_border.asp)
* [CSS 外边距合并](http://www.w3school.com.cn/css/css_margin_collapsing.asp)

CSS 外边距合并

* [CSS 外边距](http://www.w3school.com.cn/css/css_margin.asp)
* [CSS 定位概述](http://www.w3school.com.cn/css/css_positioning.asp)

**外边距合并指的是，当两个垂直外边距相遇时，它们将形成一个外边距。**

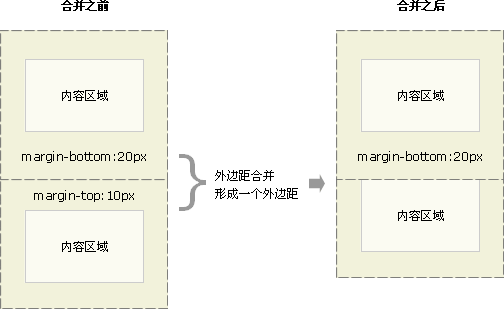
**合并后的外边距的高度等于两个发生合并的外边距的高度中的较大者。**

**外边距合并**

外边距合并（叠加）是一个相当简单的概念。但是，在实践中对网页进行布局时，它会造成许多混淆。

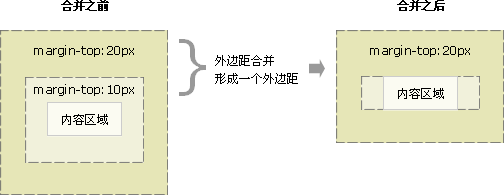
简单地说，外边距合并指的是，当两个垂直外边距相遇时，它们将形成一个外边距。合并后的外边距的高度等于两个发生合并的外边距的高度中的较大者。

当一个元素出现在另一个元素上面时，第一个元素的下外边距与第二个元素的上外边距会发生合并。请看下图：



[亲自试一试](http://www.w3school.com.cn/tiy/t.asp?f=csse_margin_collapsing1)

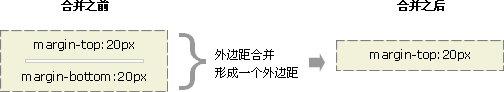
当一个元素包含在另一个元素中时（假设没有内边距或边框把外边距分隔开），它们的上和/或下外边距也会发生合并。请看下图：



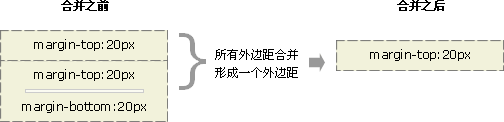
[亲自试一试](http://www.w3school.com.cn/tiy/t.asp?f=csse_margin_collapsing2)

尽管看上去有些奇怪，但是外边距甚至可以与自身发生合并。

假设有一个空元素，它有外边距，但是没有边框或填充。在这种情况下，上外边距与下外边距就碰到了一起，它们会发生合并：

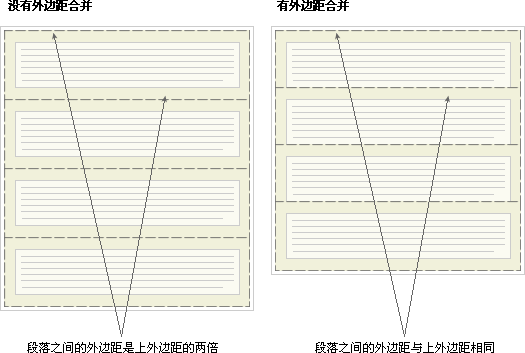


如果这个外边距遇到另一个元素的外边距，它还会发生合并：



这就是一系列的段落元素占用空间非常小的原因，因为它们的所有外边距都合并到一起，形成了一个小的外边距。

外边距合并初看上去可能有点奇怪，但是实际上，它是有意义的。以由几个段落组成的典型文本页面为例。第一个段落上面的空间等于段落的上外边距。如果没有外边距合并，后续所有段落之间的外边距都将是相邻上外边距和下外边距的和。这意味着段落之间的空间是页面顶部的两倍。如果发生外边距合并，段落之间的上外边距和下外边距就合并在一起，这样各处的距离就一致了。



**注释：**只有普通文档流中块框的垂直外边距才会发生外边距合并。行内框、浮动框或绝对定位之间的外边距不会合并。