1. 优先级:

(1)内联样式（在 HTML 元素内部）

(2)内部样式表（位于 <head> 标签内部）

(3)外部样式表

(4)浏览器缺省设置

1. 语法：selector {property: value; property: value; }

body {

color: #000;

background: #fff;

margin: 0;

padding: 0;

font-family: Georgia, Palatino, serif;

}

1. 是否包含空格不会影响 CSS 在浏览器的工作效果；

CSS 对大小写不敏感。

例外： class 和 id 名称对大小写是敏感的。

1. 选择器的分组:被分组的选择器可分享相同的声明

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

1. 继承:

子元素将继承最高级元素（如body）的属性（子元素诸如 p, td, ul, ol, ul, li, dl, dt,和 dd）。

但Netscape 4 就不支持继承，不仅忽略继承，也忽略应用于 body 元素的规则

"Be Kind to Netscape 4" 的冗余法则处理:

body {

font-family: Verdana, sans-serif;

}

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

font-family: Verdana, sans-serif;

}

创建一个针对 p 的特殊规则，它就会摆脱父元素的规则

1. 派生选择器:**依据元素在其位置的上下文关系来定义样式**

li strong {

font-style: italic;

font-weight: normal;

}

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

<ol>

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

<li>我是正常的字体。</li>

</ol>

1. id 选择器

#red {color:red;}

<p id="red">这个段落是红色。</p>

常用于建立派生选择器

#sidebar p {

font-style: italic;

text-align: right;

margin-top: 0.5em;

}

1. 类选择器

.center {text-align: center}

<h1 class="center">

This heading will be center-aligned

</h1>

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。

class 也可被用作派生选择器

1. 属性选择器：对带有指定属性的 HTML 元素设置样式

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。

[title]

{

color:red;

}

<h2 title="Hello world">Hello world</h2>

1. 属性和值选择器

为 title="W3School" 的所有元素设置样式：

[title=W3School]

{

border:5px solid blue;

}

<img title="W3School" src="/i/w3school\_logo\_white.gif" />

1. 属性和值选择器 - 多个值

（1）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; }

<h1>可以应用样式：</h1>

<h2 title="hello world">Hello world</h2>

<p title="student hello">Hello W3School students!</h1>

<hr />

<h1>无法应用样式：</h1>

<h2 title="world">Hello world</h2>

<p title="student">Hello W3School students!</p>

(2) 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; }

<h1>可以应用样式：</h1>

<p lang="en">Hello!</p>

<p lang="en-us">Hi!</p>

<hr />

<h1>无法应用样式：</h1>

<p lang="us">Hi!</p>

<p lang="zh">Hao!</p>

1. 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;

}

|  |  |
| --- | --- |
| **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。 |

1. **插入样式表的方法**
2. 外部样式表

当样式需要应用于很多页面时

<head>

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

</head>

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

1. 内部样式表

当单个文档需要特殊的样式时。可使用 <style> 标签在文档头部定义内部样式表

<head>

<style type="text/css">

hr {color: sienna;}

p {margin-left: 20px;}

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

</style>

</head>

1. 内联样式

样式仅需要在一个元素上应用一次时. 在相关的标签内使用样式（style）属性

<p style="color: sienna; margin-left: 20px">

This is a paragraph

</p>

1. 多重样式

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

例如，外部样式表：

h3 {

color: red;

text-align: left;

font-size: 8pt;

}

而内部样式表：

h3 {

text-align: right;

font-size: 20pt;

}

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

color: red;

text-align: right;

font-size: 20pt;

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

1. **CSS 背景**

允许应用纯色或背景图像

(1)背景色:如果希望背景色从元素中的文本向外少有延伸，只需增加一些内边距：

p {background-color: gray; padding: 20px;}

background-color 不能继承，其默认值是 transparent。

(2) 背景图像

body {background-image: url(/i/eg\_bg\_04.gif);}

background-image 属性的默认值是 none;不能继承

(3)背景重复

body

{

background-image: url(/i/eg\_bg\_03.gif);

background-repeat: repeat-y;

}

repeat水平垂直方向上都平铺。

repeat-x 和 repeat-y 分别导致图像只在水平或垂直方向上重复

no-repeat 则不允许图像在任何方向上平铺。

(4)背景定位

body

{

background-image:url('/i/eg\_bg\_03.gif');

background-repeat:no-repeat;

background-position:center;

}

1. background-position值:

关键字：top、bottom、left、right 和 center

长度值:如 100px 或 5cm

百分数值

2) 关键字

保证不超过两个关键字 - 一个对应水平方向，另一个对应垂直方向。

如果只出现一个关键字，则认为另一个关键字是 center。

p

{

background-image:url('bgimg.gif');

background-repeat:no-repeat;

background-position:top;// 中部上方

}

等价的位置关键字：

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

3)百分数值

将图像在其元素中居中：

body

{

background-image:url('/i/eg\_bg\_03.gif');

background-repeat:no-repeat;

background-position:50% 50%;

}

图像中心与其元素的中心对齐。百分数值同时应用于元素和图像。

图像中描述为 50% 50% 的点（中心点）与元素中描述为 50% 50% 的点（中心点）对齐。

0% 0%，图像左上角将放在元素内边距区的左上角。

100% 100%，图像的右下角放在右边距的右下角。

水平方向 2/3、垂直方向 1/3 处：

body

{

background-image:url('/i/eg\_bg\_03.gif');

background-repeat:no-repeat;

background-position:66% 33%;

}

如只提供一个百分数值，该值将用作水平值，垂直值将假设为 50%。

background-position 的默认值是 0% 0%，相当于 top left。

4)长度值

元素内边距区左上角的偏移。偏移点是图像的左上角。

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

5)背景关联

声明图像相对于可视区是固定的（fixed），因此不会受到滚动的影响：

body {

background-image:url(/i/eg\_bg\_02.gif);

background-repeat:no-repeat;

background-attachment:fixed

}

background-attachment 属性的默认值是 scroll

1. **CSS 文本**

改变文本的颜色、字符间距，对齐文本，装饰文本，对文本进行缩进

* 1. 缩进文本

将段落的首行缩进：

p {text-indent: 5em;}

(1)注意：可为所有块级元素应用 text-indent，但无法将该属性应用于行内元素，图像之类的替换元素上也无法应用 text-indent 属性。

如果一个块级元素（比如段落）的首行中有一个图像，它会随该行的其余文本移动。

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

(2)使用负值

比如“悬挂缩进”：

要当心，可能会超出浏览器窗口的左边界。建议针对负缩进再设置一个外边距或一些内边距：

p {text-indent: -5em; padding-left: 5em;}

(3)使用百分比值

可使用所有长度单位，包括百分比值。

百分数要相对于缩进元素父元素的宽度。

如果将缩进值设置为 20%，所影响元素的第一行会缩进其父元素宽度的 20%。

div {width: 500px;}

p {text-indent: 20%;}//缩进值是父元素的 20%，即 100 个像素：

<div>

<p>this is a paragragh</p>

</div>

(4)继承

text-indent 属性可以继承：

div#outer {width: 500px;}

div#inner {text-indent: 10%;}

p {width: 200px;}

<div id="outer">

<div id="inner">some text. some text. some text.

<p>this is a paragragh.</p>

</div>

</div>

段落也会缩进 50 像素，因为这个段落继承了 id 为 inner 的 div 元素的缩进值。

(5)水平对齐

西方语言text-align 的默认值是 left。文本在左边界对齐

希伯来语和阿拉伯语，text-align 则默认为 right

center 会使每个文本行在元素中居中。

提示：将块级元素或表元素居中，要通过在这些元素上适当地设置左、右外边距来实现。

(6)text-align:center 与 <CENTER>

<CENTER> 不仅影响文本，还会把整个元素居中。

text-align 不会控制元素的对齐，而只影响内部内容。justify

(7)最后一个水平对齐属性是 justify。

在两端对齐文本中，文本行的左右两端都放在父元素的内边界上。然后，调整单词和字母间的间隔，使各行的长度恰好相等。在打印领域很常见。

要由用户代理（而不是 CSS）来确定两端对齐文本如何拉伸，以填满父元素左右边界之间的空间。

(8)字间隔

[word-spacing 属性](http://www.w3school.com.cn/cssref/pr_text_word-spacing.asp)可改变字(单词)间的标准间隔。其默认值 normal 与设置值为 0 是一样的。

接受一个正长度值(字之间的间隔就会增加)或负长度值(拉近)：

p.spread {word-spacing: 30px;}

p.tight {word-spacing: -0.5em;}

(9)字母间隔

letter-spacing 属性,修改的是字符或字母之间的间隔。

可取值包括所有长度。默认关键字是 normal（这与 letter-spacing:0 相同）。

输入的长度值会使字母之间的间隔增加或减少指定的量

(10)字符转换

[text-transform 属性](http://www.w3school.com.cn/cssref/pr_text_text-transform.asp)处理文本的大小写。有 4 个值：

none

uppercase

lowercase

capitalize

默认值 none 对文本不做任何改动，将使用源文档中的原有大小写。

uppercase 和 lowercase 将文本转换为全大写和全小写字符。

capitalize 只对每个单词的首字母大写。

(11)文本装饰:text-decoration 属性

有 5 个值：

none

underline

overline

line-through

blink

underline :对元素加下划线

overline在文本的顶端画一个上划线。

line-through 则在文本中间画一个贯穿线，等价于 HTML 中的 S 和 strike 元素。

blink 会让文本闪烁，类似于 Netscape 支持的颇招非议的 blink 标记。

none 值会关闭原本应用到一个元素上的所有装饰。

通常，无装饰的文本是默认外观，但也不总是这样。例如，链接默认地会有下划线。

如果希望所有超链接既有下划线，又有上划线，则规则如下：

a:link a:visited {text-decoration: underline overline;}

如果两个不同的装饰都与同一元素匹配，胜出规则的值会完全取代另一个值。

h2.stricken {text-decoration: line-through;}

h2 {text-decoration: underline overline;}

对于给定的规则，所有 class 为 stricken 的 h2 元素都只有一个贯穿线装饰，而没有下划线和上划线，因为 text-decoration 值会替换而不是累积起来。

(12)处理空白符

[**white-space** 属性](http://www.w3school.com.cn/cssref/pr_text_white-space.asp)会影响到用户代理对源文档中的空格、换行和 tab 字符的处理。

默认的 XHTML 处理已经完成了空白符处理：会把所有空白符合并为一个空格。

以下标记在 Web 浏览器中显示时，各个字之间只会显示一个空格，同时忽略元素中的换行：

<p>This paragraph has many

spaces in it.</p>

可以用以下声明显式地设置这种默认行为：

p {white-space: normal;}

告诉浏览器按照平常的做法去处理：丢掉多余的空白符。

如给定这个值，换行字符(回车)会转换为空格，一行中多个空格的序列也会转换为一个空格。

1)值 pre

若将 white-space 设为 pre，，空白符不会被忽略。

注意：经测试，IE 7 以及更早版本的浏览器不支持该值

2)值 nowrap

会防止元素中的文本换行，除非使用了一个 br 元素。

3)值 pre-wrap 和 pre-line

pre-wrap：文本会保留空白符序列，但是文本行会正常地换行。源文本中的行分隔符以及生成的行分隔符也会保留。

pre-line：会像正常文本中一样合并空白符序列，但保留换行符。

IE7 和 FireFox2.0不支持pre-wrap 和 pre-line。

|  |  |  |  |
| --- | --- | --- | --- |
| **值** | **空白符** | **换行符** | **自动换行** |
| pre-line | 合并 | 保留 | 允许 |
| normal | 合并 | 忽略 | 允许 |
| nowrap | 合并 | 忽略 | 不允许 |
| pre | 保留 | 保留 | 不允许 |
| pre-wrap | 保留 | 保留 | 允许 |

(13)文本方向

[direction 属性](http://www.w3school.com.cn/cssref/pr_text_direction.asp)影响块级元素中文本的书写方向、表中列布局的方向、内容水平填充其元素框的方向、以及两端对齐元素中最后一行的位置。

注释：对于行内元素，只有当 [unicode-bidi 属性](http://www.w3school.com.cn/cssref/pr_unicode-bidi.asp)设置为 embed 或 bidi-override 时才会应用 direction 属性。

两个值：ltr (默认 从左到右)和 rtl(从右到左)。

**CSS 文本属性**

|  |  |
| --- | --- |
| **属性** | **描述** |
| [color](http://www.w3school.com.cn/cssref/pr_text_color.asp) | 设置文本颜色 |
| [direction](http://www.w3school.com.cn/cssref/pr_text_direction.asp) | 设置文本方向。 |
| [line-height](http://www.w3school.com.cn/cssref/pr_dim_line-height.asp) | 设置行高。 |
| [letter-spacing](http://www.w3school.com.cn/cssref/pr_text_letter-spacing.asp) | 设置字符间距。 |
| [text-align](http://www.w3school.com.cn/cssref/pr_text_text-align.asp) | 对齐元素中的文本。 |
| [text-decoration](http://www.w3school.com.cn/cssref/pr_text_text-decoration.asp) | 向文本添加修饰。 |
| [text-indent](http://www.w3school.com.cn/cssref/pr_text_text-indent.asp) | 缩进元素中文本的首行。 |
| text-shadow | 设置文本阴影。CSS2 包含该属性，但是 CSS2.1 没有保留该属性。 |
| [text-transform](http://www.w3school.com.cn/cssref/pr_text_text-transform.asp) | 控制元素中的字母。 |
| unicode-bidi | 设置文本方向。 |
| [white-space](http://www.w3school.com.cn/cssref/pr_text_white-space.asp) | 设置元素中空白的处理方式。 |
| [word-spacing](http://www.w3school.com.cn/cssref/pr_text_word-spacing.asp) | 设置字间距。 |

1. **CSS 字体**

定义文本的字体系列、大小、加粗、风格（如斜体）和变形（如小型大写字母）。

16.1 CSS 字体系列

* 通用字体系列 - 拥有相似外观的字体系统组合（比如 "Serif" 或 "Monospace"）
* 特定字体系列 - 具体的字体系列（比如 "Times" 或 "Courier"）

5 种通用字体系列：

Serif 字体

Sans-serif 字体

Monospace 字体

Cursive 字体

Fantasy 字体

16.2 指定字体系列:font-family 属性 定义文本的字体系列

(1)使用通用字体系列

如果希望文档使用一种 sans-serif 字体，但并不关心是哪一种字体：

body {font-family: sans-serif;}

用户代理就会从 sans-serif 字体系列中选择一个字体（如 Helvetica），并将其应用到 body 元素。因为有继承，这种字体选择还将应用到 body 元素中包含的所有元素指定字体系列

(2)指定字体系列

1)h1 {font-family: Georgia;}

如果用户代理上没有安装 Georgia 字体，就只能使用用户代理的默认字体来显示 h1 元素。2)可以通过结合特定字体名和通用字体系列来解决这个问题：

h1 {font-family: Georgia, serif;}

如果读者没有安装 Georgia，但安装了 Times 字体（serif 字体系列中的一种字体），用户代理就可能对 h1 元素使用 Times。

3)建议在所有 font-family 规则中都提供一个通用字体系列。

4) p {font-family: Times, TimesNR, 'New Century Schoolbook',

Georgia, 'New York', serif;}

使用引号:当字体名中有一个或多个空格（比如 New York），或者字体名包括 # 或 $ 之类的符号。单引号或双引号都可以接受。

若把font-family 属性放在HTML的style属性中，则需要使用该属性本身未使用的那种引号：

<p style="font-family: Times, TimesNR, 'New Century Schoolbook', Georgia,

'New York', serif;">...</p>

16.3字体风格： font-style

值：normal - 文本正常显示

italic - 文本斜体显示

oblique - 文本倾斜显示

italic是一种简单的字体风格，对每个字母的结构有一些小改动，来反映变化的外观。

倾斜（oblique）文本则是正常竖直文本的一个倾斜版本。

在 web 浏览器中看上去完全一样。

16.4字体变形:[font-variant 属性](http://www.w3school.com.cn/cssref/pr_font_font-variant.asp)可以设定小型大写字母。

采用不同大小的大写字母。

16.5字体加粗：font-weight

bold：粗体。

关键字 100 ~ 900指定了 9 级加粗度。100 最细，900 对应最粗。400 等价于 normal， 700 等价于 bold。

加粗设置为 bolder，浏览器会设置比所继承值更粗的一个字体加粗。

lighter 会导致浏览器将加粗度下移。

16.6字体大小：font-size

请始终使用正确的 HTML 标题，如使用 <h1> - <h6> 来标记标题，使用 <p> 来标记段落。

1)绝对值：将文本设置为指定的大小

不允许用户在所有浏览器中改变文本大小（不利于可用性）

绝对大小在确定了输出的物理尺寸时很有用

相对大小： 相对于周围的元素来设置大小

允许用户在浏览器改变文本大小

注意：如果没有规定字体大小，普通文本（如段落）的默认16 像素 (16px=1em)。

例:

h1 {font-size:60px;}

h2 {font-size:40px;}

p {font-size:14px;}

在 Firefox, Chrome, Safari 中，可重新调整以上例子的文本大小，但IE中不行。

1. em

要避免在 I E中无法调整文本的问题，许多开发者使用 em 单位代替 pixels。

1em 等于当前的字体尺寸。

若一个元素的 font-size =16 px，则对于该元素，1em =16 px。

在设置字体大小时，em 的值会相对于父元素的字体大小改变。

浏览器中默认的文本大小是 16 像素。因此 1em 的默认尺寸是 16 像素。

公式：pixels/16=em

（注：16 等于父元素的默认字体大小，假设父元素的 font-size 为 20px，那么公式需改为：pixels/20=em）

* 实例

h1 {font-size:3.75em;} /\* 60px/16=3.75em \*/

h2 {font-size:2.5em;} /\* 40px/16=2.5em \*/

p {font-size:0.875em;} /\* 14px/16=0.875em \*/

不幸的是，在 IE 中仍存在问题。在重设文本大小时，会比正常的尺寸更大或更小。

3）结合使用百分比和 EM

所有浏览器中均有效的方案是为 body 元素(父元素)以百分比设置默认的 font-size 值：

实例

body {font-size:100%;}

h1 {font-size:3.75em;}

h2 {font-size:2.5em;}

p {font-size:0.875em;}

* **CSS 字体属性**

|  |  |
| --- | --- |
| **属性** | **描述** |
| [font](http://www.w3school.com.cn/cssref/pr_font_font.asp) | 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。 |
| [font-family](http://www.w3school.com.cn/cssref/pr_font_font-family.asp) | 设置字体系列。 |
| [font-size](http://www.w3school.com.cn/cssref/pr_font_font-size.asp) | 设置字体的尺寸。 |
| [font-size-adjust](http://www.w3school.com.cn/cssref/pr_font_font-size-adjust.asp) | 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。） |
| [font-stretch](http://www.w3school.com.cn/cssref/pr_font_font-stretch.asp) | 对字体进行水平拉伸。（CSS2.1 已删除该属性。） |
| [font-style](http://www.w3school.com.cn/cssref/pr_font_font-style.asp) | 设置字体风格。 |
| [font-variant](http://www.w3school.com.cn/cssref/pr_font_font-variant.asp) | 以小型大写字体或者正常字体显示文本。 |
| [font-weight](http://www.w3school.com.cn/cssref/pr_font_weight.asp) | 设置字体的粗细。 |

1. CSS 链接

17.1设置链接的样式

例如 color, font-family, background。

能够根据它们所处的状态来设置它们的样式。

四种状态：

a:link - 普通的、未被访问的链接

a:visited - 用户已访问的链接

a:hover - 鼠标指针位于链接的上方

a:active - 链接被点击的时刻

实例

a:link {color:#FF0000;} /\* 未被访问的链接 \*/

a:visited {color:#00FF00;} /\* 已被访问的链接 \*/

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

a:active {color:#0000FF;} /\* 正在被点击的链接 \*/

17.2次序规则：

a:hover 必须位于 a:link 和 a:visited 之后

a:active 必须位于 a:hover 之后

17.3常见的链接样式

(1)文本修饰：text-decoration 属性大多用于去掉链接中的下划线：

实例

a:link {text-decoration:none;}

a:visited {text-decoration:none;}

a:hover {text-decoration:underline;}

a:active {text-decoration:underline;}

(2)背景色:background-color 属性规定链接的背景色：

实例

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

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

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

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

<http://www.w3school.com.cn/tiy/t.asp?f=css_link_advanced>

1. CSS 列表

允许你放置、改变列表项标志，或者将图像作为列表项标志

18.1 列表类型

1)例如，无序列表中，列表项的标志 (marker) 是出现在各列表项旁边的圆点

有序列表中，标志可能是字母、数字或另外某种计数体系中的一个符号。

要修改标志类型，可用属性 [list-style-type](http://www.w3school.com.cn/cssref/pr_list-style-type.asp)：

ul {list-style-type : square}

把无序列表中的列表项标志设置为方块。

2)列表项图像

ul li {list-style-image : url(xxx.gif)}

3) 列表标志位置

标志出现在列表项内容之外还是内容内部。这是利用 [list-style-position](http://www.w3school.com.cn/cssref/pr_list-style-position.asp) 完成的。

4)简写列表样式

可将以上 3 个列表样式属性合并为一个方便的属性：[list-style](http://www.w3school.com.cn/cssref/pr_list-style.asp)，就像这样：

li {list-style : url(example.gif) square inside}

值可以按任何顺序列出，值都可以忽略。只要提供了一个值，其它的就会填入其默认值。

<http://www.w3school.com.cn/tiy/t.asp?f=csse_list-style-type2>

<http://www.w3school.com.cn/tiy/t.asp?f=csse_list-style-type_all>

<http://www.w3school.com.cn/tiy/t.asp?f=csse_list-style-image> //[将图像作为列表项标记](http://www.w3school.com.cn/tiy/t.asp?f=csse_list-style-image)

<http://www.w3school.com.cn/tiy/t.asp?f=csse_list-style>

|  |  |
| --- | --- |
| **属性** | **描述** |
| [list-style](http://www.w3school.com.cn/cssref/pr_list-style.asp) | 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](http://www.w3school.com.cn/cssref/pr_list-style-image.asp) | 将图象设置为列表项标志。 |
| [list-style-position](http://www.w3school.com.cn/cssref/pr_list-style-position.asp) | 设置列表中列表项标志的位置。 |
| [list-style-type](http://www.w3school.com.cn/cssref/pr_list-style-type.asp) | 设置列表项标志的类型。 |
| marker-offset |  |

1. CSS 表格

19.1表格边框:border 属性。

table, th, td{

border: 1px solid blue;

}

上例中的表格具有双线条边框。是由于 table、th 以及 td 元素都有独立的边框。

19.2折叠边框:border-collapse 属性设置是否将表格边框折叠为单一边框：

table{

border-collapse:collapse;

}

table,th, td{

border: 1px solid black;

}

19.3表格宽度和高度

例子将表格宽度设置为 100%，同时将 th 元素的高度设置为 50px：

table{

width:100%;

}

th{

height:50px;

}

19.4表格文本对齐

text-align 属性设置水平对齐方式，比如左对齐、右对齐或者居中：

td

{

text-align:right;

}

vertical-align 属性设置垂直对齐方式，比如顶部对齐、底部对齐或居中对齐：

td{

height:50px;

vertical-align:bottom;

}

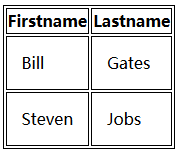
19.5表格内边距

如需控制表格中内容与边框的距离，请为 td 和 th 元素设置 padding 属性：

td{

padding:15px;

}



19.6 表格颜色

table, td, th{

border:1px solid green;

}

th{

background-color:green;

color:white;

}

|  |  |
| --- | --- |
| **属性** | **描述** |
| [border-collapse](http://www.w3school.com.cn/cssref/pr_tab_border-collapse.asp) | 设置是否把表格边框合并为单一的边框。 |
| [border-spacing](http://www.w3school.com.cn/cssref/pr_tab_border-spacing.asp) | 设置分隔单元格边框的距离。 |
| [caption-side](http://www.w3school.com.cn/cssref/pr_tab_caption-side.asp) | 设置表格标题的位置。 |
| [empty-cells](http://www.w3school.com.cn/cssref/pr_tab_empty-cells.asp) | 设置是否显示表格中的空单元格。 |
| [table-layout](http://www.w3school.com.cn/cssref/pr_tab_table-layout.asp) | 设置显示单元、行和列的算法。 |

<http://www.w3school.com.cn/tiy/t.asp?f=csse_table_fancy> //制作一个漂亮的表格

<http://www.w3school.com.cn/tiy/t.asp?f=csse_table_empty-cells> //显示表格中的空单元

1. CSS 轮廓

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

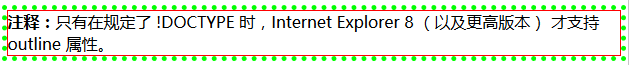
CSS outline 属性规定元素轮廓的样式、颜色和宽度。

20.2实例：

(1)在元素周围画线

http://www.w3school.com.cn/tiy/t.asp?f=csse\_outline

使用outline属性在元素周围画一条线。



(2)设置轮廓的颜色

http://www.w3school.com.cn/tiy/t.asp?f=csse\_outline-color

(3)设置轮廓的样式

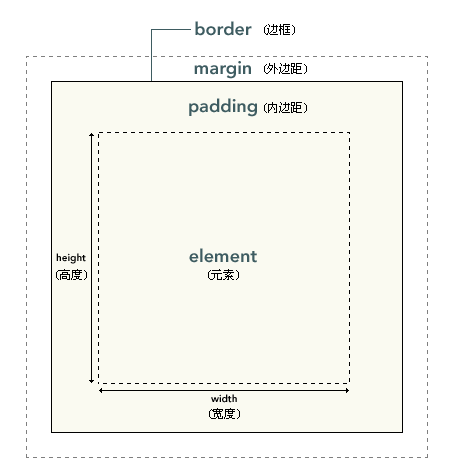
http://www.w3school.com.cn/tiy/t.asp?f=csse\_outline-style

(4)设置轮廓的宽度

http://www.w3school.com.cn/tiy/t.asp?f=csse\_outline-width

# 三．CSS 框模型

1. CSS 框模型概述



21.1 外边距默认是透明的

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。

可通过将元素的 margin 和 padding 设置为零来覆盖这些浏览器样式。

\* {

margin: 0;

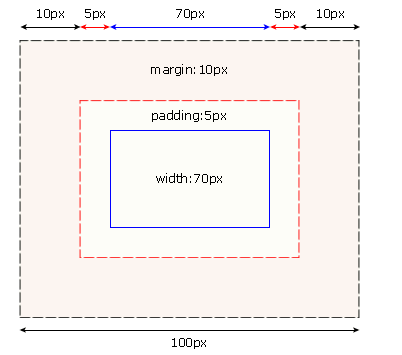
padding: 0;

}

在 CSS 中，width 和 height 指的是内容区域的宽度和高度

21.2

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：



提示：外边距可以是负值。

21.3浏览器兼容性

IE5.X 和 6 在怪异模式中使用自己的非标准模型。width是内容、内边距和边框的宽度总和。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。

* element : 元素。
* padding : 内边距，填充。
* border : 边框。
* margin : 外边距，空白或空白边。

1. CSS 内边距

22.1 CSS padding 属性

接受长度值或百分比值，但不允许使用负值。

h1 {padding: 10px;}

h1 {padding: 10px 0.25em 2ex 20%;} //上、右、下、左

22.2 单边内边距属性

[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)

h1 {

padding-top: 10px;

padding-right: 0.25em;

padding-bottom: 2ex;

padding-left: 20%;

}

22.3内边距的百分比数值

百分数值是相对于其父元素的 width 计算的

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：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

1. CSS 边框

边框有 3 个方面：宽度、样式，以及颜色

23.1边框与背景

* 边框绘制在“元素的背景之上”。 “间断的”边框（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。

23.2边框的样式：border-style

(1)如果没有样式，将根本没有边框

例如，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;}

(2)定义多种样式

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

//实线上边框、点线右边框、虚线下边框和一个双线左边框。

top-right-bottom-left 的顺序

(3)定义单边样式

[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(后面的优先级高)。

23.3边框的宽度： border-width

(1)可以指定长度值，比如 2px 或 0.1em；

3 个关键字：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。

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

或者：

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

(2)定义单边宽度

top-right-bottom-left

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

也可简写为（*值复制*）：

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

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)

23.4没有边框

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

不仅边框的样式没有了，其宽度也会变成 0。边框消失了

由于 border-style 的默认值是 none，如果没有声明样式.

因此，如果您希望边框出现，就必须声明一个边框样式。

23.5边框的颜色: border-color

(1)一次可以接受最多 4 个颜色值。

颜色值:命名颜色，十六进制和 RGB 值：

p {

border-style: solid;

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

}

(2)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;

}

(3)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。

如果元素没有任何文本，假设是一个表格，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 color 可以继承）。父元素很可能是 body、div 或另一个 table。

(4)定义单边颜色

border-top-color

border-right-color

border-bottom-color

border-left-color

(5)透明边框

边框颜色值 transparent。创建有宽度的不可见边框。

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

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

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

a:link, a:visited {

border-style: solid;

border-width: 5px;

border-color: transparent;

}

a:hover {border-color: gray;}//鼠标放上去显示边框

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。

|  |  |
| --- | --- |
| **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。 |

1. CSS 外边距

24.1设置外边距会在元素外创建额外的“空白”。

margin 属性，接受任何长度单位、百分数值甚至负值。

24.2 CSS margin 属性

margin 可设置为 auto。

h1 {margin : 0.25in;}

p {margin : 10%;}//百分数是相对于父元素的 width 计算的

margin: top right bottom left

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。

但实际中，浏览器提供了预定的样式。例如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。

24.3值复制

输入一些重复的值：

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

等价的：

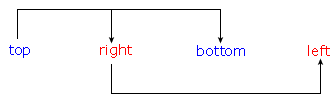
p {margin: 0.5em 1em;}

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。



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 \*/

margin-top

margin-right

margin-bottom

margin-left

24.4 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。

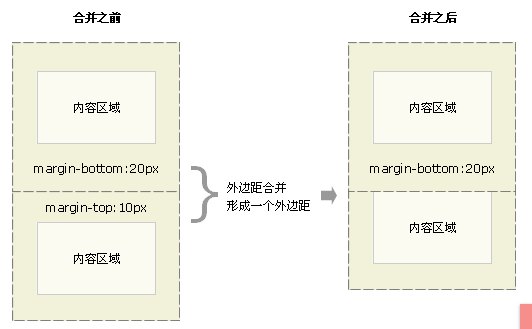
1. CSS 外边距合并

当两个垂直外边距相遇时，将形成一个外边距。

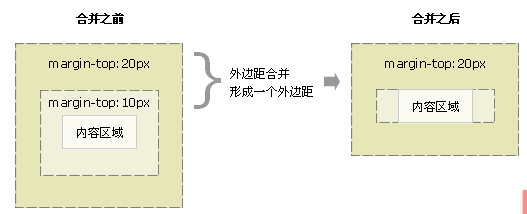
合并后的外边距的高度等于两个发生合并的外边距的高度中的较大者。

25.1外边距合并

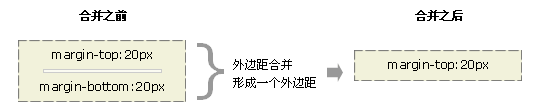
(1)



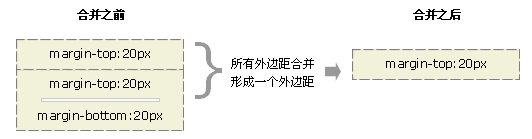
(2)当一个元素包含在另一个元素中时，它们的上和/或下外边距也会发生合并。



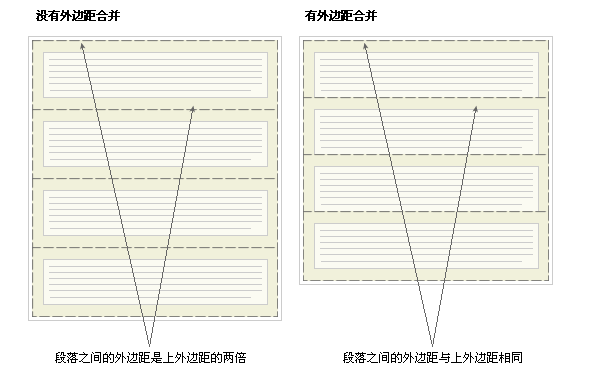
(3)假设有一个空元素，有外边距，但没有边框或填充。在这种情况下，上外边距与下外边距就碰到了一起，它们会发生合并：



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：(有点像俄罗斯方块)



(4)



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。

# 四 CSS 定位 (Positioning)

1. CSS 定位和浮动

相对于其正常位置，或父元素、另一个元素甚至浏览器窗口本身的位置。

一切皆为框

块级元素: div、h1 或 p 元素。“块框”。

行内元素: span 和 strong，内容显示在行中，即“行内框”。

用 display 属性改变生成框的类型。

display = block，可让行内元素（比如 <a> 元素）表现得像块级元素。

display = none，让生成的元素没有框。该框及其内容就不再显示，不占用文档中的空间。

把文本添加到块级元素（如 div）开头。即使没有把这些文本定义为段落，它也会被当作段落(块级元素)对待

<div>

some text

<p>Some more text.</p>

</div>

无名块框，因为它不与专门定义的元素相关联。

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。

26.1 CSS 定位机制

普通流、浮动和绝对定位

默认在普通流中定位。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。

26.2 CSS position 属性

(1)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。

(2)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。

(3)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。

(4)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。

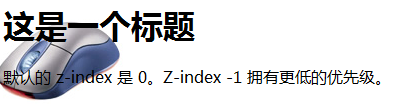
提示：相对定位实际上被看作普通流定位模型的一部分，因为元素的位置相对于它在普通流中的位置。

26.3例子

<http://www.w3school.com.cn/tiy/t.asp?f=csse_overflow> 使用滚动条来显示元素内溢出的内容

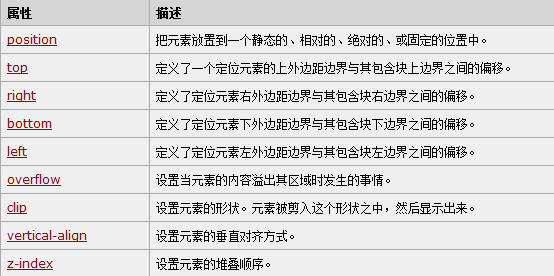
<http://www.w3school.com.cn/tiy/t.asp?f=csse_pos_overflow_hidden> 隐藏溢出元素中溢出内容

<http://www.w3school.com.cn/tiy/t.asp?f=csse_zindex2> Z-index



<http://www.w3school.com.cn/tiy/t.asp?f=csse_vertical-align> 垂直排列图象





1. CSS 相对定位

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。

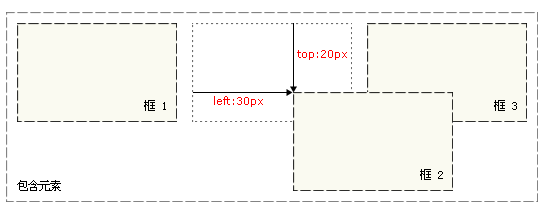
#box\_relative {

position: relative;

left: 30px;

top: 20px;

}



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。

1. CSS 绝对定位

从文档流完全删除，并相对于其包含块定位(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。

绝对定位使元素的位置与文档流无关，因此不占据空间。

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：

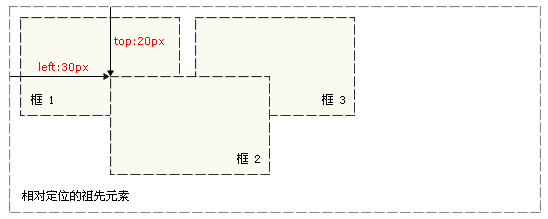
#box\_relative {

position: absolute;

left: 30px;

top: 20px;

}



绝对定位的元素的位置相对于最近的已定位祖先元素，如果没有，则相对于最初的包含块(画布或 HTML 元素)。

提示：因为绝对定位的框与文档流无关，所以可以覆盖页面上的其它元素。可通过设置 z-index 属性来控制这些框的堆放次序。

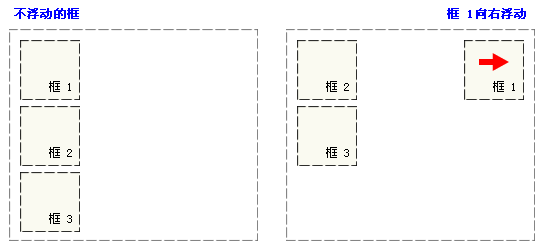
1. CSS 浮动

浮动框可向左或向右移动，直到它的外边缘碰到包含框或另一个浮动框的边框为止。

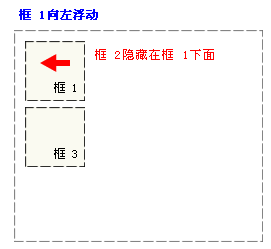
浮动框不在文档的普通流中，文档普通流中的块框表现得就像浮动框不存在一样。

29.1 CSS 浮动

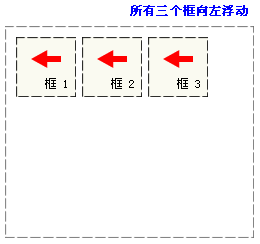
请看下图，当把框 1 向右浮动时，它脱离文档流并且向右移动，直到它的右边缘碰到包含框的右边缘



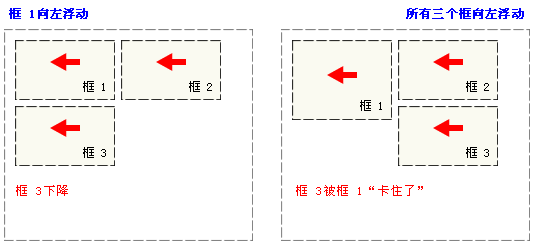
再请看下图，当框 1 向左浮动时，它脱离文档流并且向左移动，直到它的左边缘碰到包含框的左边缘。因为它不再处于文档流中，所以它不占据空间，实际上覆盖住了框 2，使框 2 从视图中消失。



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”：

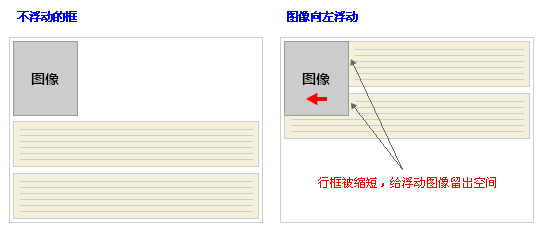


29.2 CSS float 属性

行框和清理

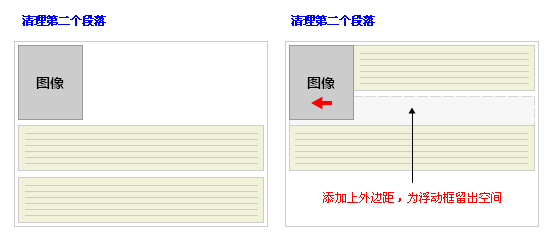
浮动框旁边的行框被缩短，从而给浮动框留出空间，行框围绕浮动框。

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">

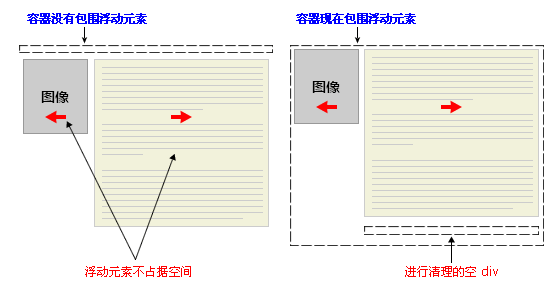
<img src="news-pic.jpg" />

<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：



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。

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，对页脚的 div 进行了清理，而页脚上面的三个 div 都向左浮动。

29.3 CSS clear 属性

实例：图像的左侧和右侧均不允许出现浮动元素：

img

{

float:left;

clear:both;

}

定义和用法：规定元素的哪一侧不允许其他浮动元素。

说明：如果声明为左边或右边清除，会使元素的上外边框边界刚好在该边上浮动元素的下外边距边界之下。

例子: <http://www.w3school.com.cn/css/css_positioning_floating.asp>

# 五 CSS 选择器

1. CSS 元素选择器

30.1 CSS 元素选择器

设置HTML样式,选择器常是某个 HTML 元素，如 p、h1、em、a，甚至可以是 html 本身：

html {color:black;}

h1 {color:blue;}

h2 {color:silver;}

30.2 类型选择器

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

h1 {font-family: sans-serif;}

(1) 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>

(2) note.css：

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;

}



1. CSS 分组

31.1选择器分组

h2, p {color:gray;}

31.2通配符选择器

\* {color:red;}

31.3声明分组

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

或h1 {

font: 28px Verdana;

color: blue;

background: red;

}

31.4结合选择器和声明的分组

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

color:gray;

background: white;

padding: 10px;

border: 1px solid black;

font-family: Verdana;

}

1. CSS 类选择器详解

32.1 CSS 类选择器

32.2结合元素选择器

类选择器可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;}

32.3 CSS 多类选择器

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>

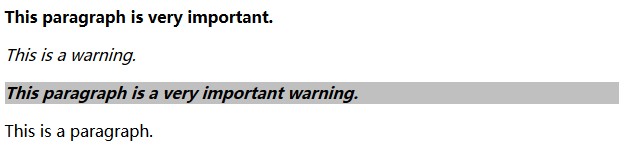
这两个词的顺序无关紧要。

假设 class 为 importat 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;}



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>

1. CSS ID 选择器详解

33.1 CSS ID 选择器

(1)# 号 -棋盘号或井号。

\*#intro {font-weight:bold;}

#intro {font-weight:bold;} //可忽略通配选择器

(2)类选择器还是 ID 选择器？

区别 1：D 选择器会使用一次

区别 2：不能使用 ID 词列表

ID 选择器不能结合使用，因为 ID 属性不允许有以空格分隔的词列表。

区别 3：ID 能包含更多含义

#mostImportant {color:red; background:yellow;}

该规则会与以下各个元素匹配（这些元素不能在同一个文档中同时出现，因它们都有相同ID 值）：

<h1 id="mostImportant">This is important!</h1>

<em id="mostImportant">This is important!</em>

<ul id="mostImportant">This is important!</ul>

(3)区分大小写

类选择器和 ID 选择器可能是区分大小写的。这取决于文档的语言。

HTML 和 XHTML 将类和 ID 值定义为区分大小写

1. CSS 属性选择器详解

可以根据元素的属性及属性值来选择元素。

34.1 简单属性选择

(1)把包含标题（title）的所有元素变为红色，可写作：

\*[title] {color:red;}

(2)只对有 href 属性的锚（a 元素）应用样式：

a[href] {color:red;}

(3) 将同时有 href 和 title 属性的 HTML 超链接的文本设置为红色：

a[href][title] {color:red;}

(4)对所有带有 alt 属性的图像应用样式：

img[alt] {border: 5px solid red;}

(5)为 XML 文档使用属性选择器

XML 语言主张要针对元素和属性的用途指定元素名和属性名。

如果我们想选择有 moons 属性的所有 planet 元素，使之显示为红色，以便能更关注有 moons 的行星，就可以这样写：

planet[moons] {color:red;}

<planet>Venus</planet> //非红

<planet moons="1">Earth</planet> //红

<planet moons="2">Mars</planet> //红

(6)根据具体属性值选择

例如，假设希望将指向 Web 服务器上某个指定文档的超链接变成红色：

a[href="http://www.w3school.com.cn/about\_us.asp"] {color: red;}

a[href="http://www.w3school.com.cn/"][title="W3School"] {color: red;}

1)XML 语言也可利用这种方法来设置样式。

planet[moons="1"] {color: red;}

<planet>Venus</planet>

<planet moons="1">Earth</planet> //红色

<planet moons="2">Mars</planet>

3)属性与属性值必须完全匹配

如果属性值包含用空格分隔的值列表，匹配就可能出问题。

<p class="important warning">This paragraph is a very important warning.</p>

如果写成 p[class="important"]，那么这个规则不能匹配示例标记。

必须这样写：p[class="important warning"] {color: red;}

(7)根据部分属性值选择

如果需要根据属性值中的词列表的某个词进行选择，则需要使用波浪号（~）。

假设您想选择 class 属性中包含 important 的元素：

p[class~="important"] {color: red;}

1)部分值属性选择器与点号类名记法的区别

p.important 和 p[class="important"]等价

"~="能用于任何属性，而不只是 class。

例如，可以有一个包含大量图像的文档，其中只有一部分是图片。对此，可以使用一个基于 title 文档的部分属性选择器，只选择这些图片：

img[title~="Figure"] {border: 1px solid gray;}

(8)子串匹配属性选择器

类型 描述

[abc^="def"] 选择 abc 属性值以 "def" 开头的所有元素

[abc$="def"] 选择 abc 属性值以 "def" 结尾的所有元素

[abc\*="def"] 选择 abc 属性值中包含子串 "def" 的所有元素

例如希望对指向 W3School 的所有链接应用样式，不必为所有这些链接指定 class，再根据这个类编写样式：

a[href\*="w3school.com.cn"] {color: red;}

(9)特定属性选择类型

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

lang 属性等于 en 或以 en- 开头的所有元素。

前三个元素将被选中：

<p lang="en">Hello!</p>

<p lang="en-us">Greetings!</p>

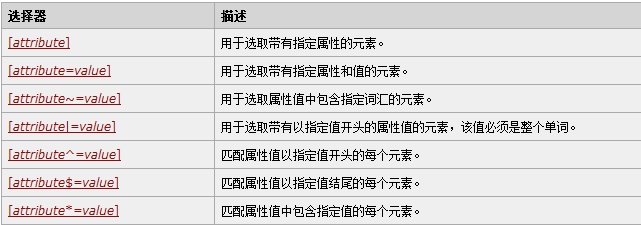
<p lang="en-au">G'day!</p>

<p lang="fr">Bonjour!</p>

<p lang="cy-en">Jrooana!</p>

假设一系列图片，文件名都形如 figure-1.jpg 和 figure-2.jpg：

img[src|="figure"] {border: 1px solid gray;}



1. CSS 后代选择器

包含选择器。可选择作为某元素后代的元素。

35.1根据上下文选择元素

h1 em {color:red;}

(1)语法解释

选择器之间的空格是一种结合符（combinator）。每个空格结合符可解释为“... 在 ... 找到”、“... 作为 ... 的一部分”、“... 作为 ... 的后代”，但是要求必须从右向左读选择器。

(2)具体应用

假设有一个文档，其中有一个边栏，还有一个主区。边栏的背景为蓝色，主区的背景为白色，两个区都包含链接列表。不能把所有链接都设置为蓝色，因为这样一来边栏中的蓝色链接都无法看到。

解决方法是使用后代选择器。在这种情况下，可以为包含边栏的 div 指定值为 sidebar 的 class 属性，并把主区的 class 属性值设置为 maincontent。然后编写以下样式：

div.sidebar {background:blue;}

div.maincontent {background:white;}

div.sidebar a:link {color:white;}

div.maincontent a:link {color:blue;}

有关后代选择器有一个易被忽视的方面，即两个元素之间的层次间隔可以是无限的。

例如，如果写作 ul em，这个语法就会选择从 ul 元素继承的所有 em 元素，而不论 em 的嵌套层次多深。

因此，ul em 将会选择以下标记中的所有 em 元素：

<ul>

<li>List item 1

<ol>

<li>List item 1-1</li>

<li>List item 1-2</li>

<li>List item 1-3

<ol>

<li>List item 1-3-1</li>

<li>List item <em>1-3-2</em></li>

<li>List item 1-3-3</li>

</ol>

</li>

<li>List item 1-4</li>

</ol>

</li>

<li>List item 2</li>

<li>List item 3</li>

</ul>