**一个 HTML 元素被CSS样式优先级**

**浏览器缺省设置<外部样式表<内部样式表（位于 <head> 标签内部）<内联样式（在 HTML 元素内部）**

**CSS的引入**

**外部样式表：可在任何编辑器中编辑，内容不得含有HTML标签，文件扩展名为.css**

**<head>**

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

**</head>**

**单个样式表**



**内联样式表**

**由于要将表现和内容混杂在一起，内联样式会损失掉样式表的许多优势。请慎用这种方法，例如当样式仅需要在一个元素上应用一次时。要使用内联样式，你需要在相关的标签内使用样式（style）属性。Style 属性可以包含任何 CSS 属性**

**多重样式的继承满足就近同名覆盖原则**

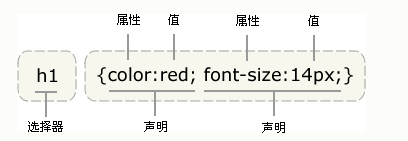


**CSS 语法**

CSS 规则由两个主要的部分构成：选择器，以及一条或多条声明

选择器通常是您需要改变样式的 HTML 元素。每条声明由一个属性和一个值组成。

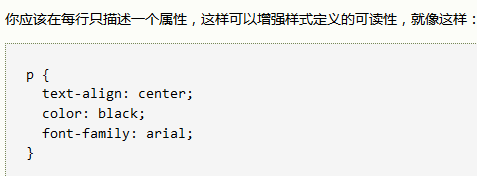
属性（property）是您希望设置的样式属性（style attribute）。每个属性有一个值。属性和值被冒号分开

**selector {property: value} **

对于一个属性多个值要加””

****

**多个属性值KV对用；分割最后一组分号可以省略**

** **

CSS 对大小写不敏感。不过存在一个例外：如果涉及到与 HTML 文档一起工作的话，class 和 id 名称对大小写是敏感的

## 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

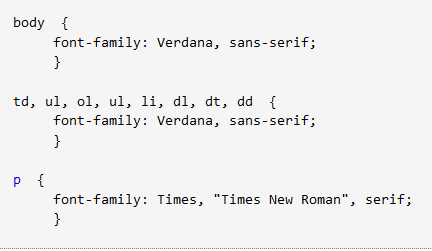
## 

## 根据 CSS，子元素从父元素继承属性

## 

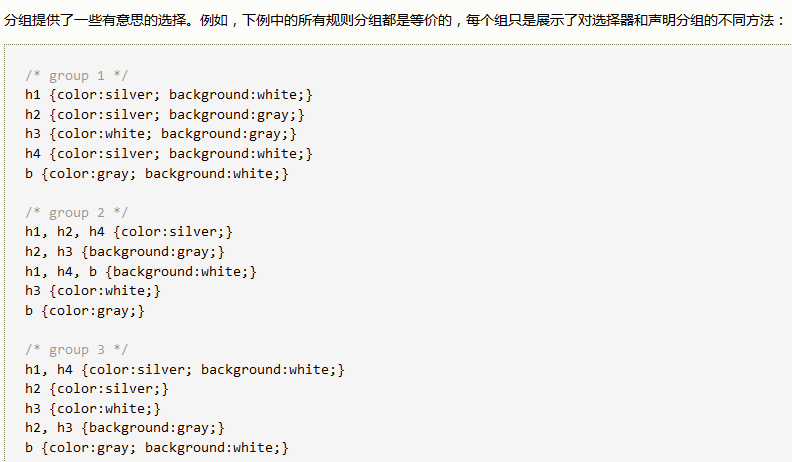
根据上面这条规则，站点的 body 元素将使用 Verdana 字体（假如访问者的系统中存在该字体的话）。

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

打断继承如果你不希望 "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



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



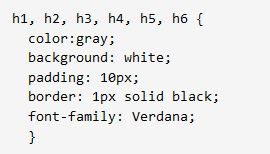
通配符选择器：\* {color:red;}

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：

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

结合选择器和声明的分组相结合



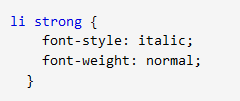
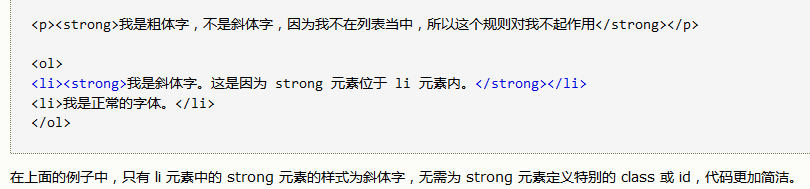
CSS选择器

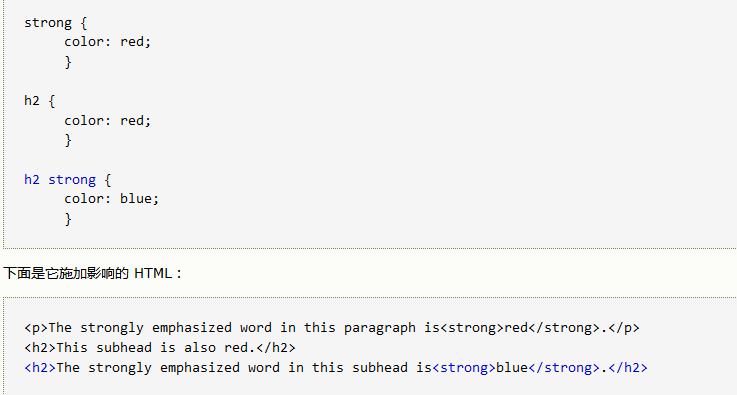
元素选择器（类型选择器）：文档的元素就是最基本的选择器。

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

CSS 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

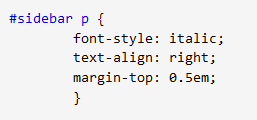
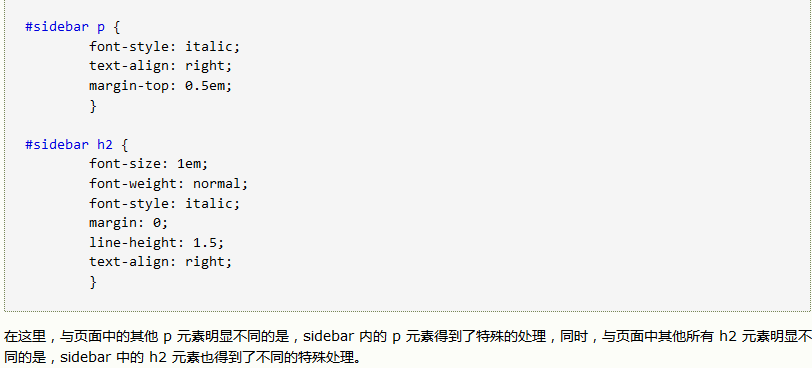


## Id 选择器：为标有特定 id 的 HTML 元素指定特定的样式（id在HTML文档中具有唯一性）

## 与类选择器一样，ID 选择器中可以忽略通配选择器

## 

**Id选择器与派生选择器**

****

不可以在内联元素 <span> 中嵌入 <p>

类选择器与ID选择器的选择

区别 1：与类不同，在一个 HTML 文档中，ID 选择器会使用一次，而且仅一次。

区别 2：不能使用 ID 词列表

不同于类选择器，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

类似于类，可以独立于元素来选择 ID。有些情况下，您知道文档中会出现某个特定 ID 值，但是并不知道它会出现在哪个元素上，所以您想声明独立的 ID 选择器。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

类选择器：**在 CSS 中，类选择器以一个点号显示，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**

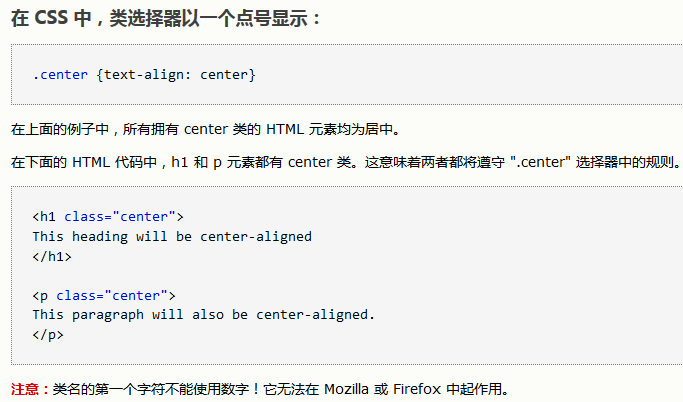
只有适当地标记文档后，才能使用这些选择器，所以使用这两种选择器通常需要先做一些构想和计划。要应用样式而不考虑具体设计的元素，最常用的方法就是使用类选择器

**对HTML的要求需要元素增加一个class属性并设置值作为标记**

**类选择器可以结合元素选择器使用**

****

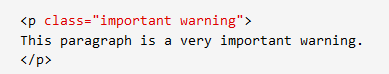
选择器现在会匹配 class 属性包含 important 的所有 p 元素，但是其他任何类型的元素都不匹配，不论是否有此 class 属性

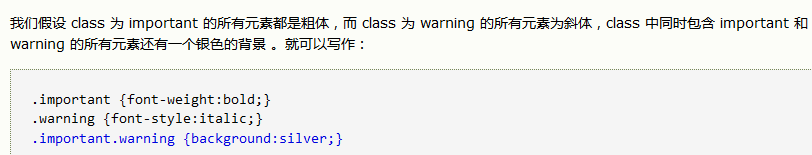


类选择器与派生选择器组合使用

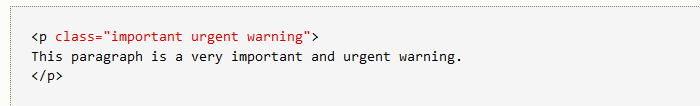
多类选择器

一个 class 值中可能包含一个词列表，各个词之间用空格分隔







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：

## 属性选择器：对带有指定属性的 HTML 元素设置样式

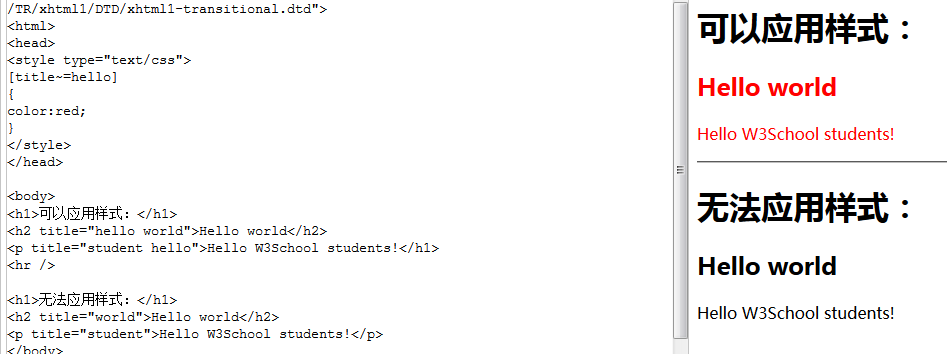
（IE7 和 IE8 才支持属性选择器。在 IE6 及更低的版本中，不支持属性选择）



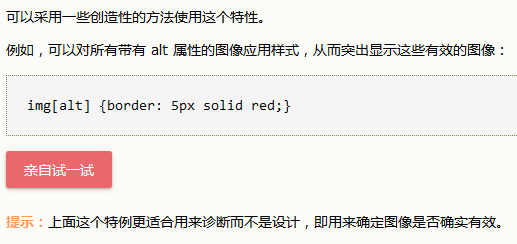
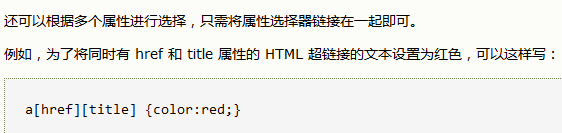
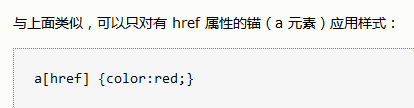
属性选择器的多个值

为属性值中含有空格分割字符串设定样式









根据具体的属性值选择

除了选择拥有某些属性的元素，还可以进一步缩小选择范围，只选择有特定属性值的元素

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，可以这样写：



与简单属性选择器类似，可以把多个属性-值选择器链接在一起来选择一个文档

属性与属性值必须完全匹配



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，可以用下面这个选择器做到这一点

### 

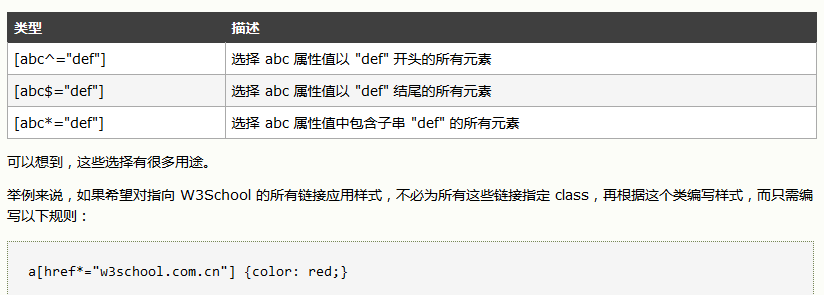
部分值属性选择器与点号类名记法的区别

p.important 和 p[class="important"] 应用到 HTML 文档时是等价的

"~=" 属性选择器呢能用于任何属性，而不只是 class

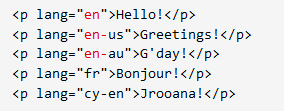
可以有一个包含大量图像的文档，其中只有一部分是图片。对此，可以使用一个基于 title 文档的部分属性选择器，只选择这些图片

子串匹配属性选择器

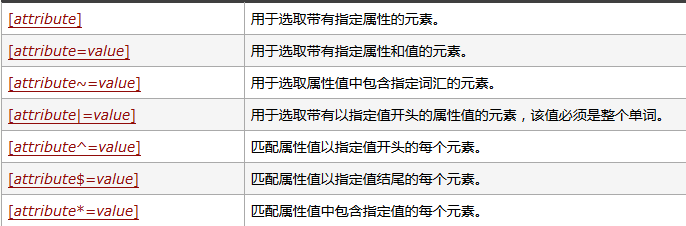


特定属性选择器



上面这个规则会选择 lang 属性等于 en 或以 en- 开头的所有元素。因此，以下示例标记中的前三个元素将被选中，而不会选择后两个元素：

一般来说，[att|="val"] 可以用于任何属性及其值。

假设一个 HTML 文档中有一系列图片，其中每个图片的文件名都形如 figure-1.jpg 和 figure-2.jpg。就可以使用以下选择器匹配所有这些图像：

CSS后代选择器（包含选择器）

根据上下行文选择元素

如果您希望只对 h1 元素中的所有后代 em 元素应用样式，可以这样写

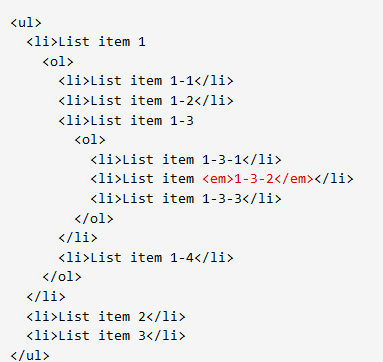


后代选择器的语法解释

在后代选择器中，规则左边的选择器一端包括两个或多个用空格分隔的选择器。选择器之间的空格是一种结合符（combinator）。每个空格结合符可以解释为“ “... 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。因此，h1 em 选择器可以解释为 “作为 h1 元素后代的任何 em 元素”。如果要从左向右读选择器，可以换成以下说法：“包含 em 的所有 h1 会把以下样式应用到该 em”。

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：



子元素选择器：**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**

**亲子关系**

**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**



后代选择器与子选择器结合使用



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。

兄弟选择器：**可选择紧接在另一元素后的元素，且二者有相同父元素。**



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”。

相邻兄弟结合符还可以结合其他结合符：



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。

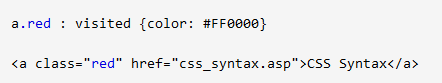
CSS伪类：**伪类用于向某些选择器添加特殊的效果**

伪类的语法



CSS 类也可与伪类搭配使用

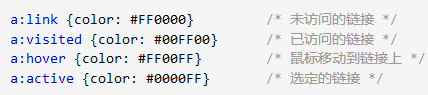




假如上面的例子中的链接被访问过，那么它将显示为红色。

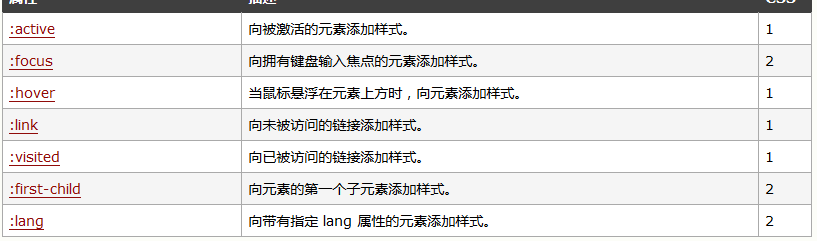
锚伪类

在支持 CSS 的浏览器中，链接的不同状态都可以不同的方式显示，这些状态包括：活动状态，已被访问状态，未被访问状态，和鼠标悬停状态。



提示

在 CSS 定义中，a:hover 必须被置于 a:link 和 a:visited 之后，才是有效的；在 CSS 定义中，a:active 必须被置于 a:hover 之后，才是有效的；伪类名称对大小写不敏感



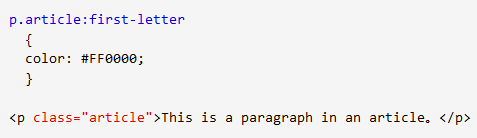
伪元素：**CSS 伪元素用于向某些选择器设置特殊效果**



CSS 类也可以与伪元素配合使用



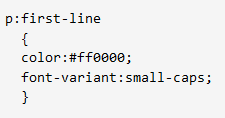
下面的例子会使所有 class 为 article 的段落的首字母变为红色



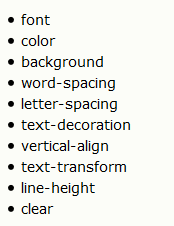
:first-line 伪元素

"first-line" 伪元素用于向文本的首行设置特殊样式。

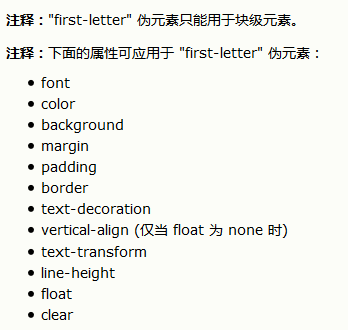
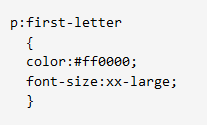
在下面的例子中，浏览器会根据 "first-line" 伪元素中的样式对 p 元素的第一行文本进行格式化：



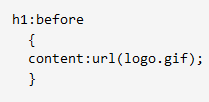
"first-line" 伪元素只能用于块级元素

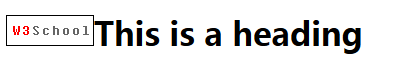


:first-letter 伪元素：伪元素用于向文本的首字母设置特殊样式

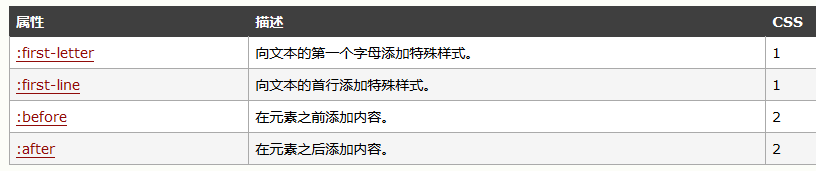


:before 伪元素：伪元素可以在元素的内容前面插入新内容





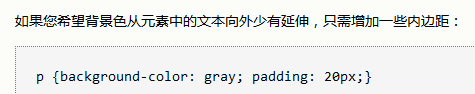
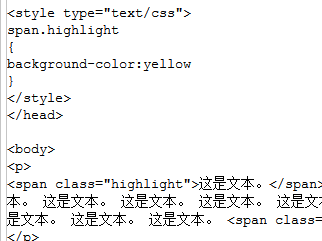
:after 伪元素：伪元素可以在元素的内容之后插入新内容



CSS背景：CSS 允许应用纯色作为背景，也允许使用背景图像创建相当复杂的效果。CSS 在这方面的能力远远在 HTML 之上

[background-color 属性](http://www.w3school.com.cn/cssref/pr_background-color.asp)为元素设置背景色。这个属性接受任何合法的颜色值

background-color 不能继承，其默认值是 transparent。transparent 有“透明”之意。也就是说，如果一个元素没有指定背景色，那么背景就是透明的，这样其祖先元素的背景才能可见



背景图像：要把图像放入背景，需要使用 [background-image 属性](http://www.w3school.com.cn/cssref/pr_background-image.asp)。background-image 属性的默认值是 none，表示背景上没有放置任何图像，background-image 也不能继承。事实上，所有背景属性都不能继承，一般的背景颜色在body上设置

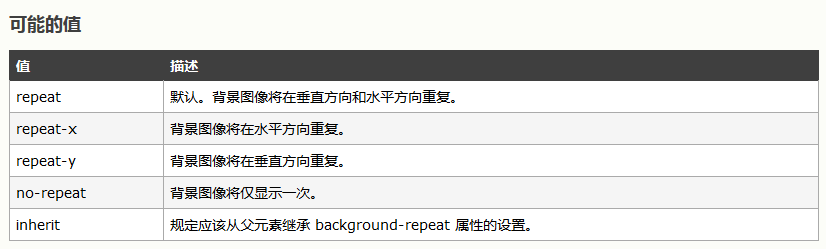


背景重复

默认地，背景图像将从一个元素的左上角开始，并在水平和垂直方向上重复

如果需要在页面上对背景图像进行平铺，可以使用 [background-repeat 属性](http://www.w3school.com.cn/cssref/pr_background-repeat.asp)

属性值 repeat 导致图像在水平垂直方向上都平铺，就像以往背景图像的通常做法一样。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



背景图像的定位[background-position 属性](http://www.w3school.com.cn/cssref/pr_background-position.asp)

为 background-position 属性提供值有很多方法。首先，可以使用一些关键字：top、bottom、left、right 和 center。通常，这些关键字会成对出现，不过也不总是这样。还可以使用长度值，如 100px 或 5cm，最后也可以使用百分数值。不同类型的值对于背景图像的放置稍有差异



CSS文本样式：**通过文本属性，可以改变文本的颜色、字符间距，对齐文本，装饰文本，对文本进行缩进，等等**

**缩进文本：**把 Web 页面上的段落的第一行缩进，这是一种最常用的文本格式化效果

CSS 提供了 [text-indent 属性](http://www.w3school.com.cn/cssref/pr_text_text-indent.asp)，该属性可以方便地实现文本缩进。通过使用 text-indent 属性，所有元素的第一行都可以缩进一个给定的长度，甚至该长度可以是负值。这个属性最常见的用途是将段落的首行缩进，下面的规则会使所有段落的首行缩进 5 em：

****

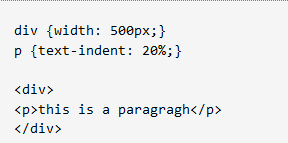
注意：一般来说，可以为所有块级元素应用 text-indent，但无法将该属性应用于行内元素，图像之类的替换元素上也无法应用 text-indent 属性。不过，如果一个块级元素（比如段落）的首行中有一个图像，它会随该行的其余文本移动。提示：如果想把一个行内元素的第一行“缩进”，可以用左内边距或外边距创造这种效果

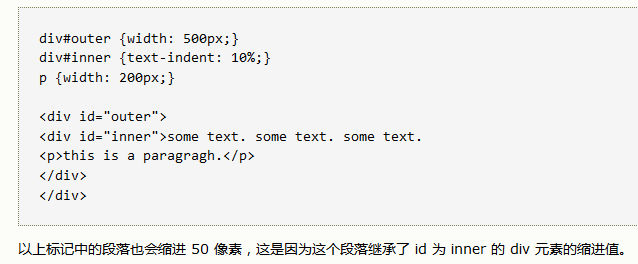
**负值的使用：达到悬挂的效果**

不过在为 text-indent 设置负值时要当心，如果对一个段落设置了负值，那么首行的某些文本可能会超出浏览器窗口的左边界。为了避免出现这种显示问题，建议针对负缩进再设置一个外边距或一些内边距

****

**使用百分比是针对父级元素的百分比**

在下例中，缩进值是父元素的 20%，即 100 个像素：

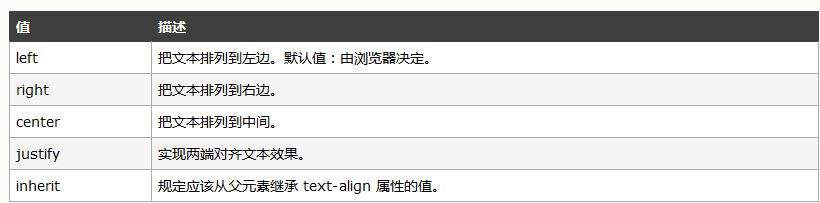
text-indent 属性可以继承

水平对其：[text-align](http://www.w3school.com.cn/cssref/pr_text_text-align.asp" \o "CSS text-align 属性) 是一个基本的属性，它会影响一个元素中的文本行互相之间的对齐方式

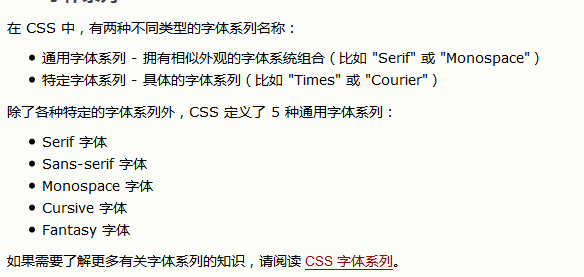
提示：将块级元素或表元素居中，要通过在这些元素上适当地设置左、右外边距来实现

text-align:center 与 <CENTER>

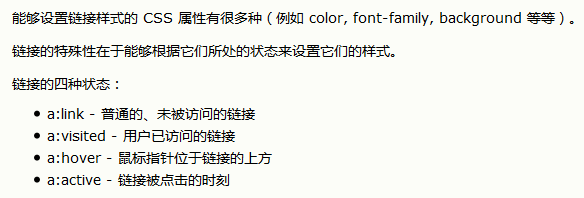
您可能会认为 text-align:center 与 <CENTER> 元素的作用一样，但实际上二者大不相同。<CENTER> 不仅影响文本，还会把整个元素居中。text-align 不会控制元素的对齐，而只影响内部内容。元素本身不会从一段移到另一端，只是其中的文本受影响。

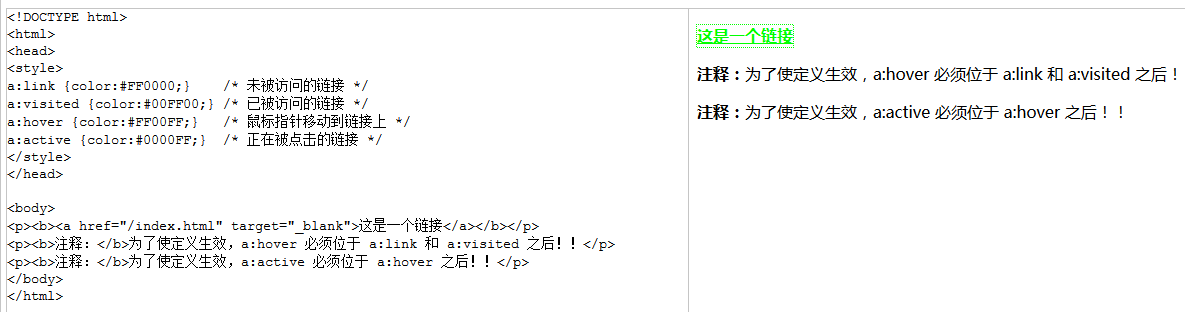
****

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，但不同的浏览器的解析效果会存在差别

**CSS 字体属性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）**

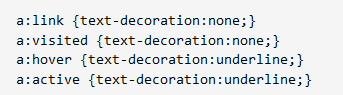
**Css链接样式**



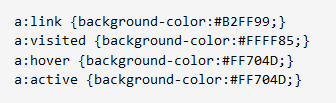
****

常见的连接样式

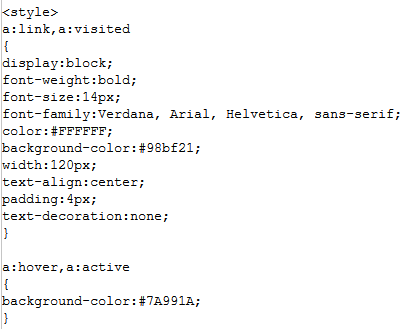
文本修饰：text-decoration 属性大多用于去掉链接中的下划线



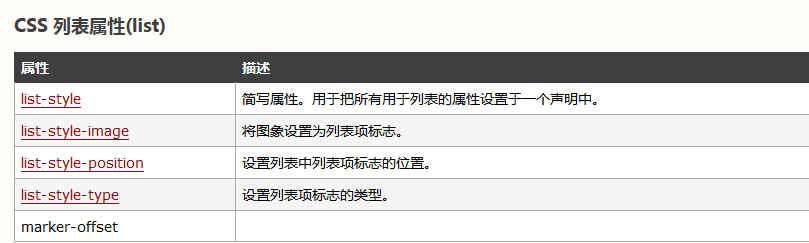
连接的背景颜色



创建连接框



css列表样式

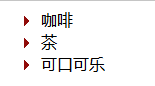


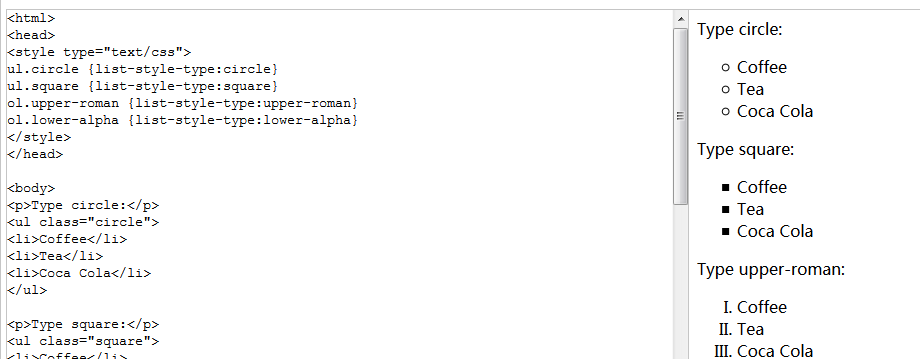
列表类型

在一个无序列表中，列表项的标志 (marker) 是出现在各列表项旁边的圆点。在有序列表中，标志可能是字母、数字或另外某种计数体系中的一个符号

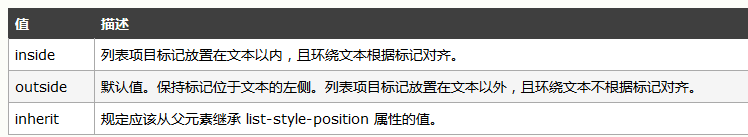
列表的图像

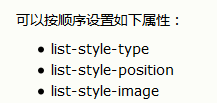
有时，常规的标志是不够的。你可能想对各标志使用一个图像，这可以利用 [list-style-image](http://www.w3school.com.cn/cssref/pr_list-style-image.asp) 属性做到，只需要简单地使用一个 url() 值，就可以使用图像作为标志

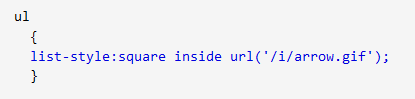


要修改用于列表项的标志类型，可以使用属性 [list-style-type](http://www.w3school.com.cn/cssref/pr_list-style-type.asp)

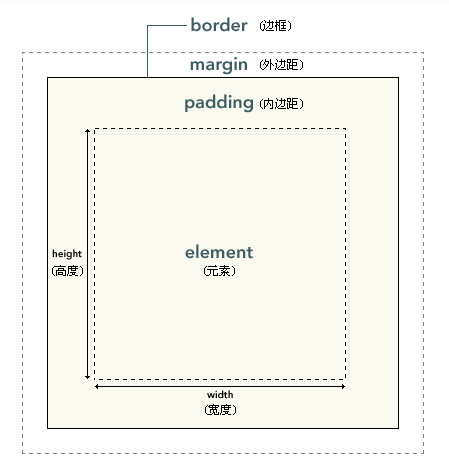
列表的标志位置：标志出现在列表项内容之外还是内容内部[list-style-position](http://www.w3school.com.cn/cssref/pr_list-style-position.asp)



简单写法可以将以上 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" \o "CSS list-style 属性)



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

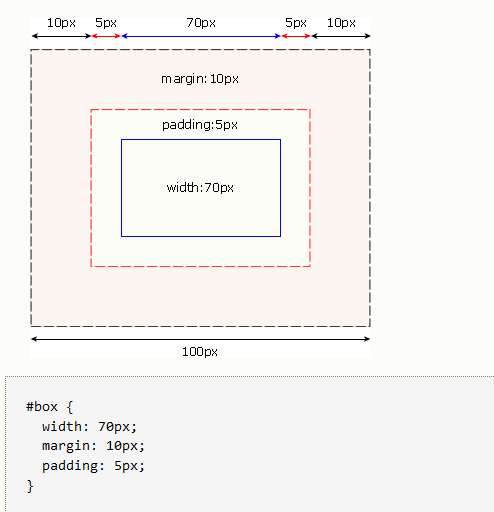
**CSS模型框：CSS 框模型 (Box Model) 规定了元素框处理元素内容、[内边距](http://www.w3school.com.cn/css/css_padding.asp" \o "CSS 内边距)、**[**边框**](http://www.w3school.com.cn/css/css_border.asp) **和** [**外边距**](http://www.w3school.com.cn/css/css_margin.asp) **的方式**

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

在 CSS 中，width 和 height 指的是（元素）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：

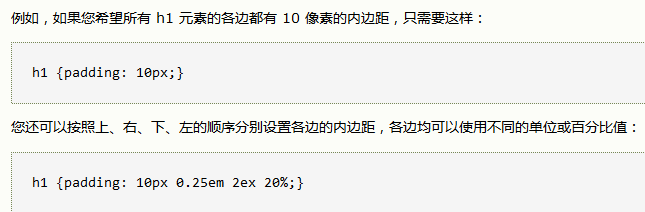


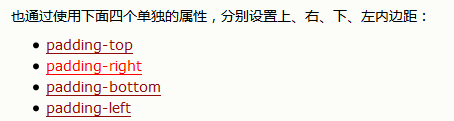
内边距、边框和外边距可以应用于一个元素的所有边，也可以应用于单独的边。

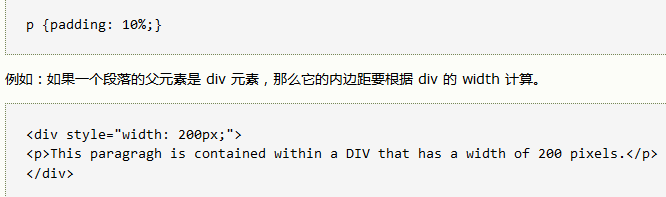
外边距可以是负值，而且在很多情况下都要使用负值的外边距

**CSS padding 属性定义元素边框与元素内容之间的空白区域**

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**

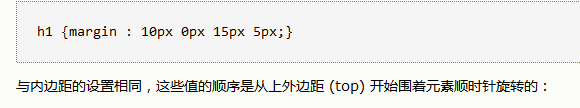


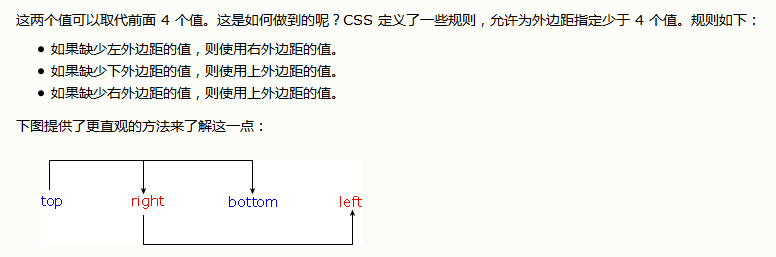


前面提到过，可以为元素的内边距设置百分数值。百分数值是相对于其父元素的 width 计算的，这一点与外边距一样。所以，如果父元素的 width 改变，它们也会改变，上下内边距与左右内边距一致；即上下内边距的百分数会相对于父元素宽度设置，而不是相对于高度

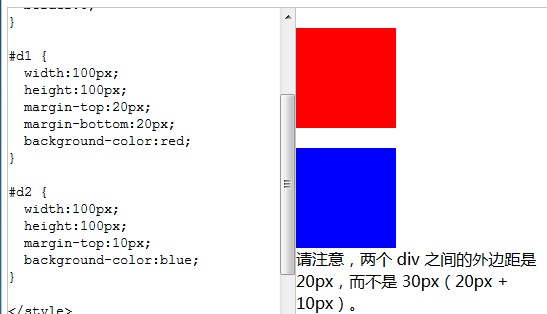
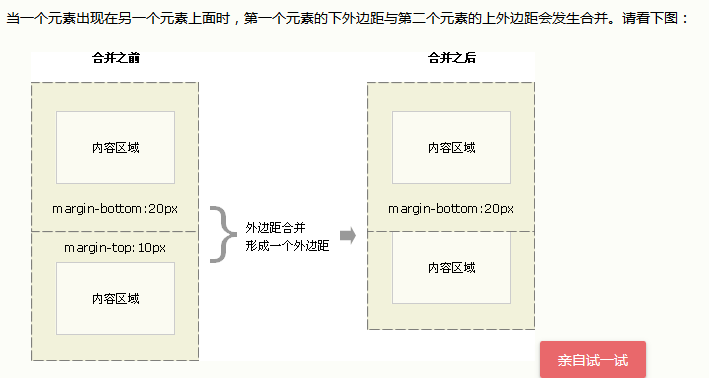
设置外边距的最简单的方法就是使用 [margin 属性](http://www.w3school.com.cn/cssref/pr_margin.asp)

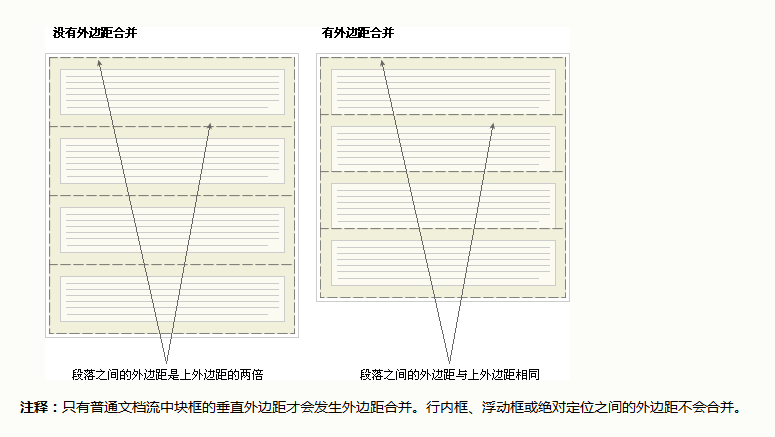
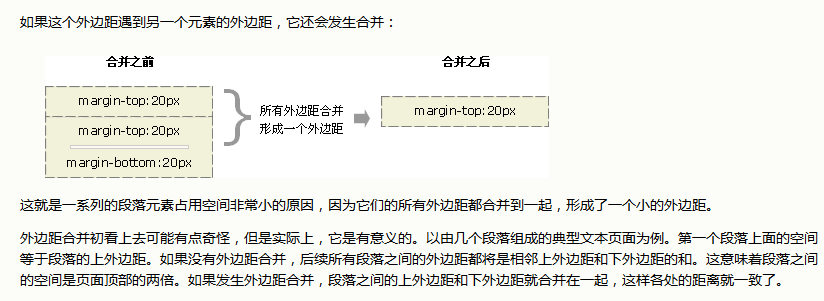
margin 属性接受任何长度单位，可以是像素、英寸、毫米或 em。margin 可以设置为 auto。更常见的做法是为外边距设置长度值



值复制

CSS外边距合并：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





CSS的定位

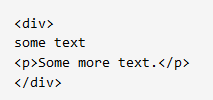
**CSS 定位 (Positioning) 属性允许你对元素进行定位：**定位的基本思想很简单，它允许你定义元素框相对于其正常位置应该出现的位置，或者相对于父元素、另一个元素甚至浏览器窗口本身的位置

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”。

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：



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

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。

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：

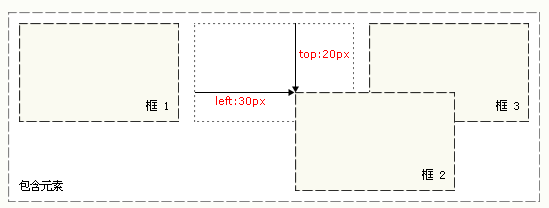
static

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

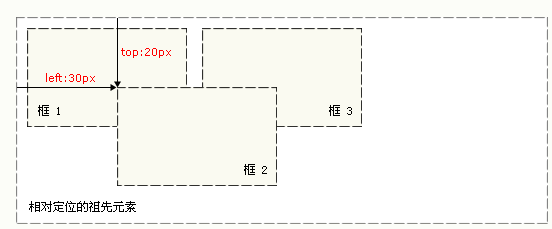
元素框偏移某个距离。元素仍保持其未定位前的形状，它原本所占的空间仍保留。相对定位实际上被看作普通流定位模型的一部分，因为元素的位置相对于它在普通流中的位置，相对定位会按照元素的原始位置对该元素进行移动

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



absolute

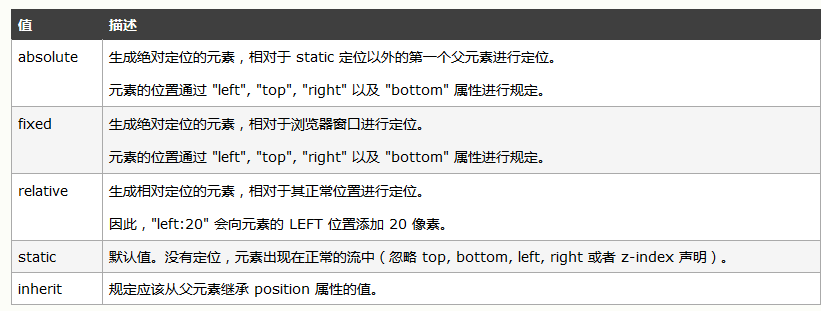
元素框从文档流完全删除，并相对于其包含块定位（父级元素）。包含块可能是文档中的另一个元素或者是初始包含块。元素原先在正常文档流中所占的空间会关闭，就好像元素原来不存在一样。元素定位后生成一个块级框，而不论原来它在正常流中生成何种类型的框。

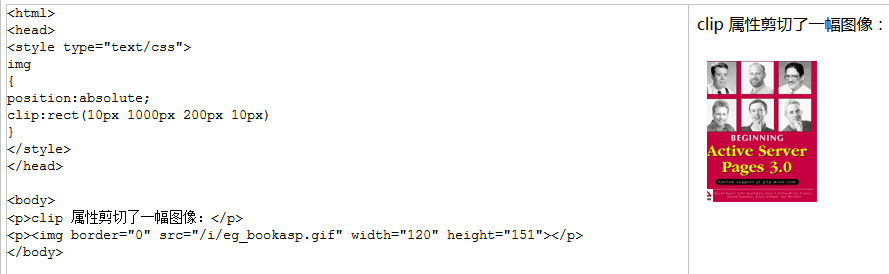


因为绝对定位的框与文档流无关，所以它们可以覆盖页面上的其它元素。可以通过设置 [z-index 属性](http://www.w3school.com.cn/cssref/pr_pos_z-index.asp)来控制这些框的堆放次序。

fixed

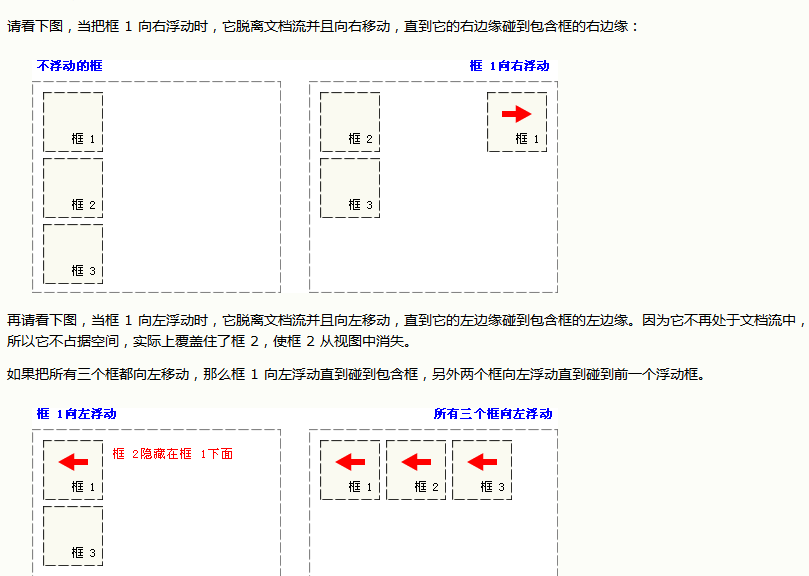
元素框的表现类似于将 position 设置为 absolute，不过其包含块是视窗本身。

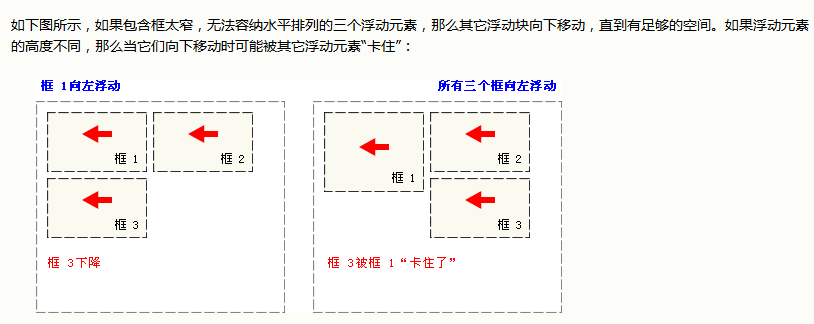




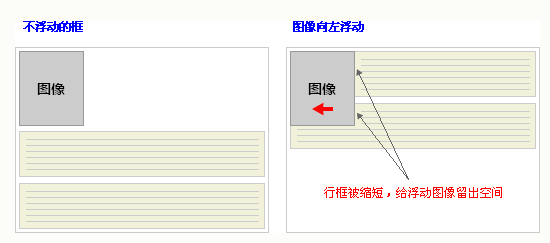
Z-index可被用于将在一个元素放置于另一元素之后形成覆盖的效果

CSS浮动**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**

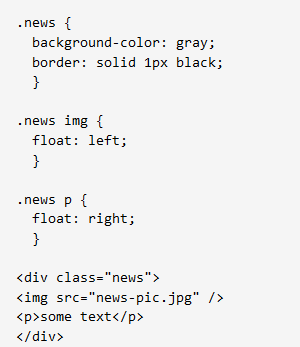


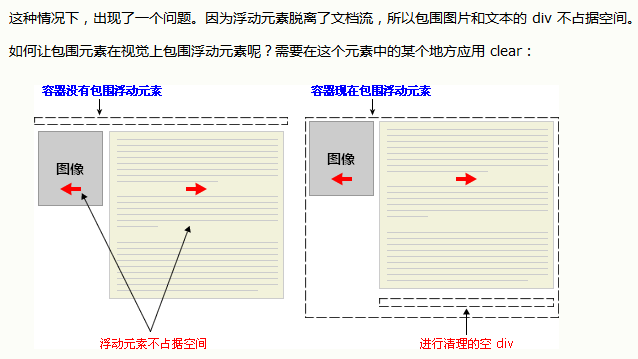


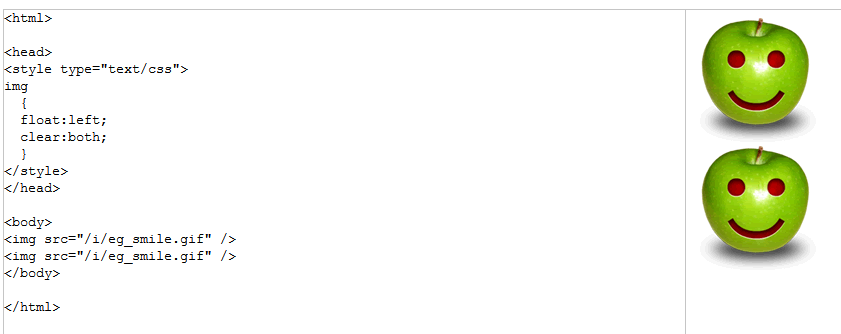
行框和清理

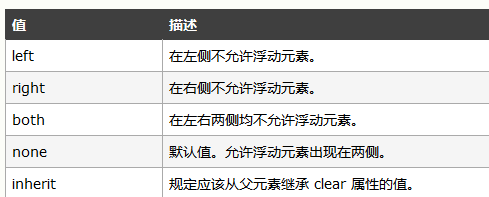
浮动框旁边的行框被缩短，从而给浮动框留出空间，行框围绕浮动框实现文字环绕图像的效果

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。



图像的左侧和右侧均不允许出现浮动元素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。



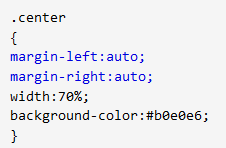
clear 属性规定元素的哪一侧不允许其他浮动元素。

**对其块级元素**

块元素指的是占据全部可用宽度的元素，并且在其前后都会换行

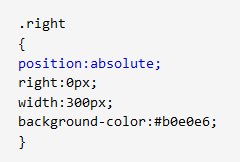
使用margin属性水平对其块级元素

可通过将左和右外边距设置为 "auto"，来对齐块元素（如果宽度是 100%，则对齐没有效果）



使用 position 属性进行左和右对齐：绝对定位

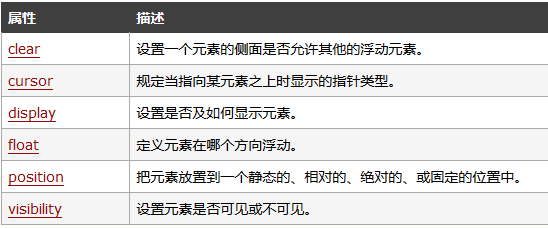
当使用 position 属性时，请始终设置 !DOCTYPE 声明

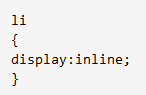


**Css尺寸**



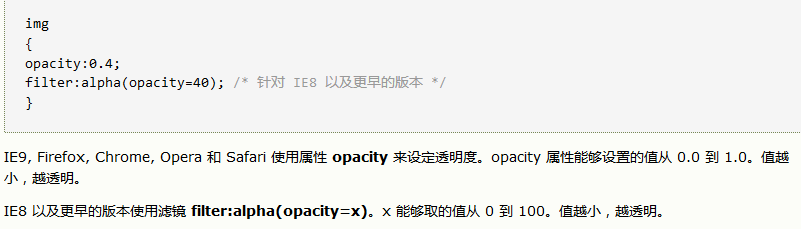
**Css分类属性**





display:inline; - 默认地，<li> 元素是块元素。在这里，我们去除了每个列表项前后的换行，以便让它们在一行中显示。

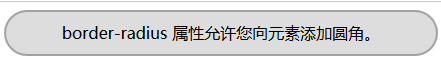
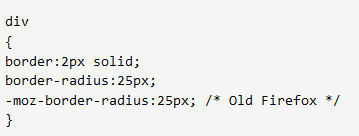
图片透明



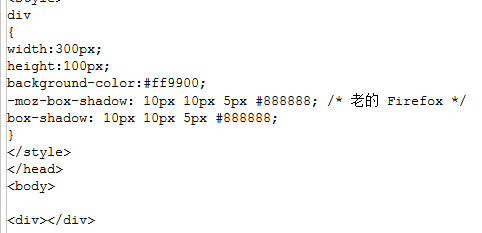
CSS3 教程

边框属性

border-radius 属性用于创建圆角



box-shadow 用于向方框添加阴影





border-image 属性，您可以使用图片来创建边框