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。

CSS声明总是以分号(;)结束，声明组以大括号({})括起来:

CSS注释以 "/\*" 开始, 以 "\*/" 结束,

HTML元素以id属性来设置id选择器,CSS 中 id 选择器以 "#" 来定义。

 ID属性不要以数字开头，数字开头的ID在 Mozilla/Firefox 浏览器中不起作用。

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

class 选择器在HTML中以class属性表示, 在 CSS 中，类选择器以一个点"."号显示：

class 选择器用于描述一组元素的样式，class 选择器有别于id选择器，class可以在多个元素中使用。

.center {text-align:center;}

指定特定的HTML元素使用class: p.center {text-align:center;}

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。

* 外部样式表

**不要在属性值与单位之间留有空格。**

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

hr {color:sienna;}  
p {margin-left:20px;}  
body {background-image:url("/images/back40.gif");}

* 内部样式表

<head>  
<style>  
hr {color:sienna;}  
p {margin-left:20px;}  
body {background-image:url("images/back40.gif");}  
</style>  
</head>

* 内联样式

<p style="color:sienna;margin-left:20px">This is a paragraph.</p>

如果某些属性在不同的样式表中被同样的选择器定义，那么属性值将从更具体的样式表中**被继承**过来。

（<由多到少>:属性从多的样式表中被组合到属性少的样式表中）

一般而言，所有的样式会根据下面的规则层叠于一个新的虚拟样式表中，其中数字 4 拥有最高的优先权。

1浏览器缺省设置

2外部样式表

3内部样式表（位于 <head> 标签内部）

4内联样式（在 HTML 元素内部）

如果你使用了外部文件的样式在 <head>中也定义了该样式，则内部样式表会取代外部文件的样式。

**背景**background

background-color

background-image 设置图片

默认情况下，背景图像进行平铺重复显示，以覆盖整个元素实体.

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

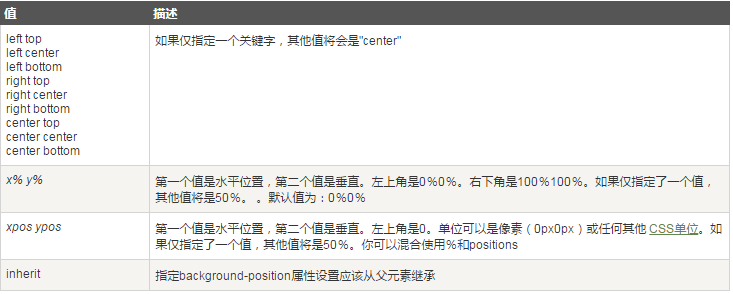
background-repeat 控制背景图片平铺

值：repeat-x;水平重复；repeat-y;垂直重复；no-repeat;不平铺，不重复背景图片

background-attachment 固定（不随滚动条滚动）

值：fixed;图片固定

background-position 设置图像位置



简写形式,顺序即以上属性顺序

eg.body {background:#ffffff url('img\_tree.png') no-repeat right top;}

文本（text）属性 描述

color 设置文本颜色

direction 设置文本方向。

letter-spacing 设置字符间距

line-height 设置行高

text-align 对齐元素中的文本

文本可居中center或对齐到左left或右right,两端对齐.

当text-align设置为"justify"，每一行被展开为宽度相等，左，右外边距是对齐（如杂志和报纸）。

text-decoration 向文本添加修饰 要是用来删除链接的下划线(none)

text-indent 缩进元素中文本的首行

text-shadow 设置文本阴影 text-shadow: *h-shadow v-shadow blur color*;

h-shadow 必需。水平阴影的位置。允许负值。

v-shadow 必需。垂直阴影的位置。允许负值。

blur 可选。模糊的距离。

color 可选。阴影的颜色。参阅 CSS 颜色值。

text-transform 控制元素中的字母

可用于所有字句变成大写或小写字母，或每个单词的首字母大写。

none 默认。定义带有小写字母和大写字母的标准的文本。

capitalize 文本中的每个单词以大写字母开头。

uppercase 定义仅有大写字母。

lowercase 定义无大写字母，仅有小写字母。

inherit 规定应该从父元素继承 text-transform 属性的值。

unicode-bidi

vertical-align 设置元素的垂直对齐 （有属性）

white-space 设置元素中空白的处理方式（有属性）

word-spacing 设置字间距 word-spacing:30px;

字体（font）属性

<http://www.runoob.com/css/css-font.html>

Property 描述

font 在一个声明中设置所有的字体属性

font-family 指定文本的字体系列

font-size 指定文本的字体大小

font-style 指定文本的字体样式

font-variant 以小型大写字体或者正常字体显示文本。

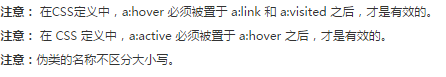
font-weight 指定字体的粗细。

链接的样式，可以用任何CSS属性（如颜色，字体，背景等）。

特别的链接，可以有不同的样式，这取决于他们是什么状态。

这四个链接状态是：

a:link - 正常，未访问过的链接

a:visited - 用户已访问过的链接

a:hover - 当用户鼠标放在链接上时

a:active - 链接被点击的那一刻

可以创建链接框 具体见HTML文档

列表 <http://www.runoob.com/css/css-list.html>

属性 描述

list-style 简写属性。用于把所有用于列表的属性设置于一个声明中

<顺序Type,position,image> list-style: square url("sqpurple.gif");

list-style-image 将图象设置为列表项标志。

list-style-position 设置列表中列表项标志的位置。

list-style-type 设置列表项标志的类型

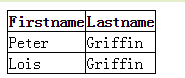
<http://www.runoob.com/cssref/pr-list-style-type.html>

表格

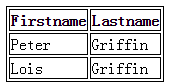
border 边框属性 粗细，颜色，线的类型

border-collapse 属性设置表格的边框是否被折叠成一个单一的边框或隔开

border-collapse: collapse;



如果不设置的话，会是这样子：



width:100%;height:50px;text-align:right;padding:15px;

vertical-align:bottom; 垂直对齐（bottom，top，center）

padding:15px; 填充，其实就是控制格子大小

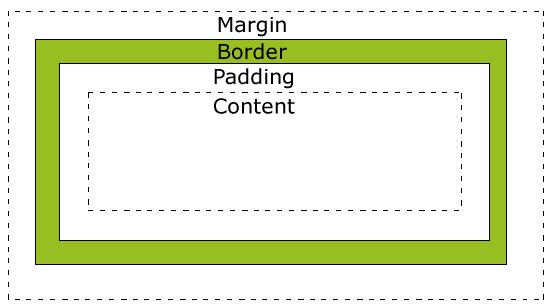
CSS盒模型本质上是一个盒子，封装周围的HTML元素，它包括：边距，边框，填充，和实际内容。

Margin(外边距) - 清除边框外的区域，外边距是透明的。

Border(边框) - 围绕在内边距和内容外的边框。

Padding(内边距) - 清除内容周围的区域，内边距是透明的。

Content(内容) - 盒子的内容，显示文本和图像。



最终元素的总宽度计算公式是这样的：

总元素的宽度=宽度+左填充+右填充+左边框+右边框+左边距+右边距

元素的总高度最终计算公式是这样的：

总元素的高度=高度+顶部填充+底部填充+上边框+下边框+上边距+下边距

**边框属性** <http://www.runoob.com/cssref/pr-border-style.html>

可单独设置每一边样式。

**属性 描述**

**border 简写属性，用于把针对四个边的属性设置在一个声明。**

**border-style 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 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 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 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 设置元素的下边框的颜色。**

**border-bottom-style 设置元素的下边框的样式。**

**border-bottom-width 设置元素的下边框的宽度。**

**border-left 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 设置元素的左边框的颜色。**

**border-left-style 设置元素的左边框的样式。**

**border-left-width 设置元素的左边框的宽度。**

**border-right 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 设置元素的右边框的颜色。**

**border-right-style 设置元素的右边框的样式。**

**border-right-width 设置元素的右边框的宽度。**

**border-top 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 设置元素的上边框的颜色。**

**border-top-style 设置元素的上边框的样式。**

**border-top-width 设置元素的上边框的宽度。**

border-style属性设置一个元素的四个边框的样式。此属性可以有一到四个值。

实例:

border-style:dotted solid double dashed;

上边框是点状

右边框是实线

下边框是双线

左边框是虚线

border-style:dotted solid double;

上边框是点状

右边框和左边框是实线

下边框是双线

border-style:dotted solid;

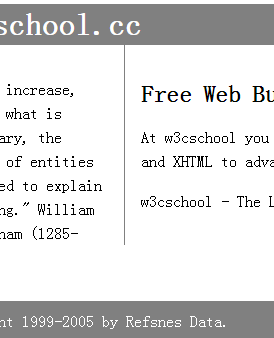
上边框和下边框是点状

右边框和左边框是实线

border-style:dotted;

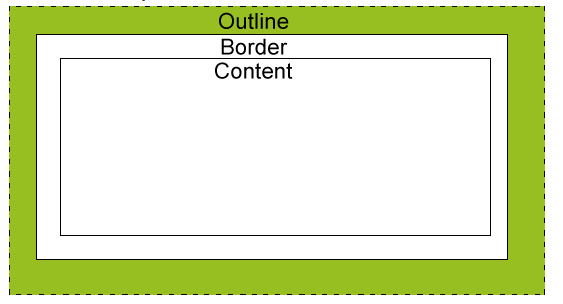
所有4个边框都是点状

border-left:1px solid gray; 可设置竖线，当分割线



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。

轮廓（outline）属性指定了样式，颜色和外边框的宽度。





外边距

margin 简写属性。在一个声明中设置所有外边距属性。顺序同border-style

三个数据时，顺序为上、左右、下

margin-bottom 设置元素的下外边距。

margin-left 设置元素的左外边距。

margin-right 设置元素的右外边距。

margin-top 设置元素的上外边距。

填充

padding 使用缩写属性设置在一个声明中的所有填充属性

顺序同border-style三个数据时，顺序为上、左右、下

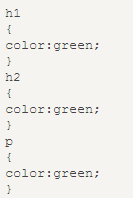
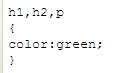
padding-bottom 设置元素的底部填充

padding-left 设置元素的左部填充

padding-right 设置元素的右部填充

padding-top 设置元素的顶部填充

分组选择器：

嵌套选择器



尺寸

height 设置元素的高度。

line-height 设置行高。

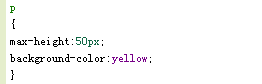
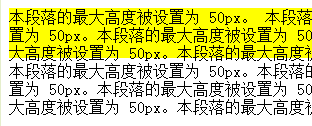
max-height 设置元素的最大高度。

max-width 设置元素的最大宽度。

min-height 设置元素的最小高度。

min-width 设置元素的最小宽度。

width 设置元素的宽度。

display属性设置一个元素应如何显示，visibility属性指定一个元素应可见还是隐藏。

隐藏一个元素可以通过把display属性设置为"none"，或把visibility属性设置为"hidden"。但是请注意，这两种方法会产生不同的结果。

visibility:hidden可以隐藏某个元素，但隐藏的元素仍需占用与未隐藏之前一样的空间。也就是说，该元素虽然被隐藏了，但仍然会影响布局。

display:none可以隐藏某个元素，且隐藏的元素不会占用任何空间。也就是说，该元素不但被隐藏了，而且该元素原本占用的空间也会从页面布局中消失。

块元素是一个元素，占用了全部宽度，在前后都是换行符。

块元素的例子：

<h1>

<p>

<div>

内联元素只需要必要的宽度，不强制换行。

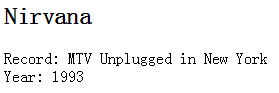
内联元素的例子：

<span>

<a>

当把块元素设置为内联inline;内联元素设置为块block；他们的换行属性改变

li{display:inline;}

IMG_256

span {display:block;}

<h2>Nirvana</h2>

<span>Record: MTV Unplugged in New York</span>

<span>Year: 1993</span>

**定位 position**

static :默认值，即没有定位，元素出现在正常的流中。

静态定位的元素不会受到top, bottom, left, right影响。

fixed:元素的位置相对于浏览器窗口是固定位置。即使窗口是滚动的它也不会移动

Fixed定位使元素的位置与文档流无关，因此不占据空间。

Fixed定位的元素和其他元素重叠

relative :相对定位元素的定位是相对其正常位置(默认位置，一般居中）。

可以移动的相对定位元素的内容和相互重叠的元素，它原本所占的空间不会改变。

相对定位元素经常被用来作为绝对定位元素的容器块。

absolute：绝对定位的元素的位置相对于最近的已定位父元素，如果元素没有已定位的父元素，

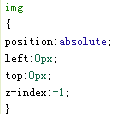
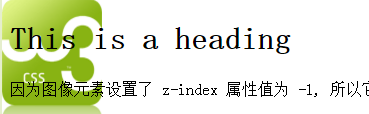
那么它的位置相对于<html>；Absolutely定位使元素的位置与文档流无关，因此不占据空间。

Absolutely定位的元素和其他元素重叠。

z-index:元素的定位与文档流无关，所以它们可以覆盖页面上的其它元素

z-index属性指定了一个元素的堆叠顺序（哪个元素应该放在前面，或后面）

一个元素可以有正数或负数的堆叠顺序

如何设置元素的外形。该元素被剪裁成这种形状，并显示出来。

[如何使用滚动条来显示元素内溢出的内容](http://www.runoob.com/try/try.php?filename=trycss_overflow" \t "http://www.runoob.com/css/_blank)

[如何设置浏览器自动溢出处理](http://www.runoob.com/try/try.php?filename=trycss_pos_overflow_auto" \t "http://www.runoob.com/css/_blank)

[更改光标](http://www.runoob.com/try/try.php?filename=trycss_cursor" \t "http://www.runoob.com/css/_blank)

<http://www.runoob.com/css/css-positioning.html>

**浮动 float**

元素的水平方向浮动，意味着元素只能左右移动而不能上下移动。

一个浮动元素会尽量向左或向右移动，直到它的外边缘碰到包含框或另一个浮动框的边框为止。

浮动元素之后的元素将围绕它。

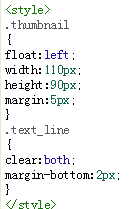
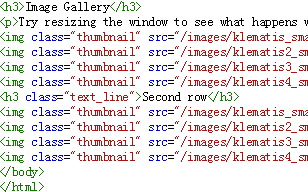
浮动元素之前的元素将不会受到影响。

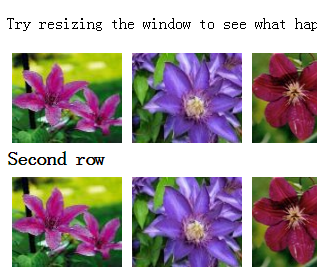
如果图像是右浮动，下面的文本流将环绕在它左边

如果你把几个浮动的元素放到一起，如果有空间的话，它们将彼此相邻.可以用外边距margin设置间距

元素浮动之后，周围的元素会重新排列，为了避免这种情况，使用 clear 属性。

clear 属性指定元素两侧不能出现浮动元素。



div

{

float:right;

width:120px;

margin:0 0 15px 20px;

padding:15px;

border:1px solid black;

text-align:center;

}

<img src="logocss.gif" width="95" height="84" /><br>

CSS is fun!

</div>



[让段落的第一个字母浮动到左侧](http://www.runoob.com/try/try.php?filename=trycss_float4" \t "http://www.runoob.com/css/_blank)

[创建一个没有表格的网页](http://www.runoob.com/try/try.php?filename=trycss_float6" \t "http://www.runoob.com/css/_blank)

<http://www.runoob.com/css/css-float.html>

清除浮动另一种方法是：使用overflow:auto/hidden,详见overflow属性

**水平对齐**

块元素是一个元素，占用了全宽，前后都是换行符。使用文本对齐

margin属性设置居中对齐，

两句

margin-left:auto;  
 margin-right:auto;

或者一句margin:auto;

Position设置左右对齐

左对齐 右对齐  
 position:absolute; position:absolute;  
 left:0px; right:0px;

Float设置左右对齐 float:right or left

后代选择器 ：选取容器内某元素

Eg.**div p**{····} 选取所有div里面的p元素设置成该样式

子元素选择器：选取容器里直接子元素，子元素的元素不选

Eg.**div>p**{····}

<div>

<h2>My name is Donald</h2>

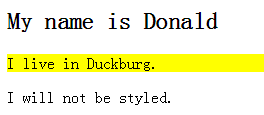
<p>I live in Duckburg.</p>

</div>

<div>

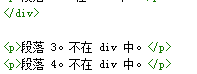
<span><p>I will not be styled.</p></span>

</div>



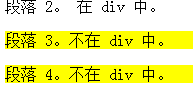
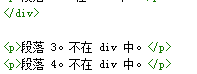
相邻兄弟选择器：可选择紧接在另一元素后的一个元素，且二者有相同父元素。

**div+p{}**



## 普通相邻兄弟选择器:可选择紧接在另一元素后的多个相同元素

**div~p{}**



伪类(Pseudo-classes)

伪类的语法：

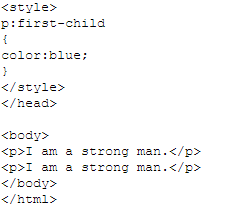
selector:pseudo-class {property:value;}

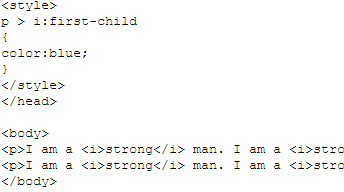
CSS类也可以使用伪类：

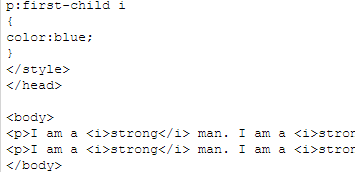
selector.class:pseudo-class {property:value;}

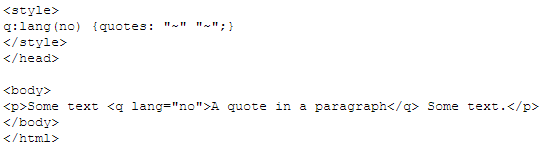
a.red:visited {color:#FF0000;}  
  
<a class="red" href="css-syntax.html">CSS Syntax</a>

如果在上面的例子的链接已被访问，它会显示为红色。

 第一个P标签会显示蓝色

 每个P标签的第一个i元素显示蓝色

 第一个P标签里面的所有i元素显示蓝色



给属性:lang为no的q标签添加~做引号

伪元素pseudo-element

伪元素的语法：

selector:pseudo-element {property:value;}

CSS类也可以使用伪元素：

selector.class:pseudo-element {property:value;}

**"first-line"** 伪元素用于向文本的首行设置特殊样式。

注意："first-line" 伪元素只能用于块级元素。

注意： 下面的属性可应用于 "first-line" 伪元素：

font properties

color properties

background properties

word-spacing

letter-spacing

text-decoration

vertical-align

text-transform

line-height

Clear

**"first-letter"** 伪元素用于向文本的首字母设置特殊样式：

注意： "first-letter" 伪元素只能用于块级元素。

注意： 下面的属性可应用于 "first-letter" 伪元素：

font properties

color properties

background properties

margin properties

padding properties

border properties

text-decoration

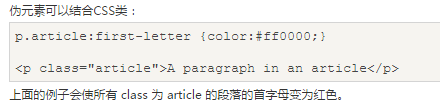
vertical-align (only if "float" is "none")

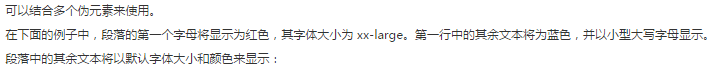
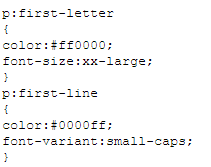
text-transform

line-height

float

Clear

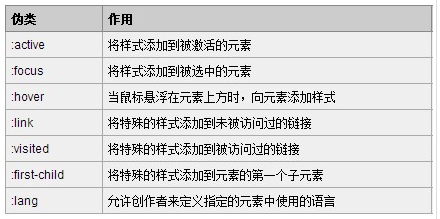


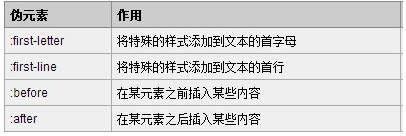


":before" 伪元素可以在元素的内容前面插入新内容。

":after" 伪元素可以在元素的内容之后插入新内容。

**伪类与伪元素区别**





这里用伪类 :first-child 和伪元素 :first-letter 来进行比较。

IMG_256p>i:first-child {color: red}

<p>

<i>first</i>

<i>second</i>

</p>

//伪类 :first-child 添加样式到第一个子元素

如果我们不使用伪类，而希望达到上述效果，可以这样做：

.first-child {color: red}

<p>

<i class="first-child">first</i>

<i>second</i>

</p>

IMG_256即我们给第一个子元素添加一个类，然后定义这个类的样式。那么我们接着看看为元素：

p:first-letter {color: red}

<p>I am stephen lee.</p>

//伪元素 :first-letter 添加样式到第一个字母

那么如果我们不使用伪元素，要达到上述效果，应该怎么做呢？

.first-letter {color: red}

<p><span class='first-letter'>I</span> am stephen lee.</p>

即我们给第一个字母添加一个 span，然后给 span 增加样式。

两者的区别已经出来了。那就是：

伪类的效果可以通过添加一个实际的类来达到，而伪元素的效果则需要通过添加一个实际的元素才能达到，这也是为什么他们一个称为伪类，一个称为伪元素的原因。

总结

伪元素和伪类之所以这么容易混淆，是因为他们的效果类似而且写法相仿，但实际上 css3 为了区分两者，已经明确规定了伪类用一个冒号来表示，而伪元素则用两个冒号来表示。

:Pseudo-classes

::Pseudo-elements

**Overflow 属性**

当页面有上下两个DIV层时，由于上面的DIV文字内容过多，超出该层的高度和宽度，此时上面的DIV层中的内容会覆盖到下面的DIV层之上。如下图：



此时该怎么解决呢？这就会用到CSS的一个很有用的属性—Overflow。该属性有四个值：

· visible（默认值）

· hidden

· scroll

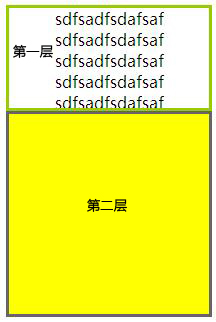
· auto

Visible

Overflow的Visible属性是默认值，也就是说，最初说到的那种情况就是Visible在起作用。这里需要注意的是，虽然第一层的溢出的内容会覆盖到第二层之上，但是并不会将第二层挤出原来的位置。如上图。

Hidden

Hidden值与默认值相反，它会将所有超出DIV层范围的所有内容都给隐藏掉。如下图：



该属性对显示动态内容、由于内容溢出而引起一些布局上的问题的确很有用。但是，要记住用此方法隐藏的内容将彻底的看不到。 比如：用户将他们的浏览器的默认字体调大，这将会使一些文字推到DIV层的外面被完全的隐藏。

Scroll

设置Overflow值为scroll将会在DIV层之内提供一个滚动条，从而可以查看到DIV层内所有的内容。如下图：



需要注意的是：使用Scroll将会同时产生水平和垂直两个滚动条，就算内容只需要其中的一个。而且如果该DIV层没有任何内容也会显示水平和垂直的禁用的滚动条。所以针对这种情况，Overflow又提供了Auto值。

Auto

Overflow的auto值与scroll很像，它唯一不同的是在DIV层中的内容不需要滚动条的时候不会出现滚动条，与普通的DIV没有区别；如果内容超出范围就会显示滚动条。

Overflow的另一种用途：清除浮动

Overflow另一个更为流行的用途是：清除浮动。假如现在有一个背景层，该层要包含两个子层，这两个子层都是浮动层（也就是说它们都设置了float属性）背景层只设置宽度，不设置高度，此时如果不设置背景层的Overflow属性，那么将不会显示背景层。

如果应用了Overflow(auto或hidden)的元素，将会清除子层的浮动，将两个子层包含到自己之内，就是说背景层将会扩展到它需要的大小以包围它里面的浮动的子元素。

代码如下：

<div id="layer1" style="width:500px; background-color:#99CC00; overflow:hidden;">

<div id="sonLayer1" style="width:100px; height:100px; float:left; background-color:#999999;">

</div>

<div id="sonLayer2" style="width:100px; height:100px; float:right; background-color:#6666FF;">

</div>

</div>

利用Overflow属性制作不带滚动条的Textarea

所有对象的Overflow属性默认值都是visible，除了Textarea和body对象的默认值是auto。设置Textarea对象此属性值为hidden将隐藏其滚动条。

利用Overflow属性还可以去除无序列表li标签前面Type，如：实心圆点、空心方框等。设置li标签的Overflow属性为Hidden即可。

**导航栏 用列表写**

垂直和水平导航栏使用的标准代码

ul

{

list-style-type:none;//移除列表前小标志。一个导航栏并不需要列表标记

margin:0;

padding:0;//移除浏览器的默认设置将边距和填充设置为0

}

垂直导航

a  
{  
display:block;  
width:60px;  
}

display:block - 显示块元素的链接，让整体变为可点击链接区域（不只是文本），它允许我们指定宽度

width:60px - 块元素默认情况下是最大宽度。我们要指定一个60像素的宽度

水平导航

## A.内嵌列表项

li  
{  
display:inline;  
}

display:inline; -默认情况下，<li>元素是块元素。在这里，我们删除换行符之前和之后每个列表项，以显示一行 。

## B.浮动列表项

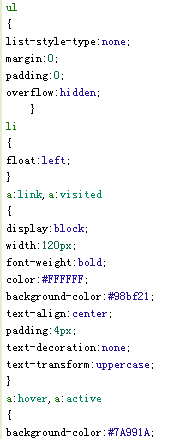
li  
{  
float:left;  
}  
a  
{  
display:block;  
width:60px;  
}

float:left - 使用浮动块元素的幻灯片彼此相邻

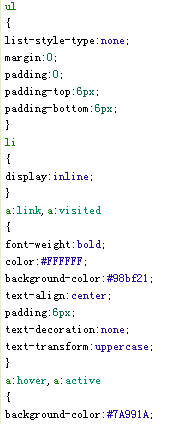
display:block - 显示块元素的链接，让整体变为可点击链接区域（不只是文本），它允许我们指定宽度

width:60px - 块元素默认情况下是最大宽度。我们要指定一个60像素的宽度

Eg.1.

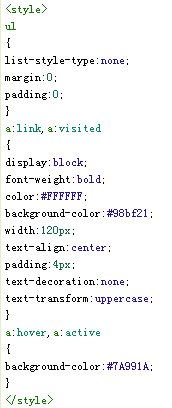
IMG_256

2.



IMG_256

3.

想选项分开，可增加li{margin-top:..px;}利用外边距分开。

**下拉菜单**

**HTML 部分：**

我们可以使用任何的 HTM，元素来打开下拉菜单，如：<span>, 或 a <button> 元素。

使用容器元素 (如： <div>) 来创建下拉菜单的内容，并放在任何你想放的位置上。

使用 <div> 元素来包裹这些元素，并使用 CSS 来设置下拉内容的样式。

**CSS 部分：**

.dropdown 类使用 position:relative, 这将设置下拉菜单的内容放置在下拉按钮 (使用 position:absolute) 的右下角位置。

.dropdown-content 类中是实际的下拉菜单。默认是隐藏的，在鼠标移动到指定元素后会显示。 注意 min-width 的值设置为 160px。你可以随意修改它。 **注意:** 如果你想设置下拉内容与下拉按钮的宽带一致，可设置 width 为 100% ( overflow:auto 设置可以在小尺寸屏幕上滚动)。

我们使用 box-shadow 属性让下拉菜单看起来像一个"卡片"。

:hover 选择器用于在用户将鼠标移动到下拉按钮上时显示下拉菜单。

<style>

/\* 下拉按钮样式 \*/

.dropbtn {

background-color: #4CAF50;

color: white;

padding: 16px;

font-size: 16px;

border: none;

cursor: pointer;

}

/\* 容器 <div> - 需要定位下拉内容 \*/

.dropdown {

position: relative;

display: inline-block;

}

/\* 下拉内容 (默认隐藏) ，//菜单样式设置\*/

.dropdown-content {

display: none;

position: absolute;

background-color: #f9f9f9;

min-width: 160px;

box-shadow: 0px 8px 16px 0px rgba(0,0,0,0.2);

}

/\* 下拉菜单的链接 \*/

.dropdown-content a {

color: black;

padding: 12px 16px;

text-decoration: none;

display: block;

}

/\* 鼠标移上去后修改下拉菜单链接颜色 \*/

.dropdown-content a:hover {background-color: #f1f1f1}

/\* 在鼠标移上去后显示下拉菜单 \*/

.dropdown:hover .dropdown-content {

display: block;

}

/\* 当下拉内容显示后修改下拉按钮的背景颜色 \*/

.dropdown:hover .dropbtn {

background-color: #3e8e41;

}

</style>

<div class="dropdown">

<button class="dropbtn">下拉菜单</button> //菜单按钮可以使用任何html元素

<div class="dropdown-content"> //菜单内容 用<div>来装

<a href="#">菜鸟教程 1</a> //为选项添加链接

<a href="#">菜鸟教程 2</a>

<a href="#">菜鸟教程 3</a>

</div>

</div>

IMG_256



如果在导航栏中添加下拉菜单，直接将该导航按钮li部分换成菜单按钮和菜单内容的<div>

菜单中加入图片可直接使用img添加



制作图片廊，可以使用display:inline使图片横向排列

<http://www.runoob.com/css/css-image-gallery.html>

图像透明/不透明

img  
{  
opacity:0.4;   
filter:alpha(opacity=40); /\* For IE8 and earlier \*/  
}

Opacity属性值从0.0 - 1.0。值越小，使得元素更加透明。

IE8和早期版本使用滤镜：alpha（opacity= x）。 x可以采取的值是从0 - 100。较低的值，使得元素更加透明

当用户将鼠标悬停在其中一个图像上时发生，原来模糊图像变清晰。

img:hover  
{  
opacity:1.0;  
filter:alpha(opacity=100); /\* For IE8 and earlier \*/  
}

透明盒子中的文字

<http://www.runoob.com/css/css-image-transparency.html>

**图像拼合技术 在一个网页中，使用一张图片的不同部分**

**好处：**降低服务器的请求数量，并节省带宽

**//列表样式**

#navlist{position:relative;}

#navlist li{margin:0;padding:0;list-style:none;position:absolute;top:0;}

#navlist li, #navlist a{height:44px;display:block;} //所有图像高44px

#home{left:0px;width:46px;} //图像在网页的位置，定位到最左边，以及图像的宽度是46px

#home{background:url('/images/img\_navsprites.gif') 0 0;}

//从图片的0,0处开始取，（0,0）是左上角

#prev{left:63px;width:43px;}//放在距离网页左侧63px处

#prev{background:url('/images/img\_navsprites.gif') -47px 0;}

//从图片47px处取43px

#next{left:129px;width:43px;} 根据以上数据依次计算，注意每图之间空隙要留出来

#next{background:url('/images/img\_navsprites.gif') -91px 0;}

IMG_256

<ul id="navlist">

<li id="home"><a href="/"></a></li>

<li id="prev"><a href="/css/"></a></li>

<li id="next"><a href="/css/"></a></li>//使用a标签给图片加链接

</ul>

如果想增加点击后效果，可在每个图片设置样式时增加.hover,但注意所有图片尽量使用一张图

eg.#home a:hover{background: url('img\_navsprites\_hover.gif') 0 -45px;

**Display 属性** 规定元素应该生成的框的类型

none 此元素不会被显示。

block 此元素将显示为块级元素，此元素前后会带有换行符。

inline 默认。此元素会被显示为内联元素，元素前后没有换行符。

inline-block 行内块元素。（CSS2.1 新增的值）

list-item 此元素会作为列表显示。

run-in 此元素会根据上下文作为块级元素或内联元素显示。

compact CSS 中有值 compact，不过由于缺乏广泛支持，已经从 CSS2.1 中删除。

marker CSS 中有值 marker，不过由于缺乏广泛支持，已经从 CSS2.1 中删除。

table 此元素会作为块级表格来显示（类似 <table>），表格前后带有换行符。

inline-table 此元素会作为内联表格来显示（类似 <table>），表格前后没有换行符。

table-row-group 此元素会作为一个或多个行的分组来显示（类似 <tbody>）。

table-header-group 此元素会作为一个或多个行的分组来显示（类似 <thead>）。

table-footer-group 此元素会作为一个或多个行的分组来显示（类似 <tfoot>）。

table-row 此元素会作为一个表格行显示（类似 <tr>）。

table-column-group 此元素会作为一个或多个列的分组来显示（类似 <colgroup>）。

table-column 此元素会作为一个单元格列显示（类似 <col>）

table-cell 此元素会作为一个表格单元格显示（类似 <td> 和 <th>）

table-caption 此元素会作为一个表格标题显示（类似 <caption>）

inherit 规定应该从父元素继承 display 属性的值。

CSS还可以根据属性值进行样式设置

[title] //包含标题（title）的所有元素变为蓝色

{

color:blue;

}

[title=w3cschool]//标题title='w3cschool'元素的边框样式  
{  
border:5px solid green;  
}

包含指定值的title属性的元素样式的例子，使用（~）分隔属性和值:

[title~=hello] { color:blue; } 只要有“hello”就可以

包含指定值的lang属性的元素样式的例子，使用（|）分隔属性和值

[lang|=en] { color:blue; } //只要含有“en”字符就可以

表单：属性选择器样式无需使用class或id的形式:

input[type="text"]

{

width:150px;

display:block;

margin-bottom:10px;

background-color:yellow;

}

居中：<http://www.blueidea.com/tech/web/2006/3231.asp>



