# 第四章 css进阶

本节所讲内容

4.1 css元素溢出

4.2 块元素、内联元素、内联块元素

4.3 浮动

4.4 定位

4.5 background属性

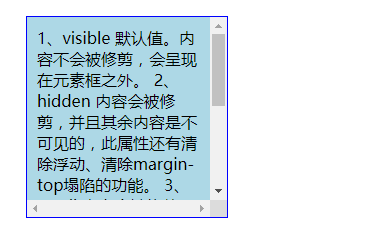
## 4.1 css元素溢出

当子元素的尺寸超过父元素的尺寸时，需要设置父元素显示溢出的子元素的方式，设置的方法是通过overflow属性来设置。

****overflow的设置项：****  
1、visible 默认值。内容不会被修剪，会呈现在元素框之外。visible [ˈvɪzəbl]： 看得见  
2、hidden 内容会被修剪，并且其余内容是不可见的，此属性还有清除浮动、清除margin-top塌陷的功能。  
3、scroll 内容会被修剪，但是浏览器会显示滚动条以便查看其余的内容。scroll[skrəʊl]:滚动  
4、auto 如果内容被修剪，则浏览器会显示滚动条以便查看其余的内容。  
5、inherit 规定应该从父元素继承 overflow 属性的值。inherit [ɪnˈherɪt]：继承

<!DOCTYPE html>  
<html lang="en">  
<head>  
 <meta charset="UTF-8">  
 <title>css元素溢出</title>  
 <style>  
 \*{  
 margin: 0;  
 padding: 0;  
 }  
 div{  
 height: 200px;  
 width: 200px;  
 border: solid 1px blue;  
 background: lightblue;  
 overflow: scroll;  
 }  
 </style>  
</head>  
<body>  
  
<div>  
 <p>  
 1、visible 默认值。内容不会被修剪，会呈现在元素框之外。  
 2、hidden 内容会被修剪，并且其余内容是不可见的，此属性还有清除浮动、清除margin-top塌陷的功能。  
 3、scroll 内容会被修剪，但是浏览器会显示滚动条以便查看其余的内容。  
 4、auto 如果内容被修剪，则浏览器会显示滚动条以便查看其余的内容。  
 5、inherit 规定应该从父元素继承 overflow 属性的值。  
 </p>  
</div>  
</body>  
</html>

运行结果如下：



## 4.2 块元素、内联元素、内联块元素

元素就是标签，布局中常用的有三种标签，块元素、内联元素、内联块元素，了解这三种元素的特性，才能熟练的进行页面布局。

### ****4.2.1 块元素****

块元素，也可以称为行元素，布局中常用的标签如：div、p、ul、li、h1~h6、dl、dt、dd等等都是块元素，它在布局中的行为：

1. 支持全部的样式
2. 如果没有设置宽度，默认的宽度为父级宽度100%
3. 盒子占据一行、即使设置了宽度



### ****4.2.2 内联元素****

内联元素，也可以称为行内元素，布局中常用的标签如：a、span、em、b、strong、i等等都是内联元素，它们在布局中的行为：

1. 支持部分样式（不支持宽、高、margin上下、padding上下）
2. 宽高由内容决定
3. 盒子并在一行
4. 代码换行，盒子之间会产生间距
5. 子元素是内联元素，父元素可以用text-align属性设置子元素水平对齐方式，用line-height属性值设置垂直对齐方式

****解决内联元素间隙的方法：****

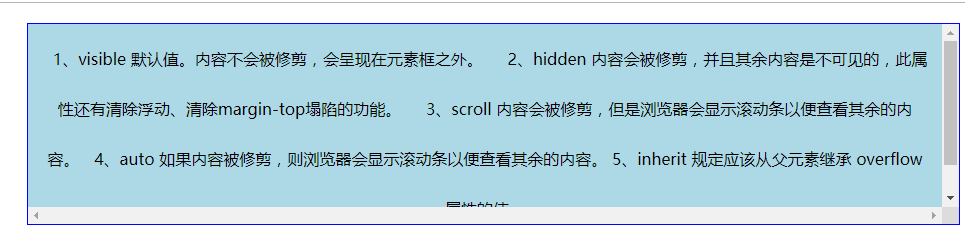
1. 去掉内联元素之间的换行
2. 将内联元素的父级设置font-size为0，内联元素自身再设置font-size

p{  
 text-align: center;  
 line-height: 50px;  
}  
span{  
 */\*display: block;\*/* height: 50px;  
}

<div>  
 <p>  
 <span>1、visible 默认值。内容不会被修剪，会呈现在元素框之外。</span>  
 <span>2、hidden 内容会被修剪，并且其余内容是不可见的，此属性还有清除浮动、清除margin-top塌陷的功能。</span>  
 <span>3、scroll 内容会被修剪，但是浏览器会显示滚动条以便查看其余的内容。</span>  
 4、auto 如果内容被修剪，则浏览器会显示滚动条以便查看其余的内容。  
 5、inherit 规定应该从父元素继承 overflow 属性的值。  
 </p>

<div>  
 <p>  
 <span>1、visible 默认值。内容不会被修剪，会呈现在元素框之外。</span>  
 <span>2、hidden 内容会被修剪，并且其余内容是不可见的，此属性还有清除浮动、清除margin-top塌陷的功能。</span>  
 <span>3、scroll 内容会被修剪，但是浏览器会显示滚动条以便查看其余的内容。</span>  
 4、auto 如果内容被修剪，则浏览器会显示滚动条以便查看其余的内容。  
 5、inherit 规定应该从父元素继承 overflow 属性的值。  
 </p>

****运行结果如下：****



### ****4.2.3 内联块元素****

内联块元素，也叫行内块元素，是新增的元素类型，现有元素没有归于此类别的，img和input元素的行为类似这种元素，但是也归类于内联元素，我们可以用display属性将块元素或者内联元素转化成这种元素。它们在布局中表现的行为：

1. 支持全部样式
2. 如果没有设置宽高，宽高由内容决定
3. 盒子并在一行
4. 代码换行，盒子会产生间距
5. 子元素是内联块元素，父元素可以用text-align属性设置子元素水平对齐方式，用line-height属性值设置子元素垂直对齐方式

注意：

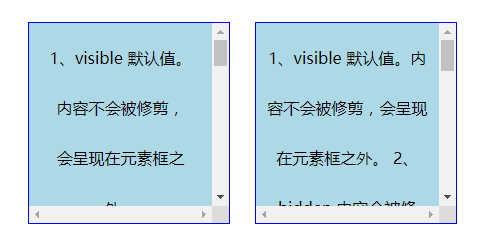
这三种元素，可以通过display属性来相互转化，不过实际开发中，块元素用得比较多，所以我们经常把内联元素转化为块元素，少量转化为内联块，而要使用内联元素时，直接使用内联元素，而不用块元素转化了。

****display属性：****

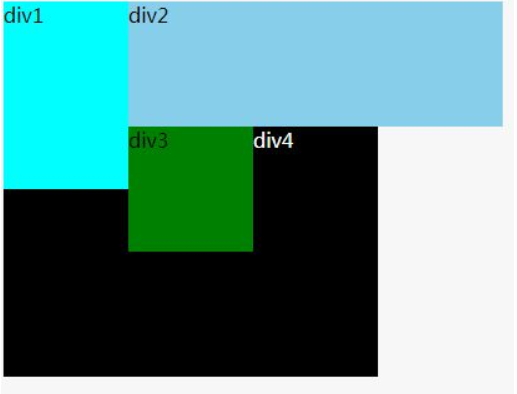
display属性是用来设置元素的类型及隐藏的，常用的属性有：  
 1、none 元素隐藏且不占位置  
 2、block 元素以块元素显示  
 3、inline 元素以内联元素显示  
 4、inline-block 元素以内联块元素显示

span{  
 display: block;  
 height: 50px;  
}

div{  
 height: 200px;  
 width: 200px;  
 border: solid 1px blue;  
 background: lightblue;  
 overflow: scroll;  
 display: inline-block;  
}



## 4.3 浮动



****文档流：****  
文档流，是指盒子按照html标签编写的顺序依次从上到下，从左到右排列，块元素占一行，行内元素在一行之内从左到右排列，先写的先排列，后写的排在后面，每个盒子都占据自己的位置。

### ****4.3.1 浮动特性****

1、浮动元素有左浮动(float:left)和右浮动(float:right)两种

2、浮动的元素会向左或向右浮动，碰到父元素边界、浮动元素、未浮动的元素才停下来

3、相邻浮动的块元素可以并在一行，超出父级宽度就换行

4、浮动让行内元素或块元素自动转化为行内块元素

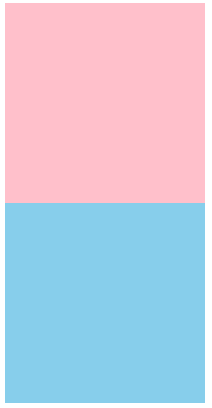
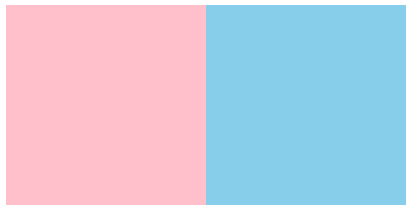
5、浮动元素后面没有浮动的元素会占据浮动元素的位置，没有浮动的元素内的文字会避开浮动的元素，形成文字饶图的效果

6、父元素内整体浮动的元素无法撑开父元素，需要清除浮动

7、浮动元素之间没有垂直margin的合并

<!DOCTYPE html>  
<html lang="en">  
<head>  
 <meta charset="UTF-8">  
 <title>float</title>  
 <style>  
 .box1{  
 float: left;  
 width: 200px;  
 height: 200px;  
 background: pink;  
 }  
 .box2{  
 float: left;  
 width: 200px;  
 height:200px;  
 background: skyblue;  
 }  
 </style>  
</head>  
<body>  
<div class = 'box1'></div>  
<div class = 'box2'></div>  
</body>  
</html>

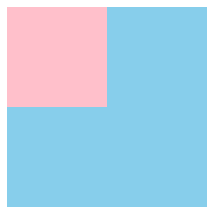
运行结果如下：

 加入浮动后—> 

在默认情况下，两个div标签是上下进行排列的。现在如果给float属性的第一个<div>标签出现了浮动，这个标签就会在另外一个层面上进行排列。而第二个<div>不给浮动，它还会在自己的层面上遵从文档标准流进行排列。

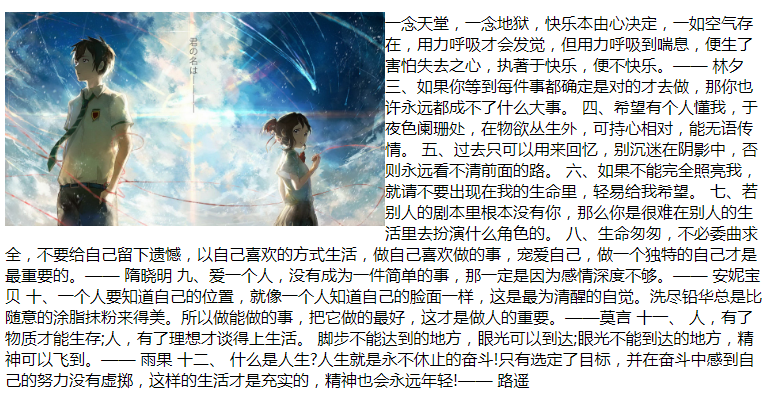
<!DOCTYPE html>  
<html lang="en">  
<head>  
 <meta charset="UTF-8">  
 <title>float</title>  
 <style>  
 .box1{  
 float: left;  
 width: 100px;  
 height: 100px;  
 background: pink;  
 }  
 .box2{  
 */\*float: left;\*/* width: 200px;  
 height:200px;  
 background: skyblue;  
 }  
 </style>  
</head>  
<body>  
<div class = 'box1'></div>  
<div class = 'box2'></div>  
</body>  
</html>

****运行结果如下：****



<!DOCTYPE html>  
<html lang="en">  
<head>  
 <meta charset="UTF-8">  
 <title>float</title>  
 <style>  
 img{  
 float: left;  
 }  
 </style>  
</head>  
<body>  
<div class ='box1'><img src="./1.jpg" width="50%" height="50%" alt=""></div>  
<p>  
 一念天堂，一念地狱，快乐本由心决定，一如空气存在，用力呼吸才会发觉，但用力呼吸到喘息，便生了害怕失去之心，执著于快乐，便不快乐。—— 林夕  
 三、如果你等到每件事都确定是对的才去做，那你也许永远都成不了什么大事。  
 四、希望有个人懂我，于夜色阑珊处，在物欲丛生外，可持心相对，能无语传情。  
 五、过去只可以用来回忆，别沉迷在阴影中，否则永远看不清前面的路。  
 六、如果不能完全照亮我，就请不要出现在我的生命里，轻易给我希望。  
 七、若别人的剧本里根本没有你，那么你是很难在别人的生活里去扮演什么角色的。  
 八、生命匆匆，不必委曲求全，不要给自己留下遗憾，以自己喜欢的方式生活，做自己喜欢做的事，宠爱自己，做一个独特的自己才是最重要的。—— 隋晓明  
 九、爱一个人，没有成为一件简单的事，那一定是因为感情深度不够。—— 安妮宝贝  
 十、一个人要知道自己的位置，就像一个人知道自己的脸面一样，这是最为清醒的自觉。洗尽铅华总是比随意的涂脂抹粉来得美。所以做能做的事，把它做的最好，这才是做人的重要。——莫言  
 十一、 人，有了物质才能生存;人，有了理想才谈得上生活。 脚步不能达到的地方，眼光可以到达;眼光不能到达的地方，精神可以飞到。—— 雨果  
 十二、 什么是人生?人生就是永不休止的奋斗!只有选定了目标，并在奋斗中感到自己的努力没有虚掷，这样的生活才是充实的，精神也会永远年轻!—— 路遥  
</p>  
</body>  
</html>

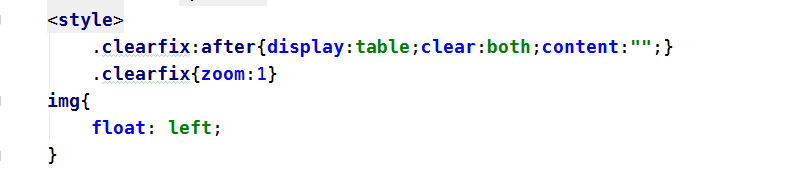
****运行结果如下：****

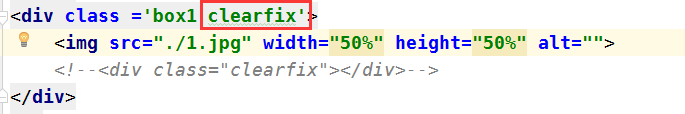


### ****4.3.2 清除浮动****

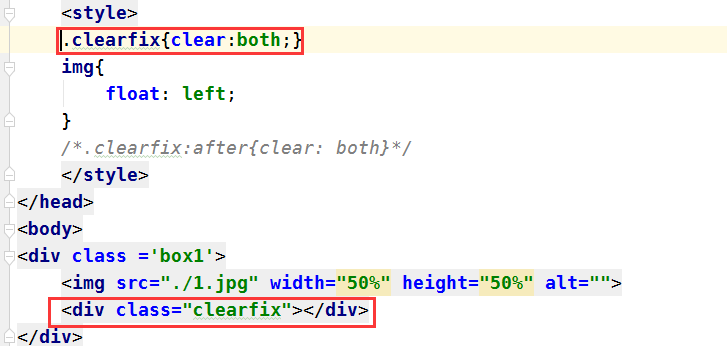
1. 父级div定义伪类：after和zoom

.clearfix:after{display:table;clear:both;content:"";}  
.clearfix{zoom:1}

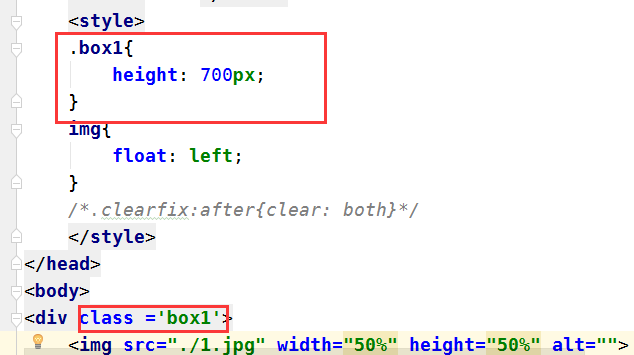




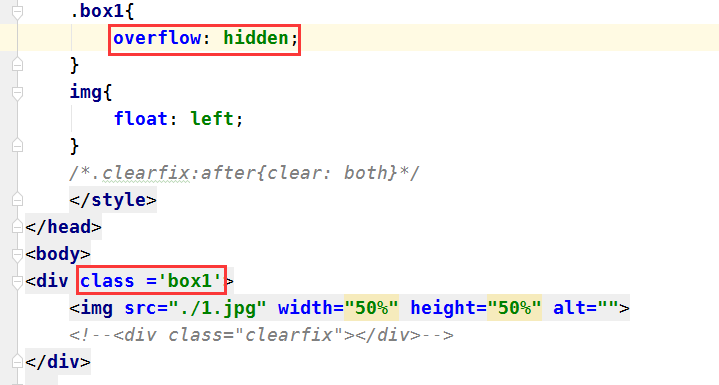
1. 在最后一个子元素的后面加一个空的div，给它样式属性 clear:both（不推荐）



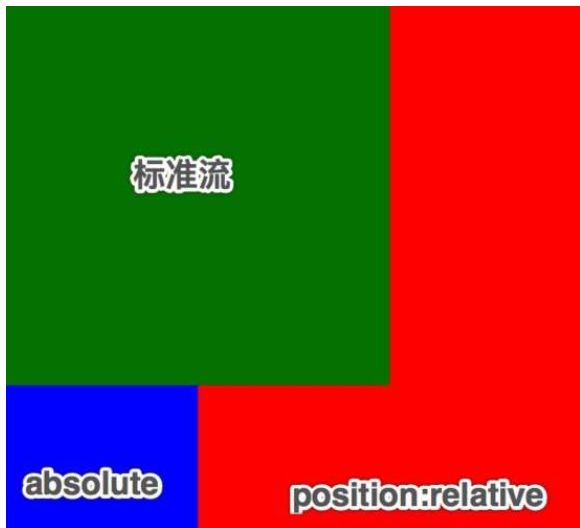
3. 父级div定义height



4. 父级div定义overflow:hidden



## 4.4 定位



****关于定位****

我们可以使用css的position属性来设置元素的定位类型，postion的设置项如下：

1. relative 生成相对定位元素，元素所占据的文档流的位置不变，元素本身相对文档流的位置进行偏移relative [ˈrelətɪv]：相对的 ；postion [pəˈzɪʃn]：位置
2. absolute 生成绝对定位元素，元素脱离文档流，不占据文档流的位置，可以理解为漂浮在文档流的上方，相对于上一个设置了相对或者绝对或者固定定位的父级元素来进行定位，如果找不到，则相对于body元素进行定位。absolute [ˈæbsəluːt]：绝对的
3. fixed 生成固定定位元素，元素脱离文档流，不占据文档流的位置，可以理解为漂浮在文档流的上方，相对于浏览器窗口进行定位。fixed [fɪkst]：固定的
4. static 默认值，没有定位，元素出现在正常的文档流中，相当于取消定位属性或者不设置定位属性
5. inherit 从父元素继承 position 属性的值inherit [ɪnˈherɪt]：继承(金钱、财产等)

****定位元素特性 ：****

绝对定位和固定定位的块元素和行内元素会自动转化为行内块元素

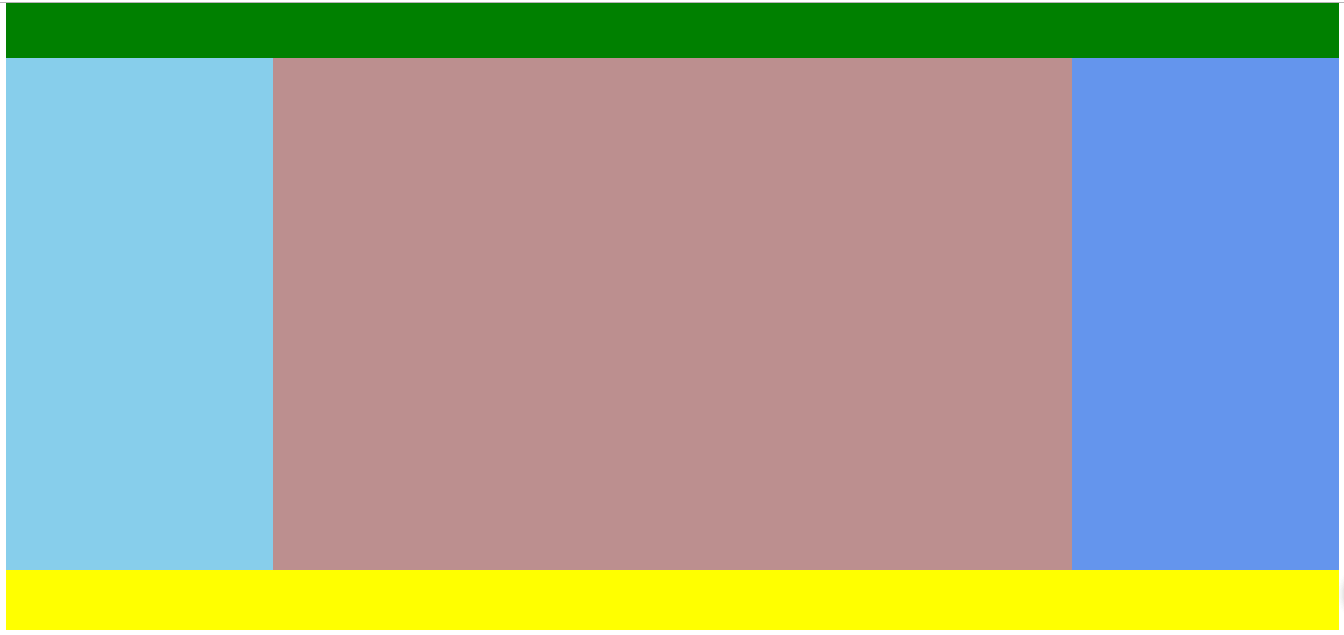
****定位元素层级 ：****

定位元素是浮动的正常的文档流之上的，可以用z-index属性来设置元素的层级

利用position实现网站界面：

<!doctype html>  
<html lang="en">  
<head>  
 <meta charset="UTF-8">  
 <meta name="viewport"  
 content="width=device-width, user-scalable=no, initial-scale=1.0, maximum-scale=1.0, minimum-scale=1.0">  
 <meta http-equiv="X-UA-Compatible" content="ie=edge">  
 <title>Document</title>  
 <style>  
 .header{  
 width: 100%;  
 height: 60px;  
 background: green;  
 }  
 .content{  
 position: relative;  
 }  
 .left{  
 width: 20%;  
 height: 512px;  
 background-color: skyblue;  
 */\*float: left;\*/* }  
 .center{  
 width: 60%;  
 height: 512px;  
 background-color: rosybrown;  
 position: absolute;  
 top: 0;  
 margin-left: 20%;  
 }  
 .right{  
 width: 20%;  
 height: 512px;  
 background-color: cornflowerblue;  
 position: absolute;  
 top: 0;  
 margin-left: 80%;  
 }  
 .footer{  
 width: 100%;  
 height: 60px;  
 background: yellow;  
 */\*position: relative;\*/* }  
 </style>  
</head>  
<body>  
<div class="header"></div>  
<div class="content">  
 <div class="left"></div>  
 <div class="center"></div>  
 <div class="right"></div>  
  
</div>  
<div class="footer"></div>  
</body>  
</html>

运行结果如下：



## 4.5 background属性



****属性解释****

background属性是css中应用比较多，且比较重要的一个属性，它是负责给盒子设置背景图片和背景颜色的，background是一个复合属性，它可以分解成如下几个设置项：

1. background-color 设置背景颜色
2. background-image 设置背景图片地址
3. background-repeat 设置背景图片如何重复平铺
4. background-position 设置背景图片的位置
5. background-attachment 设置背景图片是固定还是随着页面滚动条滚动

repeat [rɪˈpiːt]：重复

实际应用中，我们可以用background属性将上面所有的设置项放在一起，而且也建议这么做，这样做性能更高，而且兼容性更好，比如：“background: #00FF00 url(bgimage.gif) no-repeat left center fixed”，这里面的“#00ff00”是设置background-color；“url(bgimage.gif)”是设置background-image；“no-repeat”是设置background-repeat；“left center”是设置background-position；“fixed”是设置background-attachment，各个设置项用空格隔开，有的设置项不写也是可以的，它会使用默认值。

****举例：****

下面这些例子使用下面这张图片做为背景图：  

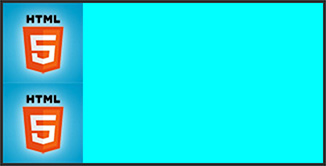

1、“background:url(bg.jpg)”，默认设置一个图片地址，图片会从盒子的左上角开始将盒子铺满。



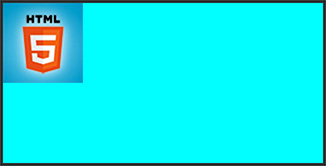
2、“background:cyan url(bg.jpg) repeat-x”，横向平铺盒子，盒子其他部分显示背景颜色“cyan”。



3、“background:cyan url(bg.jpg) repeat-y”，纵向平铺盒子，盒子其他部分显示背景颜色“cyan”。



4、“background:cyan url(bg.jpg) no-repeat”，背景不重复，背景和盒子左上角对齐，盒子其他部分显示背景颜色“cyan”。



5、“background:cyan url(bg.jpg) no-repeat left center”，背景不重复，背景和盒子左中对齐，盒子其他部分显示背景颜色“cyan”。



6、“background:cyan url(bg.jpg) no-repeat right center”，背景不重复，背景和盒子右中对齐，也就是背景图片的右边对齐盒子的右边，盒子其他部分显示背景颜色“cyan”。



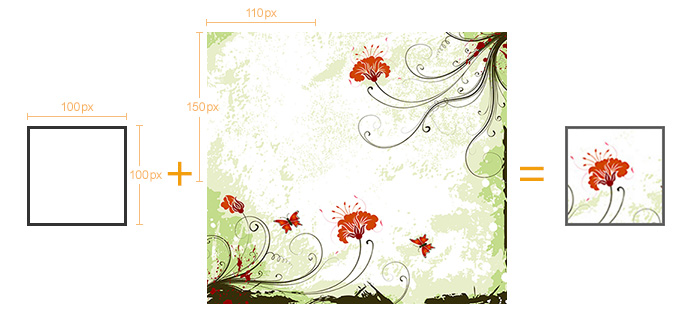
相关代码：

<!DOCTYPE html>  
<html lang="en">  
<head>  
 <meta charset="UTF-8">  
 <title>test background</title>  
 <style type="text/css">  
 .backshow{  
 width:320px;  
 height:160px;  
 border:3px solid #333;  
 float:left;  
 margin:10px;  
 }  
 .bg1{background:cyan url(bg.jpg);}  
 .bg2{background:cyan url(bg.jpg) repeat-x;}  
 .bg3{background:cyan url(bg.jpg) repeat-y;}  
 .bg4{background:cyan url(bg.jpg) no-repeat;}  
 .bg5{background:cyan url(bg.jpg) no-repeat left center;}  
 .bg6{background:cyan url(bg.jpg) no-repeat right center;}  
 </style>  
</head>  
<body>  
<div class="backshow bg1"></div>  
<div class="backshow bg2"></div>  
<div class="backshow bg3"></div>  
<div class="backshow bg4"></div>  
<div class="backshow bg5"></div>  
<div class="backshow bg6"></div>  
</body>  
</html>

例子说明：

代码中使用到的背景图片，可以直接在页面上的背景图片上点右键另存下来，命名为：“bg.jpg”，并且和网页文件存在同一个目录下。

关于背景图片的background-position的设置，设置背景图片水平位置的有“left”、“center”、“right”，设置背景图片垂直位置的有“top”、“center”、“bottom”，水平和垂直的属性值两两组合，就可以把背景图设置到对齐盒子的四个角或者四个边的中心或者盒子的正中心位置。还可以设置具体的像素值来把背景图片精确地定位到盒子的某个位置，特别是对于背景图片尺寸大于盒子尺寸的这种情况，我们可以用数值定位，截取大图片的某一块做为盒子的背景。

比如说，我们想把下边的盒子用右边的图片作为背景，并且让背景显示图片中靠近底部的那朵花：  
  
用上面中间那张图片作为左边那个比它尺寸小的盒子的背景，如果不设置background-position，默认背景图的左上角会和盒子的左上角对齐，如果设置，可以用两个数值定位背景图，上面右边的实现效果设置为：“background:url(location\_bg.jpg) -110px -150px”，第一个数值表示背景图相对于自己的左上角向左偏移110px，负值向左，正值向右，第二个数值表示背景图相对于自己的左上角向上偏移150px，负值向上，正值向下。

实现原理示意图：  


对应代码：

<!DOCTYPE html>  
<html lang="en">  
<head>  
 <meta charset="UTF-8">  
 <title>test background</title>  
 <style type="text/css">  
 */\*.backshow{\*/  
 /\*width:320px;\*/  
 /\*height:160px;\*/  
 /\*border:3px solid #333;\*/  
 /\*float:left;\*/  
 /\*margin:10px;\*/  
 /\*}\*/* .bg{width:100px;  
 height:100px;  
 border:3px solid #666;  
 background:url('locaion\_bg.jpg') -110px -150px;  
 }  
 </style>  
</head>  
<body>  
<div class="bg"></div>  
</body>  
</html>

总结：

4.1 css元素溢出

4.2 块元素、内联元素、内联块元素

4.3 浮动

4.4 定位

4.5 background属性