

2014 年辛星 web 前端教程贺岁版

第二本：css 基础

*****2014 年辛星 web 前端教程贺岁版*****

1.辛星教程最早源于夏季，夏季版中我们还很稚嫩，但是在贺岁版中我们有了不小的进步，下面就是贺岁版的书单了：

编号	书名	是否完工
1	html 基础	已完工
2	css 基础	已完工
3	css 典型应用	已完工
4	Javascript 基础	已完工
5	html5 基础	已完工
6	css3 基础	已完工
7	jQuery 基础	已完工
8	jQuery UI 基础	已完工
9	Bootstrap 基础	已完工

2.辛星系列的教程在我们的呵护下，已经有了一定的成效，它在一定程度上已经能够帮助人们的学习了。

3.当然它很多问题亟待解决，它需要我们的呵护才能走得更远，给更多的人带来实实在在的益处。因此，我们相当期待您的建议奥。

*****交流平台*****

我们的 QQ 群：392328871

我们的贴吧：辛星

希望我们一起构建一个学术的净土，梦想的舞台。

千言万语，汇成一句话：一起努力。

*****致辞*****

1.百度搜索“辛星 web 前端”，相信会有收获奥。

2.辛星----期待您的关注。

*****寄语*****

特色：更新更全更实用。

目标：专业权威可信赖。

纲领：传播编程知识，振兴中华软件。

前进的道路，辛星陪伴您。

只要星哥在，编程充满爱。

*****加入我们吧*****

我们的 QQ 群：392328871

我们的贴吧：辛星

我的个人 QQ：1808347923

我的个人邮箱：xinguimeng@163.com

我的个人博客：blog.csdn.net/xinguimeng

*****共同努力*****

1.每周五都会有新一批的教程出现，风雨中，我们一起努力。

2.当学习成为一种习惯，那么进步就是一种必然。

3.我们是一个图案家友爱奋进的大家庭，愿意加入我们吗？

*****纲领*****

传播编程知识，振兴中华软件。我心永恒，始终如一。

目录

第一节：CSS 简介.....	4
第二节：选择器.....	10
第三节：继承、层叠.....	25
第四节：文字样式.....	28
第五节：盒模型.....	42
第六节：标准流.....	53
第七节：浮动.....	60
第八节：定位.....	73
第九节：布局初体验.....	83
第十节：应用图片.....	91
十一节：杂项知识(调色、注释).....	99
附录：意见反馈.....	103

第一节：CSS 简介

[illegible]

1. 所谓 CSS 也就是 **Cascading Style Sheets** 的缩写，即**层叠式样式表**，也可以被称为**级联样式表**，它是用于指定 **html** 的样式。
2. 在 1996 年 12 月 17 日，CSS 的 1.0 版本发布。
3. 在 1998 年 5 月 12 日，CSS 的 2.0 版本发布。
4. 在 2001 年 5 月 23 日，CSS 的 3.0 版本发布。
5. 在 2007 年 7 月 19 日，CSS 的 2.1 版本发布。
6. 目前还在用的技术就是 CSS 的**两个版本**为 **2.1** 和 **3.0**，不过由于 3.0 的完善工作还在进行，虽然它离最初的发布已经有 13 年之久。

>>>>>>>>> CSS 的优势 <<<<<<<<<<<<<<<<

- 1.如果没有 CSS，那么我们会把网页的格式直接写到 html 中去，这样会造成源代码难以阅读和修改，如果我们想给网页换一个样式，就更麻烦了。
- 2.于是，把网页内容与其表现形式相分离的思想就出现了，CSS 就是这样一门技术。
- 3.我们的 CSS 能对网页中的对象提供精确的控制，支持几乎所有的字体字号样式，还可以编辑网页对象和模型样式，并且具有较强的交互设计。
- 4.而且 CSS 的可读性也很强，而且易学易用。

>>>>>>> CSS 快速入门 <<<<<<<<<<<<<<<<<<<

- 1.对于 CSS 的入门，并不比 html 难，也是一句话入门。
- 2.我们看一下 CSS 的语法吧：

选择器{声明 1; 声明 2;声明 n}

- 3.这里的声明都是使用“属性：值”的形式，而且多个声明之间使用逗号分隔。
- 4.下面我们就来看一个实例吧。

>>>>> 第一个实例 <<<<<<<<<<<<<<<<<<

1.我们先写一个html文件，代码如下：

```
1 <html>
2 <head>
3     <style type="text/css">
4         p{color: green;text-align: center}
5     </style>
6 </head>
7 <body>
8     <p>辛星CSS，传播知识，传递温情</p>
9 </body>
10 </html>
```

2. 那么我们在第 3 行是用 `<style>` 标签使用内嵌的方式写入了 CSS 代码，而第 4 行则是样式的内容，它把 `p` 标签的颜色设置为绿色，而且把文本格式设置为居中。

3.那么我们看一下效果吧:



4.我们可以看到，这里的文字内容变成了绿色，而且居中显示。

5.从上面的例子中，我们也看出来 CSS 的语法很简洁。

>>>>>>>>>开发工具<<<<<<<<<<<<<<<<

1.和开发 html 一样，我们的开发工具依然只需要两个：①文本编辑器②浏览器。

2.而我这里的文本编辑器依旧使用的是 **Sublime Text 2**，而浏览器使用的依然是**谷歌浏览器**。

3.包括我们之后写 Javascript 教程的时候，也会一直使用 Sublime Text 2,仅仅是因为它比较帅气。


3.它比第一种方式有优点，可以把 CSS 代码拿出来统一管理，但是它仍然和 HTML 文件是混杂在一起的。

4.如果一个站点的 html 文件很多，那么 CSS 文件也就很难统一管理，而想换一套主题也很难。

>>>>>第三种===链接式<<<<<<<<<<<<<<<<<<<

1.第三种方式就是**链接式**，它是在一个<link>标签中把css文件导入进来。

2.我们先看一个例子，我们首先写一个文件，命名为 my.css，然后写入如下代码：

A screenshot of a code editor with two tabs: 'my.html' and 'my.css'. The 'my.css' tab is active, showing the following CSS code:

```
1 p{  
2     color: blue;  
3     text-align: center;  
4 }  
5
```

The code is syntax-highlighted: 'p{' is orange, 'color:' is light blue, 'blue;' is green, 'text-align:' is light blue, 'center;' is green, and '}' is orange. Line numbers 1 through 5 are on the left.

3. 然后我们在同级目录下新建一个 my.html 文件，写入如下代码：

```
1 <html>
2 <head>
3     <link rel="stylesheet" type="text/css" href="my.css">
4 </head>
5 <body>
6     <p>
7         辛星CSS，传播知识，传递温情
8     </p>
9 </body>
10 </html>
```

4.我们看到，这里的<link>标签中的rel被指定为“stylesheet”也就是样式表，而type则是加载的类型，而href则是说明了要导入的文件的路径。

5.那么该网页的效果图如下:



7.我们要导入多个css文件的时候，直接使用多个<link>标签就可以了。

1. 第四种就是**导入式**，它之所以被命名为导入式，很大程度上依赖于它的**import** 这个单词。

```
1 <html>
2 <head>
3     <style type="text/css">
4         @import "my.css";
5     </style>
6 </head>
7 <body>
8     <p>
9         辛星CSS，传播知识，传递温情
10    </p>
11 </body>
12 </html>
```

4.效果和上一个是一样的，就不给大家截图展示了。

第二节：选择器

[illegible]

1.我们再回忆一下 CSS 的语法吧，也就是那句：

选择器{声明 1; 声明 2;声明 N}

2.也就是说，这个语句分为两部分，第一部分指明了给谁的样式，而大括号中的部分则说明了使用什么样的样式。

3.因此，我们需要了解一下 CSS 中的各种选择器，其中选择器的用法还是非常灵活的，我们逐一来认识。

[illegible]

1.要说id选择器，就必须说一下id这个属性。在html的元素中，每个元素都可以设置一个属性，那就是id，它的作用就是唯一的标识了这个标签。

2.其中 id 的取名还是很自由的，通常来说建议使用全部小写的英文字母，可以带有下划线和数字。

3.我们在 CSS 中就可以使用#号加 id 名的格式来作为选择器。

4.我们看一个例子吧，首先我们在 my.html 中写入如下内容：

```
1 <html>
2 <head>
3     <link rel="stylesheet" type="text/css" href="my.css">
4 </head>
5 <body>
6     <p id="xin">辛星CSS，传播知识，传递温情</p>
7     <p id="star">我是宅男，你是吗</p>
8     <p>我就想知道，谁能比我更宅</p>
9 </body>
10 </html>
```

5.我们在这个html文件中定义了三个p元素，并且前两个都指定了相应的id属性，而最后一个则没有指定。

6.那么我们在同级目录下新建一个 my.css 文件，写入代码如下：

```
1 #xin{
2     color: blue;
3     text-align: center;
4 }
5 #star{
6     color: red;
7     text-align: center;
8 }
9
```

7. 我们可以看出，这两个 CSS 语句直接写就可以了，它们之间并不需要其他的什么符号。

8.那么我们看一下效果吧:



9.对于 id 选择器，我们就介绍到这里了。

>>>>>>类选择器<<<<<<<<<<<<<<<<<<<

1.对于类选择器，也有人叫 **class 选择器**，其实都是一回事。

2.所谓类选择器，首先说一下很多html的标签都可以设置class属性，它不能像id那样唯一的标识一个标签，但是它能唯一的标识一类标签。

3.也就是说那些具有相同 class 属性的标签使用同样的 CSS 代码来渲染。

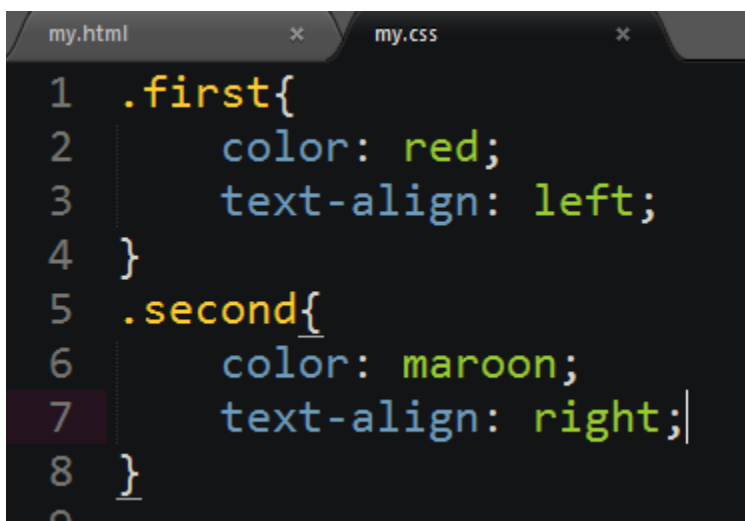
4.在 CSS 文件中，我们使用一个点号.加类名的方式来表示选择器。

5.首先我们写一个 my.html 文件吧：



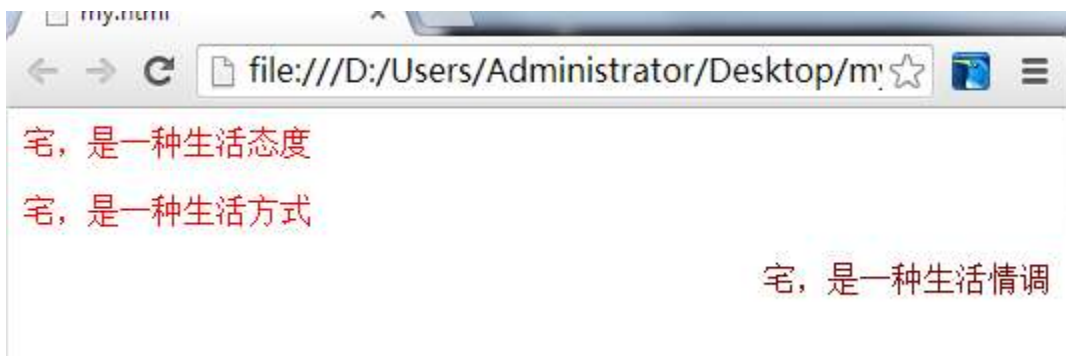
```
1 <html>
2 <head>
3   <link rel="stylesheet" type="text/css" href="my.css">
4 </head>
5 <body>
6   <p class="first">宅，是一种生活态度</p>
7   <p class="first">宅，是一种生活方式</p>
8   <p class="second">宅，是一种生活情调</p>
9 </body>
10 </html>
```

6.然后我们在同级目录下修改 my.css 文件为：



```
1 .first{
2   color: red;
3   text-align: left;
4 }
5 .second{
6   color: maroon;
7   text-align: right;
8 }
```

7.那么我们看一下效果吧：



8.是不是和读者朋友们想的一样呢?

>>>>>> 多风格的类选择器 <<<<<<<<<<<<<<<<<<<<<

- 1.对于一个标签来说,我们可以使用多个类选择器的属性。
- 2.也就是说,我们的html的元素可以在class属性中使用多个class的值。
- 3.我们首先看一下my.html中的代码吧:

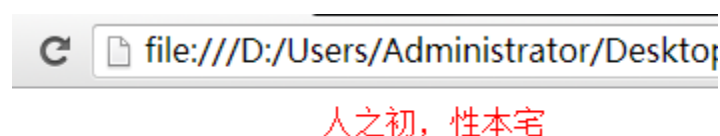
```
1 <html>
2 <head>
3     <link rel="stylesheet" type="text/css" href="my.css">
4 </head>
5 <body>
6     <p class="one two">
7         人之初, 性本宅
8     </p>
9 </body>
10 </html>
```

4.可以看出,这里的<p>标签中的class属性是使用了one和two两个类的,因此它会同时拥有这两个类的样式。

5.然后我们看一下此时的my.css中的代码:

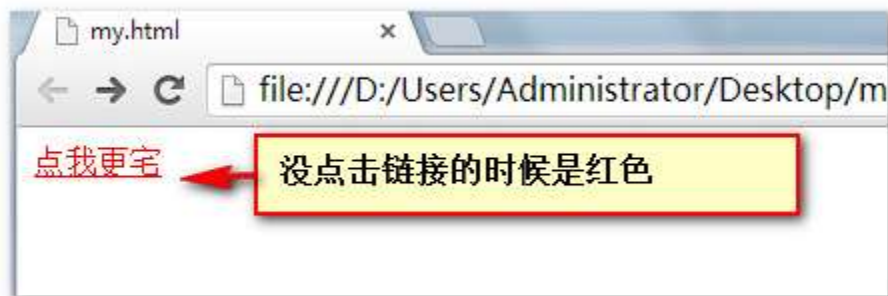
```
1 .one{
2     color: red;
3 }
4 .two{
5     text-align: center;
6 }
7
```

6.那么它的效果图:



6.此时我们看一下效果吧:

首先就是没有点击该超链接的时候是红色--->



然后就是鼠标放到这上面的时候变成棕色---->



接下来就是我们鼠标进行点击的时候变成橘红色:



最后就是当我们访问完该超链接之后变成绿色:



7.那么我们看完了上述的效果之后,相信大家开始了解伪类的作用了,但是我们还是给出书面说明吧,看下面的表格:


```

<html>
<head>
  <link rel="stylesheet" type="text/css" href="my.css">
</head>
<body>
  <p>
    <span>我是辛星</span>
    -----
    <span>典型技术宅</span>
  </p>
  <span>宅男的生活，你不懂</span>
</body>
</html>

```

6.然后我们看一下最终效果吧：



7.我们可以看到，前两个由于是在<p>标签中的，因此被设置为红色，而最后一个标签中的内容则是无样式，也就是使用默认的样式，因此是黑色的。

8.这里的父选择器和后代的选择器都是使用的标记选择器，其实也可以使用我们前面介绍过的类选择器和 id 选择器的，只要说清楚选择器的父子关系，就是没有问题的。

9.那么我们再来看一个例子吧。

>>>>>>> 后代选择器的另一个例子 <<<<<<<<<<<<

1. 为了帮助朋友们更好的理解后代选择器，我们在写一个关于后代选择器的范例。

2. 我们在 my.html 中书写如下代码：

```
<html>
<head>
  <link rel="stylesheet" type="text/css" href="my.css">
</head>
<body>
  <p>
    <span class="one">我是辛星</span>
    -----
    <span class="two">典型技术宅</span>
  </p>
</body>
</html>
```

3. 然后在 my.css 中书写如下代码：

```
p .one{
  color: blue;
}
p .two{
  color: red;
}
```

4. 那么它的效果就是下面这样子的：



5. 应该很好理解吧，可能有人说，这里不写前面的 p 不也一样吗？确实对于这个例子来说，是一样的。但是，当我们的 较多之后，就会看出差别了。

6. 希望大家可以慢慢体会一下后代选择器，应用很广泛。


```

<html>
<head>
  <link rel="stylesheet" type="text/css" href="my.css">
</head>
<body>
  <p class="xin">我是辛星---小小技术宅</p>
  <p>猫儿---辛星得力助手</p>
  <h2 class="xin">技术宅---技术狂</p>
</body>
</html>

```

6.其实这个 my.html 和上一小段的 my.html 是完全一样的，那么我们再看一下此时的 my.css 文件吧：

```

p,.xin{
  color: red;
}

```

7.可以看到我们在 p 和 .xin 之间加了一个逗号，那么此时我们看一下效果吧：



8.其实很简单啦，前两行中有<p>标签，而最后一行的 class 属性是“xin”，因此就会有这样的效果了。

9.聪明的读者朋友们理解了没？


```
#p1{
  font-family: Arial;
}
#p2{
  font-family: "Times New Roman";
}
```

8.那么我们可以看出这是两个不同的字体，我们看一下界面效果吧：



9.大家可以看到，上面这两行使用的是两种字体，单纯的看一下冒号就能看出端倪，第二个占据的空间更宽一些。

10.在 font-family 的属性中可以指定多个字体，它们之间使用逗号分隔即可，比如：

```
font-family:Arial,"Times New Roman";
```

11.那么究竟哪些字体好看呢？一般来说设置上面的两个再加上一个“宋体”、“微软雅黑”这几个，是比较常用一些的。

12.当然，更加漂亮的字体，我们也可以使用图片的方式来代替。

>>>>>>>文字大小<<<<<<<<<<<<<<<<<<<

1.在 css 中，我们通过 font-size 来控制字体大小，那么这里要说一下 css 使用的单位了。

2.一般我们可以用 px 来指定单位为“像素”，1px 表示一个像素。

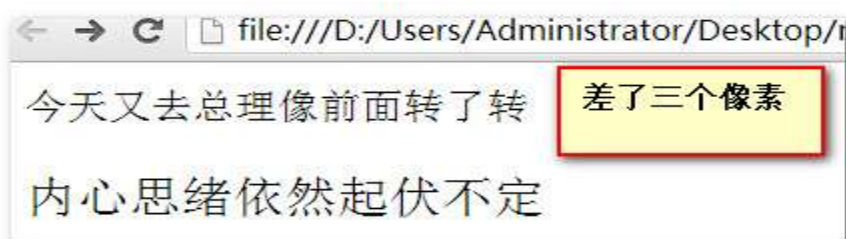
- 3.此外还有 em 和 ex 两个单位，其中 1em 表示的长度是其父元素中字母 m 的标准宽度，1ex 则是父元素中字母 x 的标准高度。
- 4.也就是说 em 和 ex 都是相对父元素的大小改变而改变的，这就使得页面的风格更加一致。
- 5.当然还有 in 表示英寸，cm 表示厘米，mm 表示毫米等等，我感觉不是很常用，就不介绍了。
- 6.那么我们首先写一个 my.html 代码如下：

```
<html>
<head>
  <link rel="stylesheet" type="text/css" href="my.css">
</head>
<body>
  <p id="xin">今天又去总理像前面转了转</p>
  <p id="star">内心思绪依然起伏不定</p>
</body>
</html>
```

- 7.然后我们写 my.css 代码如下：

```
#xin{
  font-size: 12px;
}
#star{
  font-size: 15px;
}
```

- 8.那么我们看一下效果吧：



9. 这里我们可以看到，这两行文字差了三个像素，还是能够看出大小的差别的。

10.那么我们一般设置为多少合适呢？一般来说，12px 是一个比较均衡的选择。

11.当然，我们也要具体情况具体分析，在合适的情况下选择合适的大小，是最重要的。

12.其实也可以使用百分比的形式，不过我还是建议统一使用像素，也就是 px 的格式来调整大小。

[illegible]

1.行高还是蛮重要的，它确定了我们的段落中两行文字之间的大小。

2.行高使用 **line-height** 这个属性来设置，它可以使用 px 为单位，也就是使用像素的形式，比如 12px。

3.行高的值，也可以不加任何单位，此时行高就是与字体的比值，比如 `line-height:1.5`；表示行高是字体高度的 1.5 倍。

4.为了更好的演示行高的作用，我们看如下的 my.html 代码：

```
<html>
<head>
    <link rel="stylesheet" type="text/css" href="my.css">
</head>
<body>
    <p id="p1">
        宅，是一种生活态度<br />
        宅，是一种生活方式<br />
        宅，是一种生活习惯<br />
    </p>
    <p id="p2">
        当学习成为一种习惯<br />
        那么进步就成为必然<br />
        前途就会愈加光明<br />
    </p>
</body>
</html>
```


2.对于是否加粗，我们使用 `font-weight` 来表示，如果属性取值为 `bold`，则表示为加粗，如果为 `normal`，则表示不加粗。

3.我们来看一下 `my.html` 中的代码：

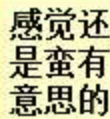
```
<html>
<head>
  <link rel="stylesheet" type="text/css" href="my.css">
</head>
<body>
  <p id="p1">
    宅，是一种生活态度
  </p>
  <p id="p2">
    当学习成为一种习惯
  </p>
</body>
</html>
```

4.那么我们再设置一下 `my.css` 代码吧：

```
#p1{
  font-weight: bold;
}
#p2{
  font-weight: normal;
}
```

5.那么究竟有何不同呢？我们看下效果吧：





>>>>>>删除下划线<<<<<<<<<<<<<<<<<

3.我们先看如下的 my.html 代码:

```
<html>
<head>
  <link rel="stylesheet" type="text/css" href="my.css">
</head>
<body>
  <a href="#">辛星</a>
</body>
</html>
```

```
a{
  color: red;
  text-decoration: none;
  font-size: 12px;
}
```


4.我们先看一下 my.html 中的代码吧：

```
1 <html>
2 <head>
3   <link rel="stylesheet" type="text/css" href="my.css">
4 </head>
5 <body>
6   <span>卧薪尝胆十六年，背卧沙场风雨旋</span><br />
7   <span>多少良言对你说，多少风雨为你遮</span><br />
8   <span>笔墨三千断笔墨，只为兄弟和游客</span>
9 </body>
10 </html>
```

5.然后我们看一下 my.css 中的代码吧：

```
1 body{
2   background: #eee;
3 }
4 span{
5   font-size: 18px;
6   line-height: 18px;
7   color: #777;
8 }
```

6.然后我们看一下效果吧：




```

<html>
<head>
  <link rel="stylesheet" type="text/css" href="my.css">
</head>
<body>
  <p>
    辛星<br />
    但使龙城飞将在，传播小伙就是帅。<br />
    万里长城万里长，传播小伙就是狂。<br />
  </p>
</body>
</html>

```

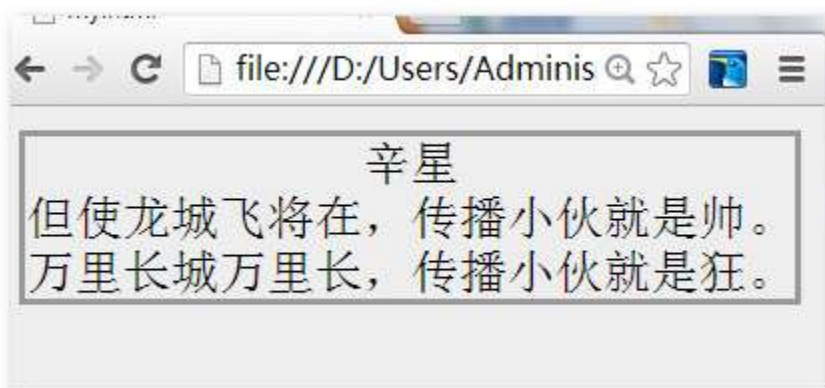
7.那么我们看一下相应的 my.css 中的内容吧：

```

body{
  background: #eee;
}
p{
  text-align: center;
  border-color: #999;
  border-style: solid;
  border-width: 2px;
}

```

8.那么我们看一下效果吧：



8.然后我们设置一下 my.css 的值:

```
body{
    background: #eee;
}
p{
    text-align: center;
    border-color: red blue yellow orange;
    border-style: solid dotted;
    border-width: 2px;
}
```

9.那么最终的结果就是这样了:



10.看到了吧，我们的格式设置还是蛮灵活的。

>>>>>简写形式<<<<<<<<<<<<<<<<<<<

1.如果四条边框取值相同的话，也就是 border-xxx 都只有一个值，那么我们就可以简写为一个属性。

2.也就是使用 **border** 属性来设置，它的格式是首先写 **border-width**，然后写 **border-color**，最后写 **border-style**。

3.比如我们可以这么设置:

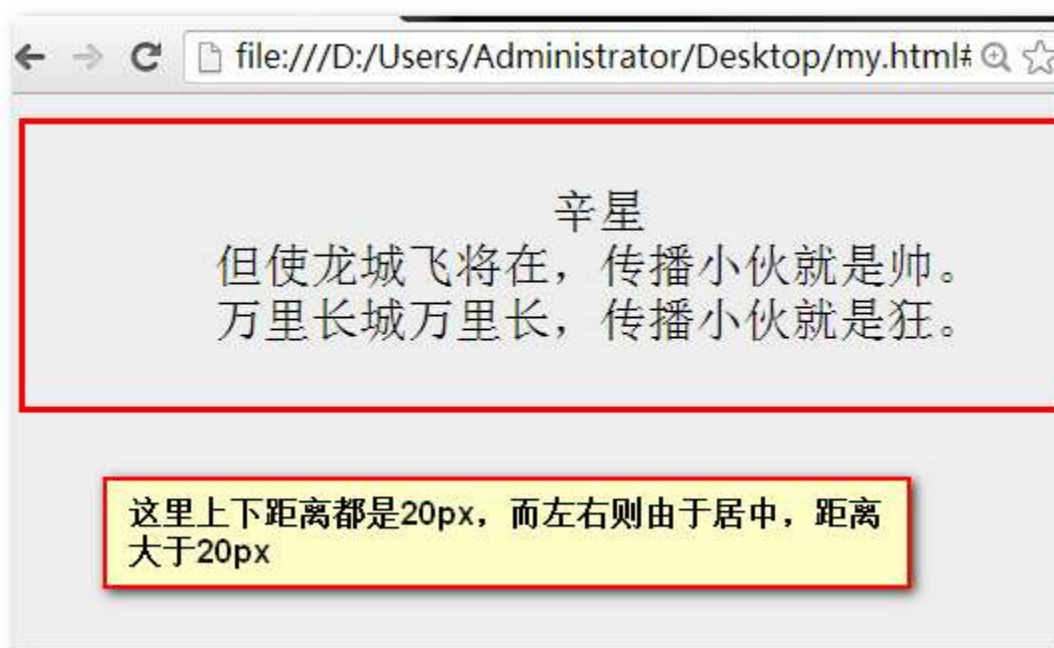
border:2px red solid;

4.上面设置会比单独设置要方便一些。

5.然后我们看一下 my.css 的内容:

```
1 body{
2     background: #eee;
3 }
4 p{
5     text-align: center;
6     border-color: red ;
7     border-style: solid;
8     border-width: 2px;
9     padding: 20px;
10 }
```

6.上面代码的第 9 行就说明了内容到边界的距离为 20px，那么我们看一下效果吧：



7.我们可以看到，对于内容的顶部和边框的距离为 20px，而内容的底部和边框的距离也为 20px，而左右的距离，由于内容居中的原因，它的距离大于 20px。

8.当我们拖拽页面的时候，我们会发现文字和边框至少会有 20px 的距离，这也就是 padding 在起作用。

盒模型从内向外的内容	
属性	汉语描述
content	元素内容
padding	内边距
border	边框
margin	外边距

3.然后是边框的属性介绍:

盒模型边框的属性	
属性	汉语描述
border-color	边框颜色
border-width	边框宽度
border-style	边框线条风格

4.然后是取值从1个到4个的含义:

属性取值从1个到4个表示的含义

取值数目	取值意义
1	四个方向都使用此值
2	上下方向使用第一个值 左右方向使用第二个值
3	上方向使用第一个值 下方向使用第三个值 左右方向使用第二个值
4	上方向使用第一个值 右方向使用第二个值 下方向使用第三个值 左方向使用第四个值

5.对于盒模型的内部结构,我们大致就讲到这里吧。

6.本来想结束的,后来想想,还是给大家推荐一下使用浏览器自带的工具来研究盒模型吧。

第六节：标准流

>>>>>>>说明<<<<<<<<<<<<<<<<<<<

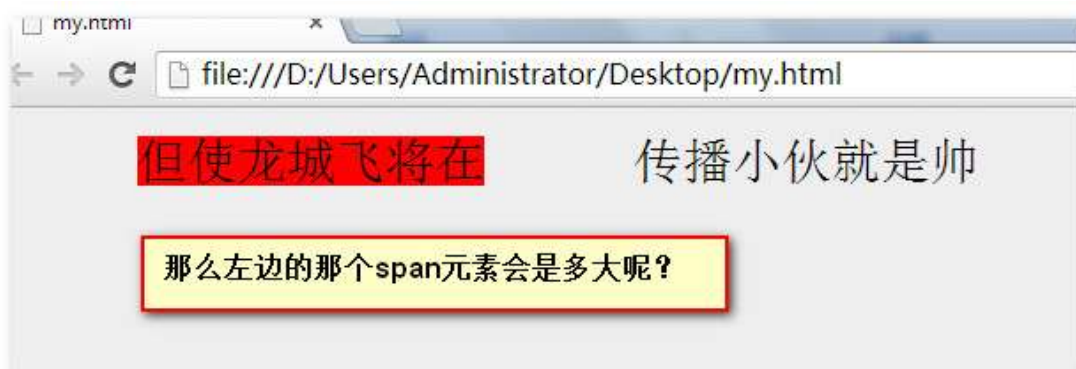
- 1.本节课我们主要讲的是多个盒子的显示问题。
- 2.对于 DOM 这个概念，大家可以考虑不去了解，等后面学习 Javascript 的时候，我们还会介绍。

>>>>>>DOM<<<<<<<<<<<<<<<<<

- 1.第一个概念就是 DOM，也就是 Document Object Model 的缩写，翻译成汉语也就是“文档对象模型”。
- 2.一个网页的所有元素组织在一起，就构成了一颗 DOM 树。而 DOM 树是由若干节点组成的，而 html 中的每个元素都是一个节点。
- 3.比如咱们以前经常写的 body 就是浏览器窗口中显示的队友对象的根节点，而 p 又是 body 的子节点。
- 4.其实每一个 html 的元素都可以作为一个节点存在，对于这个节点，html 控制它的内容结构，css 控制它的表现形式。
- 5.对于 DOM 的概念，我们在学习 Javascript 的时候会深入讲解，这里只是暂时提一下。

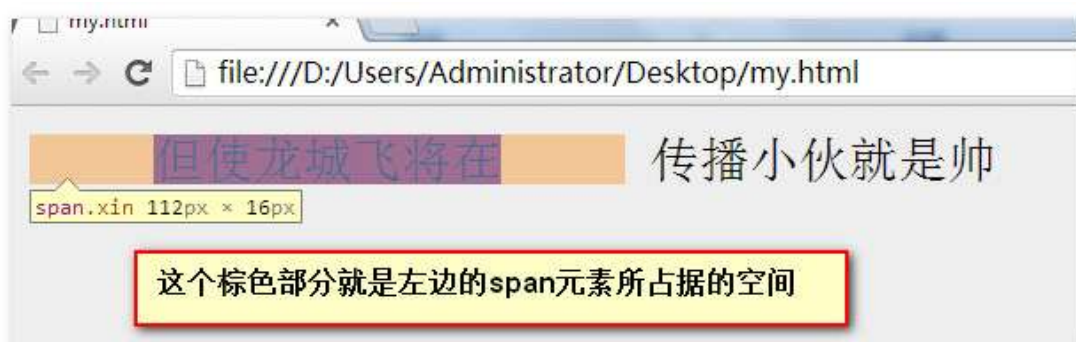
>>>>>>>标准流<<<<<<<<<<<<<<<<<<

- 1.有了上面的概念，我们引入“标准流”的概念，它是“文档标准流”的简称，也有人称之为“普通流”。
- 2.所谓标准流，也就是在不使用任何的 CSS 的情况下，各种元素的排列规则。
- 3.标准流的排列非常简单，总的来说八个字：从上到下，从左到右。
- 4.可能大家对标准流的理解还只是停留在概念上，为了让大家有一个更加深入的认识，我们用代码实战的形式来理解一下吧。
- 5.本节我们研究的是盒子之间的互相作用，因此我们接下来的代码主要研究位置关系。



5.这里由于第一个span的margin是40px，而第二个span的margin是没有指定的。

6.其实我们可以看一下span的效果简图：

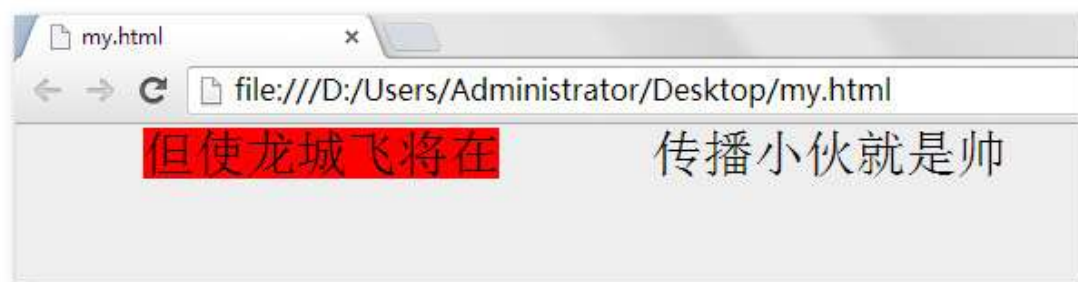


7.这里我使用的是谷歌浏览器的盒模型查看，那么我们可以看出，左边的span元素的左边和body是有一定的距离的，这里默认是8px，而右边和右边的span元素之间也有一些距离，当然也是8px了。

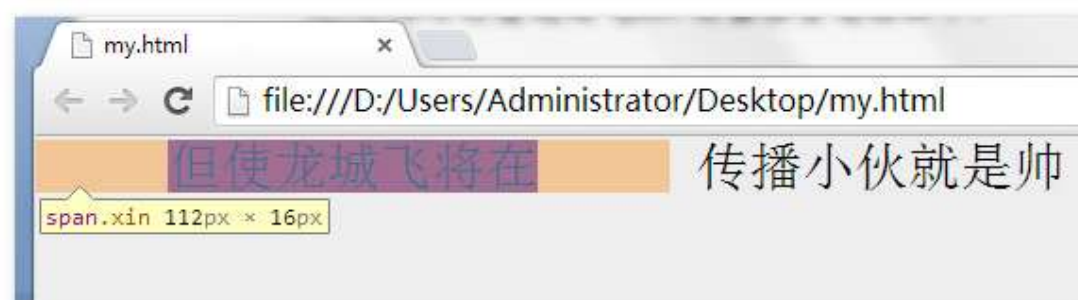
8.那么我们在my.css中设置一下body的padding和margin属性为0，即：

```
body{
    background: #eee;
    padding: 0px;
    margin: 0px;
}
.xin{
    background: red;
    margin: 40px;
}
```

9.此时我们会看到效果如下:



10.而此时的左边的 span 的盒模型简图如下:



11.这里我想说的是 body 本身也是一个盒子, 而它又包含两个 span 的盒子, 那么这些盒子的横向排布就是上面介绍的内容。

>>>>>>>>>>>>>横向排布继续<<<<<<<<<<<<<<

1.那么如何去计算两个盒子之间的距离呢? 比如两个盒子是左右分布的, 如下图:



2.那么这两个盒子的距离怎么计算呢?其实计算方式还是蛮简单的, 注意这里计算的标准是**两个盒子的边框之间的距离**。

3.这两个盒子的**边框之间的距离**=**盒子 1 的 margin-right-width 值+盒子 2 的 margin-left-width 值**。

4.应该很好理解吧, 而且从上一小段我们看出, 所谓的 **background** 其实就是 **content+padding**.注意是**不加 border** 的奥。

5.这是左右的计算, 其实对于上下的计算也是一样的, 关键是计算两个 border(也就是边框)之间的距离。


```
<html>
<head>
  <link rel="stylesheet" type="text/css" href="my.css">
</head>
<body>
  <div class="xin">
    <div class="star">内部内容</div>
    <div class="new">底部内容</div>
  </div>
</html>
```

4.然后我们看一下它的css的内容:

```
.xin{
  background: #feb;
  padding: 10px;
  border: 1px solid red;
  margin: 0px;
}
.star{
  background: #adf;
  margin-top: 30px;
  margin-bottom: 0px;
  padding: 15px;
  border: 1px solid #049;
}
.new{
  background: #396;
  margin: 0px;
  padding: 0px;
  border: 1px solid #786;
}
```


第七节：浮动

>>>>>>>>>说明<<<<<<<<<<<<<<<<<<<

- 1.在讲布局之前，必须得讲一下浮动。
- 2.而浮动确实也是最容易让新手朋友们迷茫的，往往自己想好了浮动的效果，但是代码却很难去实现。
- 3.那么我们在学习阶段必须重视原理这一块，也就是我们的重点还是在学习理论上。
- 4.其实我的html、css教程夏季版中，我感觉里面对浮动讲解的就已经比较清晰了，因此这一节很大程度上和那部分很相似。

>>>>>>>>>>说明<<<<<<<<<<<<<<<<

- 1.对于浮动，我们使用css的float属性来定义，值为left表示向左浮动，值为right表示向右浮动。
- 2.本节为了简单起见，使用四个div作为例子来研究浮动，前面也说过，div只是一个单纯的块级元素，对于不同的div，我们给予不同的背景色来显示。
- 3.然后我们通过几种比较常见的浮动形式作为范例，来研究一下浮动的基本内容。

[illegible]

- 1.星哥看过N个人的总结，浏览过N个帖子之后，把关于浮动的经验总结为四个原则，希望对您有所帮助。
- 2.这四个原则是：①一旦某个元素设置了浮动，它就脱离了标准流。
②标准流是从上到下、从左到右排列的。③浮动确定元素的左右位置。
④浮动的元素默认在标准流之上。
- 3.随着学习的深入，我们会时刻用这四条原则来验证，来判断。
- 4.当然，随着时代的进步，期待更简单的原则出现，这样我们学习起来会更加轻松简单。

>>>>>>>不浮动的情况<<<<<<<<<<<<<<<<

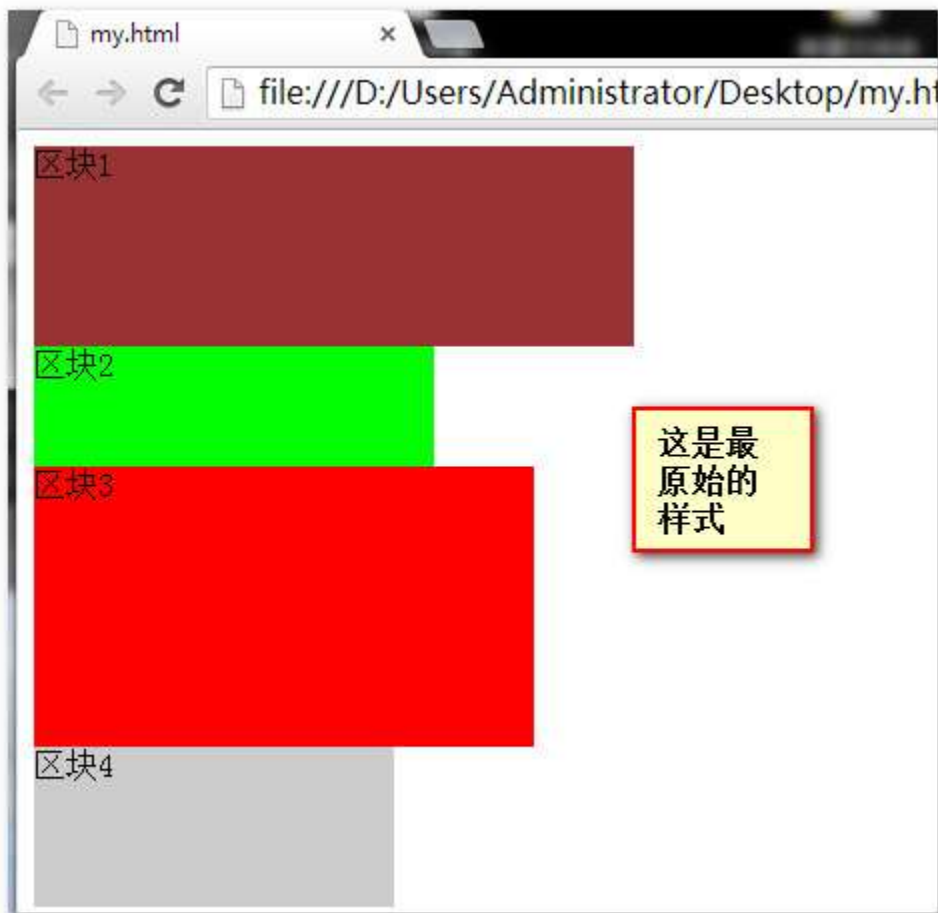
1.我们首先写一个 my.html 代码如下:

```
<html>
<head>
  <link rel="stylesheet" type="text/css" href="my.css">
</head>
<body>
  <div id="xin1">区块1</div>
  <div id="xin2">区块2</div>
  <div id="xin3">区块3</div>
  <div id="xin4">区块4</div>
</body>
</html>
```

2.然后我们看一下 my.css 中的内容吧:

```
#xin1{
    background: #933;
    height: 100px; width: 300px;
}
#xin2{
    background: #0F0;
    height: 60px; width: 200px;
}
#xin3{
    background: #F00;
    height: 140px; width: 250px;
}
#xin4{
    background: #CCC;
    height: 80px; width: 180px;
}
```


3.也就是说我们这里只是给它们设置了不同的大小和背景色,然后我们看看不浮动的时候的效果吧:



4.上面的四个区块是在标准流中的，它们是从上到下的顺序排列，接下来我们就要考虑浮动的情况了。

>>>>>>>一个元素的浮动<<<<<<<<<<<<<<<<

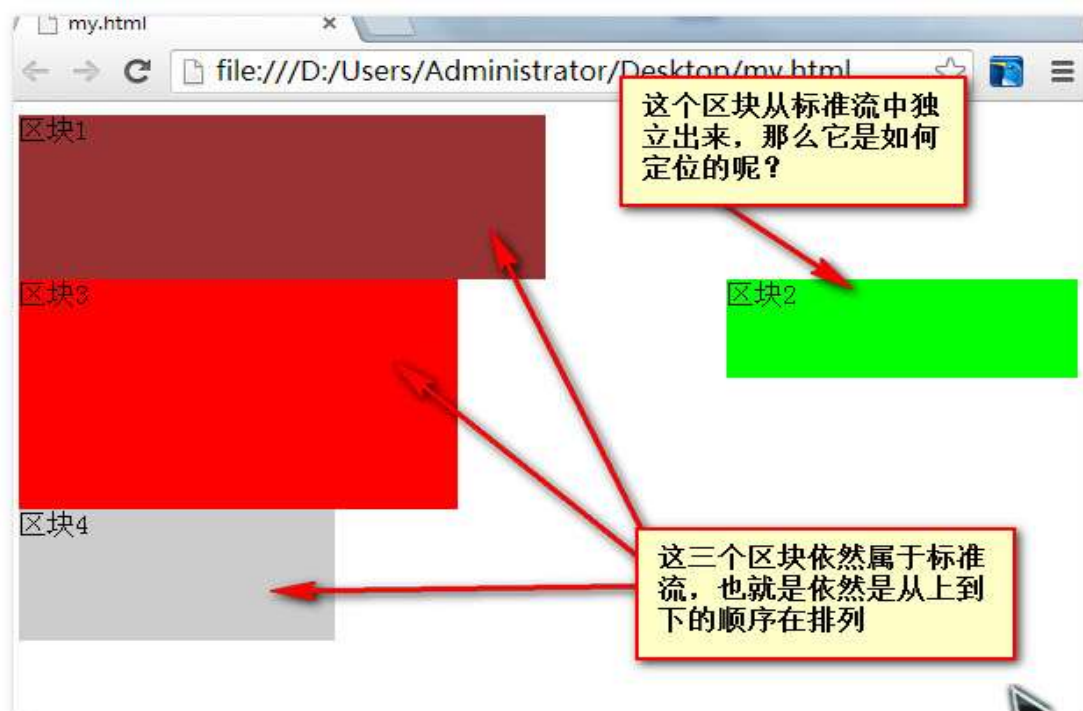
1.现在我们首先让一个元素进行浮动吧，也就是我们让第二个div右浮动。

2.我们在 my.css 中修改#xin2 的代码如下:

```
4 }
5 #xin2{
6     float:right;
7     background: #0F0;
8     height: 60px; width: 200px;
9 }
```

添加这一行代码表示使用右浮动的属性

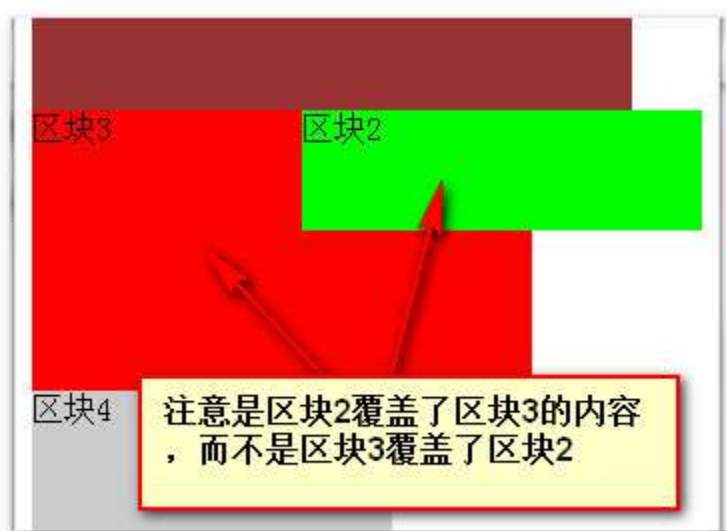
3.然后我们看一下效果吧:



4.那么为什么会出现这样的效果呢? 这里我们说明一下, 首先是区块1、区块3、区块4依然是在标准流中, 它们还是从上到下的顺序进行排列。

5.而区块2从标准流中脱离出来, 它由于是右浮动, 因此它的最右边肯定是和浏览器的右边框是对齐的。它的顶部是和标准流中的上一个元素的底部对齐的。

6.那么我们拖动浏览器的大小, 看下效果吧:



7.注意这里的区块 2 会遮住区块 3，也就是上述四条原则中的第四条：**浮动的元素位于标准流之上。**

8.可能有人会说：那么区块 2 如果左浮动呢？这里我们在 my.css 中修改代码为：

```
3     height: 100px; width: 300px;
4 }
5 #xin2{
6     float: left;
7     background: #0F0;
8     height: 60px; width: 200px;
9 }
```

把右浮动改为左浮动

9.那么我们看一下效果吧：



10.可以看到，这里的标准流中还是正常显示，这里区块 2 的纵向位置是不变的，只是横向位置跑到了左边。

11.当然，**浮动的元素会遮住不浮动的元素。**

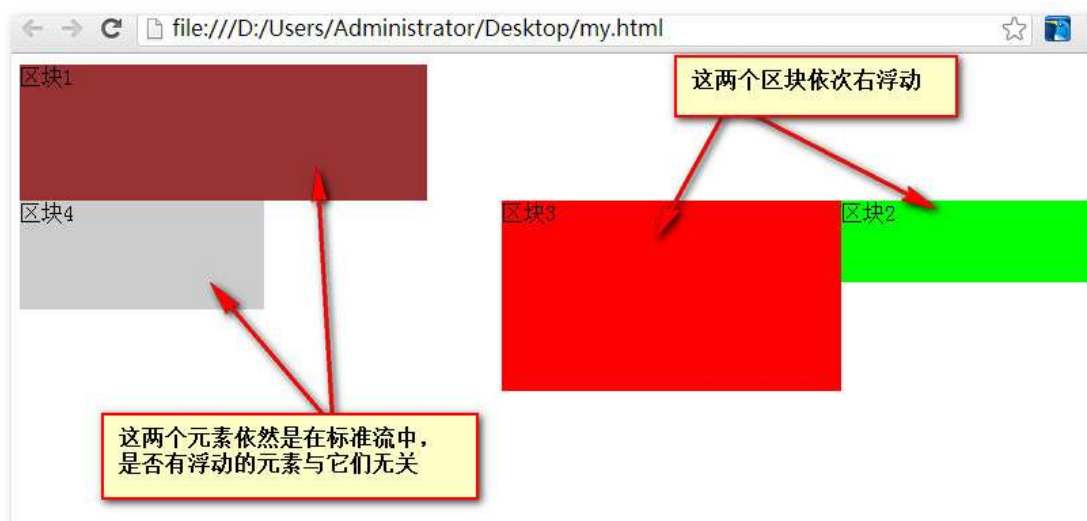
>>>>>>>两个元素一起浮动<<<<<<<<<<<<<<<<<<<<

1.我们经常会遇到多个元素一起浮动的例子，那么本段内容先研究两个元素一起浮动的情况。

2.首先我们修改 my.css 样式如下：



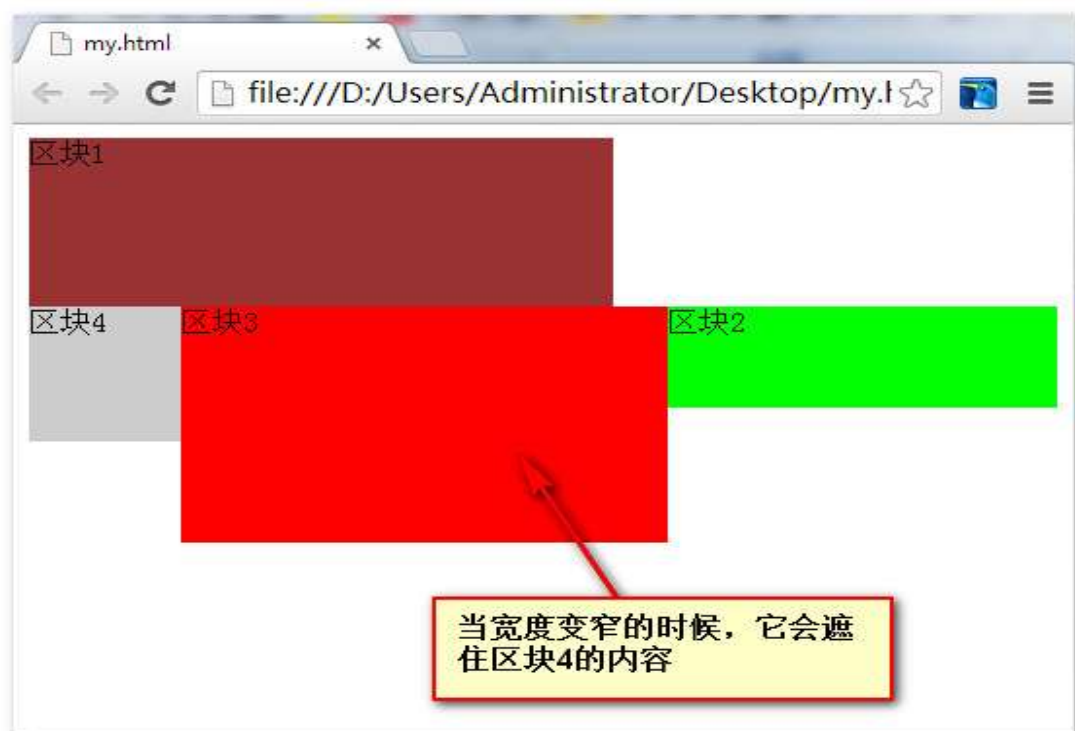
3.然后我们看一下效果吧：



4.这里我们看到区块1、区块4依然在标准流中，它们的布局是不受外界影响的，而区块2和区块3都是右浮动，也就是它们的上一个在标准流中的元素就是区块1，它们的顶部默认都对齐区块1的底部。

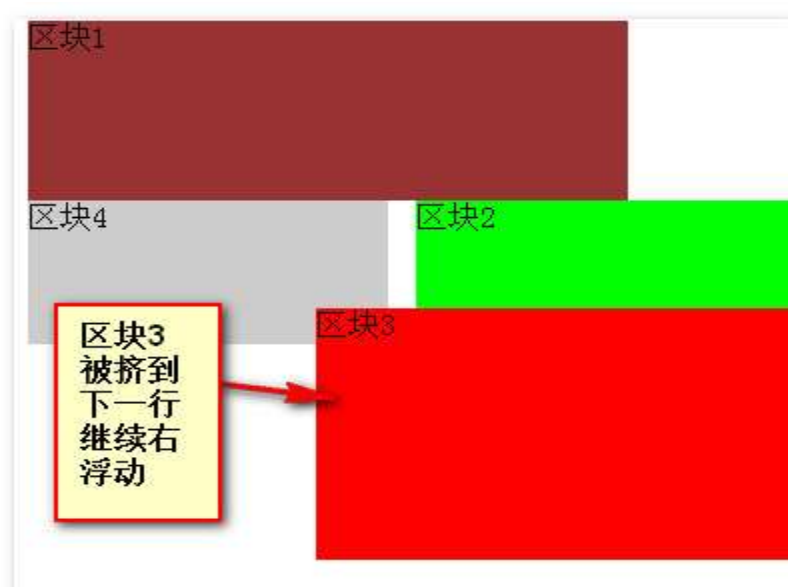
5.由于区块2和区块3都是右浮动，不过区块2首先浮动，然后才是区块3再浮动，也就有了上面的效果。

6.那么我们让区块变窄，发现效果如下：



7.也就是说，浮动的元素会遮住不浮动的元素，但是只要浮动的元素能够在同一行中显示，它的高度是不会变的。

8.我们可以让它变窄，发现此时区块3被挤到下一行去了：



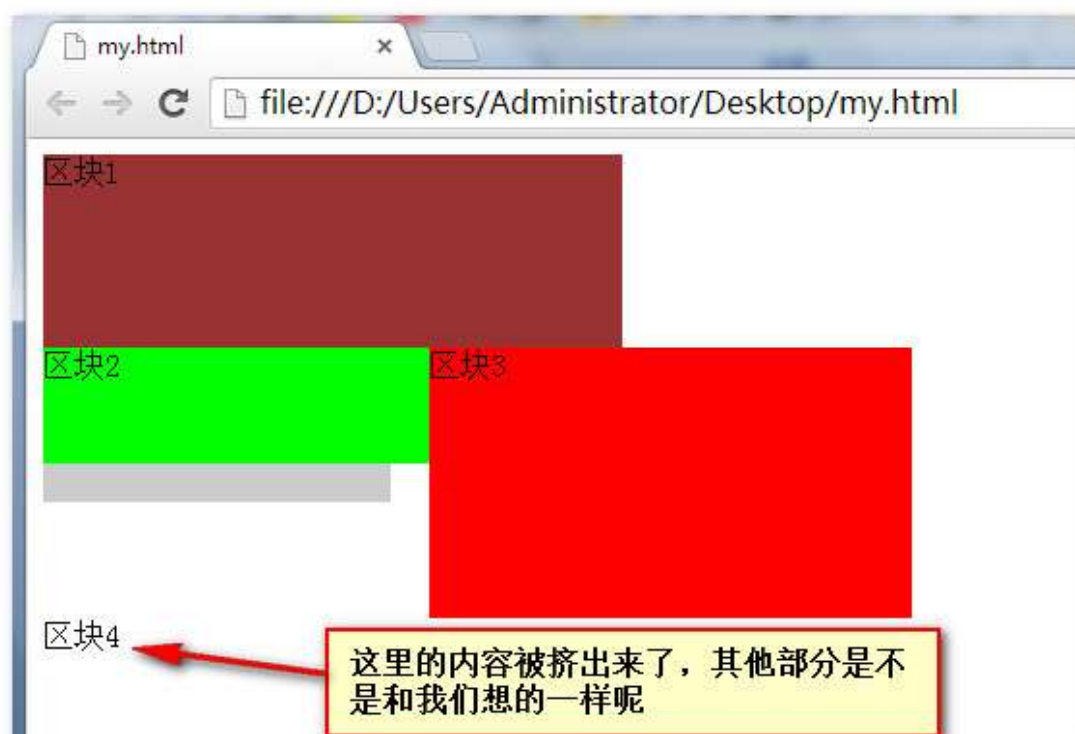
9.这里我们可以看到区块1、区块4依然是在标准流中的，而区块3被挤到了下一行，但是依然会选择右浮动。

10.上面我们使用的是右浮动，对于左浮动也很类似，这里我们让区块2和区块3左浮动吧：

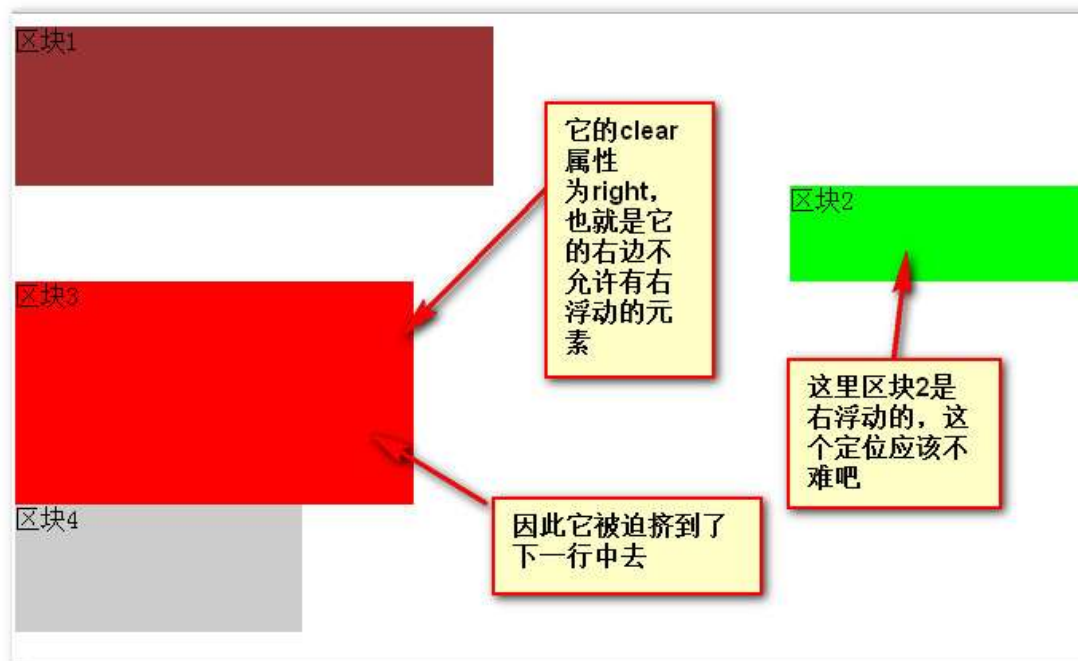
```
5 #xin2{
6     float:left;
7     background: #0F0;
8     height: 60px;width: 250px;
9 }
10 #xin3{
11     float: left;
12     background: #F00;
13     height: 140px;width: 250px;
```



11.此时它的效果如下：



12.对于多个元素的浮动，读者朋友们也可以自己联系下奥，我们相信只有动手才是最有效的学习方式。



4.那么上面的图片中已经解释的比较清楚了, 首先我们给区块2进行定位, 它的位置很好找, 然后区块3的clear属性为right, 也就是它的右边不允许有浮动的元素, 因此它的纵向位置被迫下移, 来满足“右边没有右浮动的元素”这一特性。

5.如果我们把区块3的clear属性设置为left呢? 也就是my.css中写成下面这个样子:

```

5  #xin2{
6      float:right;
7      background: #0F0;
8      height: 60px; width:
9  }
10 #xin3{
11     clear:left;
12     background: #F00;
13     height: 140px; width:
14 }

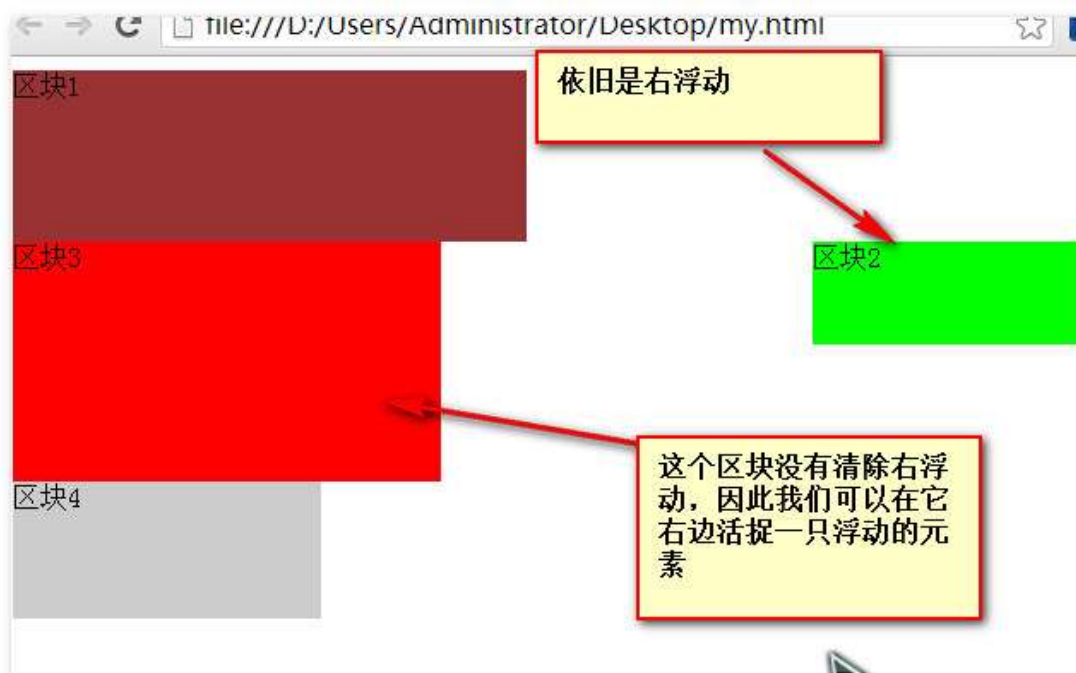
```

这一这里的#xin2还是右浮动

但是我们的#xin3清除的是左浮动

6.我们根据定义来看一下, 首先#xin2是右浮动的, 但是#xin3是清除了左浮动, 但是它没有清除右浮动, 因此, 在这里就相当于没有清除浮动。

7.因此它的效果图如下:



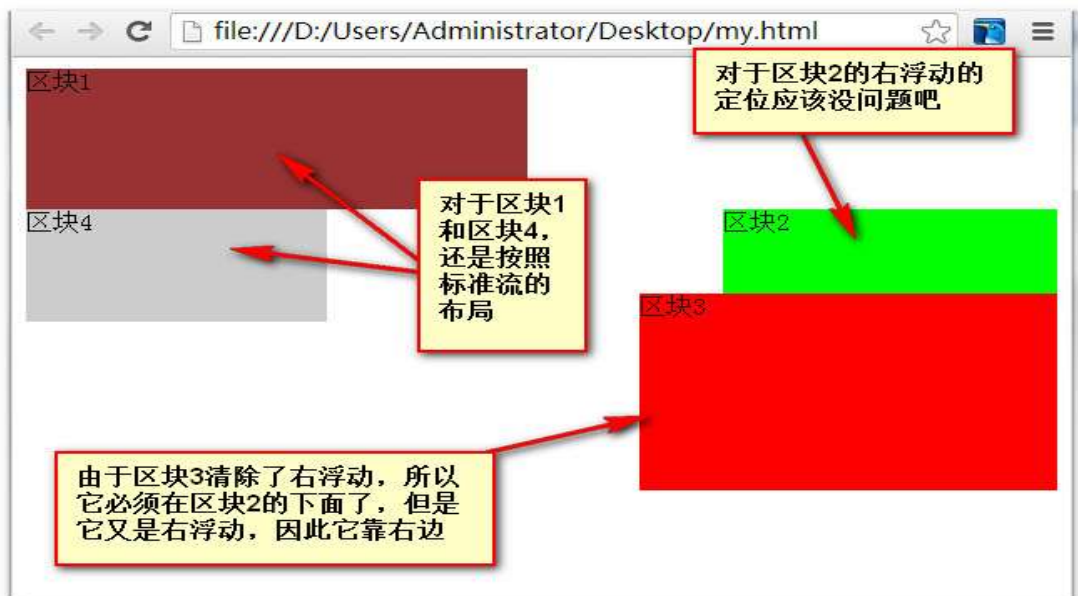
8.对于清除右浮动与左浮动的情形，大家只要牢牢把握定义 `clear:left`；是指“该元素的左边不能有左浮动的元素”即可，对于右浮动也同理。

>>>>>较为复杂的情况<<<<<<<<<<<<<<<<<<<

- 1.上面我们都是一个元素在浮动，另外一个元素会清除浮动。
- 2.下面我们来看一下稍微复杂的情况，也就是涉及到三个元素的情形，其实只要我们牢牢把握定义，一切都不难。
- 3.首先我们修改 my.css 文件如下：



4.然后我们看一下效果吧:



5.那么这个效果怎么出现的, 我们已经介绍的差不多了。

6.如果此时我们让区块4清除右浮动呢? 也就是我们让区块2右浮动, 让区块3清除右浮动并且右浮动, 让区块4清除右浮动。

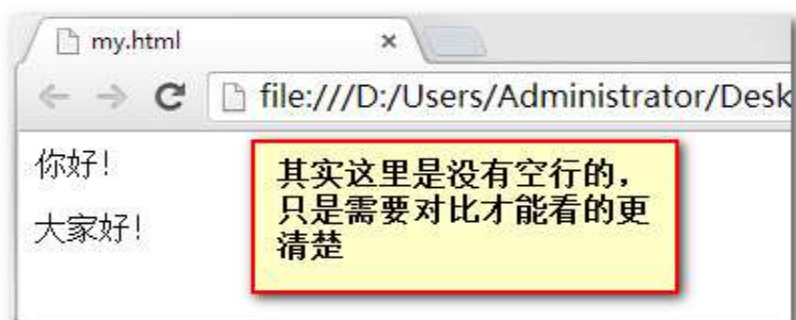
7.此时我们把 my.css 代码修改为如下内容:



4.然后我们写 my.html 内容如下:

```
<html>
<head>
  <link rel="stylesheet" type="text/css" href="my.css">
</head>
<body>
  <p>你好! </p>
  <p id="xin">我好! </p>
  <p>大家好! </p>
</body>
</html>
```

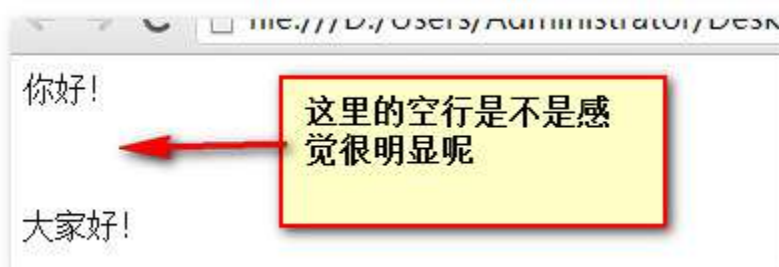
5.此时我们看一下效果吧:



6.然后我们把 my.css 的内容修改为下面的样式:

```
#xin{
  visibility: hidden;
}
```

7.然后我们看下效果吧:



3.说明一下，这里的 top 为 30px 即距离浏览器的顶部有 30 个像素，而 left 为 50px 则表示它距离浏览器的左边有 50 个像素。

4.那么此时的 my.html 中的内容为：

```
<html>
<head>
  <link rel="stylesheet" type="text/css" href="my.css">
</head>
<body>
  <p id="xin">我是辛星</p>
  <p>
    丘吉尔是英国的历史学家、政治家、画家、演说家、作家、
    记者。它的《不需要的战争》获得了1953年的诺贝尔文学奖
    ，并且著有《第二次世界大战回忆录》6卷，《英语民族史》
    4卷。曾经两度任英国首相，被认为是20世纪最重要的政治领
    袖之一，并且带领英国获得了第二次世界大战的胜利。并且
    在2002年的调查中被选为有史以来最伟大的英国人。
  </p>
</body>
</html>
```

5.然后我们看一下它的效果吧：



6.由于它有 900px 高，因此我们继续向下拉滚动条，看到效果如下：

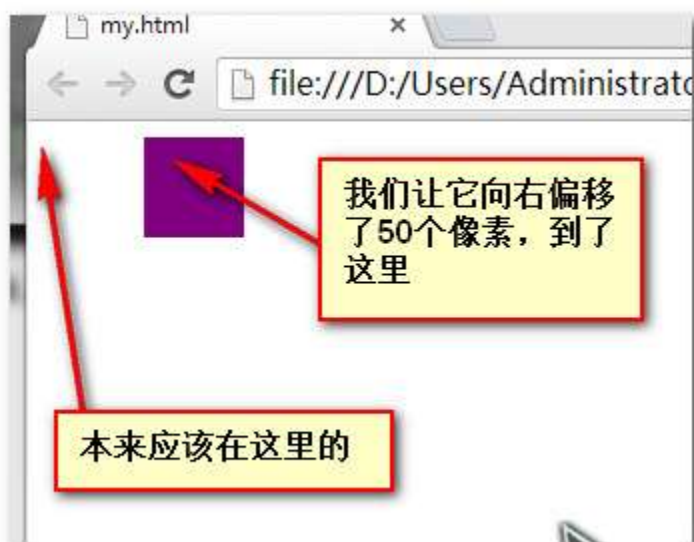
3.我们首先来写 my.html 内容如下：

```
<html>
<head>
  <link rel="stylesheet" type="text/css" href="my.css">
</head>
<body>
  <div id="xin"></div>
</body>
</html>
```

4.然后修改 my.css 内容如下：

```
#xin{
  background: purple;
  position: relative;
  width: 50px;
  height: 50px;
  left: 50px;
}
```

5.最后我们看一下效果吧：



5.然后我们看一下效果吧:

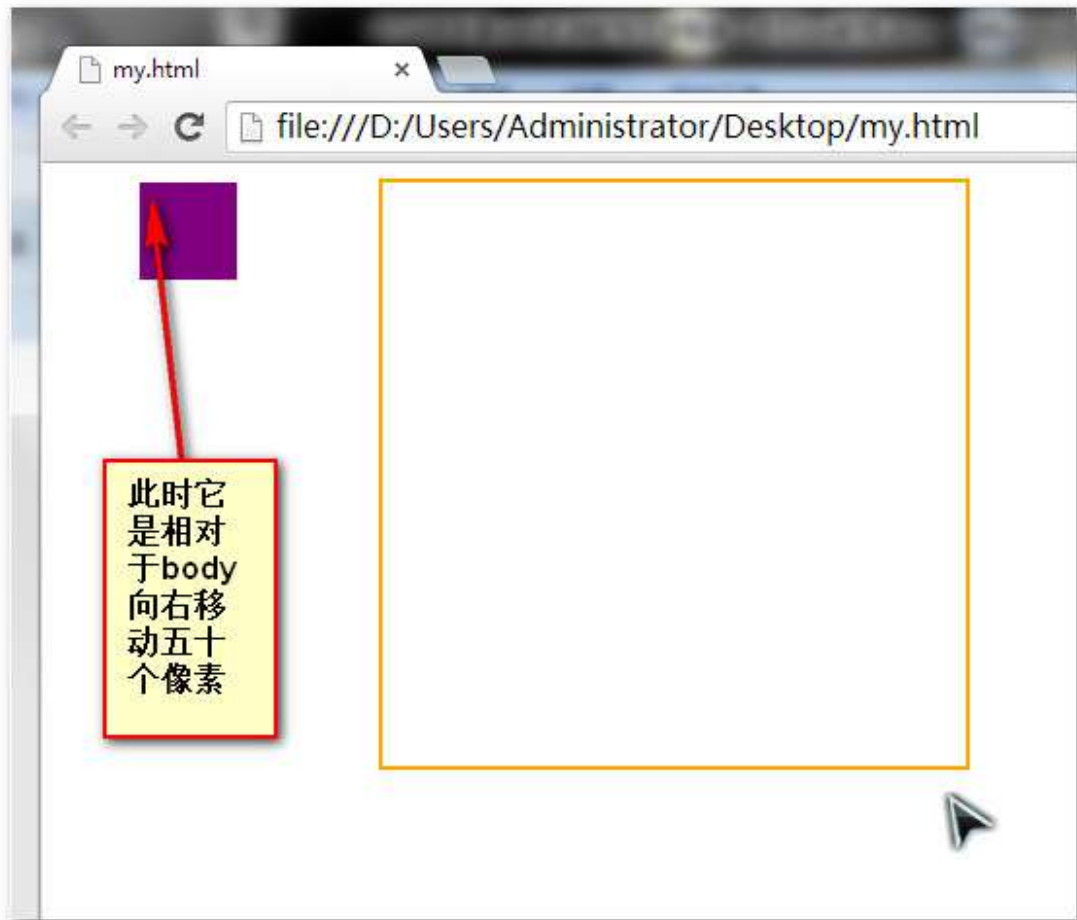


6.那么我们可以看到，它相对于父元素向右移动了五十个像素。

7.如果我们把 my.css 中的定位那行代码给注释掉，也就是变成下面这样:



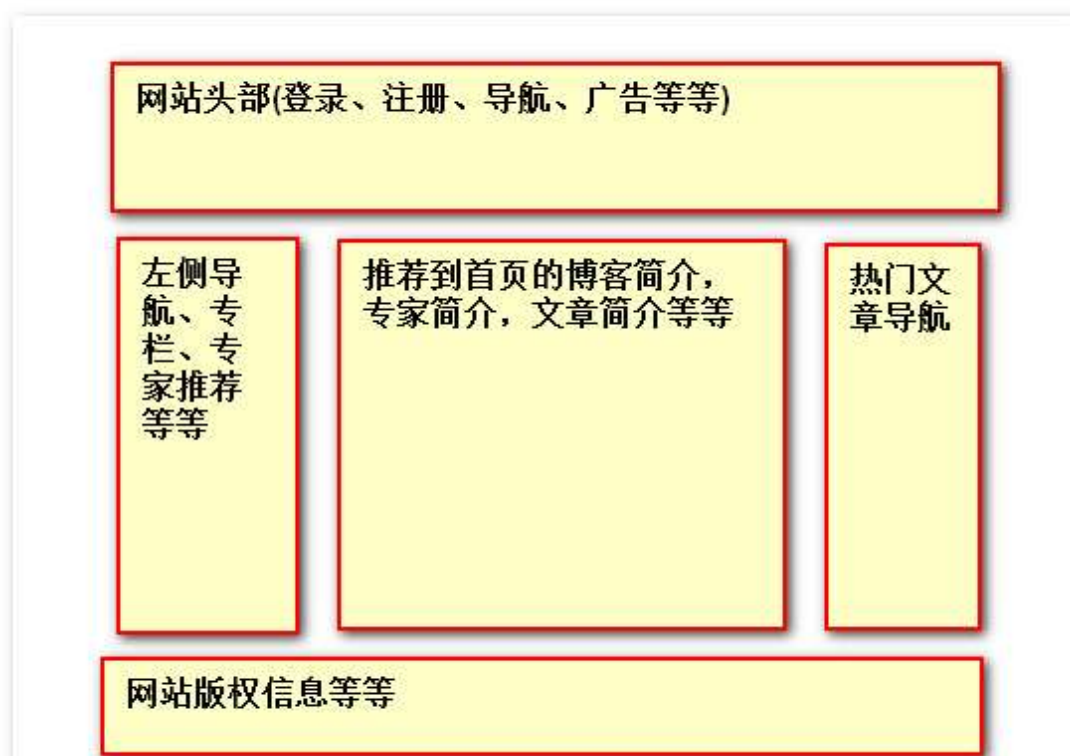
8.那么我们看一下效果吧:



2.然后对后半部分来个截图：



3.大家可以去这个网站亲自看一下，它的地址是：blog.csdn.net，那么我们可以得到这个网站的布局，它是一个典型的1-3-1模式，下面是这个布局的简图：



4.也就是说，虽然这个网站看起来貌似内容很多的样子，但是它的总体布局就可以看成这样的。

5.当然，每个小块内还可以再分，不过这里我就不分那么细了，我们到实战的时候再去讲解。

[illegible]

1.鉴于刚开始学布局，我们就不搞的那么高深了。

2.第一种布局方式就是单列布局，它就是那种直接下来的，现在很多论坛仍然使用这种布局。

3.比如说我们看一下 chinaunix 这个论坛的大致布局:

aunix.net			
 虚拟化与私有云 (0) 最后发表: 10-10 23:08 by donaldunix RSS订 主题: 帖数: 阅 10953 4294978345	分布式文件系统 (FastDFS) (0) 最后发表: 10-10 17:50 by haoyunkgq RSS订阅 主题: 1694 帖数: 7723	 数据安全 (0) 最后发表: 10-10 22:06 by shang2010 RSS订阅主题: 20022帖数: 100114	
 服务器及硬件技术 (0) Power服务器 最后发表: 10-10 17:38 by 445644178 RSS订阅主题: 12928 帖数: 63276	 存储备份 (0) Symantec专区 最后发表: 10-10 21:01 by lshlffy RSS订阅主题: 50516帖数: 198686	 网络技术 (0) 网络社区 最后发表: 10-10 23:57 by yunas RSS订阅主题: 49264帖数: 169496	
 行业应用和管理 (0) 最后发表: 10-09 09:28 by loving_you2000 RSS订阅 主题: 6087 帖数: 28037	 中间件技术 (0) 最后发表: 09-28 22:41 by cssnjjy RSS订阅 主题: 3106 帖数: 11304		
操作系统			
 Linux新手园地 (2) 最后发表: 10-11 00:30 by kle13 商城 RSS订 主题: 帖数: 阅 181801 4295289479	 Linux系统管理 (0) 最后发表: 10-10 22:58 by hanyi412 RSS订 主题: 帖数: 阅 121181 660676	 Windows系统 (0) 最后发表: 10-10 23:54 by yunas RSS订阅主题: 10150 帖数: 53826	本版新帖 -
 BSD (0) FreeBSD社区 最后发表: 10-10 19:07 by action08 RSS订 主题: 帖数: 阅 45805帖数: 289181	 AIX (0) Power中国用户组 最后发表: 10-09 17:33 by q1208c RSS订阅主题: 46781帖数: 236800	 AS400 (0) 最后发表: 10-08 17:13 by bendy88 RSS订阅主题: 14054 帖数: 94017	
 Solaris (0) Solaris文档中心 最后发表: 10-10 16:09 by mjlzg RSS订阅主题: 79846帖数: 449241	 HP-UX (0) HP文档中心 最后发表: 10-10 17:24 by bc3 RSS订阅 主题: 19785 帖数: 99096	 其他UNIX (0) SCO文档中心 最后发表: 10-03 07:23 by cgweb RSS订阅主题: 48899帖数: 247880	
 移动操作系统 (0) 最后发表: 10-10 17:28 by pacocai RSS订阅 主题: 8480 帖数: 14592	 Mac OS X (0) 最后发表: 10-09 19:24 by send_linux RSS订阅 主题: 2478 帖数: 14409		
程序设计			
 C/C++ (0) 最后发表: 10-10 23:24 by BetonArmEE	 Linux环境编程 (0) 最后发表: 10-10 16:52 by 剑魂策心	 内核源码 (0) 最后发表: 10-10 20:22 by ly365 RSS订阅主题: 24225帖数: 145499	本版新帖 -

4.这种布局的好处就是简单，简洁，清爽。

5.第二种就是典型的1-2-1和1-3-1这种布局方式，第一个1就是网站头部，比如登录、退出、网站logo等信息，后面的2或者3表示分为两列或者三列，每一列中可以列举出若干内容，最后一个1则是网站的尾部，一般就是写网站的版权信息之类的。

6.第三种布局也是基于第二种布局的，只是分的更细，比如我举个例子：1-((1-2)+1)-1 这个布局，那么它是什么意思呢？看下面简图：

2.然后我们看一下 my.css 中的代码:

```
body{
    background: silver;
    margin: 0px;
    padding: 0px;
    height: 500px;
}
#header{
    background: purple;
    height: 100px;
}
#con1{
    background: orange;
    height: 150px;
}
#con2{
    background: silver;
    height: 150px;
}
#footer{
    background: maroon;
    height: 100px;
}
```

3.这里的代码都比较容易理解吧,虽然有点长,但是真正的项目中的代码都是蛮长的,因此坚持住奥。

4.这里我们只是给出其布局,因此我们这里统一使用纯色背景来表示,而不向里面输入任何内容,等真正开发的时候,在第三本和第四本都会有介绍的,我们再用相应的内容替代。

5.废话少说,直接看下面的效果吧:


```
<html>
<head>
  <link rel="stylesheet" type="text/css" href="my.css">
</head>
<body>
  <div id="header">网站头</div>
  <div id="nav">左侧导航</div>
  <div id="con">中间内容</div>
  <div id="side">侧边栏</div>
  <div id="footer">网站尾</div>
</body>
</html>
```

6.然后看一下 my.css 的代码吧:

```
#header{
  background: purple;height: 100px;
}
#nav{
  float: left;background:blue;
  width: 20%;height: 400px;
}
#con{
  float: left;background: silver;
  width: 60%;height: 400px;
}
#side{
  float: right;background: orange;
  width: 20%;height: 400px;
}
#footer{
  clear: both;background: maroon;
  height: 100px;
}
```


第十节：应用图片

[illegible]

- 1.由于网速越来越快，而对于网页的美观性的要求也越来越高，因此，我们的网页中对图片的需求有增无减。
- 2.那么对于图片的边框、对齐方式、图文混排等等，这些知识的掌握也就非常重要。
- 3.在接下来的几个小段中，我们会依次介绍。

>>>>>>>>说明<<<<<<<<<<<<<<<<<<<

- 1.首先我们找一张图片，命名为 star.jpg，来个特写吧：



- 2.此人名叫黄美姬，韩国第一车模，关于 ta 的更多信息请自行百度。
- 3.我们接下来就以此图片为例，来开始我们的图片操作。

>>>>>>>>>>基本设置<<<<<<<<<<<<<<<<

1. 我们在引入图片之后，可以对其进行一些基本设置，比如设置 border 的一系列属性来决定其边框啦，比如设置 padding、margin 的值，比如设置 float 来修改其浮动等属性。当然我们还可以设置它的大小。
2. 也就是说我们的 html 文件只负责引入这个文件，而 css 文件负责它的具体样式。

3.我们看一下 my.html 中的内容吧：

```
<html>
<head>
  <link rel="stylesheet" type="text/css"
    href="my.css">
</head>
<body>
  
  <p>黄美姬，韩国当红模特，出生于1992年
    2月14日。她的特点就是丰满挺拔、面容俊
    俏，是一个天生的尤物。不论是在台球、
    足球，还是在赛车，都能看到她的火辣形
    象。</p>
</body>
</html>
```

4.然后我们看一下 my.css 中的内容吧：

```
img{
  border: 1px gray dashed;
  margin: 10px 10px 10px 0;
  padding: 5px;
}
p{
  text-indent: 2em;
  width: 200px;
}
```

5.那么它的效果是什么样的呢？让我们来看一下：

2.然后我们在上面的 my.css 中加这么一行：



3.然后我们看一下效果吧：



4.这里不知道读者是否注意到，它是默认铺满整个窗口的，也就是它会自动向水平和竖直两个方向平铺。

5.我们可以根据自己的需要来设置它的平铺方式，我们可以用 background-repeat 属性来控制，它有如下四个值可供选择：

取值	说明
repeat	按照水平和竖直两个方向平铺
no-repeat	不平铺，只显示一次
repeat-x	沿水平方向平铺
repeat-y	沿竖直方向平铺

6.然后比如说我们设置为水平方向平铺，也就是我们的 my.css 做如下改动：



```
body{
  background-image: url(space.jpg);
  background-repeat: repeat-x;
}
img{
  border:
  margin:
  padding: 5px;
}
p{
```

7.注意这里我们是在 body 里面设置的，也就是它是作为 body 的背景图片出现的。

8.读者朋友们可以尝试在不同的 html 的元素中来试一试，看看效果图。(不要问我为什么这么多废话，因为我在补空)

9.我们这里看一下效果吧：



10.读者朋友们是否会设置背景了呢?那么是不是感觉上面的设置有点麻烦呢? 没错，我们可以使用简化的方式，也就是通过background 属性，我们直接对它进行赋值就可以了。

11.那么究竟应该怎么赋值呢？看我操作：



十一节：杂项知识

>>>>>>何谓杂项知识<<<<<<<<<<<<<<<<<<

- 1.所谓**杂项知识**，就是感觉它归类到上面那一类都不合适。
- 2.但是不介绍这些知识，感觉又不合适，因此就索性放到这一节中。

[illegible]

- 1.说实话，在编程语言中注释非常重要，特别是涉及到算法的时候，没有注释很难看懂。
- 2.但是在 html 和 css 这种标记语言中，注释的作用就没有那么大了，不过，必要的注释还是得有的。
- 3.所谓注释，就是作者看得懂，读者看得懂，机器看不懂，在注释中的内容不会被执行，只是用来给人阅读。
- 4.在 css 中可以使用 C++ 风格的单行注释：

```
border: 1px gray dashed;  
margin: 10px 10px 10px 0;  
padding: 5px;  
}  
p{  
    //首行缩进2字符  
    text-indent: 2em;  
    width: 200px;  
    color: green;
```

5. 它也支持 C 语言风格的注释, 也就是如下形式:

/*

这中间是注释

✶/

- 6.也就是上面的风格以/*开头，以*/结尾，中间的都是注释：

这也是一个
注释

[illegible]

2.自己配色的格式，首先写一个#号，然后跟6个从0到f的16进制数，比如#23df98就是一个正确的颜色格式。

4. 由于 RGB 中本来是用两位表示的，如果相同，可以简化为一位，不过这里需要 RGB 都是使用一位才行，也就是 #336699 和 #369 表示是一样的。

[illegible]

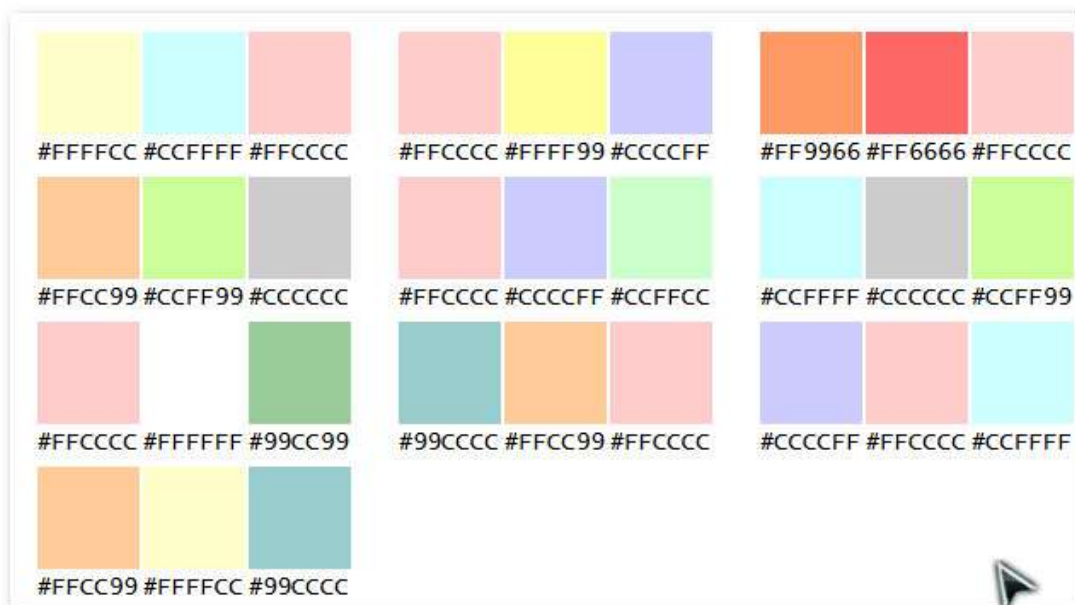
2. 对于一个前端开发人员来说，记住常见的色彩是很有必要的。

3. 可以从色相和印象的角度来分，我们单纯的从印象的角度去介绍。

4.一般来说数据越低，那么色泽就会越暗，直到都是0的时候表示黑色，数值越高，则色就越亮。

>>>>>>>>>>>>> 常见色泽分类 <<<<<<<<<<<<<<<<<<<<

1. **第一类：柔和型**，它的特点就是比较色泽大多都是暖色，而且偏淡，来看一下吧：



2. **第二类：冷淡型**，它的特点就是大多使用冷色，而且相对浓烈一些，来看看效果：



3. **第三类：高贵优雅型**，可能部分读者要笑了，其实我感觉还是蛮科学的，它的主要特点是偏向灰、紫、红三种颜色，那么我们看一下吧：

附录：意见反馈

>>>>>>>我们的愿望<<<<<<<<<<<<<<<<<<<

1. 因为我们致力于做出一部真正优秀的教程出来，我们相信肯定有太多的地方需要改进。
2. 我们希望尽可能早的解决这些问题，从而会让它的成长速度更快更给力。
3. 因此，我们必须广泛征求意见才行，因为我们相信一点：足够多的眼睛，就会找出足够的缺点，从而它更加优秀。

>>>>>>>期待您的建议<<<<<<<<<<<<<

- 1.我们相信：**您的建议是我们进步的最有力的保障。**
- 2.因此我们的每本书都会加这么一个附录，征求您的意见奥。
- 3.联系方式可以发邮件(推荐)给：**xinguimeng@163.com**，也可以加我的**QQ:1808347923**,然后给我交流也不错。
- 4.如果您想和其他可爱的朋友们一起交流的话，也可以考虑加我们的**QQ 群奥：392328871**，和我们一起讨论交流吧。

>>>>>>>>>>如何给出建议呢<<<<<<<<<<<<<<<<

- 1.可以从以下几个方面来给出建议:

- ①知识结构。
- ②讲解的方式
- ③文档的书写
- ④具体的知识点

- 2.当然您也可以给出让我们参考的东西,包括但不限于:

- ①网站
- ②其他教程
- ③自己期望看到的教程
- ④自己期望的学习模式

