

## 2014 年辛星 HTML、CSS 教程夏季版

\*\*\*\*\*说明\*\*\*\*\*

1. 如果您在一年后看到了该教程，那么可以考虑搜索它的更新的版本了，毕竟它的更新速度还是蛮快的。
2. 建议百度搜索“辛星 html”或者“辛星 css”，肯定不会让您失望的。
3. 如果您也经常用 CSDN 博客的话，可以留言给：  
<http://blog.csdn.net/xinguimeng>，有什么问题也可以发邮件给我：[xinguimeng@163.com](mailto:xinguimeng@163.com)。
4. 由于是第一版写 HTML、CSS，难免会有知识疏漏、没讲清楚、系统性不够的问题，期望您的参与，我们一起完善它，可以吗？

目标：更新更全更实用。

前进的道路，辛星陪伴您。

只要星哥在，编程充满爱。

可能当前版本您并不满意，  
下个版本绝对超乎您想象。

目录：

1 / 120

前言.....	4
第一部分：HTML 快速掌握	
第一节：HTML 快速入门.....	5
第二节：head 标签和文本标签.....	8
第三节：图片.....	11
第四节：超链接.....	16
第五节：表单.....	21
第六节：列表.....	25
第七节：区块.....	30
第八节：小结.....	32
第二部分：CSS 快速掌握	
第一节：CSS 简介.....	35
第二节：CSS 语法快速入门.....	38
第三节：CSS 中的颜色与文字.....	45
第四节：超链接与列表以及简单导航栏.....	49
第五节：盒模型.....	52
第六节：定位.....	56
第七节：浮动.....	65
第八节：图片、背景和布局.....	77

## 第三部分：CSS 经典应用

第一节：导航条.....	84
第二节：气泡.....	91
第三节：面包屑导航.....	100
第四节：新闻列表.....	113
第五节：结语.....	120

## 前言

\*\*\*\*\*我需不需要学习它\*\*\*\*\*

1. 我认识的很多开发 web 的朋友，他们或许有人专注于后台程序的编写，有人专注于前台的制作。
2. 写后台的程序员大都很怕编写前台界面，认为写网页是一个很累的活儿，感觉又没有技术含量。
3. 于是，很多人会说：HTML、CSS 是前台的事，我不需要去学，我是做后台的。
4. 其实，我的看法是：由于它的学习这么简单，那我们为什么不跟着星哥快速的掌握它呢？

\*\*\*\*\*我怎么学好它\*\*\*\*\*

1. 老实说，HTML 和 CSS 绝对不是靠“学”出来的，它是靠练出来的。
2. 它和算法不一样，它更像绘画和射击，理论就那么点知识，学习完了理论就是无尽的实战和自己的创意。
3. 如果说写后台是一门“技术活”，那么写前台绝对是一门“艺术活”，我相信你有艺术细胞的。
4. 其实最好是从心里喜欢它，就一定能做好它，就想有些人喜欢绘画一样，做出一个漂亮的前台界面，和绘画出一幅画一样有成就感，不是吗？

\*\*\*\*\*推荐\*\*\*\*\*

1. 其实我最推荐的就是博客园了，这里是一个高手云集的地方：<http://www.cnblogs.com/>，我也从中受益匪浅。
2. 另外，号称新手乐园的 CSDN 也不错，至少我在这里还是感觉很有收获的：<http://blog.csdn.net/xinguimeng>。

## 第一部分：HTML 快速掌握

### 第一节：HTML 快速入门

首先推荐一个网站，就是 <http://www.w3cschool.cc>，这是一个公开的教学网站，但是它有一个问题，那就是虽然很全面，但它不是一个入门网站，而是一个手册性质的网站，也就是适合我们去随时查询，但是不适合我们随时随地的阅读和学习。本教程则是一个适合入门的教程，如果是新手，可以学习下，如果是高手，这种 html 的知识就不用看了。

#### \*\*\*\*\*快速入门\*\*\*\*\*

1. html 是 HyperText Markup Language 的缩写，即超文本标记语言，所谓的超文本，也就是可以包含了图片、超链接等的文本，而不仅仅是只能显示文字，它是一个标记语言，不是一个编程语言，因此大家学习起来会非常的快速。
2. 我们可以新建一个 xinxing.html，在编辑器中写入如下代码：



```
<html>
  <head>
    <title>辛星 CSS，期待您的关注</title>
  </head>
  <body>
    <p>辛星 HTML，期待您的关注</p>
  </body>
</html>
```

然后保存，然后用浏览器打开，会发现浏览器的界面上显示了几个文字：辛星 HTML，期待您的关注。下面我们来讲解一下这段代码。

3. 我运行一下这个页面把，它的界面是这样的：



#### \*\*\*\*\*标签\*\*\*\*\*

1. 这里大家会发现我们有一个<html>和一个</html>标记，它们两个组成了一个标签对，每一个都被称之为一个标签。当然下面的 head 和 body 也是标签对，而且开头都用尖括号包裹起来，并且结尾的地方加一个斜线。

2. 其实 HTML 既然被称为一个标记语言，它本身也是由若干的标签组成的，我们的最初始的<html>说明这是一个 html 文档的开始，而</html>则说明停止解析该文档，也就是 html 标签对中间的部分都是 html 的内容。

3. 当然，我们还有 head 标签对和 body 标签对，它们的功能是一样的，这里说一下 head 标签，它包含了所有的头部标签元素，在 head 元素中我们可以添加一些设置，我们到后面再细说。

4. 下面的 title 标签说明的是在浏览器的工具栏的标题的内容，它还会协助搜索引擎来搜索页面，而且当我们把该网页添加到收藏夹的时候，它还会显示在收藏夹中的标题。

5. 上面的 body 标签则是浏览器的显示内容，我们通常也是更多的把内容写到这里面来，它的内容也就愈加广泛。

6. body 标签的 p 标签则是指明了一个段落，它里面的文本信息就是该段落所拥有的内容，当然我们这里只是显示了一句话，没有什么特别的意义。

\*\*\*\*\*备注\*\*\*\*\*

1. 并不是所有的标签都是由开始标签和结束标签组成的，比如用于换行的标签<br />, 其实我们要换行，只需要写一个<br>就可以了，但是出于标准，我们在 br 后面加一个空格和一个斜线。

2. 标签还可以有属性，这些我们后面再说，我的原则就是：用到的时候再说。

\*\*\*\*\*小结\*\*\*\*\*

1. 本小节我们主要认识了一下 HTML 的基础知识，然后演示了一下 HTML 文件的写法。

2. 虽然 HTML5 的脚步越来越快了，但是本教程并不是为 HTML5 准备的，而是为那些学习 PHP、JavaEE、Python 的程序员快速入门前台，不再为前台编写界面而一筹莫展而写的一本教程。

## 第二节：head 标签和文本标签

上面一节中我们介绍了 HTML 文件的书写和几个标签，接下来我们来认识几个其他的标签，这里我们主要介绍一下 head 标签和文本标签。

### \*\*\*\*\*head 标签\*\*\*\*\*

1. 所谓 head，就是头的意思，它可以包含若干子标签，它的内容是不会再浏览器的文本区域显示的，但是它的作用非常广泛，下面我们重点介绍一下它的子标签。

2. head 标签可以包含 title 标签，上面我们已经看到了，而 title 标签就是在浏览器的标题栏显示的内容，这点我们上节说了。

3. link 标签可以用于导入外部的资源文件，通常用于导入 css 文件和 js 文件，对于 css 和 js，我们后面会介绍的，这里先给一个导入 css 的例子：

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

4. 它还可以通过使用 style 标签来内置 css，不过我本人不是很推荐这么做，这会使得文件过于臃肿。

5. meta 标签，它是指的元数据，所谓元数据，就是可以用于描述网页的一些信息，比如指定编码方式、作者等等，下面这个例子就指定网页的编码格式使用 gbk：

```
<head>
    <meta http-equiv="Content-Type"
content="text/html"; charset="gbk">
</head>
```

比如下面我们定义了一个关键字，方便搜索引擎来抓取我们的页面：



```
<head>
  <meta name="keywords" content="辛星">
</head>
```

6. script 元素，它用于加载 Javascript 脚本，我们后面再说。

\*\*\*\*\*body 标签\*\*\*\*\*

1. 我们上面讲解了 head 标签，那么我们接下来讲解 body 标签，body 标签中的内容是浏览器的主要显示部分。

2. 除了我们上面讲解的几个标签，下面的标签都是 body 的子标签，就不一一介绍了。

\*\*\*\*\*标题\*\*\*\*\*

1. 我们用<h1></h1>标签中的内容来表示最大的标题，用<h2></h2>标签对来表示小一号的标题，这个数字是可以继续加下去的，一直加到<h6></h6>，它表示最小字号的标题。

2. 标会使得文字变粗而且也会变大，但是不要单纯的让文字变粗变大而使用标题。

3. 标题标签出现在 body 标签里面，因为它是可以直接显示在浏览器中的。

\*\*\*\*\*段落\*\*\*\*\*

1. 我们前面也说过，我们用 p 标签表示段落。

2. 我们可以用<strong></strong>标签对或者<b></b>标签对来让文本加粗，用<em></em>标签对或者<i></i>标签对来让文本倾斜。

3. 此外我们还可以用 sub 标签来定义下标字，用 sup 标签来定义上标字，用 ins 标签来定义插入字，用 del 标签来定义删除字。

4. 下面是一个代码示例：

```
<body>  
    以前价格：<del>400 元</del>现在价格：  
    <strong>200 元</strong>  
</body>
```

5. 它的效果如下：



6. 此外，我们可以用<br />来表示一个换行，当然老版的书籍上也会用<br>，我更推荐前者。

7. 对于这种一个标签就可以表达清楚，而不是需要一个标签对的，我们通常建议在第一个标签的后面加一个空格和一个斜线。

\*\*\*\*\*小结\*\*\*\*\*

1. 本节我们认识了两个比较重要的标签，一个是 head，另一个是文本标记的。

2. 例子比较简单，希望读者也可以跟着做一下。

### 第三节：图片

我们前面也涉及了 HTML 中的一些东西，接下来我们要涉及到图像了，如果没有图像，即使文字的样式再多，再复杂，终归还是单调的，我们就需要用图片来给我们的网页增加更多的表现形式。

#### \*\*\*\*\*图片\*\*\*\*\*

1. 我们用<img>标签来给我们的 HTML 文件导入一个图片，它是一个空标签，即它只包含属性，而且它没有相应的闭合标签，即它不用成对出现。
2. 我们很多时候还是需要图片去作为装饰甚至是主要内容来呈现的，而后台技术里面对图片的常见操作就是加水印，它是维护版权的一种比较有力的方式，当然，我会出一期 PHP 的视频，里面就有操作图片的，有兴趣的读者可以去看一下。
3. 它的 src 属性应用于指定源属性地址，它可以是绝对路径，也可以是相对路径，还可以是一个网址，即一个 URL，比如我们要引入的图片是百度的一张图片，那么可以写如下代码：

```

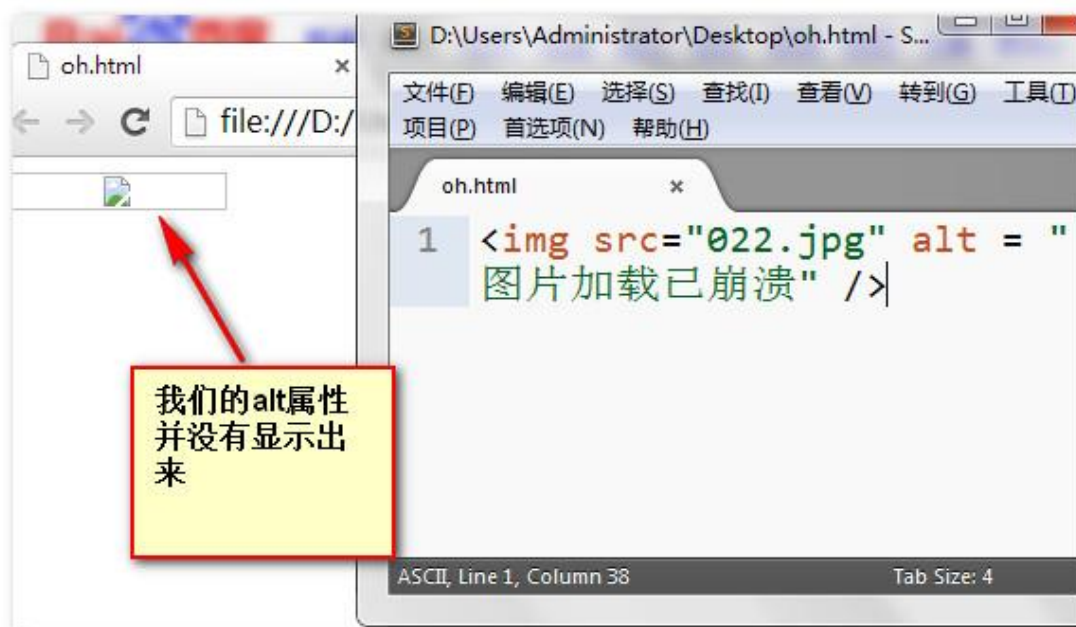
```

4. 当然上述代码有点长，主要是百度的地址比较长，既然地址都发出来了，于是干脆把运行效果一起粘贴出来把：

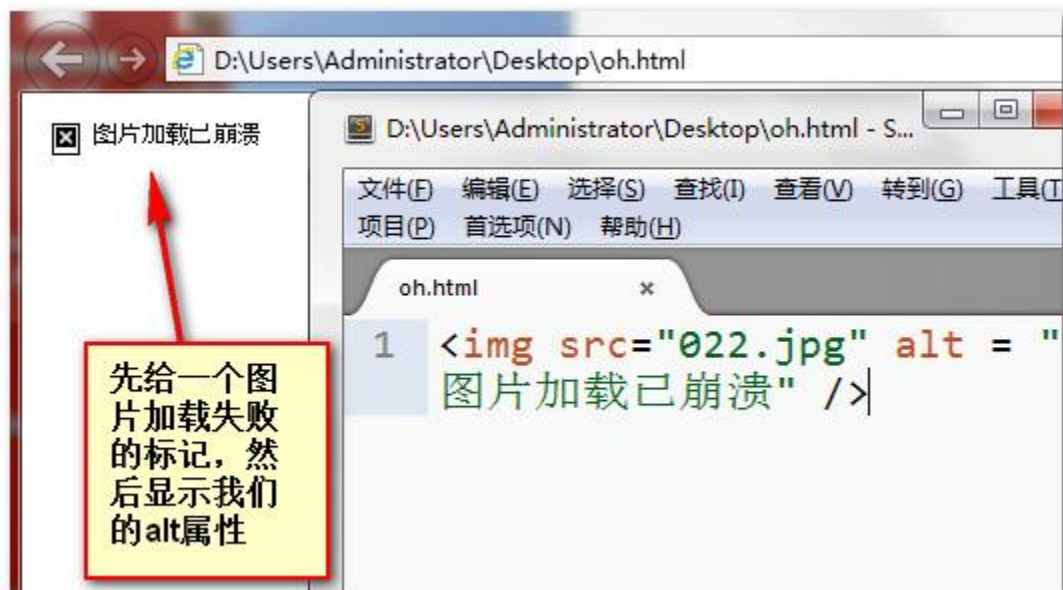


5. `img` 标签还可以包括 `alt` 属性，它是当图片无法载入的时候用来显示的。

6. 这个时候浏览器的区别就看出来了，各大浏览器的支持是不一样的，首先是全球用户量已经第一的谷歌浏览器：



7. 还是上面的代码，用微软老大哥的 IE 浏览器浏览效果如下；



8. img 标签还有一个 width 属性用于设置显示图像的宽度，用 height 属性来设置显示图片的高度。

9. 如果一个标签有多个属性，那么需要用空格进行分开，并且属性的值需要用双引号引起来，以明确表示它的属性。

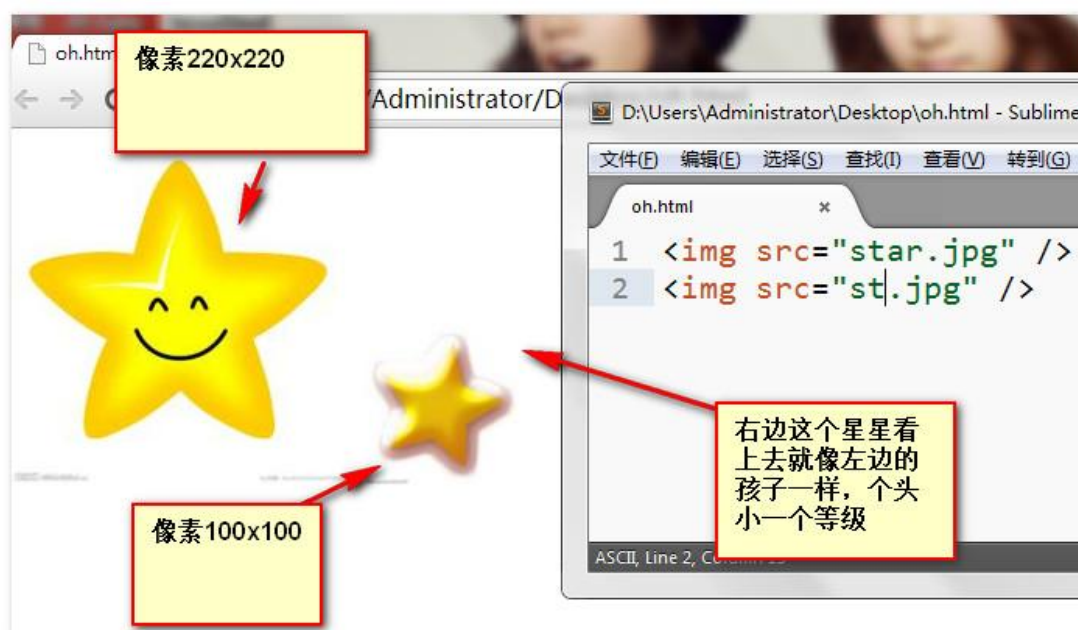
10. 示例代码：

```

```

11. 设置图片高度和宽度是非常重要的，它可以让我们的图片排列更加整齐，比如我们不指定大小的时候如下：





12. 但是我们使用下面这种方式是不是就更整齐一些了呢？

13. 我们对来源大小不一样的图片可以通过人为的改变它的显示大小：



\*\*\*\*\*路径问题\*\*\*\*\*

1. 会想到我初学 HTML 的一些经历，在这里提醒大家一下关于路径的问题吧，img 标签的 src 属性可以用绝对路径，也可以用相对路径。

2. 比如我们编写的 html 和图片在同一目录下，且图片叫 x.gif，那么就可以写代码为：<img src = "x.gif" />

\*\*\*\*\*小结\*\*\*\*\*

1. 本节课我们介绍了 HTML 中关于图片的一些知识，希望大家可以理解。
2. 但是真正的项目中，添加合适的图片却往往是一件比较费时费力的事情，我们往往要进行各种各样的切图，这些后面再说。
3. 其实 HTML 本身非常简单，现在的 CSS 技术则是越来越复杂了，因此学好 CSS 和 Javascript 显得越来越关键。

\*\*\*\*\*另外\*\*\*\*\*

1. 由于每一节都是一页的开头，这里还有一些空隙，我废话几句把，那就是，图片的使用并不难，但是图片的布局，还有图片的切图等部分，还是一个比较累的活的。
2. 另外附上一张小星星图片（作为我的吉祥物把）：



Tan001.com

## 第四节：超链接

接下来我们继续学习 HTML 中的标签，希望大家能够再接再厉，同时辛星也会支持大家，我们一起努力，一起加油。我们本小节来认识另外几个标签。

\*\*\*\*\*空格和换行\*\*\*\*\*

1. 我们在 HTML 文件中的那么多的空格和换行在浏览器上显示的时候和我们在文本编辑器中看到的并不一样，我们可能 HTML 代码中有大量的空格和换行，但是输出到浏览器上的时候却好像没有空隙。
2. 浏览器会把我们多余一个的空格都解析为一个空格，如果我们想要多加几个空格的话，需要使用  它表示字符实体，每当我们加上一个  它就会多加一个空格，值得注意的是，这里的  总共是六个字符，两个符号和四个字母，缺一不可。
3. 至于换行，我们前面也看到了，我们通常用<br />表示。
4. 比如：



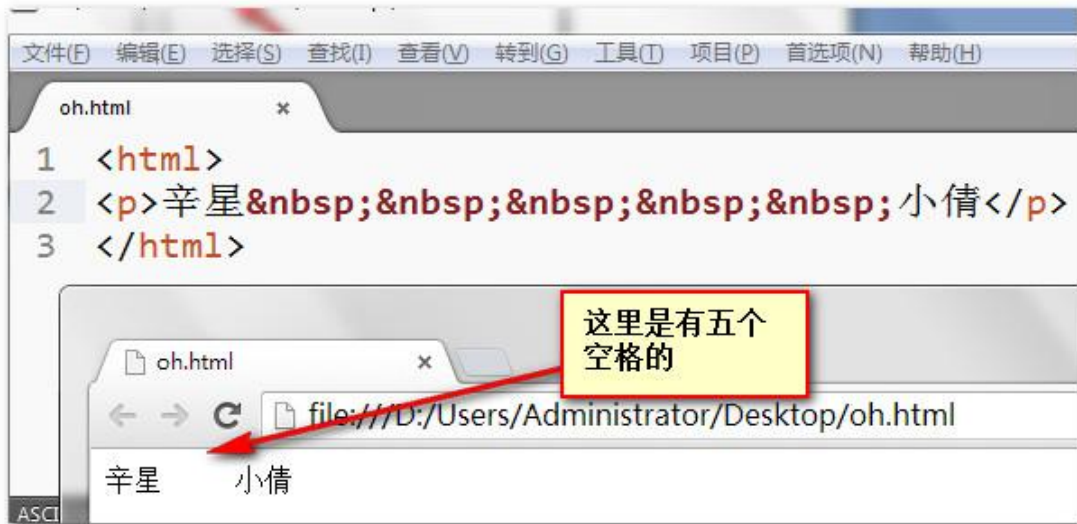
\*\*\*\*\*实例演示\*\*\*\*\*

1. 比如我们在文本编辑器中输入“辛                      星”，虽然我们输入了很多空格，但是只是显示一个空格，截图：





2. 因此我们使用 的话，效果如下：

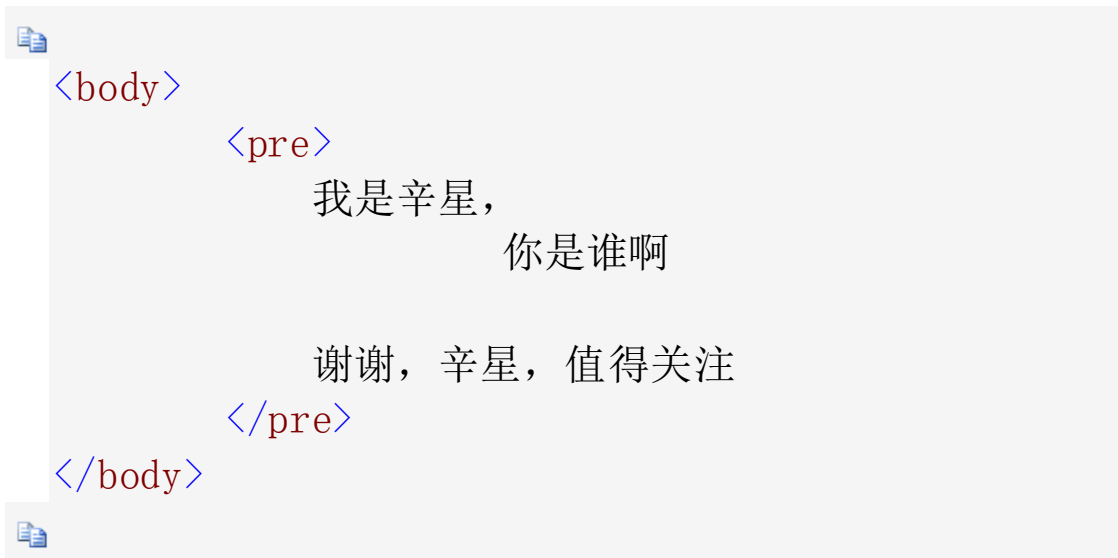


3. 就像 这种在 HTML 中叫做“字符实体”，感兴趣的童鞋可以去百度下字符实体。

\*\*\*\*\*pre\*\*\*\*\*

1. 所谓的 pre，就是定义预格式文本，在该标签中间的部分会原封不动的输出到浏览器上，它的多于一个的空格不会被当做一个空格，而且换行也会正常显示。

2. 示例代码：



3. 我们看一下它的执行效果：



\*\*\*\*\*超链接\*\*\*\*\*

1. 所谓超链接，就是当我们点击某个东西的时候，它会自动跳转到别的网址，这个“东西”可以是文字，也可以是

图片，而且当我们把鼠标指针移动到网页的某个超链接上的时候，鼠标通常会变成一只小手的样子。

2. 我们用 a 标签来表示超链接，而且它有一个 href 属性，这个 href 属性中我们填写连接到的地址。

3. 我们光用 a 标签定义一个超链接往往还不够，我们还需要定义一个载体，我们点击这个载体，跳转到连接的地址，如果我们用文字作为载体，则在 a 标签中书写相应的文字即可。

4. 下面我们用文字作为载体，来演示一下超链接的使用：

```
<body>
    <a href="http://www.baidu.com">百度</a>
    <a href="http://www.sina.com">新浪</a>
</body>
```

需要说明的是，我们超链接的 href 属性中的 http:// 不能省略，否则会导致链接错误。

5. 上面的截图：

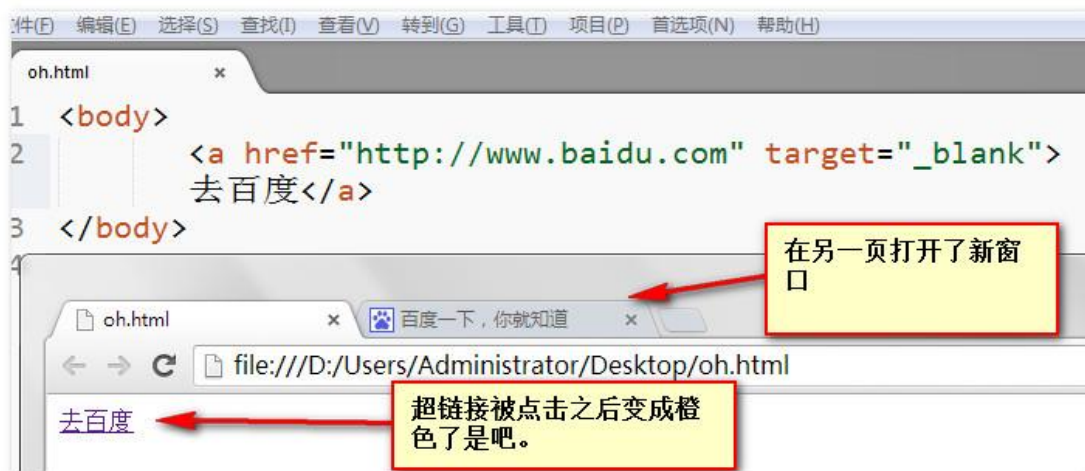


6. 我们也通常指定 a 标签的 href 属性为 # 号表示一个空链接。

7. a 标签的另一个常用属性就是 target 属性了，如果没有该属性，我们点击它，默认是在本网页加载该连接到的网页，如果我们使用了它，那么可以在新的标签页中打开，示例代码：

```
<a href="http://www.baidu.com" target="_blank">去百度</a>
```

8. 比如我们点击该超链接，并且效果如下：



\*\*\*\*\*小结\*\*\*\*\*

1. 本节介绍的内容首先介绍了 HTML 网页中的回车和换行的问题，然后又介绍了一下 pre 标签。
2. 然后就是介绍了超链接，超链接在 HTML 应用中异常的广泛，因为 HTML 网站要进行各种各样的页面跳转，就必然要涉及到 url，而使用 url 很方便的一种方式就是使用超链接，因此，我们以后学 CSS 的时候，对应于超链接的美化是很大的一块内容。

## 第五节：表单

如果读者是一位后台开发者，那么肯定会知道什么叫表单，这里我们就介绍一下前台如何使用表单，表单的使用也是我们编写网页的必须经历的一关，而且，表单也往往是我们网站的漏洞和弱点出现的地方。

### \*\*\*\*\*表单\*\*\*\*\*

1. 所谓表单，就是一个包含表单元素的区域，所谓表单元素，可以是输入文本的文本框，也可以是用于输入密码的密码框，还可以是各种各样的下拉框，还有可能是单选框和复选框。
2. 表单通常需要定义一个方法，用属性 `method` 表示，它表示我们把数据提交给服务器后台所使用的方法，通常来说，`method` 有 `get` 和 `post` 两种方式，如果是 `get` 方式的提交，那么表单的数据可以在 `url` 中出现，而 `post` 方式就安全的多了，不会再 `url` 中出现。
3. 表单通常还需要定义一个 `action` 属性，它的属性值是服务器后台的处理脚本，服务器后台可以用 `php` 写，可以用 `java`，还可以用 `python` 等等，这一点确实不好说了。

### \*\*\*\*\*表单元素\*\*\*\*\*

1. 所谓表单元素，通常我们使用输入标签来表示，即 `input` 标签，它也是一个空标签，它没有与之对应的结束标签，我们通过指定它的 `type` 属性来让该表单元素具有不同的功能。
2. 比如我们可以指定 `type` 的属性为 `text`，那么此时它表示一个文本输入框。
3. 比如我们可以指定 `type` 的属性为 `password`，那么此时它表示一个密码输入框，其实密码输入框和文本输入框很像，

如果非要说区别，那就是它的显示特别单一，我们无法从它的显示信息来获取密码的内容。

4. 我们可以通过指定 `type` 属性为 `radio` 来设置为一个单选框，它需要传递一个 `value` 属性，表示当我们选中该单选框的时候向后台传递的值是多少。

5. 我们还可以通过指定 `type` 属性为 `checkbox` 来设置为一个复选框，它可以同时选择。

6. 我们的表单是需要提交给后台处理程序的，此时我们定义 `type` 为 `submit`，它还可以接受一个 `value` 属性来指定该按钮的显示文本信息。

#### \*\*\*\*\*name 属性\*\*\*\*\*

1. 我们上面提到表单中可以有 `input` 标签，而 `input` 表单都可以拥有一个 `name` 属性，它表示当我们把数据提交给后台的时候，用于区分这些内容，比如我们把一个表单的数据提交给后台的 PHP 去处理的时候，`name` 的值就是 `$_POST` 数组的下标。

2. 当我们点击 `type` 为 `submit` 的时候，它会提交表单，并且同时把数据发送过去。

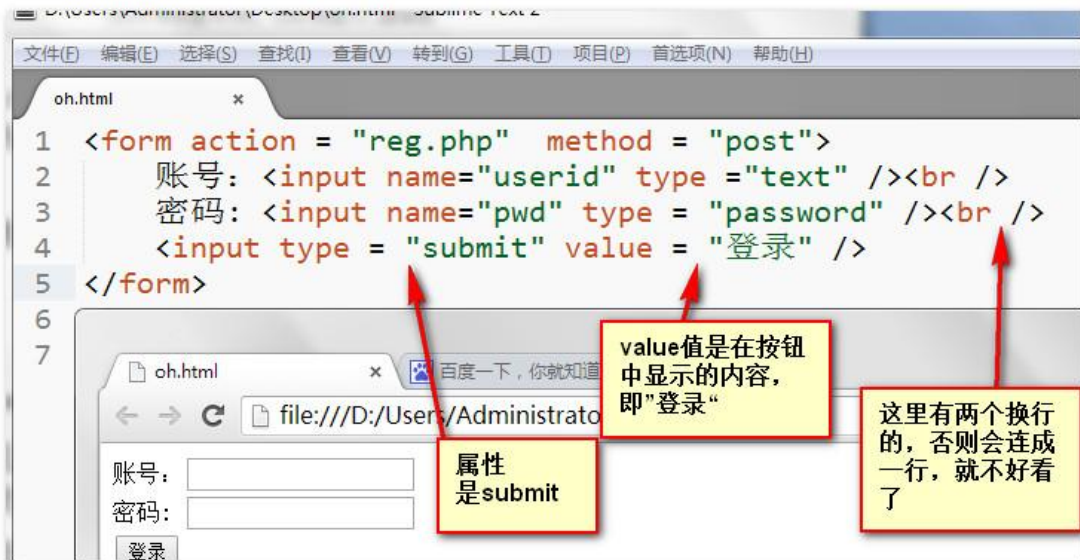
3. 下面是定义了一个表单的代码：

```
<form action = "reg.php" method = "post">
    账号: <input name="userid" type = "text" /><br
/>
    密码: <input name="pwd" type = "password"
/><br />
    <input type = "submit" value = "登录" />
</form>
```



4. 至于 post 和 get 是什么意思，这个对于学习后台的童鞋是一定要熟练掌握的，对于前台来说，一般用 post 就可以了。

5. 它的效果图：

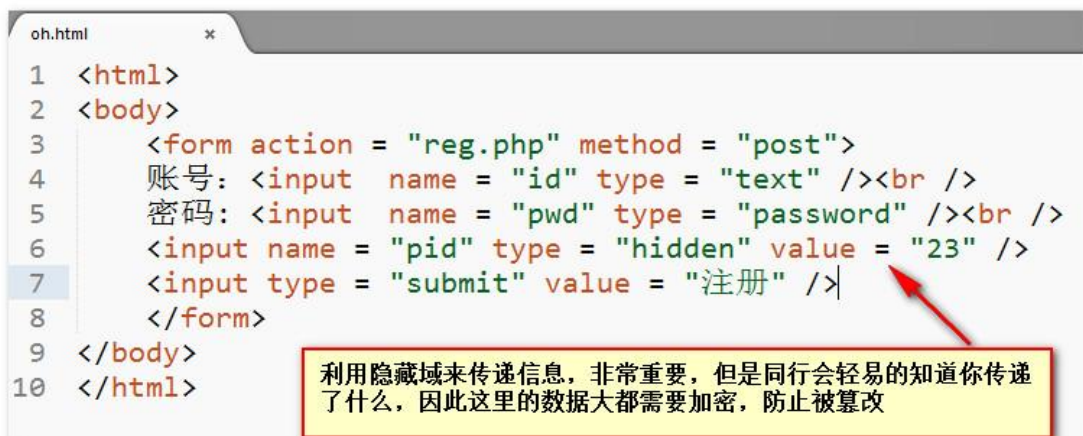


### \*\*\*\*\*常见的表单元素\*\*\*\*\*

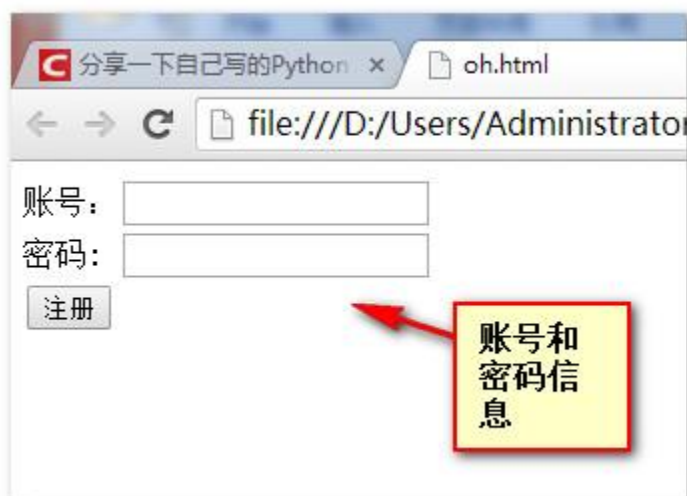
1. 比较常见的就是输入框、密码框和隐藏域了，所谓隐藏域，就是我们需要发送的额外信息，但是不让用户看到。

2. 比如下面表单中我们就使用了隐藏域发送了一条额外的信息，但是用户是看不到的。

3. 我们的 HTML 代码如下：



4. 当然小白级别的用户和老太太是不知道隐藏域是神马的，因为我们的界面只有两个框框可以输入信息：



5. 那么怎么加密呢？这就需要后台程序猿的配合了，到底采用什么加密方式不是前台一个人的事。

#### \*\*\*\*\*小结\*\*\*\*\*

1. 这里我们定义了表单，如果读者会诸如 java 或者 php 等服务器端的语言，就很容易理解了。
2. 如果读者没有后台方面的基础，可能该小节有点吃力，也建议学习前台的朋友们研究一下后台的知识，毕竟这两者的结合态紧密了。



## 第六节：列表

不知道这样大家会不会感觉枯燥，总之不管大家怎么样，辛星始终会陪伴大家的，期待您的关注和支持，也衷心的希望我的博客能让您少走弯路，获得更系统的知识。

\*\*\*\*\*表格\*\*\*\*\*

1. 在早些时候，那时候大家为了布局上的美观，会经常使用表格来布局，因为表格的布局比较整齐，我们只需要把表格的边框设置为 0 就可以让内容显示的很有条理了。
2. 随着 div+css 的流行，表格布局越来越不被人所喜欢，甚至有人特别排斥表格，本来该用表格的地方也不用表格了，其实大可不必那么偏激，它也有它有用的一面，当然也有它不可取的一面。
3. 我们 table 标签来实现一个表格，然后我们可以指定它的 border 属性，默认是无边框的，即默认情况下表格是用于布局的。
4. 我们用 tr 来表示表格的每一行，即 tr 标签是嵌套在他 table 标签里面的。
5. 而我们用 td 标签来表示一个单元格，即 td 标签是嵌套在 tr 标签里面的，我们通常把内容写到我们的 td 标签中，而 tr 标签仅仅相当于起到一个换行符的作用。

\*\*\*\*\*示例\*\*\*\*\*

1. 比如下面我们定义一个三行两列的表格：

```
<table border = "1">
  <tr><td>姓名</td><td>性别</td></tr>
  <tr><td>辛星</td><td>男</td></tr>
  <tr><td>小倩</td><td>女</td></tr>
</table>
```

2. 不知道读者是否感觉有点别扭，因为上述表格的”姓名“和”辛星“、”小倩“的作用是不同的，它有点类似于标题的感觉。

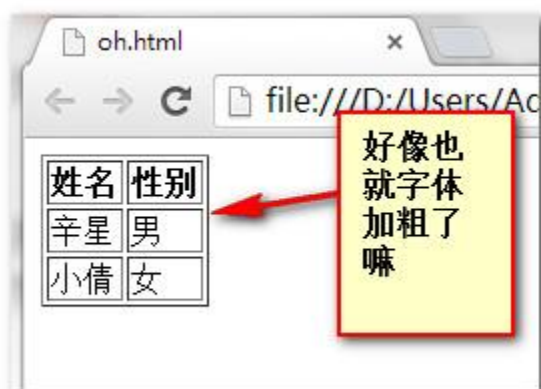
3. 看一下上面的运行效果：



4. 对于上述表格中的”姓名“和”性别“这种单元格，我们可以用 th 标签来代替 td 标签，即我们把代码写成如下样式：

```
<table border = "1">
  <tr><th>姓名</th><th>性别</th></tr>
  <tr><td>辛星</td><td>男</td></tr>
  <tr><td>小倩</td><td>女</td></tr>
</table>
```

5. 我们来看一下上述代码的运行效果把：



6. 其实表格还有诸如表格的标题、页眉、页脚等属性，个人感觉不是很常用，就不讲解了。

#### \*\*\*\*\*列表\*\*\*\*\*

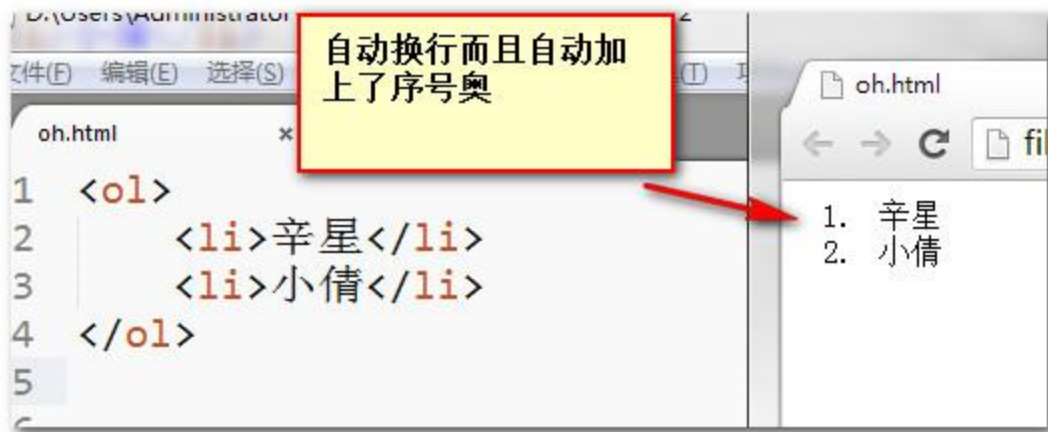
1. HTML 中我们支持有序列表、无序列表和定义列表。
2. 所谓有序列表即一系列内容是用序号进行标识的，它用 `ol` 标签来表示一个有序列表，用 `li` 来表示每一个列表项。
3. 这里的 `ol` 是 `order list` 的意思，即”有序 列表“，而 `li` 是”list item“的意思，即”列表 元素“。
4. 我们可以用 `ul` 来表示一个无序列表，每一个列表项用 `li` 标签来表示，这里的 `ul` 中的 `u` 是 `un` 的第一个字母，表示否定，即没有顺序。
5. 我们可以用 `dl` 来进行一个自定义列表，其中的 `d` 是 `define` 的意思，即”定义“。
6. 而 `dt` 标签表示列表中的上层项目，`dd` 标签通常用来表示列表中的下层项目，而 `dt` 标签和 `dd` 标签都必须放在 `dl` 标签对之间。

#### \*\*\*\*\*代码示例\*\*\*\*\*

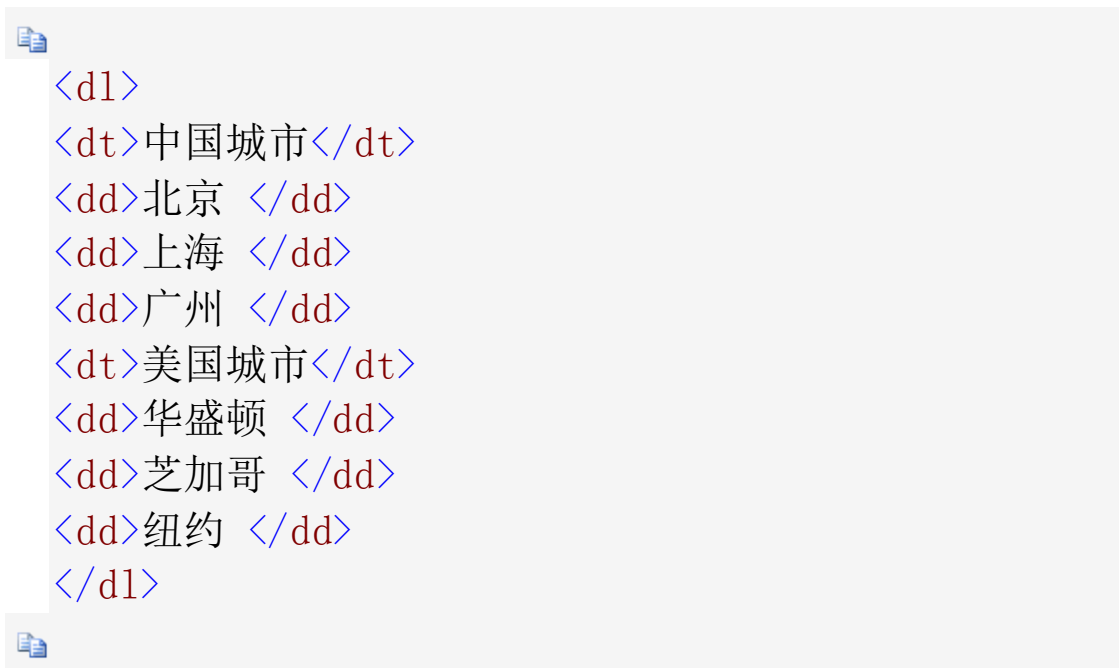
1. 首先是有序列表和无需列表，这里仅以有序列表为例子进行说明：

```
<ol>
  <li>辛星</li>
  <li>小倩</li>
</ol>
```

2. 我们看一下运行效果把：



3. 然后就是对定义列表的代码实例：



4. 我们来看一下它的运行效果吧：



#### \*\*\*\*\*小结\*\*\*\*\*

1. 本小节的前部分我们介绍了表格，后部分我们介绍了列表。
2. 列表的应用，特别是无序列表，其应用比较广泛，但是由于我们没有涉及到 CSS，因此先不讲了，等后面再讲解。
3. 无序列表不论在制作导航，还是商品列表、文章列表等功能的时候都显得异常强大，因为我们很多时候需要显示重复的信息，而最适合的就是列表（当然也得分情况）。
4. 而且懂后台编程的话，会发现列表配合循环将会是一个非常强大的组合，二者很有默契，不是吗？
5. 当然，也不全对，也要看情况。

## 第七节：区块

经过前面六节的学习，我们大致清楚了 HTML 教程中的基础内容，那么接下来我们开始继续向后推进，可以说，下面我们介绍一下 HTML 中的区块。

### \*\*\*\*\*区块\*\*\*\*\*

1. HTML 中的大多数元素都被定义为块级元素或内联元素，块级元素在显示的时候，会以新行开始，比如我们前面讲到的 h1 标签，p 标签，ol 标签和 table 标签。
2. 而内联元素则在显示的时候是不会另起一行的，它们会在后面显示，比如我们前面讲到的 td 标签，img 标签，strong 标签等等。

### \*\*\*\*\*div\*\*\*\*span\*\*\*\*\*

1. div 元素是一个块级元素，而且这个标签没有什么特定的含义，它就是表示一个块，由于是块级元素，因此每当我们遇到一个 div，都会换行。
2. div 通常需要配合 css 一起用，才能显示出它强大的威力，我们在后面讲 css 的时候也会经常用到 div，这里就先不说了。
3. span 元素是典型的内联元素，它也没有特定的含义，它就是定义一个内联元素，然后我们可以配合 css 来修改其内部样式。

### \*\*\*\*\*布局\*\*\*\*\*

1. 布局，翻译成英语通常就是 layout，一个网站的布局是非常重要的，一个合理的布局就像我们所说的”这个人身材很好“，而一个不好的布局，就会给人一种很别扭的感觉。

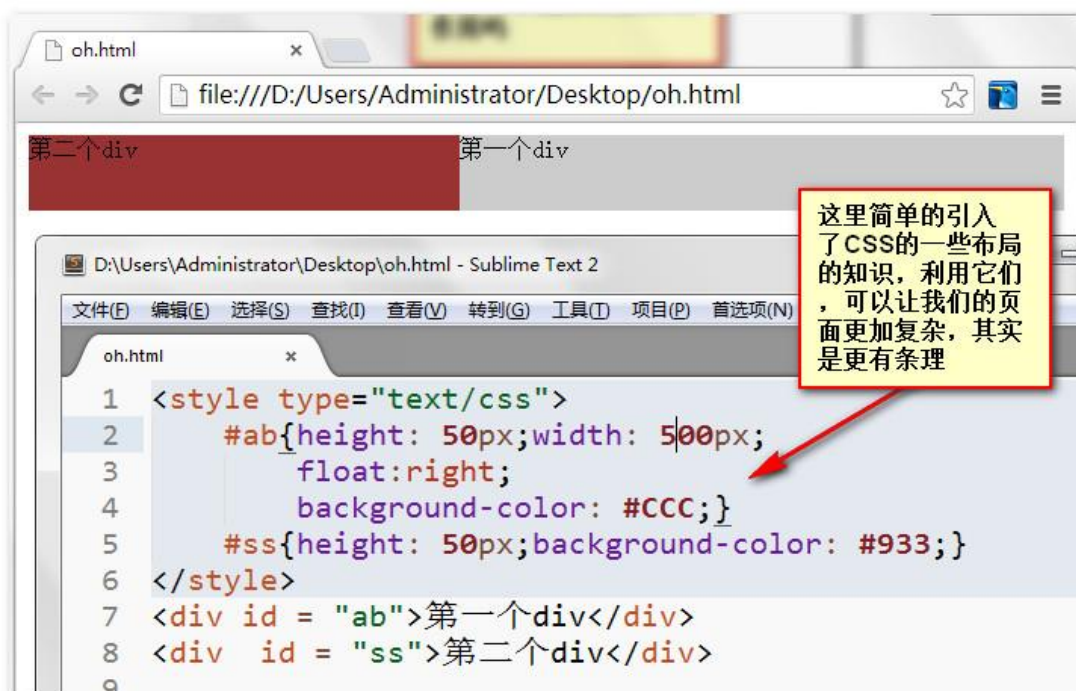
2. 关于布局的更多知识，我都放到后面的 css 教程里面去了，这里就先略过去了。

\*\*\*\*\*提前说一下\*\*\*\*\*

1. 让我们来看下一个 div 是什么把，代码和演示效果如下：



2. 我们加入背景色和大小来看一下：





## 第八节：小结

经过前面七节的学习，我感觉大家的 HTML 的功底也差不多了，而且我特别的删去了某些东西，比如框架，如果回到几年前，那么框架是非常流行的，但是现在都到了 2014 年了，这些东西早就该被遗忘了，因此，我果断的把这些我感觉过时的东西全部扔掉。

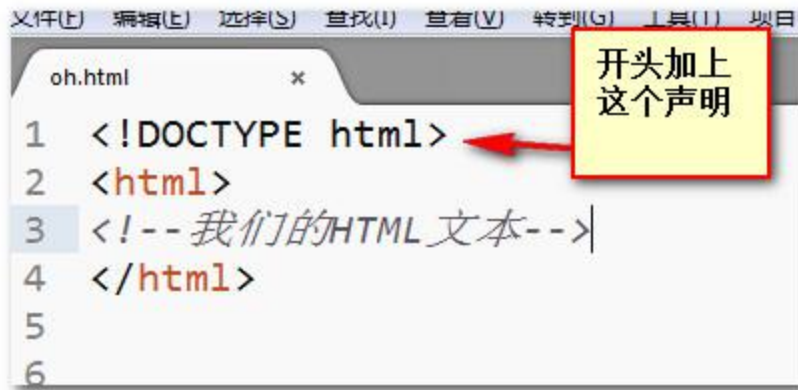
\*\*\*\*\*标准写法\*\*\*\*\*

1. 如果读者是初学，经过上面的学习，可能会乱了，比如我需要不需要写 html 标签呢？
2. 我想说的是，由于 HTML 语言是一门非常松散的语言，就是我们随便写两个汉字，然后保存为.html 格式的文件，它也会给我们去解析，比如：



3. 而且各大浏览器对 HTML 文件的解析大致是一样的，但是也有很多不同，比如我们在图片那一节就看到了谷歌浏览器和 IE 浏览器的区别，那么我们该支持哪一种呢？按理说，我们要尽可能的做到兼容性，剩下的，就让老板决定吧。
4. 其实标准写法也分为好几种，不过如果按照 HTML5 的写法一般情况下是这样的格式：





5. 如果我们遵循 XHTML1.0，写法大致是这个样子：



6. 目前来说，我还是推荐标题为 4 的写法。

#### \*\*\*\*\*HTML5\*\*\*\*\*

1. 而 HTML5 的话，本教程还是没有涉及，我认为现在学习 HTML5 已经可以了，毕竟 HTML5 的教程也有一些人出了，但是本教程不涉及，并不是我不建议大家学、不建议大家用，只是本教程没有提及而已。

2. 还有些东西，比如样式的设置，比如标签的 id 和 name 什么的，凡是我认为可以放到 CSS 教程里面的，都统一全部从 HTML 教程中划掉了，可能有些读者会持反对态度，但是我的经验告诉我：这么做更好一些。

3. 不管怎么说，HTML 教程告一段落，辛星 HTML，期待您的支持。

\*\*\*\*\*说明\*\*\*\*\*

1. 由于是第一版，虽然我经过了一些修改和校正，避免了错误的出现。
2. 但是还是肯定还会有这样或那样的不足，如果您喜欢这种教程的风格（更新快、截图展示、使用序号、使用星号分割内容等等），那么为何不加入到我们，一起把这个教程做好呢？
3. 期待您的支持。

## 第二部分：CSS 快速掌握

### 第一节：CSS 简介

CSS 是 Cascading Style Sheets 的缩写，即层叠样式表，它用于表现 HTML 的样式，即 HTML 只是去写该网页有哪些内容，至于如何去表现它们，由 CSS 去定制。

\*\*\*\*\*时代的呼唤\*\*\*\*\*

1. 在之前，我们直接把网页的格式写到 HTML 中去，这样会造成格式的混乱，而且难以阅读和修改，不过这不是最大的问题，最大的问题在于我们想给网页换一个表达样式的时候，需要修改的地方不仅很多关键是非常零散，需要花费很大的力气去定位这些样式。
2. 于是，把网页内容与其表现形式相分离的想法就出现了，CSS 能够实现对网页中的对象的精确控制，而且支持几乎所有的字体字号样式，还拥有对网页对象和模型样式进行编辑的能力，并且能够进行一定的交互设计，因此功能还是比较强大的。
3. CSS 的可读性也很强，稍微熟悉一点英语的，也能够很轻松的读懂 CSS 代码所要表述的意思，并且取修改它的样式。

\*\*\*\*\*CSS 的历史\*\*\*\*\*

1. 1996 年 12 月 17 日，CSS1 发布，1999 年，此推荐被修订。
2. 1999 年 1 月 1 日，CSS2 发布，并且增加了对媒介和一些字体的支持。
3. 2001 年 5 月 23 日，CSS3 发布，但是只是一项草案，并没有成为 W3C 的推荐标准，W3C 的推荐标准还是 1998 年 5 月 12 日退出的 CSS2。

\*\*\*\*\*CSS 快速入门\*\*\*\*\*

1. CSS 的入门比 HTML 并不高多少，也是一句话入门，剩下的就是实战了。

2. CSS 的语法如下：

```
选择器 { 声明 1; 声明 2; 声明 3; ..... 声明 N }
```

3. 这里的声明是以 “属性:值” 的形式，而且多个声明之间用分号进行分割，即 C 语言的语句分隔符。

4. 这里的属性是我们希望设置的样式属性，至于具体的属性，我们后面会说。

5. 先看一个例子：

```
h1 {color:red; font-size:14px;}
```

这里的 h1 就是选择器，当然这里我们直接用标签当做选择器了，后面的 color 属性取值为 red，而后面的 font-size 属性取值为 14 个像素。

6. 其中 color: red 就可以理解为一个声明。

7. 我们的 CSS 由多个这样的规则组成。

\*\*\*\*\*示例\*\*\*\*\*

1. 我们先写一个 HTML 文件，取名为 “xin.html”，内容如下：

```
<html>
<head>
  <title>2014 年辛星 CSS 教学夏季版</title>
  <link rel="stylesheet" type="text/css"
href="my.css">
</head>
<body>
  <p>辛星 CSS，期待您的关注，分享知识，传递温情
</p>
```

```
</body>  
</html>
```



2. 可以从第四行代码看出，它导入了一个 my.css 文件，然后在同一目录下新建一个 my.css 文件，书写内容如下：

```
p{color: green;font-size: 14px;text-align:  
center;}
```

3. 我们都保存完毕之后，就可以双击打开 my.html 文件，发现文字都在中间并且是绿色的。

4. 下面是它们的效果截图：



5. 至此，我们大致理解了 css 的工作机理，下面我们开始深入 css 的细节处开始学习。

\*\*\*\*\*小结\*\*\*\*\*

1. 我们理解了 CSS 的由来，以及我们演示了 CSS 的语法。
2. 然后我们通过一个小型的例子，演示了 CSS 的大致使用方式。

## 第二节：CSS 语法快速入门

第一节我们简介了一下 CSS 的工作流程，我相信读者会有一个大概的认识，那么接下来我们将会深入的研究一下 CSS 的细节问题，这些问题的涉及将会使我们的工作更加完善。

### \*\*\*\*\*注释\*\*\*\*\*

1. CSS 也是需要注释的，注释中的内容不会被解释执行，它也和 C 语言一样，用/\*开头，用\*/结尾，其中的内容会被当做注释，我们想怎么写就怎么写。

2. 下面给个例子：

```
p{    color: green;
    /* 我们在这里书写注释
    * 辛星 CSS，与博客园共同成长
    */
    font-size: 14px;
    text-align: center;}
```

### \*\*\*\*\*选择器\*\*\*\*\*

1. 我们第一节就介绍了 CSS 的基础语法了，其中就说到了选择器，在 CSS 中有两类稍微特殊点的选择器，我们称之为 id 选择器和 class 选择器。

2. 先说 id 选择器，我们在 HTML 的元素中，每一个元素都可以设置一个属性，该属性就是 id，然后我们在 CSS 中使用#号外加这个 id 的数值型来作为选择器，后面的声明就和普通的写法一样。

3. 光说是有点抽象的，我们下面给个例子：

首先新建一个 my.html 文件，书写内容如下：



```
<html>
```

```
<head>
  <title>2014 年辛星 CSS 教学夏季版</title>
  <link rel="stylesheet" type="text/css"
href="my.css">
</head>
<body>
  <p id = "para">辛星 CSS，分享知识，传递温情</p>
</body>
</html>
```



然后我们在 my.css 中这么写就可以了：

```
#para{    color: green;
font-size: 14px;
text-align: center;}
```

上面我们就通过在 HTML 文件中指定了 p 标签的 id，然后在 css 文件中根据 id 来设置了它的属性。

4. 它的效果图如下：



5. 所谓 id，那就是唯一的，在同一个 HTML 文件中，不能有两个 id 都是 para，这样会导致混乱，我们可以用下面的 class 来解决。

6. 如果说 id 是用来描述单个的样式，那么，class 则用于描述一组标签的样式，class 可以用在多个元素中使用。

7. id 使用#号来加上属性值，而 class 则使用. 号加上属性值，由于两者的使用及其的类似，我就不赘述了，想必聪明的读者已经可以仿造出来了。

### \*\*\*\*\*伪类\*\*\*\*\*

1. CSS 中的选择器是比较灵活的，比较常用的还有伪类的说法，下面我来解释一下，伪类的使用官方给出的语法格式是这样的：

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

2. 什么意思呢，就是 selector 就是选择器嘛，而 pseudo-class 就是相应的伪类了，后面跟的就是它的属性和值。

3. 其实伪类的最经典的例子之一就是超链接，比如我们可以书写如下代码：

首先是创建一个 my.html 文件，代码如下：



```
<html>
<head>
  <title>2014 年辛星 CSS 教学夏季版</title>
  <link rel="stylesheet" type="text/css"
href="my.css">
</head>
<body>
  <a href="#">博客园辛星</a>
</body>
</html>
```



然后就是 my.css 文件，代码如下：

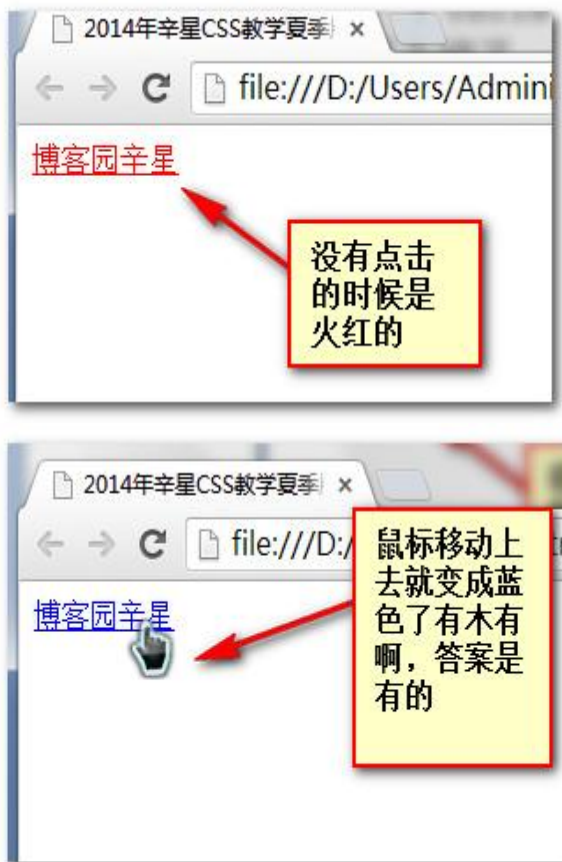
```
a:link{color:red;}
a:visited{color: green;}
```

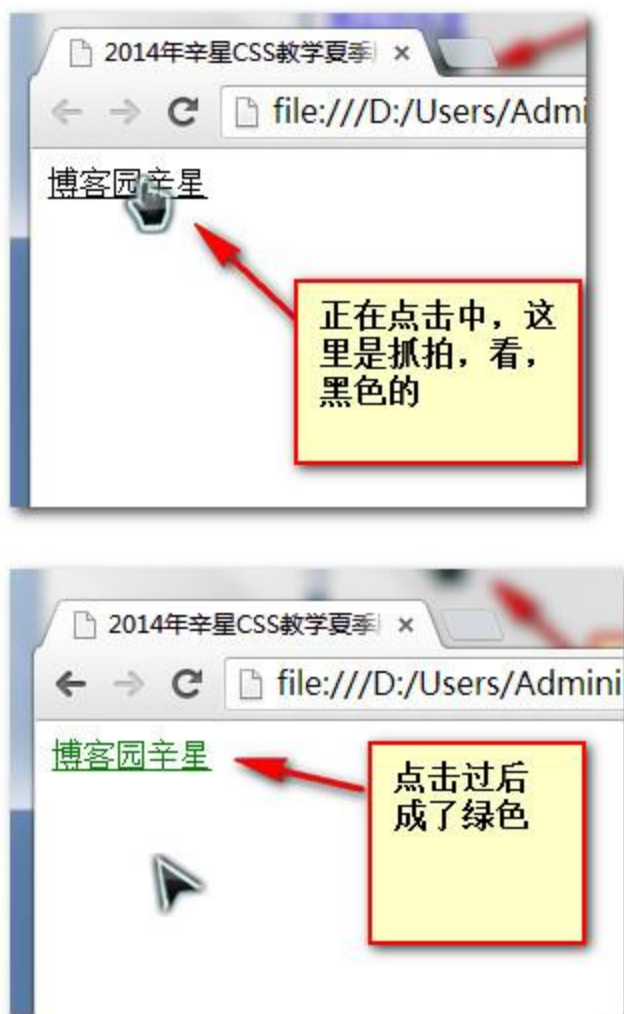


```
a:hover{color:blue;}  
a:active{color:black;}
```

4. 上面的 css 文件中我们就设置了当我们没有点击超链接的时候是红色的，当我们的鼠标经过的时候是蓝色的，当我们的鼠标点击它而又没有松开的时候，它是黑色的，当我们访问过的连接，它会呈现绿色。

5. 下面我们实例演示一下吧：





\*\*\*\*\*进一步说明\*\*\*\*\*

1. 而且 `a: hover` 必须被放到 `a: link` 和 `a: visited` 之后，才会有效。
2. 而 `a: active` 必须被当道 `a: hover` 之后，才有效。
3. 我们还可以让伪类与上面提到的 `class` 选择器一起用，我感觉不是很常用，就不介绍了。
4. 还有一个 `child` 伪类，但是对浏览器的兼容性上不好，因此也不介绍了。

5. 此外还有伪元素的说法，如果大家做的事英文网站，还是比较重要的，但是对于汉语网站来说，意义不大，因此也就不说了。

#### \*\*\*\*\*其他选择器\*\*\*\*\*

1. CSS 还支持几种其他的选择器类型，第一种就是后代选择器，比如我们把 a 插入到一个 div 里面去，那么可以用 div a 这样的选择器来表示 a 的样式。
2. 第二种就是子元素选择器，它会选择父元素中的所有子元素，比如 div>a，即它和第一个的区别就是把空格换成了大于号。
3. 另外的相邻兄弟选择器和普通相邻兄弟选择器，总体感觉意义不大，只会增加记忆量和学习的难度，实用性并不高。
4. CSS 还支持属性选择器，但是由于也是对浏览器的兼容性上不好，也就不介绍了。
5. 还有所谓的群组选择器的说法，是指几个选择器使用同一个样式，它用逗号把各个选择器进行区分，并且连接起来，就像我们 C 语言中函数的参数列表中的多个参数一样，示例：

```
p, td, li {color: red;}
```

6. 还有通用选择器的说法，就是用一个\*表示，如下代码：

```
p* {color: red;}
```

会把所有的段落的颜色都设置为红色。

\*\*\*\*\*小结\*\*\*\*\*

1. 本小节我们讲述了 CSS 选择器和注释。
2. CSS 常用的选择器就是标签选择器、id 选择器、class 选择器和伪类选择器，其他选择器的方式都可以用上面这几种方式来搞定。

\*\*\*\*\*说明\*\*\*\*\*

1. 我们后面的实例中也会大量的使用选择器，没办法，要操纵 HTML 元素，必须得借助于选择器。
2. 而且大家在学习 jQuery 的时候，也必然会接触选择器，因为选择器的使用太广泛了。

### 第三节：CSS 中的颜色与文字

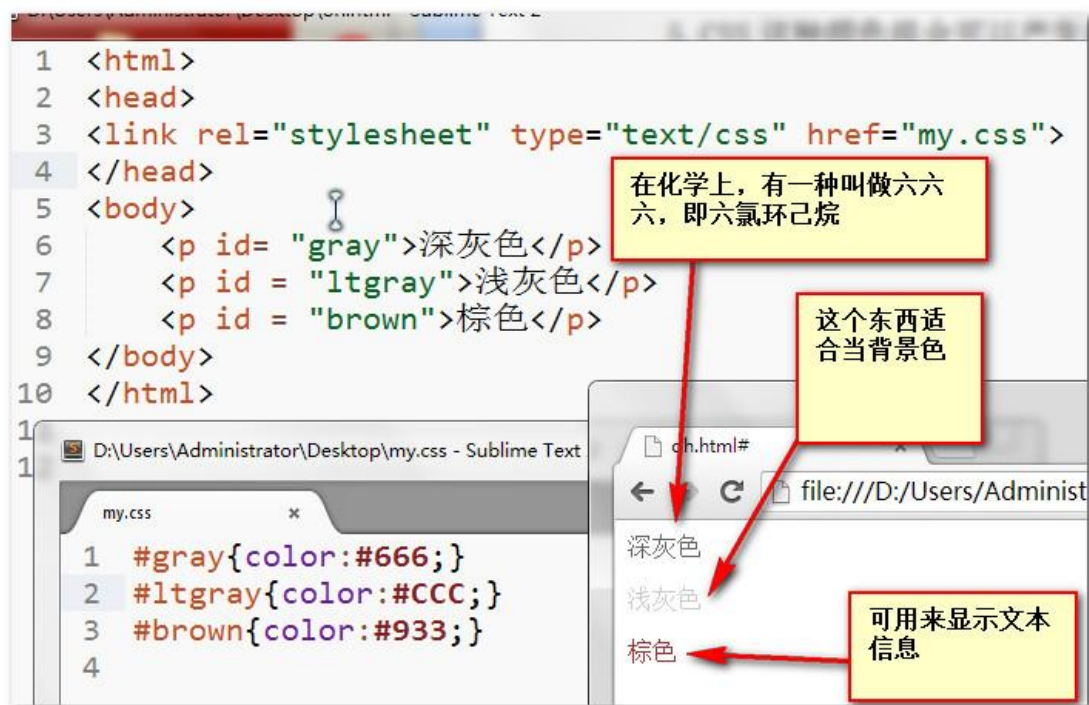
第二节我们讲述的几乎全是 CSS 的选择器，那么下面这一节我们来讲一下 CSS 的颜色和文本的一些东西，虽然我对调色不大敏感，但是对于颜色还是比较感兴趣的。

#### \*\*\*\*\*CSS 中的颜色\*\*\*\*\*

1. 说到颜色，不得不一下牛顿，简直是伟大，光的色散，非常大的突破，可以说它一个人完成了几代人没有完成的工作。
2. 下面说一下三原色：即红绿蓝，即 RGB，我们在 CSS 中定义一个颜色可以用这种三原色的方式来自己配色，还可以用一些比如 red, green 等这种比较经典的颜色。
3. 我们要配色的话，首先需要以#开头，然后跟上三个双位数字的十六制的写法，所谓十六进制，即每一位从 0 到 F，即包括 0、1、2、3、4、5、6、7、8、9、A、B、C、D、E、F。
4. 其中#000000 是黑色，#FFFFFF 是白色，#FFFF00 是黄色，#CCCCCC 是一款比较经典的灰色，关于如何配色，我还没有很好的经验可以和大家分享，总之数值越大，颜色越淡，数值越小，颜色越深。
5. CSS 这种颜色组合可以产生出大约十六万种颜色，而且这些大多数还是可以被浏览器正确的解释的。

#### \*\*\*\*\*几种常见颜色\*\*\*\*\*

1. 在说这个之前，其实颜色是支持简写方式的，即我们在#号后面不写六个字符，而是写三个，比如写 FFF，它就自动当成 FFFFFFFF，即六个里面每两个都是一样的。
2. 下面我们通过 CSS 文本的方式介绍几种我喜欢的颜色：



### \*\*\*\*\* 文本的样式 \*\*\*\*\*

1. 虽然网站的布局追求多样化，但是，文字信息仍然是不可少的，下面我们就文字说一下 CSS 中的使用。
2. 关于颜色，用 color 属性，可以指定文字的颜色，这一点我想大家应该知道怎么用了吧。
3. text-align 属性是指文本的对其方式，可以设置为 left（居左）、right（居右）、center（居中）、justify（每一行被展开且宽度相等，左右外边距对齐）。
4. text-decoration 是指用来设置或者删除文本的装饰，它的最经典的应用就是删除超链接的下划线，如下代码：

```
a {text-decoration:none;}
```

5. 上面我们就是删除了超链接的下划线，比如我们还可以设置如下：

```
1 p {text-decoration:overline;}
2 p {text-decoration:line-through;}
3 p {text-decoration:underline;}
```

第一个是添加了上划线，第二个是中划线，第三个是添加了下划线，不过下划线用的比较广泛，大家比较好接受，上划线和中划线有点怪怪的感觉。

6. 我们还可以设置大小写字母、缩进之类的，我感觉这些对汉语意义不大，倒是英文网站蛮适合的，就不讲了。

#### \*\*\*\*\*字体\*\*\*\*\*

1. CSS 中我们可以相对自由的设置字体，我们可以用 `font-family` 来指定字体名称，我对字体的研究不多，也就知道那么几个。

2. 我们可以用 `font-size` 来指定字体大小，我们可以用 `px` 表示的方式，即像素的格式。浏览器时有默认的大小的，普通文本的大小是 `16px`，我们可以以用 `font-size` 来指定该字体的大小。

3. 我们除了用 `px` 来指定字体的大小，还可以用 `em` 来指定大小，`em` 的尺寸单位由 W3C 建议，`1em` 相当于 `16px`，即我们想用 `40px` 的时候，就是使用 `2.5em`。

4. 我们可以通过 `font-style` 来设置字体的风格，它的两个经典样式就是 `italic`，即倾斜，如下代码：

```
p {font-style:italic;}
```

5. 下面就来一个大杂烩把：

首先创建一个 `my.html`，内容如下：



```
<html>
<head>
  <title>2014 年辛星 CSS 教学夏季版</title>
  <link rel="stylesheet" type="text/css"
href="my.css">
</head>
```



```
<body>
  <p>博客园辛星，传递温情</p>
</body>
</html>
```



然后创建一个 my.css，内容如下：

```
p{color: red;text-align: center;
   font-size: 13px;font-style: italic;}
```

6. 下面是效果图：



\*\*\*\*\*小结\*\*\*\*\*

1. 本节我们主要涉及的就是颜色和字体的排版以及字体的样式。
2. 这里需要说明的是一个 text 类的，它用于排版，是一群文字的排布方式。
3. 而 font 则是每个字的格式，比如字体，字号，字体是否倾斜等等。

## 第四节：超链接与列表以及简易导航栏

接下来的这一节我计划讲解的是超链接和列表的样式，然后我们做出一个导航栏出来，其实导航栏是非常常见的，但是我们这里做得这个有点并不那么完善，等我们学完了 CSS 之后再完善它。

### \*\*\*\*\*超链接\*\*\*\*\*

1. 我们讲伪类的时候已经讲到了，一个超链接可以有 `link`（正常状态）、`visited`（已经访问过）、`hover`（获得鼠标焦点）、`active`（被鼠标点击中）这样四个状态，我们前面也分别设置过不同的颜色了，使用选择器：伪类的形式。
2. 我们还可以使用 `background-color` 来设置其背景颜色，我们可以用 `text-decoration: none` 来去掉超链接的下划线。

### \*\*\*\*\*列表\*\*\*\*\*

1. 我们前面讲 HTML 的时候讲过，列表分为有序列表和无序列表，我们这里讲解它的一个属性：`list-style-type`。
2. `list-style-type` 属性是什么意思呢，就是列表的风格类型，对于无序列表，可以用 `circle` 和 `square`，即分别用圆形和方形来作为无序列表的开头。
3. `list-style-type` 属性还可以取值为 `upper-roman` 来设置为大写罗马数字，还可以用 `lower-alpha` 来设置为小写的英文字母。
4. 它还有一个属性是 `list-style-image`：用图片来作为列表项的开头，它需要用 `url` 导入一个图片，这个我们后面再说。


### \*\*\*\*\*实现一个导航栏\*\*\*\*\*

1. 第一步，我们先来一个无序列表，然后给他们加上超链接，由于这里我不知道连接到哪里，用#表示一个空连接，下面我们新建一个 my.html 代码如下：



```
<html>
<head>
  <title>2014 年辛星 CSS 教学夏季版</title>
  <link rel="stylesheet" type="text/css"
href="my.css">
</head>
<body>
  <ul>
    <li><a href="#">辛星</a></li>
    <li><a href="#">小倩</a></li>
    <li><a href="#">小楠</a></li>
    <li><a href="#">辛勇</a></li>
    <li><a href="#">辛强</a></li>
  </ul>
</body>
</html>
```

2. 然后新建一个 my.css 文件如下：



```
ul{list-style-type: none;}
a{text-decoration: none;}
li{display: inline;}
```

3. 它的效果如下：



4. 我们的这个横向的导航条就算完成了，但是它还是太简单了，它只是单纯的列出了五个连接，并且水平排列，我们到后面再去完善。

5. 这里说明一下，这个 `li` 属性的 `display: inline` 表示设置格式为内联，即它不是以块级元素出现的，是内联元素，即向右排列而不是向下排列。

#### \*\*\*\*\*小结\*\*\*\*\*

1. 本小节我们认识了超链接和列表的常见样式。
2. 然后我们做出来了一个简易的导航条，读者可以先去完善它，如果读者不去做，我后面再做。

## 第五节：盒模型

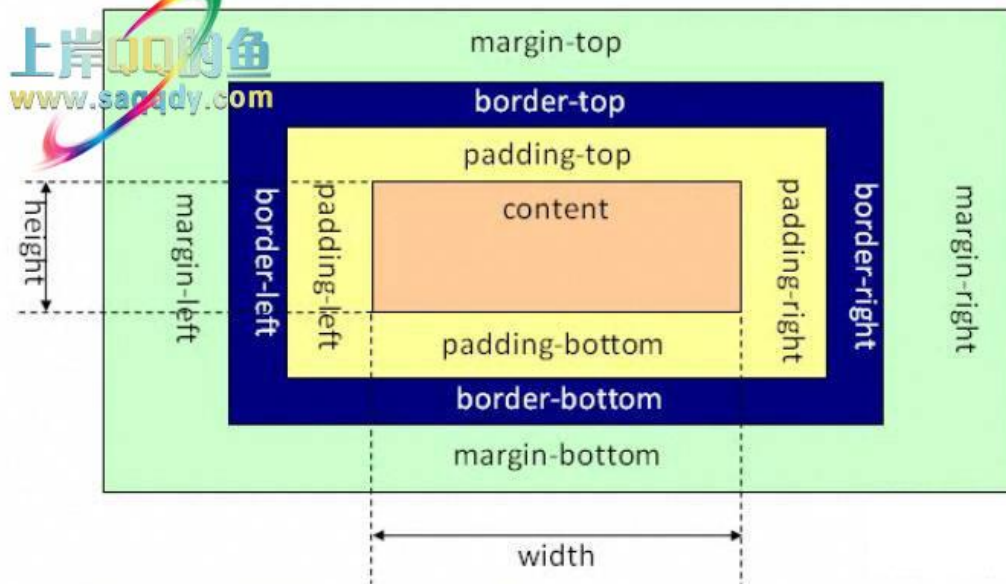
本小节我们讲解 css 中的“盒模型”，即“box model”，它通常用于在布局的时候使用，这个“盒模型”也有人成为“框模型”，其实原理都一样。

它的大致原理是这样的，它把一个 HTML 元素分为了这么几个部分：边距、边框、填充和实际内容，我们通过设置这几个内容可以设置它的一些现实形式。

\*\*\*\*\*盒模型\*\*\*\*\*

1. 盒模型从内向外依次是实际内容（Content）、内边距（Padding）、边框（Border）、外边距（Margin），其中我们通常也说内边距为边距，外边距为填充。
2. 下面是我找的一个图片来解释这个盒模型：

### ■ 标准盒子模型



3. 这里还是用文字来描述一下把，Content 是盒子的内容，比如我们需要显示的文本信息，Padding 是它守卫的区域，但是它的大小会受到 border 的约束，而 border 则是这个边框，而 margin 则是边框的周围区域，即该盒子与外界的元素距离。

### \*\*\*\*\*进一步说明\*\*\*\*\*

1. 因此我们在计算一个元素的宽度和高度的时候，这里我们以宽度为例，其实是 content 的宽度加上 padding 的宽度乘以 2，再加上 border 的宽度乘以二，在加上 margin 的宽度乘以 2。

2. 这些宽度和高度我们直接用 px 来表示就可以了，唯一比较特殊的就是这个 border，我们还需要指定它的一些样式和颜色，它的第一个参数是大小，第二个参数是风格（实现还是虚线等），第三个参数是颜色。

3. 代码演示：

```
#par{width:220px;
padding:10px;
border:5px solid gray;
margin:0px;
}
```

4. 其实很好理解的，它的盒模型只是说法稍显专业化，其理解很简单。

### \*\*\*\*\*边框\*\*\*\*\*

1. 上面我们只是简单的讲解了一下 border，其实 border 有几个属性可以分开设置的，首先是 border-style，它指的是边框的线的绘制方式，可以用 none 来表示无边框，可以用 dashed 来定义一个虚线框，可以用 solid 来定义一个实线框，可以用 double 来定义一个双线框，还可以定义带有 3D 样式的边框，比如用 groove 来定义一个 3D 沟槽边界。

2. 我们可以用 border-width 来定义边框的大小，可以用 px，也可以用 em，这些前面都讲过了，还有三个选项，个人不建议用。

3. 还可以用 border-color 来设置边框的颜色，单独的设置 border-color 是无意义的，必须在设置了 border-style 之后再设置 border-color 才有意义。

4. 这些边界属性可以接受 1 个或者四个值，如果是接受四个值，则是按照上右下左的顺序去实现，如果是三个值，则设置上右下，如果是两个值，则上下采用第一个值，左右采用第二个值，如果是一个值，则所有的边界都采用同一个值。

5. 示例代码，首先是 my.html 代码：



```
<html>
<head>
  <title>2014 年辛星 CSS 教学夏季版</title>
  <link rel="stylesheet" type="text/css"
href="my.css">
</head>
<body>
  <p>博客园辛星，无限温情</p>
</body>
</html>
```



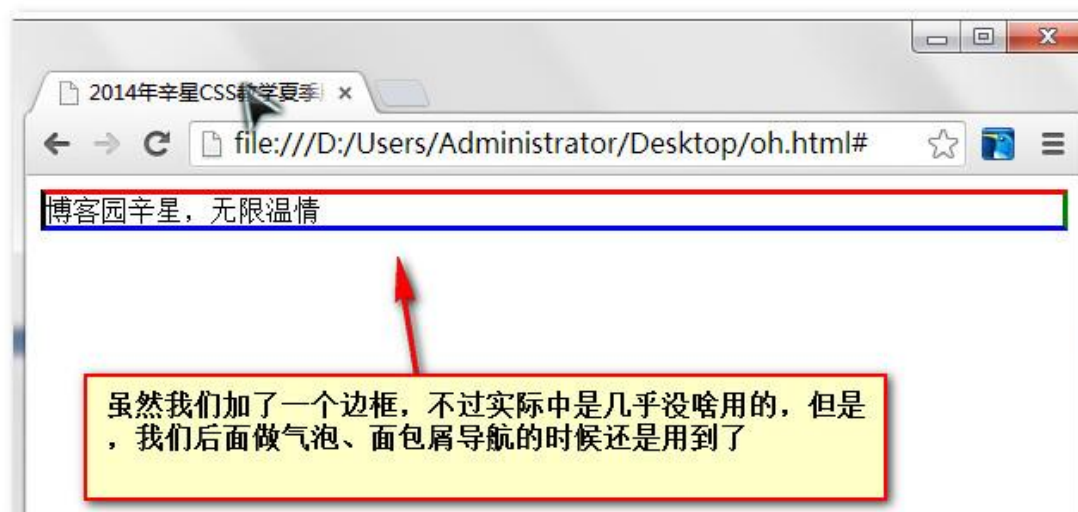
然后是 css 代码：

```
p{border-size:9px ;
  border-style: solid;
  border-color: red green blue black;}
```

6. 其实不光是 border，padding 和 margin 也是接受一个到四个值，用法一样。

7. 下面是它的效果图：





\*\*\*\*\*小结\*\*\*\*\*

1. 本小节我们主要讲解的是布局这部分，也就是盒模型。
2. 希望我能够表达的足够清楚。

## 第六节：定位

这一节我们就要讲到布局了，其实布局本身特别简单，但是要合理的布好局就不那么简单了，就像我们写文章一样，写一篇文章非常简单，但是要写一篇名著就很难了，这需要我们扎实的功底和对文学的理解，但是，千里之行，始于足下，我们开始吧。

### \*\*\*\*\*显示方式\*\*\*\*\*

1. 有时候我们会设置一个元素的显示方式，比如我们在做导航条的时候，就会使用 `display: inline` 来使它们成为内联的样式。
2. 下面先不说这个 `display`，先说一下 `visibility` 属性把，它可以设置为 `hidden`，即不显示，那么该块区域就显示一片空白。有一个类似的方法就是使用 `display: none`，它也是设置不显示，但是它的作用却是让该元素消失，后面的元素会自动占据它的空间。
3. 我们可以使用 `display: block` 使得它的显示成为一个块级元素，可以理解为强制性的换行，而 `display: inline` 则是强制性的使其不换行，向右排列。
4. 变更元素的显示类型看该元素如何显示，看它是什么样的元素，比如一个内联元素设置为 `display: block` 是不允许有它的内部嵌套块元素的。

### \*\*\*\*\*大小\*\*\*\*\*

1. 对于一个 HTML 元素，我们可以通过设置它的 `width` 属性和 `height` 属性来设置其大小，也就是其宽度和高度。
2. 还有一些其他设置大小的属性，感觉不是很常用，即没列举。

### \*\*\*\*\*定位\*\*\*\*\*

1. 第一种定位方式就是默认的方式，它是自上而下，从左到右的一个模式，也是最简单的一种模式。
2. 第二种就是固定定位，它的位置相对于浏览器的窗口位置是固定的，即使窗口滚动，它也不会动，这种可以用来做那种广告条之类的，我们下面做一个示例，首先是 HTML 文件中书写代码如下：



```
<html>
<head>
  <title>2014 年辛星 CSS 教学夏季版</title>
  <link rel="stylesheet" type="text/css"
href="my.css">
</head>
<body>
  <p id = "fix">博客园辛星，无限温情</p>
  <p>这些意义可以是政治思想方面的，也可以表达某种哲理，或某种情趣；写景，要通过景物描写表现出个人某种感情或深刻的感悟；咏物，或透露出世间人生的某种乐趣，或托物言志，表现对社会上某种人某种现象的情感。因此，我们写记叙文总会有一定的目的，总要表达一定的思想和感情，实际上就是我们平常所讲的“文章的主题”，主题是文章的灵魂，它像一根红线贯串于文章的始终。这些意义可以是政治思想方面的，也可以表达某种哲理，或某种情趣；写景，要通过景物描写表现出个人某种感情或深刻的感悟；咏物，或透露出世间人生的某种乐趣，或托物言志，表现对社会上某种人某种现象的情感。因此，我们写记叙文总会有一定的目的，总要表达一定的思想和感情，实际上就是我们平常所讲的“文章的主题”，主题是文章的灵魂，它像一根红线贯串于文章的始终。</p>
```

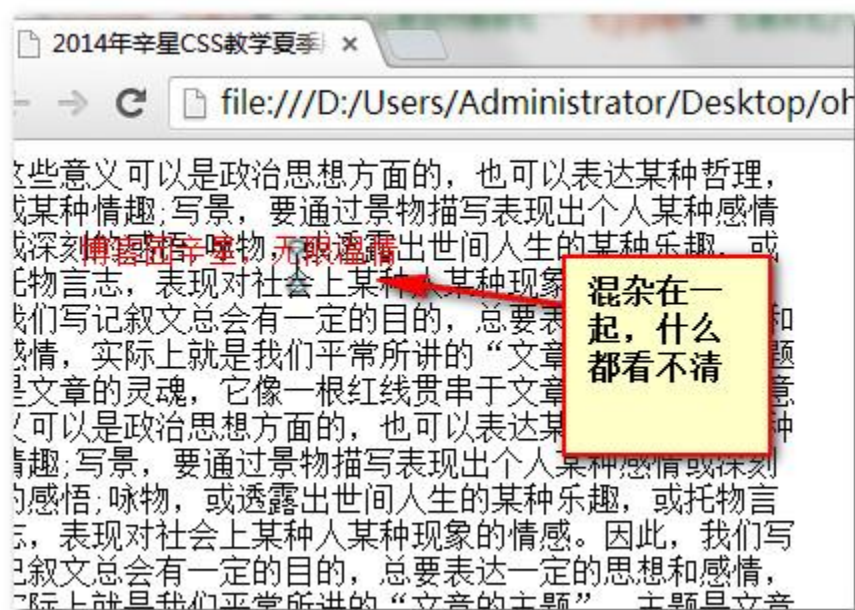
```
</body>
</html>
```



然后就书写 my.css 文件，这里我们为了看出随着滚动条的滚动，该段文字的位置不变的特性，我们需要给浏览器窗口一定的大小：

```
body{width:400px;
height: 900px;}
p{position: fixed;
top:30px;
left: 50px;}
```

3. 下面是它的效果图：





4. 上面这种方式对浏览器的兼容不好，在 IE8 以下需要描述!DOCTYPE 才可以，这里的 top 指的是它的上面到浏览器的顶部的距离，left 是指它的右端到浏览器的右端的距离。

\*\*\*\*\*绝对定位和相对定位\*\*\*\*\*

1. 相对定位：这里需要清楚相对“谁”的定位，这里的相对定位是相对于第一种方式的定位，即默认方式的定位，它会在默认方式的基础上进行一些修改，可以设置 left 等属性，如果设置 left 为 -20px，则表示它相对于原始位置向左移动二十个像素，它也需要在 position 属性中赋值为 relative。

2. 绝对定位：它需要指定 position 为 absolute，也可以设置 left 和 top 或 right 等属性，如果该元素没有父元素，则它的位置是相对整个浏览器窗口来说的。

\*\*\*\*\*进一步解释\*\*\*\*\*

1. 我发现有些童鞋对此很迷茫，因此我进一步解释一下吧。
2. 所谓相对定位，就是设置为相对定位的元素会偏移某个距离，元素仍然保持原来的样式，它会保留该 HTML 元素的空间，比如我们看如下代码：

[css] [view plaincopyprint?](#)

```
1. .demo{
2. position: relative;
3. width:50px;
4. height: 50px;
5. left:50px;
6. }
```

3. 它会出现现在相对原来的地方向右 50px 的地方，而它原来所在的地方为空。
4. 下面说一下绝对定位，所谓绝对定位，即不占用空间，如果原来这个地方有 HTML 元素，那么就会重叠，这里需要特别之处的，也是我发这篇博文的目的，就是想告诉大家，绝对定位如果被父元素包含，如果父元素没有定位属性，它会根据 body 进行定位，因此，如果它有父元素，必须在父元素中指定定位属性，如下代码：

[html] [view plaincopyprint?](#)

```
1. <html>
2. <head>
3.     <title>辛星解读绝对定位和相对定位
   </title>
4.     <style type="text/css">
5.         .father{
6.             position: relative;
7.             width:300px;
8.             height: 300px;
```

```
9.             margin: 0 auto;
10.             border: 1px solid orange;
11.         }
12.     .sun{
13.         position: absolute;
14.         background: purple;
15.         width: 50px;
16.         height: 50px;
17.         left: 50px;
18.     }
19.         </style>
20.     </head>
21.     <div class = "father">
22.         <div class = "sun"></div>
23.     </div>
24. </html>
```

5. 效果图:





6. 可能有童鞋会问，如果我们不指定父元素的定位又会如何呢？请看下面代码，我把 css 代码中的哪一行给注释掉之后，源代码如下：

[html] view plaincopyprint?

```

1. <html>
2. <head>
3.     <title>辛星解读绝对定位和相对定位
   </title>
4.     <style type="text/css">
5.         .father{
6.             //position:  relative;
7.             width:300px;
8.             height:  300px;
9.             margin:  0  auto;
10.            border:  1px  solid  orang
   e;
11.        }
12.        .sun{

```

```
13.                position:absolute;
14.                background:purple;
15.                width:50px;
16.                height: 50px;
17.                left: 50px;
18.                }
19.            </style>
20.    </head>
21.    <div class = "father">
22.        <div class = "sun"></div>
23.    </div>
24. </html>
```

7. 然后是它的效果图:



\*\*\*\*\*小结\*\*\*\*\*

1. 本小节讲解了显示方式和定位方式，为后面的布局打下一定的基础。
2. 我们下一小节将会讲解布局方面的问题。

## 第七节：浮动

\*\*\*\*\*浮动\*\*\*\*\*

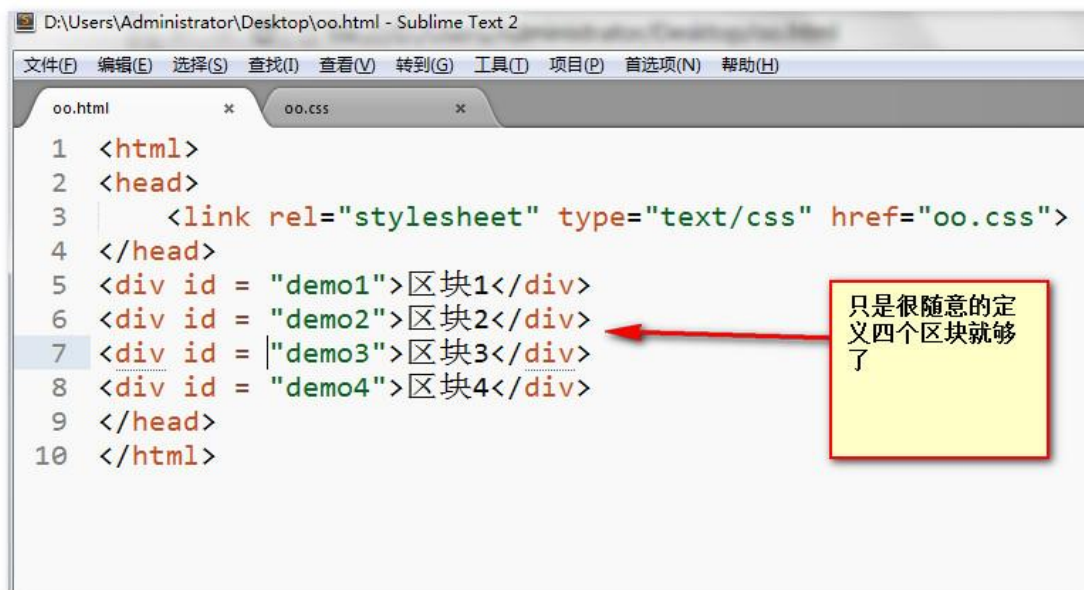
1. 浮动，是 CSS 布局中的必修课，如果浮动学不好，那么 CSS 的布局就像空中楼阁，不够稳定，而谈到浮动，其实也是结合 div 来说的，所谓 div，就是一个块级元素。

2. 我们可以用 CSS 的属性 float 来定义浮动属性，left 是向左浮动，right 是向右浮动。

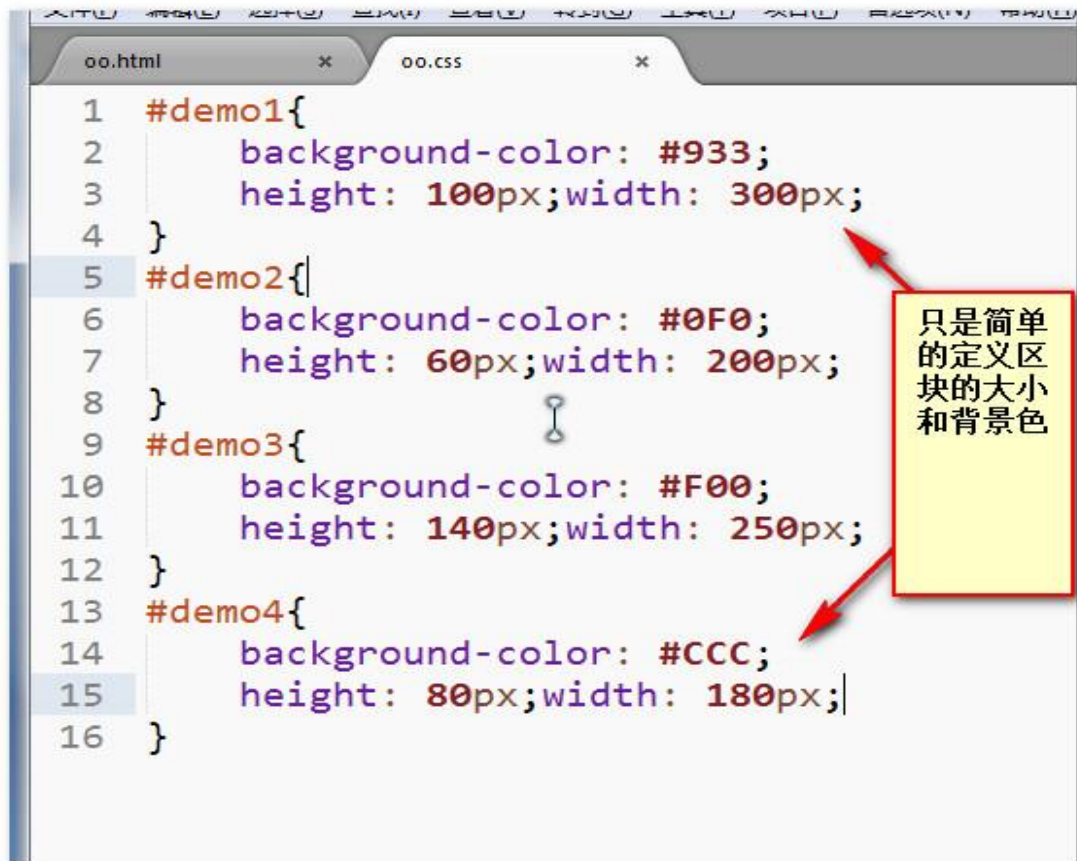
3. 我有两篇博客是专门介绍浮动的，大家可以通过搜索“辛星 浮动”来找到它们，也可以通过 <http://blog.csdn.net/xinguimeng/article/details/38091685> 去查看。其实这一节我就是按照这篇博客来讲的。

\*\*\*\*\*不浮动的情况\*\*\*\*\*

1. 首先我们先探究不浮动的情况，我们先写一个 oo.html 文件如下：



2. 然后我们看一下 oo.css 文件，内容如下：



3. 下面是它的运行效果:



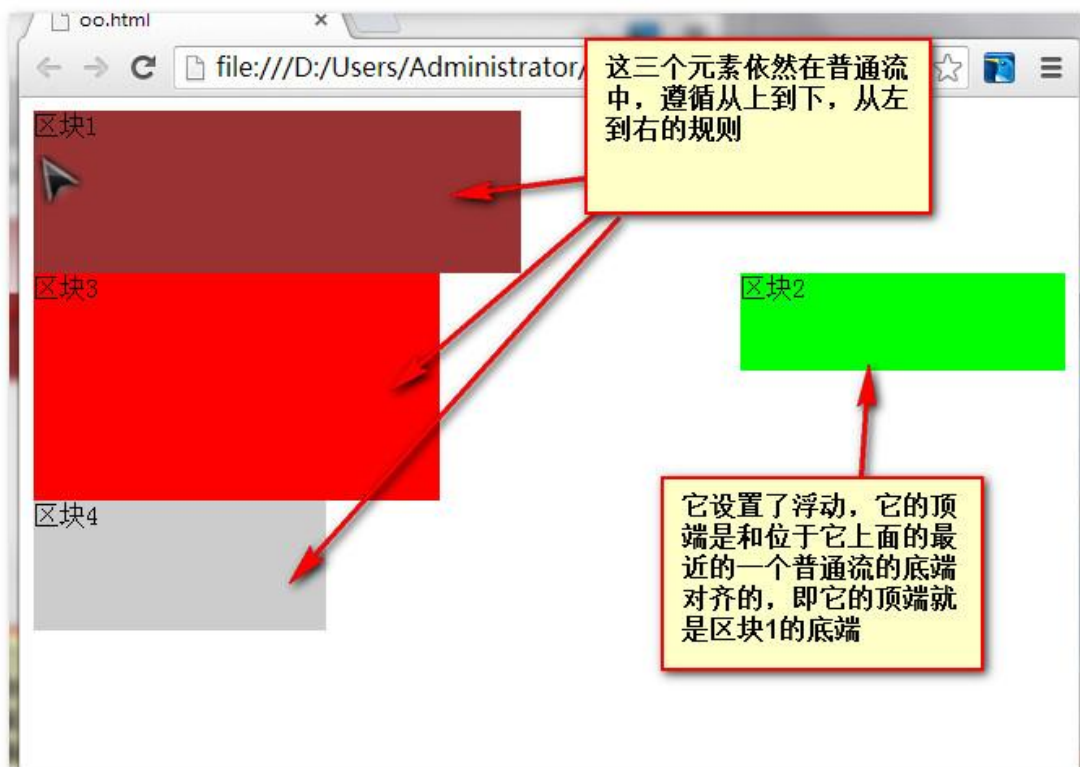
### \*\*\*\*\*一个元素的简单浮动\*\*\*\*\*

1. 下面我们开始设置浮动，在这之前我先说一下：“一旦某个元素设置了浮动，它就脱离了普通流，而普通流的文本是从上到下进行排列的，浮动只是确定他们的左右位置。且浮动的元素默认在普通流之上显示。”

2. 好，下面我们通过添加一行代码来让区块 2 进行右浮动，首先给 css 代码截个图把：



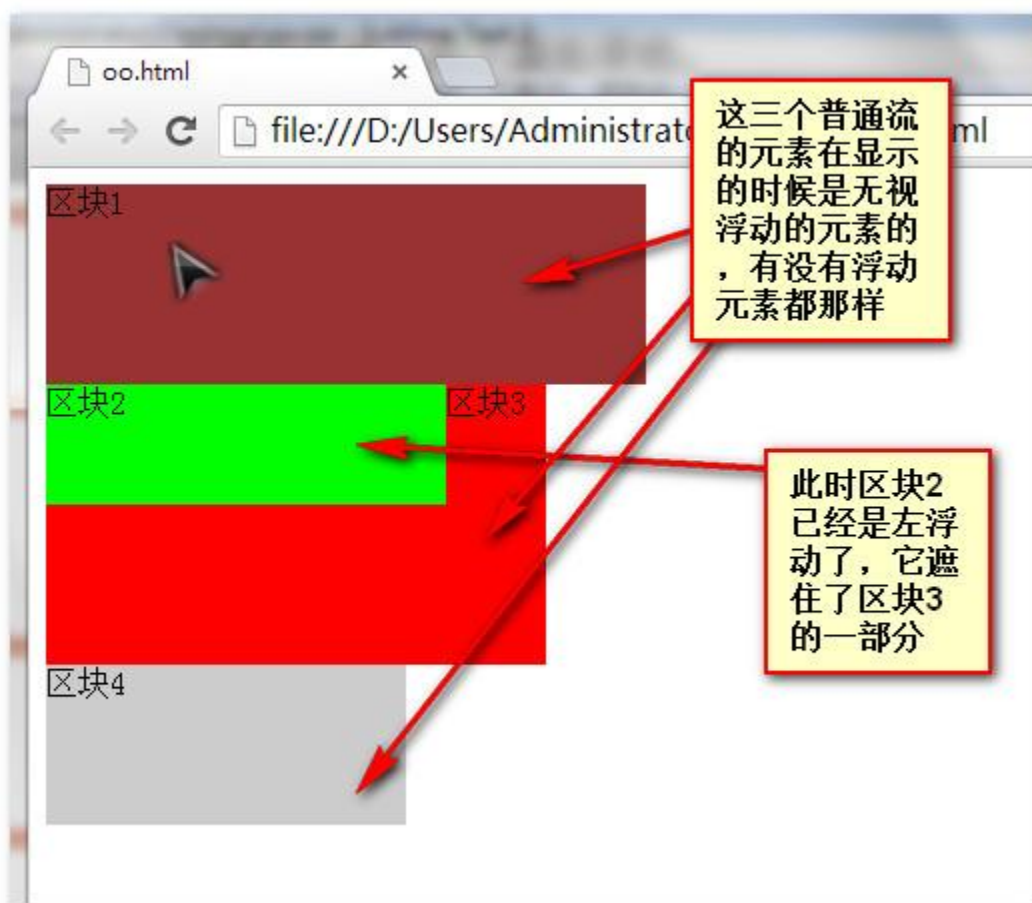
3. 然后我们看一下界面效果：



4. 我想经过上面的讲解大家应该明白了吧，其实是浏览器在渲染的时候首先把普通流的元素显示出来，然后查看一

下区块 2，发现它的上面一个在普通流中的元素是区块 1，于是就在区块 1 的下面右浮动。

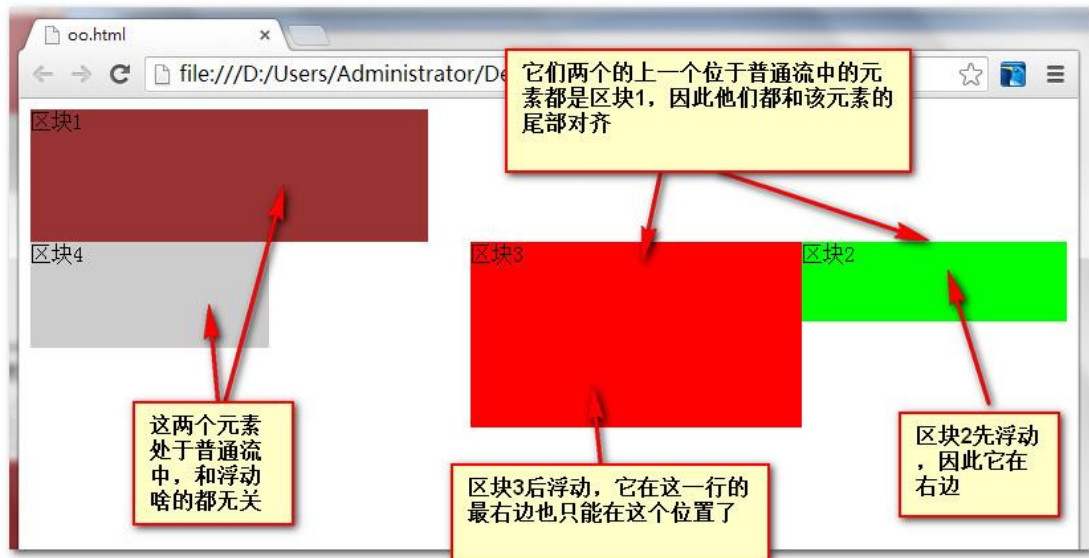
5. 那么区块 2 左浮动呢？答案是类似的，我们首先把普通流中的内容显示出来，然后显示区块 2，发现它的上一个位于普通流中的元素是区块 1，于是就紧挨着区块 1，下面是截图：





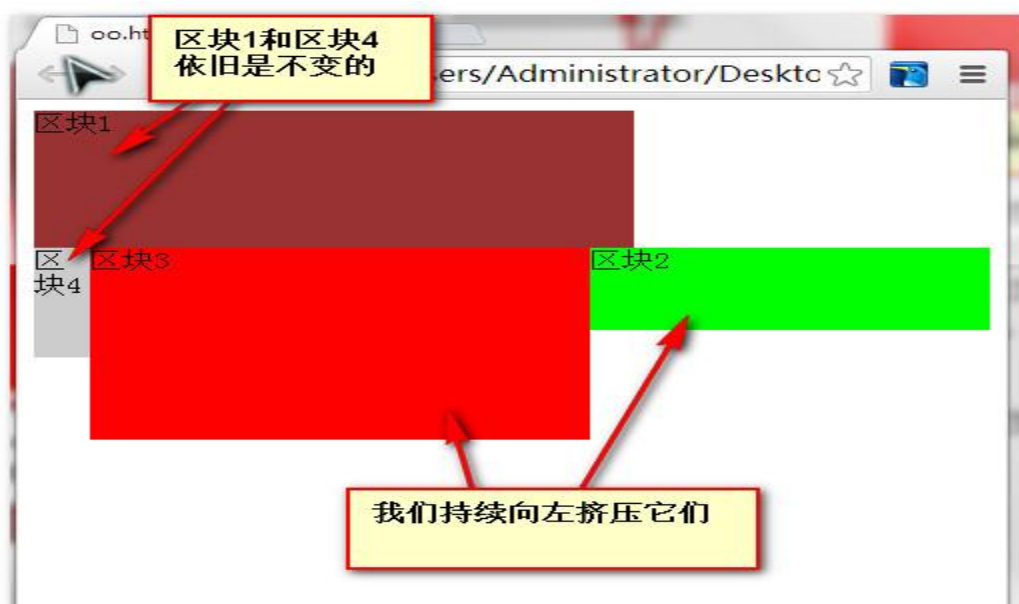
\*\*\*\*\*两个元素一起浮动\*\*\*\*\*

1. 下面我们让区块 2 和区块 3 一起浮动，首先我们让区块 2 和区块 3 都是右浮动，那么截图如下：

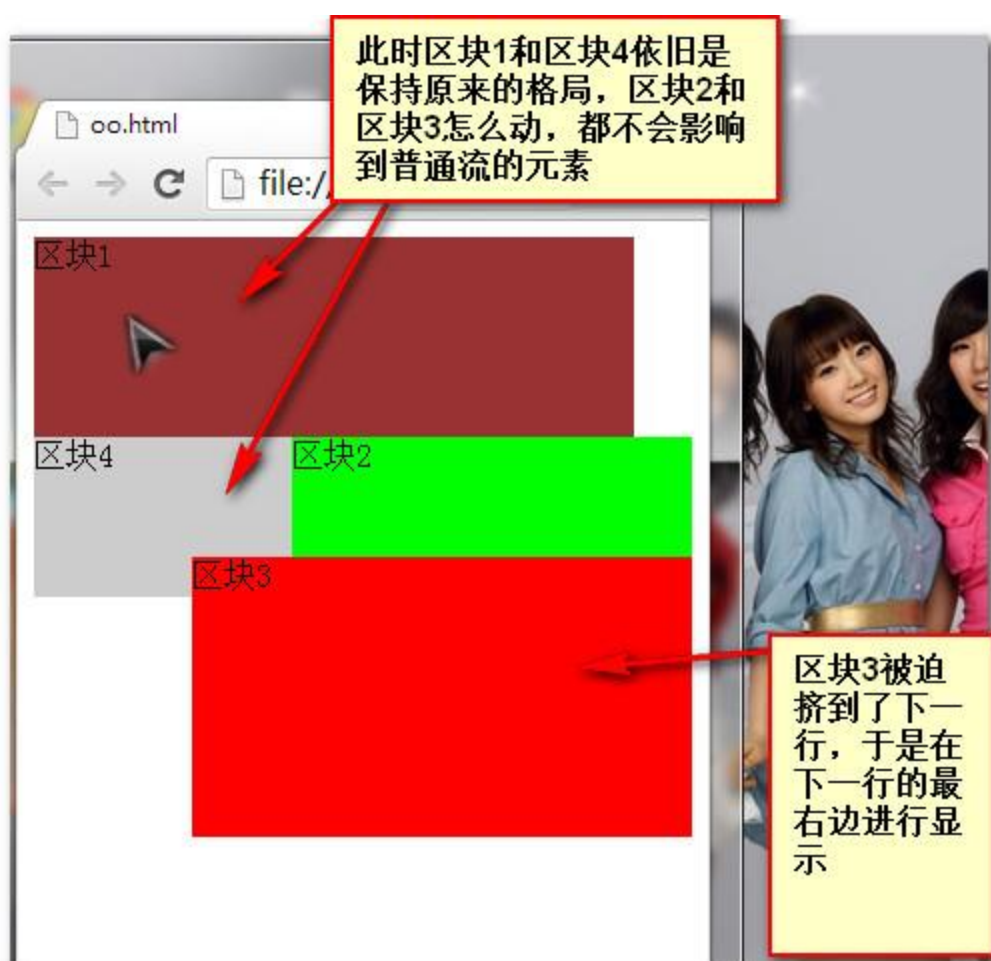


2. 我想通过上面的解释大家都清楚了吧，那就是区块 1 和区块 4 都是出于普通流中的，浏览器先放置它们两个，然后再从浮动的元素中先放置区块 2，然后再放置区块 3，由于区块 2 先浮动，区块 3 后浮动，所以，区块 3 肯定在区块 2 的左边啦。

3. 那么当空间不够的时候呢？看下面例子：

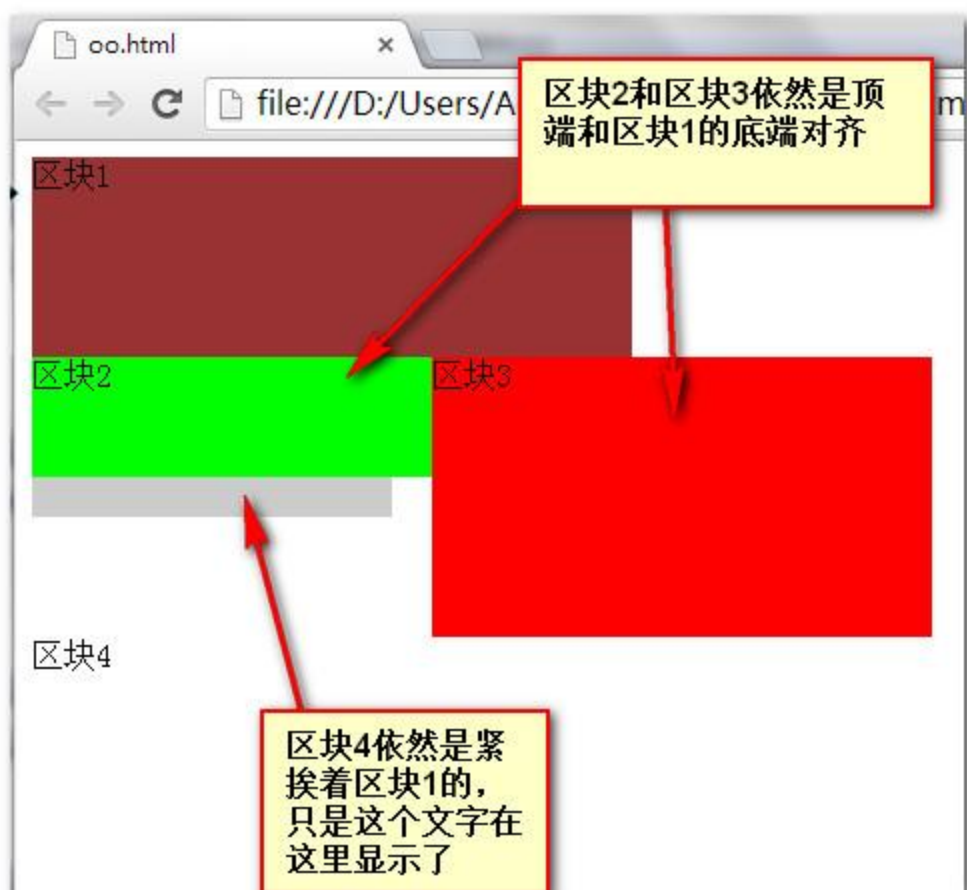


4. 那么我们继续向左挤压呢？请看下图：



5. 区块 3 被挤到了下一行，从最后边开始显示，但是这一切都和区块 1 还有区块 4 无关。

6. 右浮动清除了，那么左浮动也就同理了，下面我们让区块 2 和区块 3 都是左浮动，截图：



7. 当然如果我们一直挤压区块 2 和区块 3，它就会是这样：

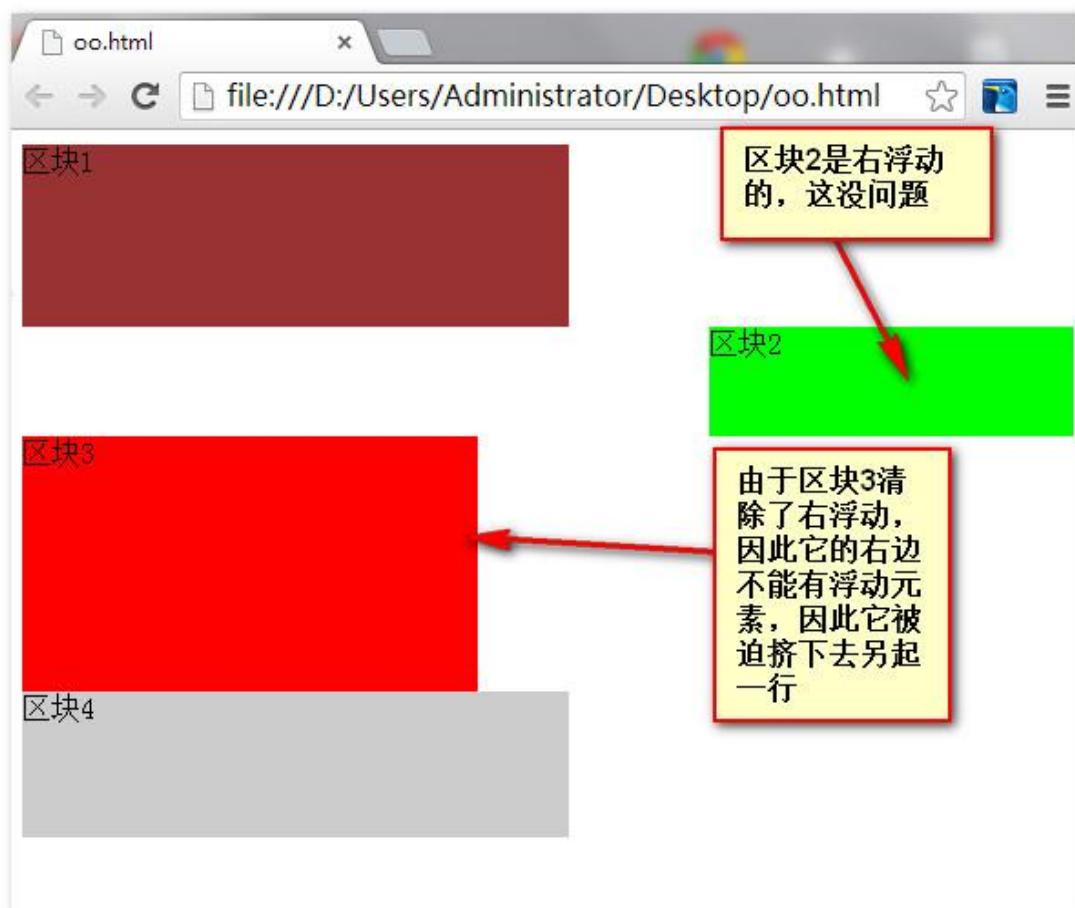


### \*\*\*\*\*清除浮动\*\*\*\*\*

1. 所谓清除浮动，只要把握一个核心概念就可以了：clear 属性可以取的值有四个，分别是 none, left, right, both。
2. 取值为 none 的时候没有效果，取值为 left 则表示该元素的左边不能有浮动元素，取值为 right 则表示该元素右边不能有浮动元素，取值为 both 则表示两边不能有浮动元素。
3. 我们看如下 css 代码：



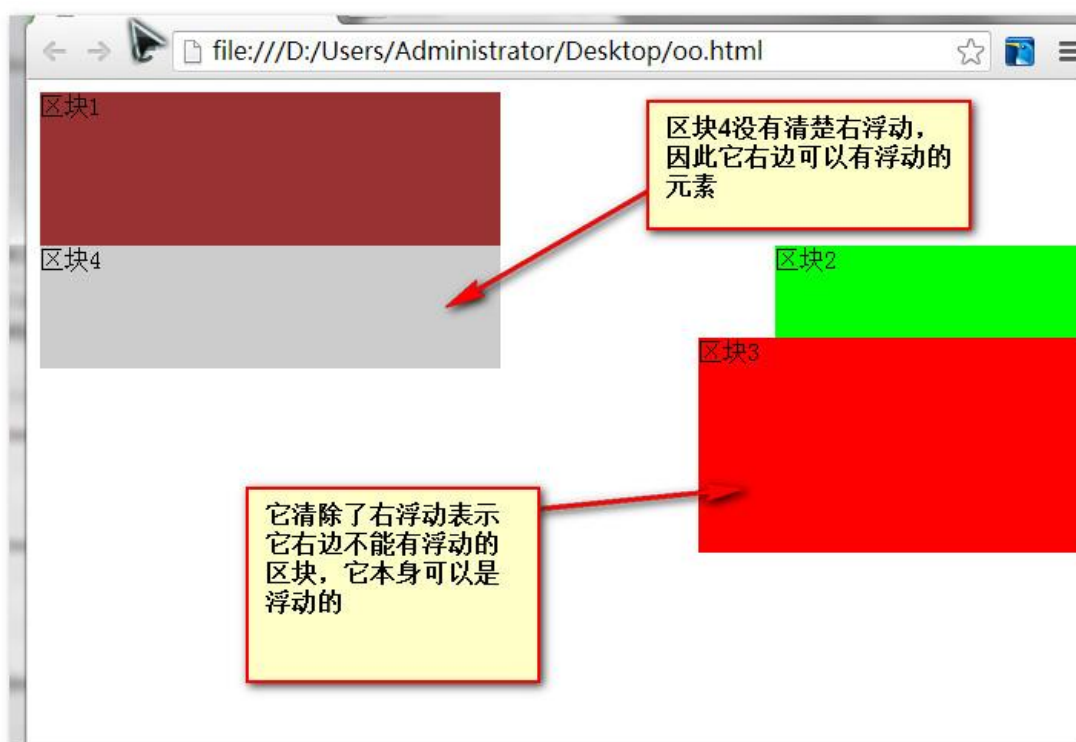
4. 我们会发现效果如下：



5. 那如果我们把区块 3 也右浮动呢，其实大家仔细想想，应该能猜到结果：

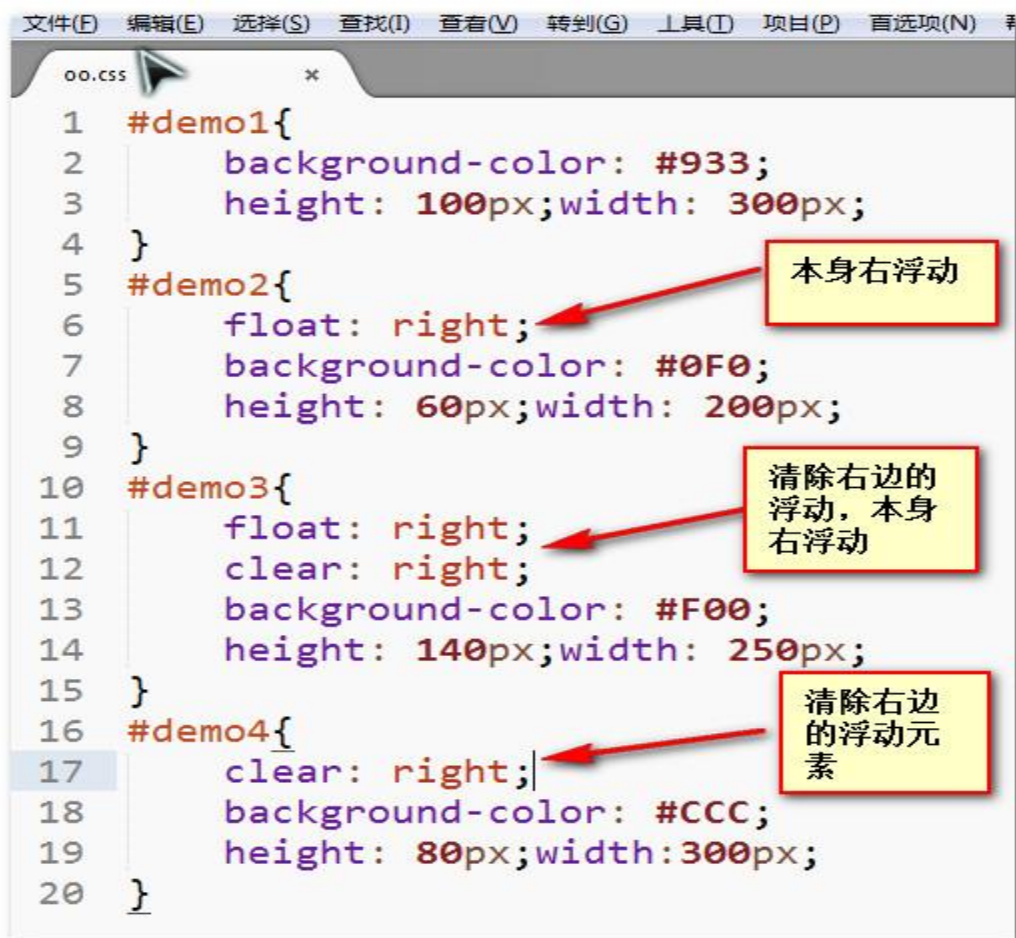


6. 下面我们让区块 3 清除右浮动，当然区块 3 本身依旧右浮动，效果如下：

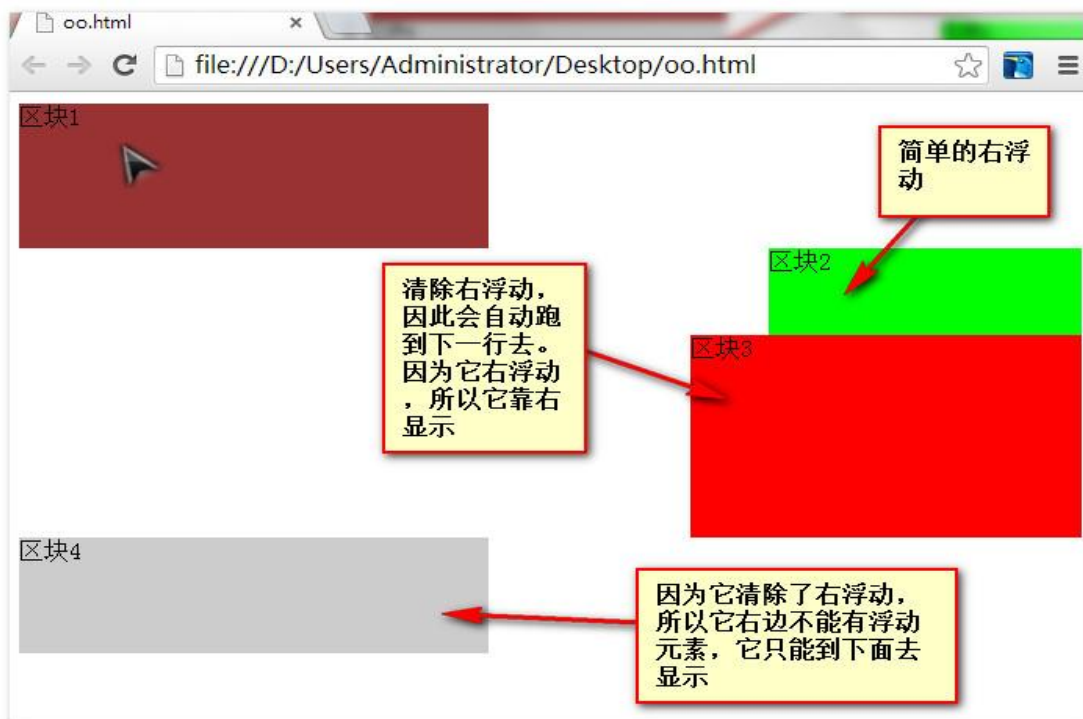


7. 如果我们让区块 4 清除右浮动呢？这里有点乱了，我给大家看一下现在的 css 样式把：





下面是效果图：





\*\*\*\*\*说明\*\*\*\*\*

1. 不知道大家晕了没有，反正我初学的时候是晕了，不过多研究几遍就好多了。
2. 我也是在 N 多博客和 N 多论坛的帖子里学习到的，另外，对于这种事情，实践是最好的老师，因为它能够让你亲身体会。
3. 希望我的表达是清晰的，如果我讲的大家不是很清楚的话，也可以去搜索下别人的教程看看，毕竟我感觉浮动还是很有意思的。

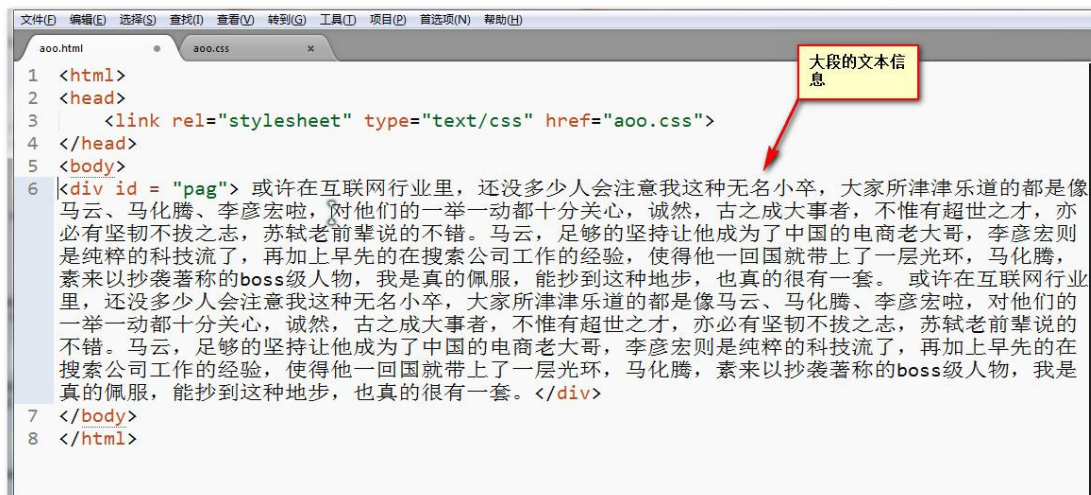
## 第八节：图片、背景和布局

### \*\*\*\*\*背景\*\*\*\*\*

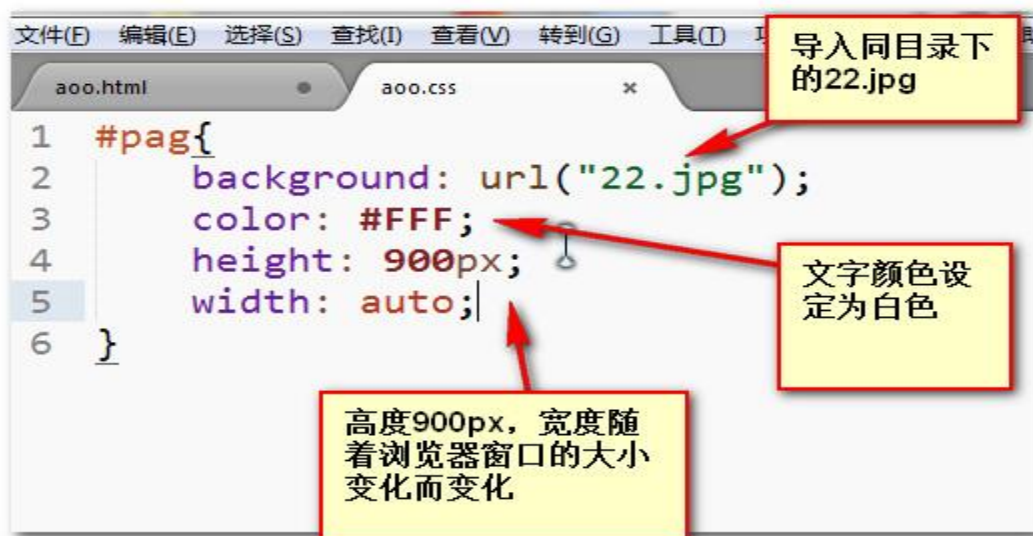
1. 这里我们讲一下怎么把图片用作背景，至于原因，我就不介绍了。
2. 我们引用图片做背景的话，需要在 background 属性中用 url 把图片地址括起来就可以了，然后可以指定重复的方式，这个我们下边说。

### \*\*\*\*\*代码示例\*\*\*\*\*

1. 首先是 html 代码的截图：



2. 然后我们在它导入的 css 文件中添加如下代码：

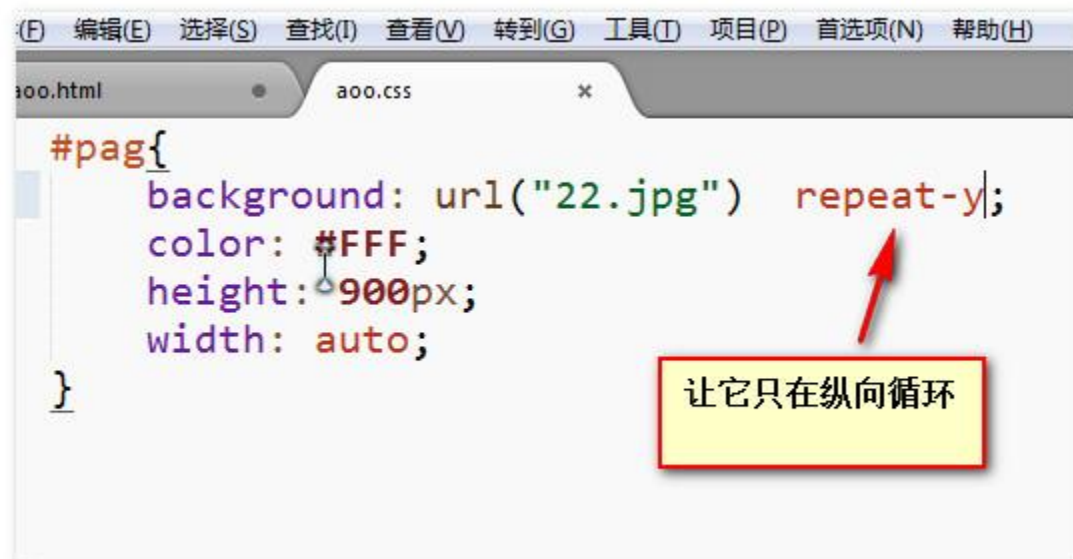


### 3. 最后效果如下：



4. 这里默认该图片在横向和纵向都是循环的，其实我们可以指定 `repeat-x` 为只在横向循环，`repeat-y` 只在纵向循环，`no-repeat` 只显示一次。

5. 比如下面就会只在纵向循环：



6. 最终效果图如下：



7. 关于背景图片我们就先介绍这么多吧，至于其他的，童鞋们可以查阅其他资料或者提意见给我：

[xinguimeng@163.com](mailto:xinguimeng@163.com)，在后续版本中更新。

### \*\*\*\*\*布局\*\*\*\*\*

1. 布局是一个很大的话题，足够写一本书出来，我一时间想要讲完也不大现实。

2. 我们先以 csdn 为例，分析一下它的布局：



3. 然后继续向下走，接下来就是：



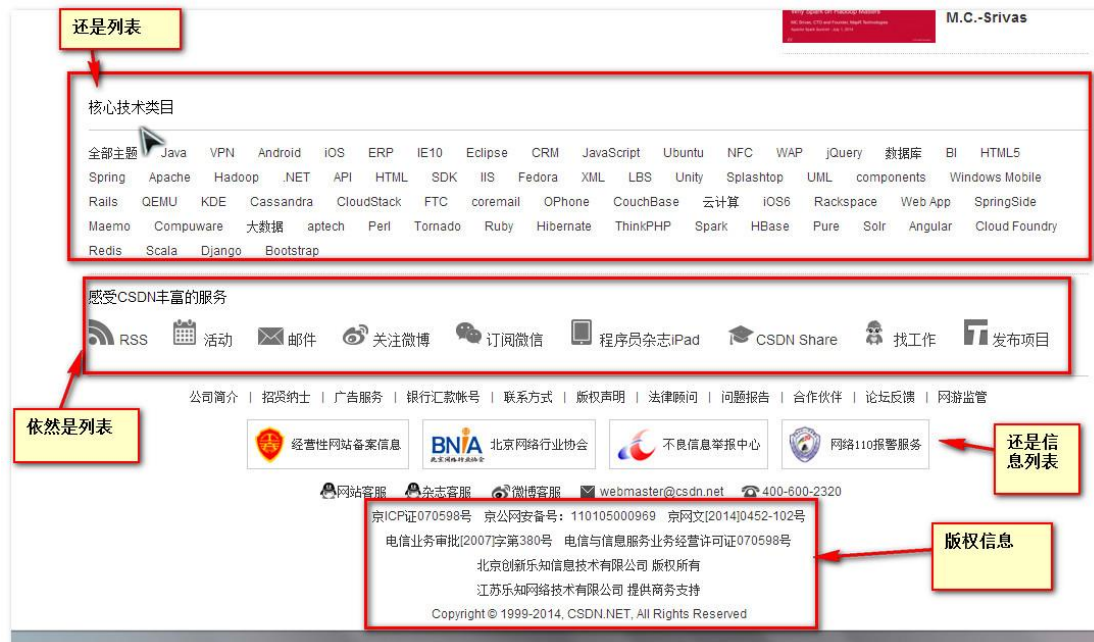


#### 4. 继续向下分析：



5. 其实下面也就是不断的用新闻列表，只是样式变化了些，有的把数字放到前面，有的没有数字，还有的 xxx 等等，也就是说，我们必须得把握好列表的制作。

6. 各种信息之后，来了个版权声明：

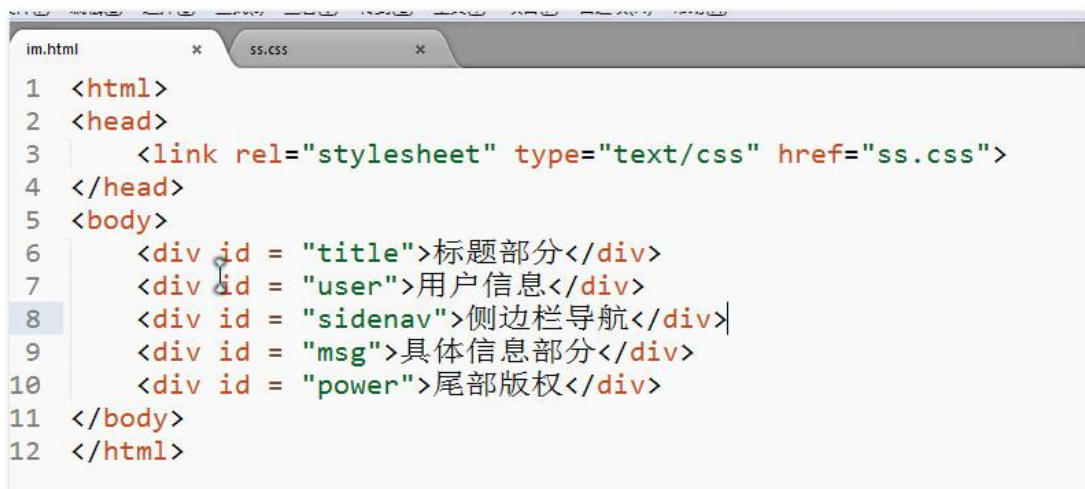


7. 其实，向 CSDN 这种网站，就是把整个版面先用 div 进行划分，然后每个 div 内用列表的形式去展示信息，值得注意的是，这里不可避免的会用到浮动来布局。

\*\*\*\*\*简易布局\*\*\*\*\*

1. 我们也给自己的网页规划一个布局，我这里为了简单，只是做了一个草图，如果读者做真正的网站的话，考虑的还应该更多。

2. 首先我们写 html 文件：



3. 然后我们写相应的 css 文件：



4. 然后我们网页的雏形就出来了：





5. 当然读者可能说它不够美观，比如设置一下 margin 和 padding 为 0 啦，什么的，这些事情，就需要读者的努力啦。

#### \*\*\*\*\*小结\*\*\*\*\*

1. 我们 CSS 的理论部分到这里就要说 bye bye 了，它简要的介绍了 CSS 的骨架。
2. 由于我考虑到我们太缺乏实战，于是，我特意在下面加了四节的实战，希望读者可以跟着做一做。

## 第三部分：CSS 经典应用

### 第一节：导航条

\*\*\*\*\*开始动手\*\*\*\*\*

1. 我们新建一个 my.html 文件，填写内容如下，这个 html 文件直到最后都不用动了，它就是这些内容：



```
1 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
2 <html xmlns="http://www.w3.org/1999/xhtml">
3 <head>
4   <link rel="stylesheet" type="text/css" href="xin.css">
5 </head>
6 <body>
7   <div class = "nav">
8     <ul>
9       <li><a href="#">辛星PHP</a></li>
10      <li><a href="#">辛星Python</a></li>
11      <li><a href="#">辛星CSS</a></li>
12      <li><a href="#">辛星HTML</a></li>
13      <li><a href="#">辛星jQuery</a></li>
14      <li><a href="#">辛星Java</a></li>
15    </ul></div>
16 </body>
17 </html>
18
19
```

此时效果图 如下：



2. 我们开始新建一个 xin.css 文件，然后开始进行调整这些导航条的显示格式，我们要做的第一步就是先把列表的小圆点给去掉，我们先写如下代码：

```
1. .nav ul {list-style-type: none;}
```

我们发现小圆点没了，但是，现在这些列表元素都是竖向排列的，我们想让它们横向排列。

此时效果图如下：



3. 我们通过设置 li 的浮动来让它们横向排列，我们在第二行添加代码，因此整个代码成为：

```
1. .nav ul {list-style-type: none;}  
2. .nav li {float: left;}
```

此时效果图如下：



4. 我们发现现在它们是到了一起了，但是是连在一起的，这是为什么呢，因为它们没有设置宽度，我们只要给他们设置了一定的宽度，它们就会分开啦，但是我们同时还可以设置一下背景色，因此该代码就成为了如下代码：

```
1. .nav ul {list-style-type: none;}
2. .nav li {float:left;width: 100px;background:#CCC;}
```

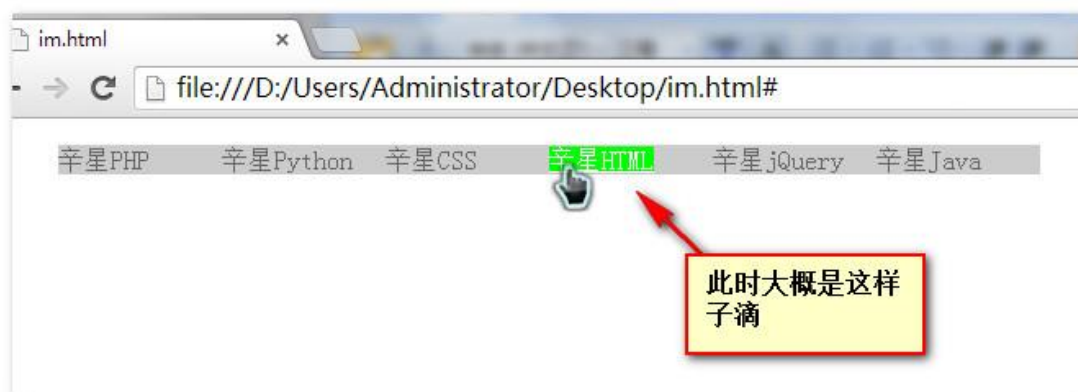
此时效果图如下：



5. 我们就开始设置鼠标的样式了，顺便我们把下划线给去掉，我们前面介绍过伪类的概念，如果读者不熟悉，也可以到前面翻翻我的教程，我们同时添加背景颜色什么的，因此它的代码成为下面这个样子：

```
1. .nav ul {list-style-type: none;}
2. .nav li {float:left;width: 100px;background:#CCC;}
3. .nav a:link {color:#666;background: #CCC;text-decoration: none;}
4. .nav a:visited {color: #666;text-decoration: none;}
5. .nav a:hover {color: #FFF;font-weight: bold;text-decoration: underline;background: #0F0;}
```

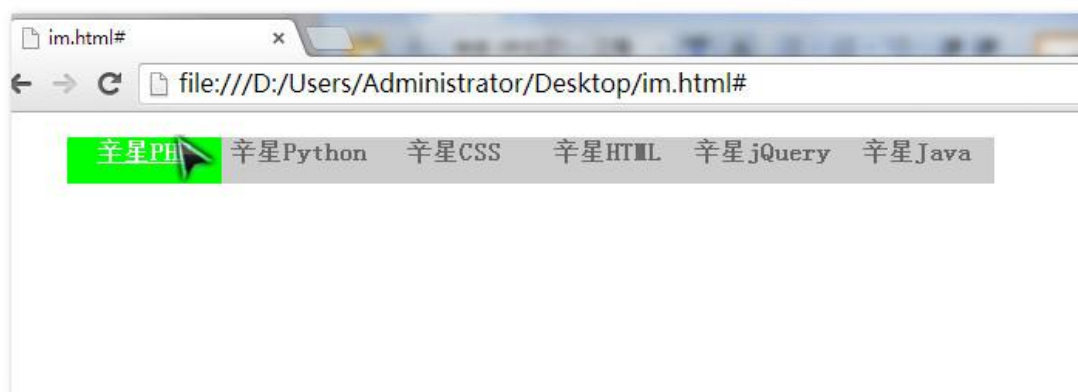
此时它的效果图：



6. 这里我们调整一下大小，即我们设置一下 .nav a 的属性值，比如文本居中啦，设置高度啦，字体加粗啦，代码如下：

```
1. .nav ul{list-style-type: none;}
2. .nav li{float:left;width: 100px;}
3. .nav a:link{color:#666;background: #CCC;text-decoration: none;}
4. .nav a:visited{color: #666;text-decoration: none;}
5. .nav a:hover{color: #FFF;font-weight: bold;text-decoration: underline;background: #0F0;}
6. .nav a{display: block;text-align: center;height: 30px;font-weight: bold;}
```

然后它的效果如下：



7. 还有一个问题就是我们的这个导航条是黏在一起的，我们需要分开菜看上去更加舒服一点，因此我们这里设置一下 `li` 的属性，让它的边距设置为 `3px`，然后在让字体的大小适合该导航条的大小，最后的代码如下：

```
1. .nav ul{list-style-type: none;}
2. .nav li{float:left;width: 100px;margin-left: 3px;line-height: 30px;}
3. .nav a:link{color:#666;background: #CCC;text-decoration: none;}
4. .nav a:visited{color: #666;text-decoration: none;}
5. .nav a:hover{color: #FFF;font-weight: bold;text-decoration: underline;background: #0F0;}
6. .nav a{display: block;text-align: center;height: 30px;font-weight: bold;}
```

8. 这就是它的效果啦：

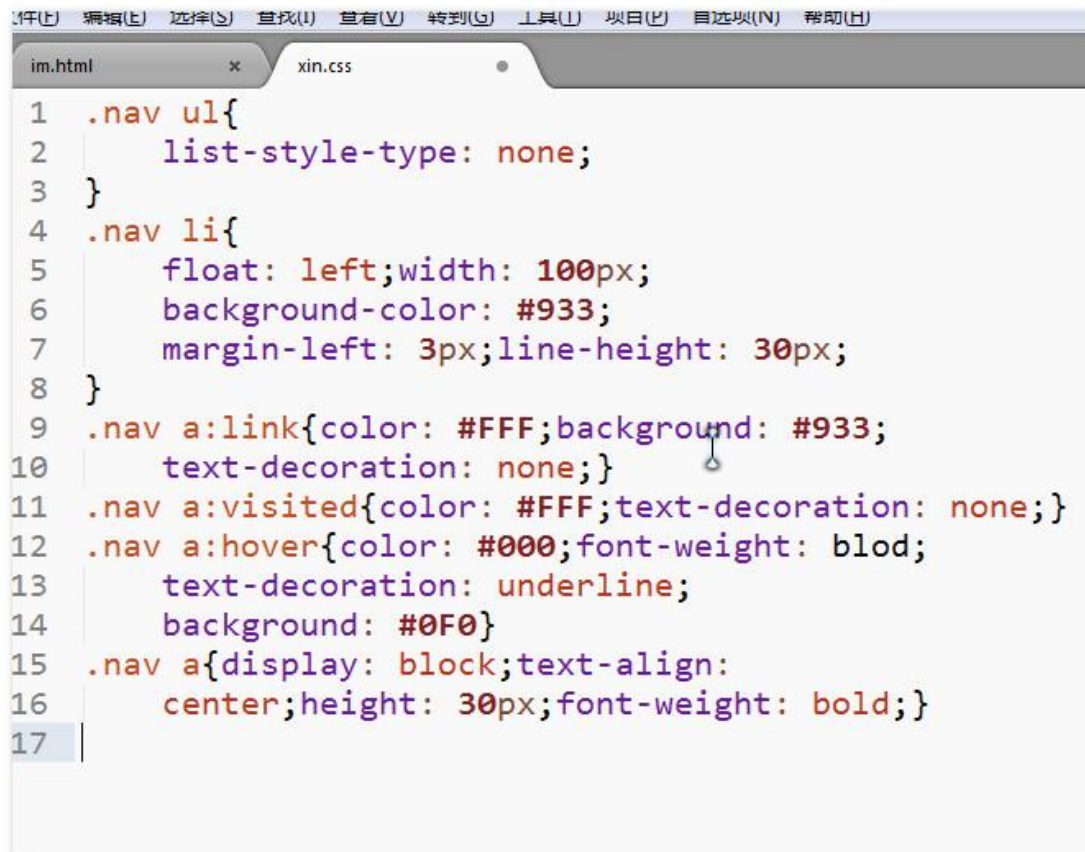


9. 是不是感觉不够漂亮，其实我现在也这么觉得，不过当时没感觉到，我们换一下背景色，在 a 标签中我们把背景色调一下，现在是这样子滴：



10. 它的 css 代码最终如下：



A screenshot of a web browser window with two tabs: 'im.html' and 'xin.css'. The 'xin.css' tab is active, displaying CSS code. The code is as follows:

```
1 .nav ul{
2     list-style-type: none;
3 }
4 .nav li{
5     float: left; width: 100px;
6     background-color: #933;
7     margin-left: 3px; line-height: 30px;
8 }
9 .nav a:link{color: #FFF; background: #933;
10     text-decoration: none;}
11 .nav a:visited{color: #FFF; text-decoration: none;}
12 .nav a:hover{color: #000; font-weight: bold;
13     text-decoration: underline;
14     background: #0F0}
15 .nav a{display: block; text-align:
16     center; height: 30px; font-weight: bold;}
17
```

11. 我承认，它的书写格式还可以优化一下，这个任务我就不去做啦。(☹o☹)

## 第二节：气泡

说到气泡，那么什么是气泡呢？先给那些刚入门的童鞋一个截图，来更好的认识一下什么是气泡把：



这就是一个简单的气泡啦，那么它主要用来干什么呢？他可以用来当我们点击某段文字的时候再显示出来，当然，这个可能需要用到 Javascript 的知识，我们先不提了，我们这里先给大家看一下如何去做一个 CSS 气泡把。

\*\*\*\*\*思路分析\*\*\*\*\*

1. 它的实现无非就是一个矩形的边框，然后下面是一个三角形，但是这个三角形是一个镂空的，也就是是一个空心三角形，并且它还是一个只有两条边的三角形，那么我们先来看一下如何去做出这个三角形，其实实现这个三角形还是蛮简单的，我先给出代码，然后给出解释：

```
1. <html>
2. <head>
3. <style type="text/css">
```

```
4.      #demo{width: 0px;height: 0px;border-  
width: 75px;border-style:solid;border-  
color:#F00 transparent transparent; }  
5.  
6.  
7. </style>  
8. </head>  
9. <body>  
10.    <div id = "demo"></div>  
11.  </body>  
12.  </html>
```

上面这段代码大家去运行的话，发现只有一个红色的倒立的三角形，效果图如下：



2. 那么它是如何实现的呢？首先我们创建了一个 div，然后设置其 id 为 demo，然后我们在 css 中把它的宽度和高度都设置为 0，把边框设置为 75 像素。

3. 这一点很关键，即其实我们是看不到它的内容的，我们所看到的只有它的边框，而它的四个边框的颜色都是红色的，但是就但是在，它的下边框和左右边框都是透明的。

4. 因此，我们就只能看到上边框了，也就是我们只能看到一个倒立的三角形，看不到整个的矩形区域了，就是这个三角形的实现原理，如果大家想要做出来向右的三角形，可以自己写一写试试奥。

\*\*\*\*\*动手写一个气泡\*\*\*\*\*

1. 我们先写一个 my.html 文件，内容如下：

```
1. <!DOCTYPE html>
2. <html lang="zh-cn">
3. <head>
4.     <title>辛星手写 CSS 气泡</title>
5.     <link rel="stylesheet" type="text/css"
      href="xing.css">
6. </head>
7. <body>
8. <div class="tag">
9.     <div class="arrow">
10.         <em></em><span></span>
11.     </div>
12.     辛星 CSS 手写气泡
13. </div>
14. </body>
15. </html>
```

此时它的效果是这样的：



2. 我们新建一个 xing.css 文件，开始我们的 css 编辑工作，我们要在 em 中实现这个三角形，然后用 span 去镂空它，因此我们的 tag 来实现文本显示，我们用下面的 arrow 来实现箭头，我们先写 tag 的样式：

```
1. .tag{ width:300px; height:100px;  
2.      border:5px solid #09F; position:  
   relative;  
3.      background-color:#FFF;  
4. }
```

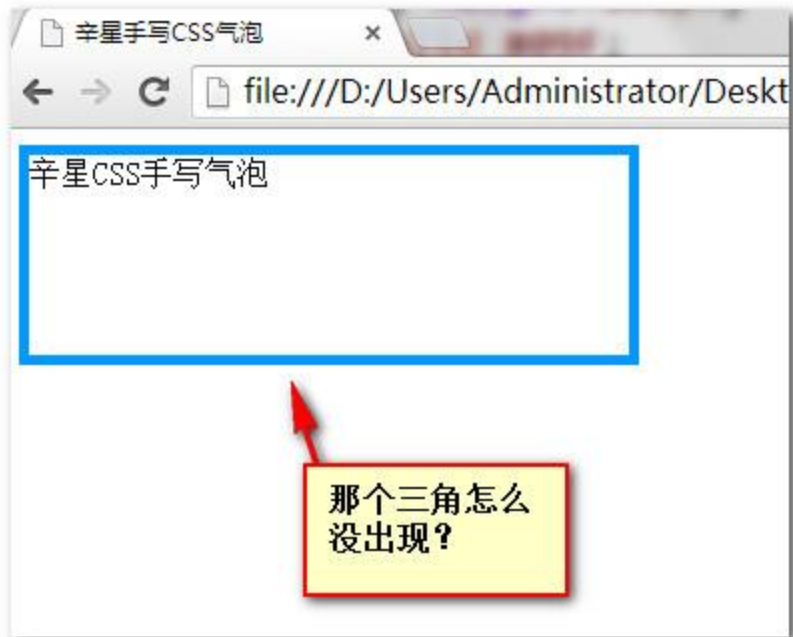
写完之后发现加了个边框：



3. 我们开始写下面的这个箭头，我们就需要用到 `em` 这个标签了，在写之前，有一个问题：那就是它的布局怎么办，这里我们采用绝对布局，关于绝对布局，如果大家不清楚，可以翻翻我前面的教程，里面有讲解。关于如何实现这个箭头，上面也有讲解，如果读者不清楚，可以跟着代码多敲敲，试验几遍，我们添加后的代码如下：

```
1. .tag{ width:300px; height:100px;
2.      border:5px solid #09F;
3.      background-color:#FFF;
4. }
5. .tag em{display:block; border-
6.      width:20px;
7.      position:absolute; bottom:-
8.      40px; left:100px;
9.      border-style:solid ;
10.     border-
11.     color:#09F transparent transparent;
12. }
```

此时，它的效果如下：



4. 可能有童鞋会感觉有问题，刷新界面之后这个下拉箭头没有实现，那又是什么原因呢？这就是我们在一个子元素中使用绝对布局的话，如果父元素中没有指定布局模式，那么子元素就会相对于 body 进行定位，在 body 的下面 40 个像素，当然什么都不会看到了，于是我们在 tag 中变化代码如下，即添加它的定位属性，变化后的代码如下：

```
1. .tag{ width:300px; height:100px;
2.      border:5px solid #09F; position:
   relative;
3.      background-color:#FFF;
4. }
5. .tag em{display:block; border-
   width:20px;
6.      position:absolute; bottom:-
   40px; left:100px;
7.      border-style:solid ;
```

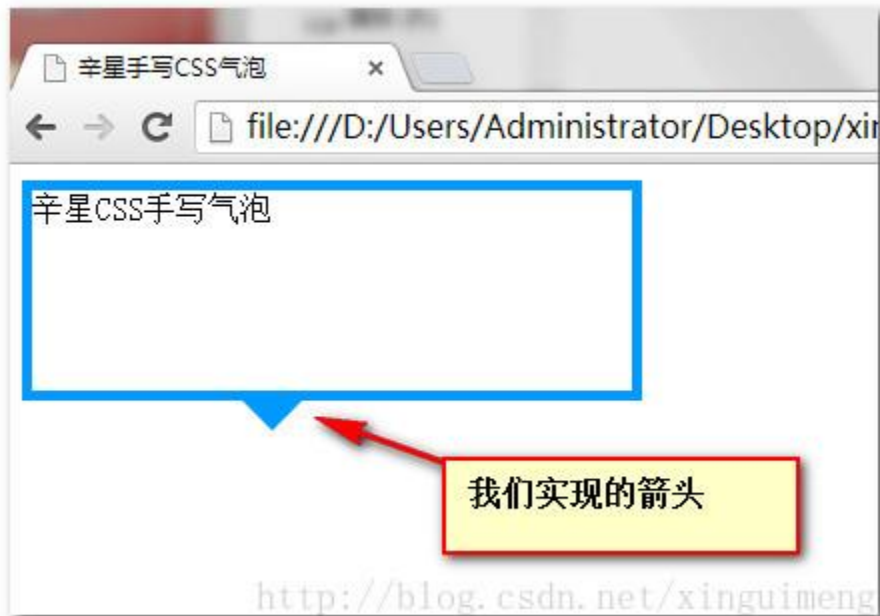


```

8.         border-
   color:#09F  transparent  transparent;
9. }

```

此时，让我们来看一下效果把：



其实，此时已经可以说是一个气泡了，如果读者要求不高的话，现在已经可以打完收工了。

5. 进一步的修饰，其实这个时候就很简单了，我们下面的 span 几乎就是照抄上面的 em 标签，我们只需要让该元素的向下的箭头是一个白色的三角形就 Ok 了，因此添加之后的代码如下：

```

1. .tag{  width:300px;  height:100px;
2.         border:5px  solid  #09F;  position:
   relative;
3.         background-color:#FFF;
4. }
5. .tag  em{display:block;  border-
   width:20px;
6.         position:absolute;  bottom:-
   40px;  left:100px;

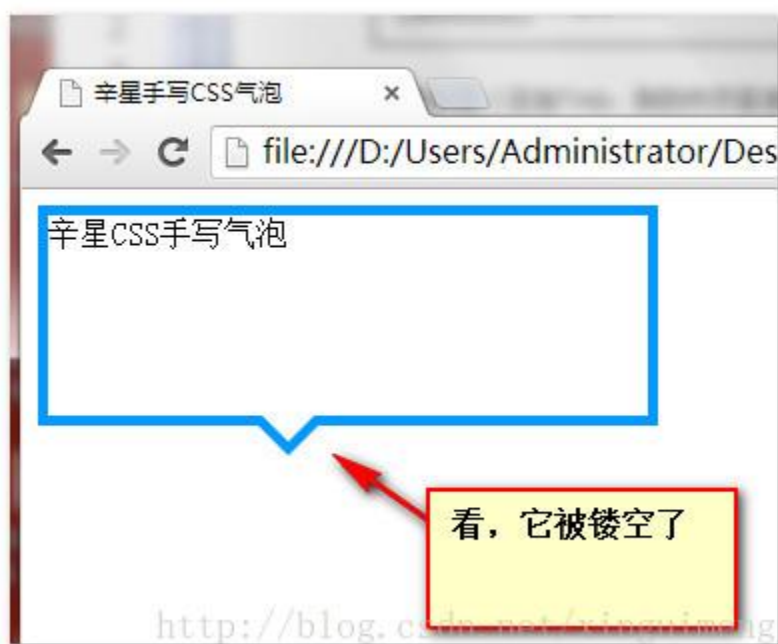
```

```

7.         border-style:solid  ;
8.         border-
color:#09F  transparent  transparent;
9.    }
10.    .tag  span{display:block; border-
width:20px;
11.        position:absolute;  bottom:-
33px;  left:100px;
12.        border-style:solid  ;
13.        border-
color:#FFF  transparent  transparent;
14.    }

```

好啦，我们来看一下效果：



好啦，这就完成了我们的用纯 CSS 手写气泡的效果。

\*\*\*\*\*说明\*\*\*\*\*

1. 该气泡的制作并不复杂，希望读者可以耐心看完。

2. 如果有什么问题，可以发邮件给我  
[xinguimeng@163.com](mailto:xinguimeng@163.com) 。

### 第三节：面包屑导航

#### \*\*\*\*\*简单介绍\*\*\*\*\*

1. 首先讲一下面包屑导航为什么叫做面包屑导航，它来自于一个童话故事，这个童话故事的名字也非常有特色，叫做“汉塞尔和格莱特”，有一天他们去森林里游玩，但是走着走着发现迷路了，大家都知道，森林里都是数，不管往哪边走，还是茫茫的一片树，于是他们在沿途走过的地方都撒下面包屑，他们利用这些面包屑帮助他们走出了森林，因此，面包屑导航的意思就是说我们去过哪些地方，以及我们如何回到我们之前的位置。

2. 有时候我们特别需要面包屑导航，比如我们在某论坛或者某网上商城闲逛的时候，很容易被茫茫的帖子和众多的商品搞的头昏绕张，想要回去的时候，就需要借助于面包屑了，因此，面包屑在我们掌握一个网站的结构和用户更好的使用我们的网站，比如我们可以通过面包屑从帖子回到版块，然后通过版块回到首页，这就是面包屑的经典使用，比如我们可以在某网上商城中从商品界面回到商品列表界面，然后回到商品分类界面，然后重新查看其它的分类。

3. 下面让我们看看我刚才做出来的这个面包屑导航把：



4. 当然，面包屑导航有很多种样式，这里我们只介绍这一种，而且它比较简单，我做它总共用了不到五分钟。

\*\*\*\*\*基础知识\*\*\*\*\*

1. 不过，在做它之前，先给不懂得如何做三角形的童鞋补习一下基础知识，让您能够快速的做出自己的三角形。
2. 我们新建一个 html 文件，书写内容如下：

```
1. <html>
2. <head>
3.     <style type="text/css">
4.         #demo{
5.             width: 0px;height: 0px;
6.             border-
7.             color: #0FF #00F #F00 #933;
8.             border-style: solid;
9.             border-width: 50px;
10.        }
11.     </style>
12. </head>
13. <div id = "demo"></div>
14. </html>
```

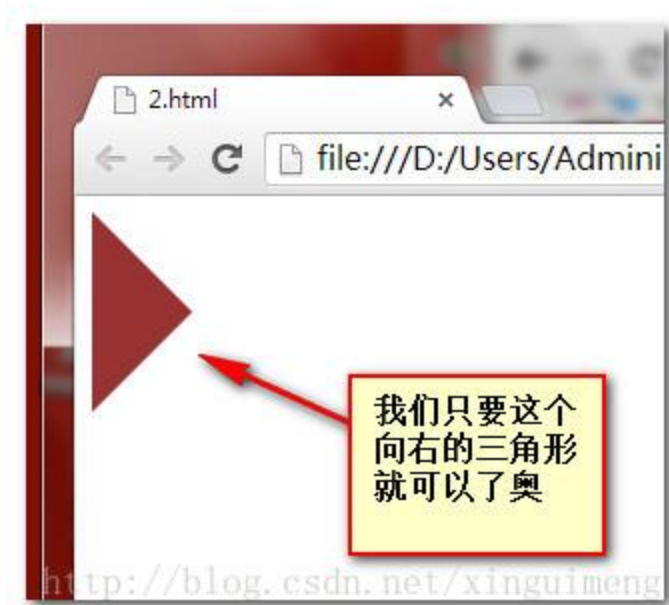
此时我们得到的界面是这样的：



2. 但是实际上我们不要那么多，我们只要其中的一个三角形就可以了，于是，我们把这个边界的颜色修改一下，于是代码如下：

```
1. <html>
2. <head>
3.     <style type="text/css">
4.         #demo{
5.             width: 0px;height: 0px;
6.             border-
7.             color: transparent transparent transparent
8.             #933;
9.             border-style: solid;
10.            border-width: 50px;
11.        }
12.    </style>
13. </head>
14. <div id = "demo"></div>
15. </html>
```

于是我们看到的界面如下：



3. 上面既然我们学会了制作三角形，那么下面我们就开始做我们的面包屑把，其实这个时候读者完全可以自己考虑试一下制作了，因为最重要的知识已经讲完了，当然直接看我的教程也不错奥。

#### \*\*\*\*\*制作面包屑\*\*\*\*\*

1. 就是先新建一个 html 文件，把主要内容写进去，这个文件我们直到最后都不用去修改它，读者如果没有把握，不要自行修改奥，我们在里面填写如下内容：

```

1. <html>
2. <head>
3.         <title>辛星手写 CSS 面包屑</title>
4.         <link rel="stylesheet" type="text/css"
           href="my.css">
5. </head>
6. <div class = "nav">
7.         <ul>
8.                 <li><a href="#">网站首页
                   </a><em></em><i></i></li>

```



```

9.          <li><a href="">手机专柜
    </a><em></em><i></i></li>
10.          <li><a href="">三星专栏
    </a></li>
11.          </ul>
12.      </div>
13.  </html>

```

2. 我们新建一个 my.css 文件，然后我们需要干什么呢，我们先把列表前面的小圆点去掉，然后设置一下 margin 和 padding，代码如下：

```

1. *{margin: 0px;padding: 0px;}
2. .nav ul{list-style-type: none;}

```

此时它的效果如下：



3. 但是现在我们发现我们的列表是竖向排列的，下面我们让他们向左浮动就可以横向排列了，于是我们添加之后代码如下：

```

1. *{margin: 0px;padding: 0px;}
2. .nav ul{list-style-type: none;}
3. .nav ul li{float: left;}

```

此时效果变成如下效果：



4. 我们发现一个问题就是所有的列表都挤到一起了，什么原因呢，因为我们没有设置这些列表的大小，下面我们设置其宽度，并且我们设置一下字体布局，让他们居中：

```
1. *{margin: 0px;padding: 0px;}
2. .nav ul{list-style-type: none;}
3. .nav ul li{float: left;
4.           width: 200px;text-align: center;
5. }
```

此时它的效果如下：



5. 我们就开始设置这个 nav 的背景了，因为我们要一个棕色的背景色，并且我们设置好它的宽度和高度以及字体大小，我们添加一些代码，它变成如下样式：

```

1. *{margin: 0px;padding: 0px;}
2. .nav{background-color: #933;
3.     width: 600px;height: 32px;
4.     line-height: 32px;
5. }
6. .nav ul{list-style-type: none;}
7. .nav ul li{float: left;
8.     width: 200px;text-align: center;
9. }

```

此时它的效果如下：



6. 其实我们做完上面的样式，会发现有点不舒服，为什么呢，因为我们的文字太晃眼了，于是，我们修改一下文字的字体和样式，其实我们主要做的事情就是去掉超链接的下划线，并且把文字的颜色改变成黑色，添加代码如下：

```

1. *{margin: 0px;padding: 0px;}
2. .nav{background-color: #933;
3.     width: 600px;height: 32px;
4.     line-height: 32px;
5. }
6. .nav ul{list-style-type: none;}
7. .nav ul li{float: left;

```

```

8.         width: 200px;text-align: center;
9.     }
10.     .nav ul li a{
11.         color:black;
12.         text-decoration:none;
13.     }

```

此时它的效果如下：



7. 我们就开始做我们的箭头了，直到这里，我们并没有用到我们的新知识，下面，我们先构造一个空白的三角形出来，它需要用到绝对定位，我们知道，如果一个子元素用到了绝对定位，那么父元素一定要使用定位才行，于是，我们在所有的 li 里面加入一条：position: relative，它不为别的，就是为了能够让子元素显示出来，代码如下：

```

1. *{margin: 0px;padding: 0px;}
2. .nav{background-color: #933;
3.     width: 600px;height: 32px;
4.     line-height: 32px;
5. }
6. .nav ul{list-style-type: none;}
7. .nav ul li{float: left;
8.     position: relative;
9.     width: 200px;text-align: center;

```

```

10.     }
11.     .nav ul li a{
12.         color:black;
13.         text-decoration:none;
14.     }

```

此时它的效果和上面是一样的。

8. 我们先加入一个空白的三角形，我们上面已经知道了怎么去做三角形了，于是这里我们就是做一个白色的三角形就可以了，但是，需要注意的是，由于我们设置了该三角形的边框是 24 像素，所以我们需要抬高  $24 - (32/2) = 8$  像素，向右偏移的像素量是我推测出来的，并不具有严格的数据意义，于是我们添加代码如下：

```

1. *{margin: 0px;padding: 0px;}
2. .nav{background-color: #933;
3.     width: 600px;height: 32px;
4.     line-height: 32px;
5. }
6. .nav ul{list-style-type: none;}
7. .nav ul li{float: left;
8.     position: relative;
9.     width: 200px;text-align: center;
10. }
11. .nav ul li a{
12.     color:black;
13.     text-decoration:none;
14. }
15. .nav ul li em{
16.     width: 0px;height: 0px;
17.     border-
18.     color: transparent transparent transparent
19.     #FFF;

```

```

18.         border-style: solid;
19.         border-width: 24px;
20.         position: absolute;
21.         right: -24px; top: -8px;
22.     }

```

此时，它的效果如下：



10. 我们就开始添加一个棕色的三角形，值得注意的是它需要向左移动 8 个像素，我们这里设置其大小为 16px，于是，代码如下；

```

1. margin: 0px; padding: 0px;
2. .nav {background-color: #933;
3.         width: 600px; height: 32px;
4.         line-height: 32px;
5.     }
6. .nav ul {list-style-type: none;}
7. .nav ul li {float: left;
8.             position: relative;
9.             width: 200px; text-align: center;

```

```
10.     }
11.     .nav ul li a{
12.         color:black;
13.         text-decoration:none;
14.     }
15.     .nav ul li em{
16.         width: 0px;height: 0px;
17.         border-
18.         color: transparent transparent transparent
19.         #FFF;
20.         border-style: solid;
21.         border-width: 24px;
22.         position: absolute;
23.         right: -24px;top: -8px;
24.     }
25.     .nav ul li i{
26.         width: 0px;height: 0px;
27.         border-
28.         color: transparent transparent transpa
29.         rent #933;
30.         border-width: 16px;
31.         border-style: solid;
32.         position: absolute;
33.         right:-8px;top: 0px;
34.     }
```

11. 现在，让我们看一下我们的成品把，它的截图如下：



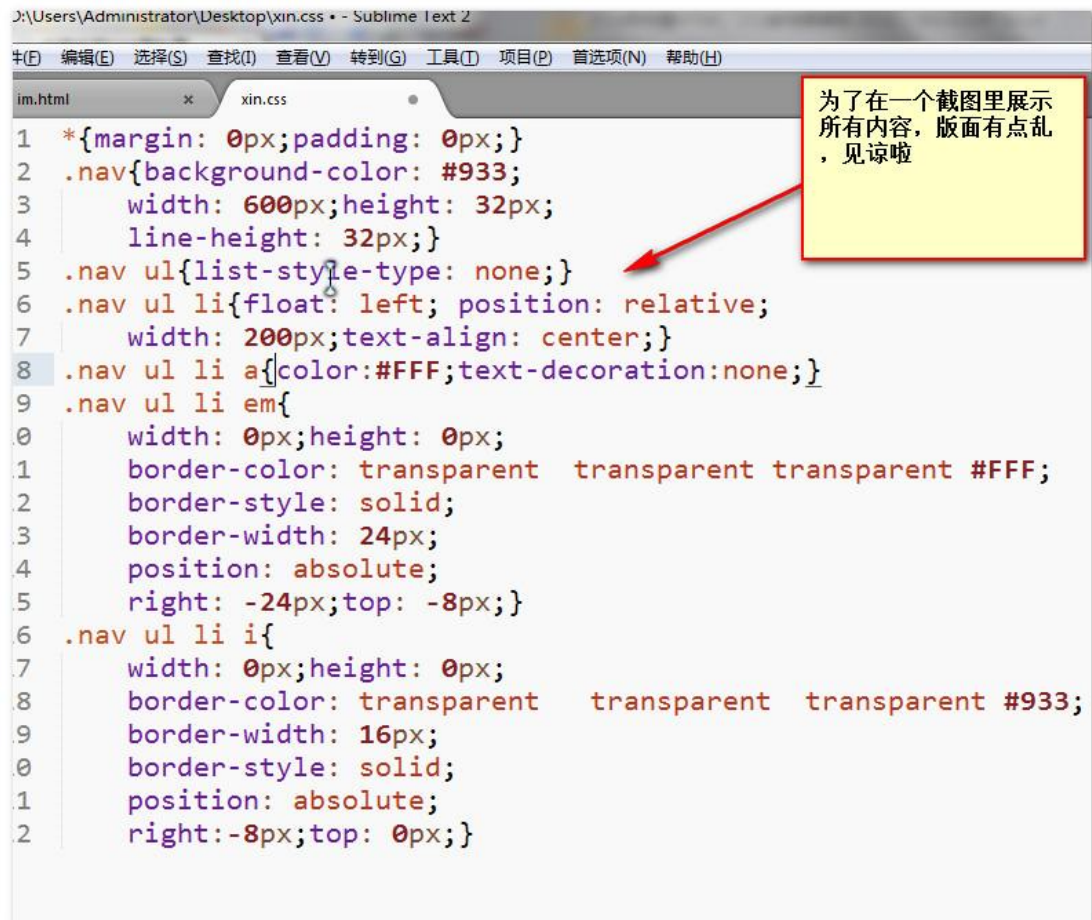


好啦，一个简单的面包屑就做好了，如果我们配合 Javascript 的知识，会做的更好看，这里我们就不介绍啦，喜欢吗？

12. 可能你会说：不好看，没错，我现在突然也这么感觉，我就简单修改下字体颜色，它就是这样了：



13. 下面是它的 css 代码截图：



14. 这篇文章是在我的博客上发布的:

<http://blog.csdn.net/xinguimeng/article/details/38069269> , 其中有一个博友对我的其中的定位问题产生了疑问, 我又写了一篇博客对其问题进行了 详细的回答, 如果读者也不是很明白三角形是如何定位的, 不妨读读, 地址给出:

<http://blog.csdn.net/xinguimeng/article/details/38121977> 。

15. 如果读者点击该超链接不起作用, 那么可以搜索“辛星 csdn”, 然后找到我的博客, 在 css 那一分类里, 会找到的, 这里也是打个小小的广告, 还希望读者不要嫌烦。

## 第四节：新闻列表

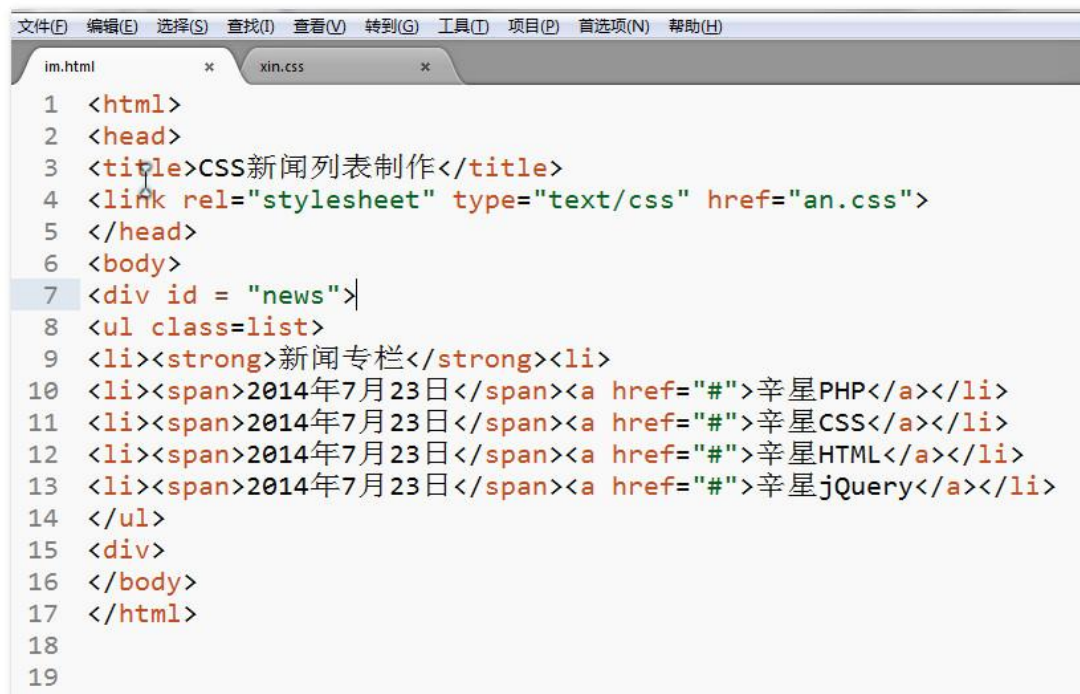
\*\*\*\*\*说明\*\*\*\*\*

1. 本来我想写一个商品列表的，但是考虑到一般商品都需要截图，并不能让读者有效的参与进来，但是使用新闻列表就不一样了。
2. 它只需要文字，也是特别的简单，因此，我决定使用新闻列表的样子。
3. 前面我们在讲布局的时候，通过 csdn 的网页也看到，我们需要大量的使用信息列表的形式，那么接下来，先给大家看一下我的这个列表：



\*\*\*\*\*开始制作\*\*\*\*\*

1. 由于它并不难，因此我希望读者和我一起动手，亲自做一做，首先我们在 html 中写入如下内容：



```
文件(F) 编辑(E) 选择(S) 查找(I) 查看(V) 转到(G) 工具(T) 项目(P) 首选项(N) 帮助(H)
im.html x xin.css
1 <html>
2 <head>
3 <title>CSS新闻列表制作</title>
4 <link rel="stylesheet" type="text/css" href="an.css">
5 </head>
6 <body>
7 <div id = "news">
8 <ul class=list>
9 <li><strong>新闻专栏</strong><li>
10 <li><span>2014年7月23日</span><a href="#">辛星PHP</a></li>
11 <li><span>2014年7月23日</span><a href="#">辛星CSS</a></li>
12 <li><span>2014年7月23日</span><a href="#">辛星HTML</a></li>
13 <li><span>2014年7月23日</span><a href="#">辛星jQuery</a></li>
14 </ul>
15 <div>
16 </body>
17 </html>
18
19
```

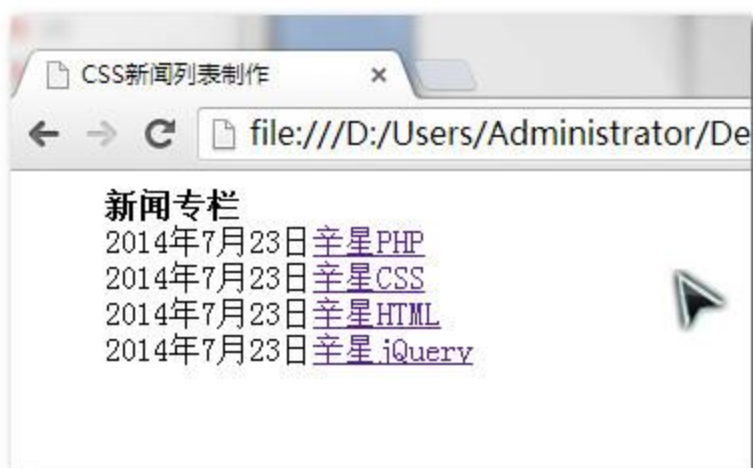
2. 然后我们在 css 文件中先写如下内容：



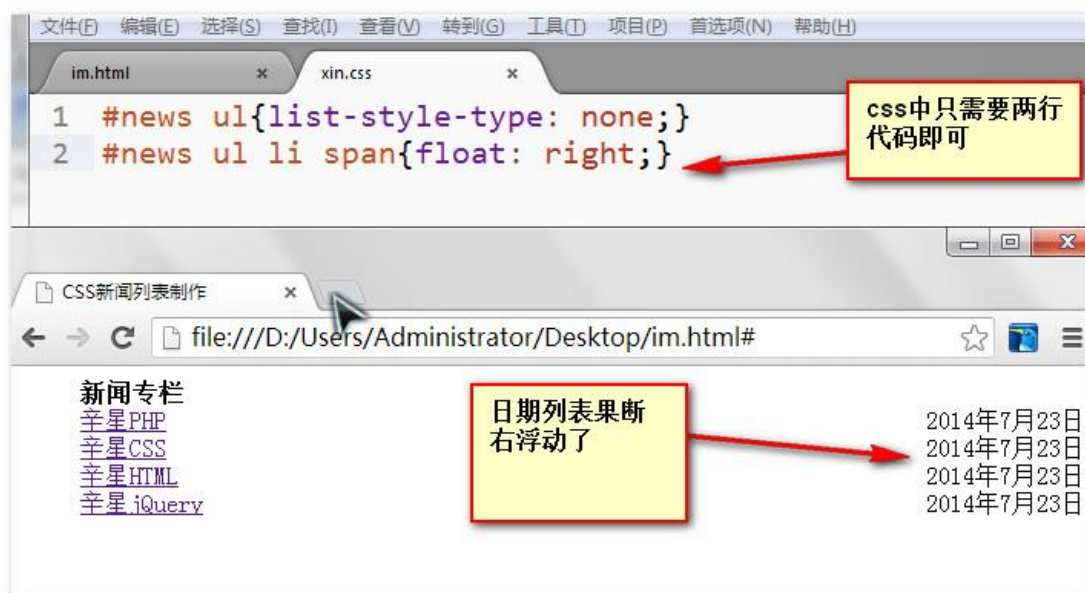
```
im.html x xin.css
1 #news ul{list-style-type: none;}
2
```

就是为了干掉列表前面的小圆点

此时的效果图如下：

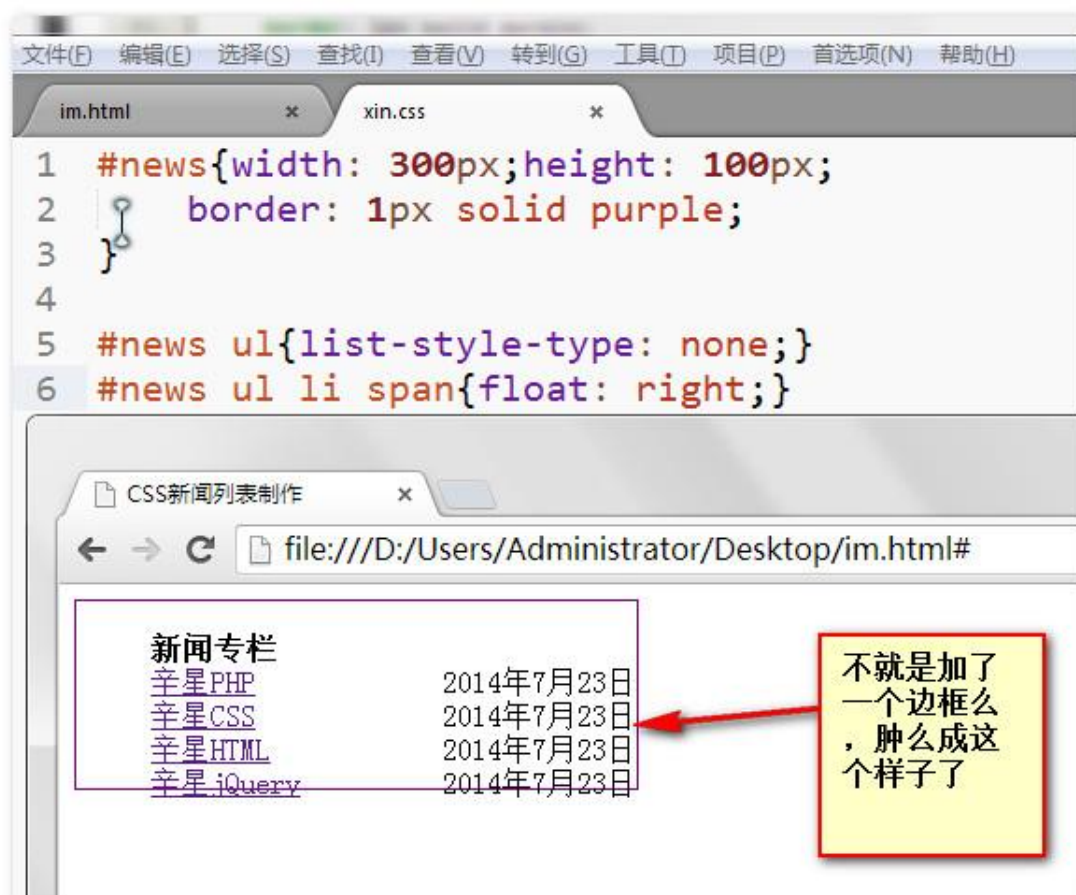


3. 我们干掉了小圆点之后，我们接下来需要让它的日期右浮动，于是 css 文件和效果图如下：

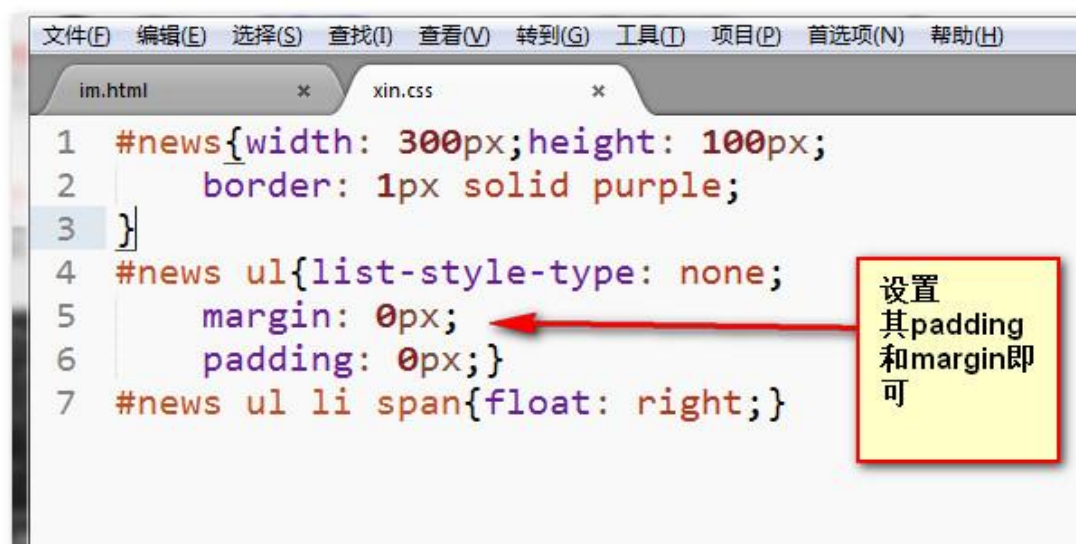


4. 接下来我们要做的就是设置它的范围，我们给 news 这个 div 加一个大小即可，截图如下：

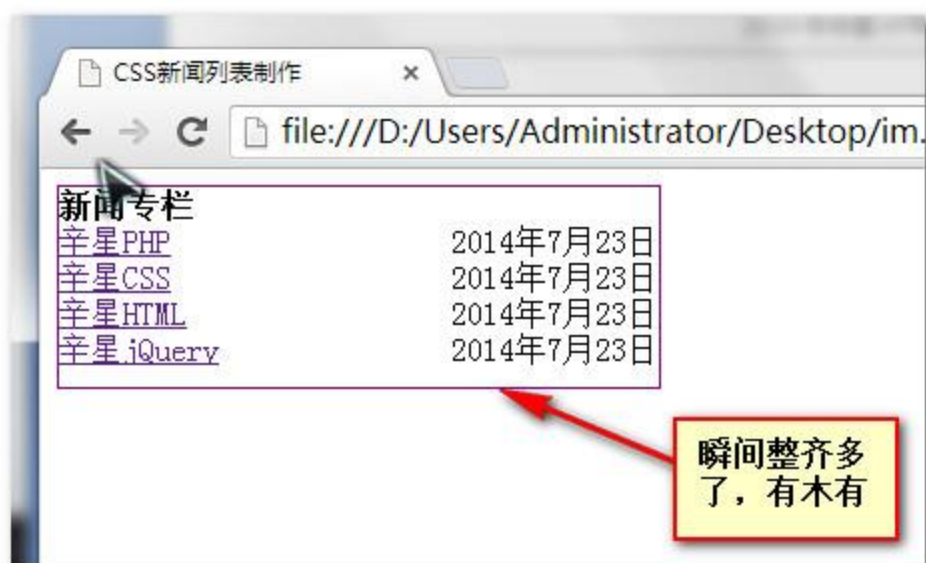




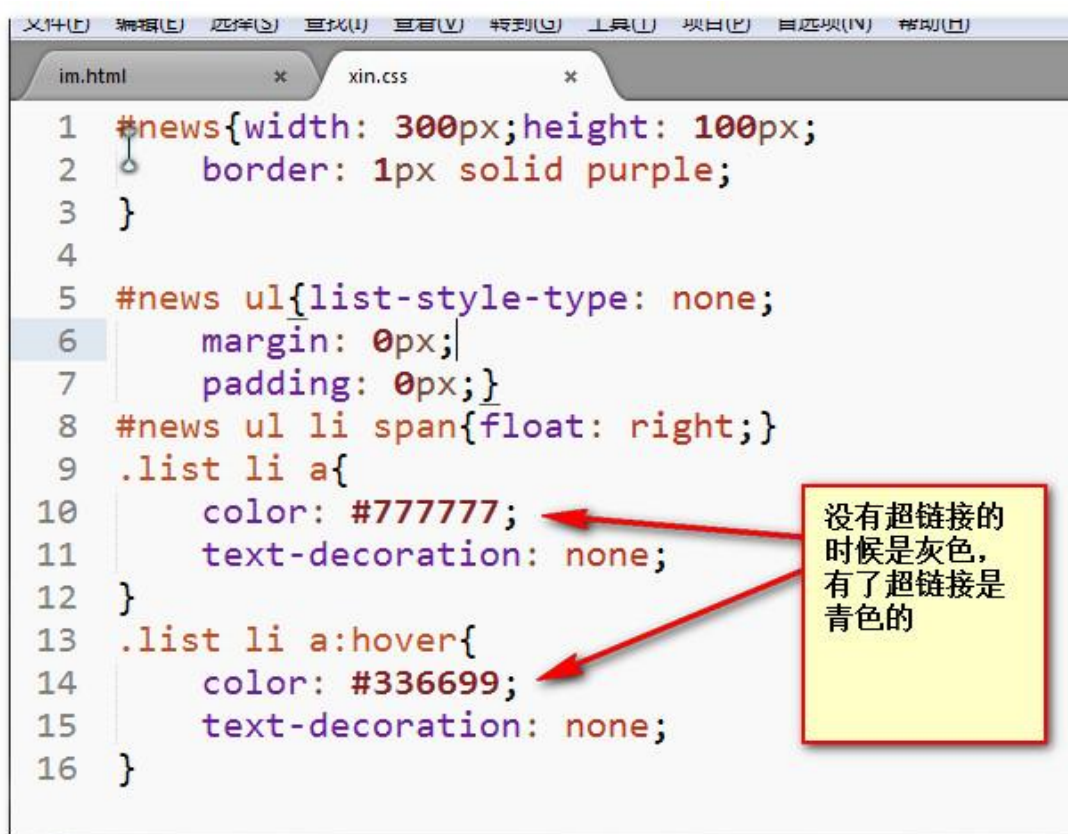
5. 其实很简单啦，只需要设置一下 ul 的距离就够了，css 截图如下：



然后我们得到的效果图如下：

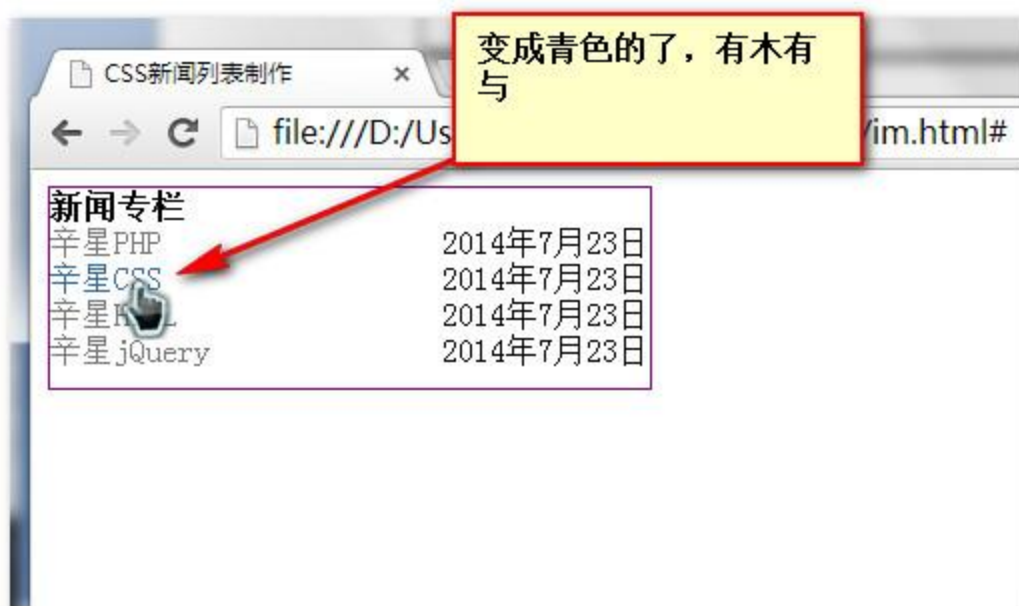


6. 接下来，我们设置下它的字体，css 截图如下：

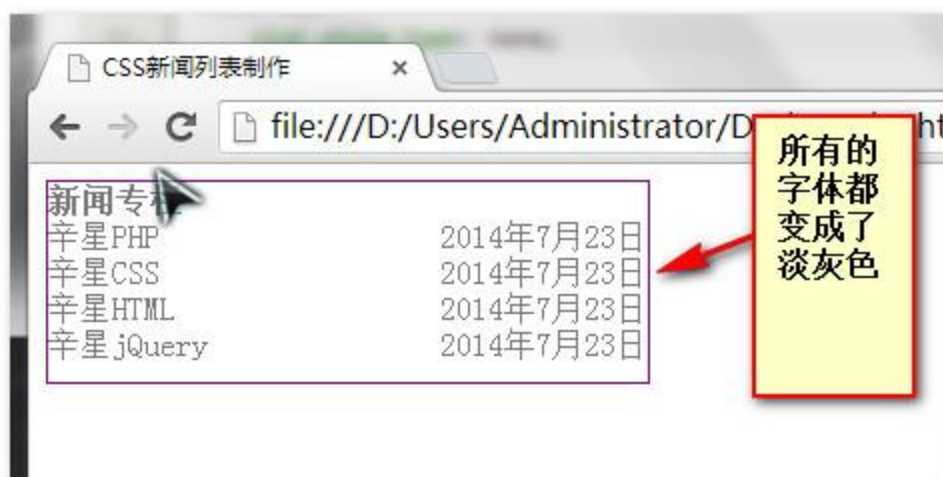


我们看一下效果把：





7. 到这里，我们基本上就做完了，最后再修改下其他的字体，然后美化一下格式，就完工了，最终的效果图：



经过整理后的 css 文件截图：



## 第五节：结语

\*\*\*\*\*暂时的停顿\*\*\*\*\*

1. “辛星”是一个致力于公益教学的人，是一个致力于传播知识的人。
2. 由于这是我第一次写 HTML、CSS 教程，难免会有一些疏漏和不足之处，希望大家可以原谅。
3. 有什么建议也可以给我指出，发送到我的邮箱：  
[xinguimeng@163.com](mailto:xinguimeng@163.com)。
4. 当然，如果想找我的更多的教程，可以搜索“辛星 html”来得到。
5. 如果想查看一下我的动态，那就可以去  
<http://blog.csdn.net/xinguimeng>，这个人就是我奥，在下面的 css 那一栏里，有本教程的网页版，期待您的关注。

\*\*\*\*\*永恒的前进\*\*\*\*\*

1. 当然了，一篇教程或许解决不了什么问题，但是至少它能夯实我们的一些基础。
2. 我们还需要继续前进，我的 PHP 系列教程写了两篇了，分别是“2014 年辛星 PHP 教程夏季版”和“2014 年辛星 ThinkPHP 教程夏季版”，有兴趣的可以搜索下。
3. 关于 Python，我则是写了五六本了，毕竟我最喜欢的一门语言。
4. Javascript 和 jQuery 的也正在写，bootstrap 的也正在计划动工，期待您的关注。