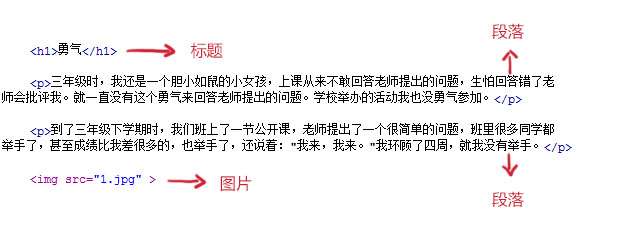
# 1 .html介绍

1.1 Html、css、js三者之间的关系：1. ****HTML是网页内容的载体****。内容就是网页制作者放在页面上想要让用户浏览的信息，可以包含文字、图片、视频等。

2. ****CSS样式是表现****。就像网页的外衣。比如，标题字体、颜色变化，或为标题加入背景图片、边框等。所有这些用来改变内容外观的东西称之为表现。

3. ****JavaScript是用来实现网页上的特效效果****。如：鼠标滑过弹出下拉菜单。或鼠标滑过表格的背景颜色改变。还有焦点新闻（新闻图片）的轮换。可以这么理解，有动画的，有交互的一般都是用JavaScript来实现的。

1.2标签

标签可以互相嵌套：但必须顺序一致，

1.3 文本基本结构

<html>

<head>...</head>

<body>...</body>

</html>

代码讲解：

1. <html></html>称为根标签，所有的网页标签都在<html></html>中。

2. <head> 标签用于定义文档的头部，它是所有头部元素的容器。头部元素有<title>、<script>、 <style>、<link>、 <meta>等标签，头部标签在下一小节中会有详细介绍。

3. 在<body>和</body>标签之间的内容是网页的主要内容，如<h1>、<p>、<a>、<img>等网页内容标签，在这里的标签中的内容会在浏览器中显示出来。

1.4 head标签 文档的头部描述了文档的各种属性和信息，包括文档的标题等。绝大多数文档头部包含的数据都不会真正作为内容显示给读者。

下面这些标签可用在 head 部分：

<head>

<title>...</title>

<meta>

<link>

<style>...</style>

<script>...</script>

</head>

<title>标签：在<title>和</title>标签之间的文字内容是网页的标题信息，它会出现在浏览器的标题栏中。网页的title标签用于告诉用户和搜索引擎这个网页的主要内容是什么，搜索引擎可以通过网页标题，迅速的判断出网页的主题。每个网页的内容都是不同的，每个网页都应该有一个独一无二的title。

1.5 ****<!--注释文字 -->****

# 2.认识标签

## ****2.1<body>标签，网页上显示的内容放在这里****

2.2 <p>标签，添加段落

<p>段落文本</p>

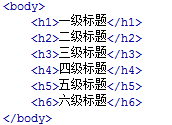
 注意一段文字一个<p>标签，如在一篇新闻文章中有3段文字，就要把这3个段落分别放到****3****个<p>标签中。

2.3 <hx>标签，为你的网页添加标题

标题标签一共有6个，h1、h2、h3、h4、h5、h6分别为一级标题、二级标题、三级标题、四级标题、五级标题、六级标题。并且依据重要性递减。<h1>是最高的等级。  
****语法：****  
<hx>标题文本</hx> (x为1-6)

****h1-h6标签的默认样式：****

标签代码：



## 2.4 **<strong>和<em>标签**

浏览器中<em> 默认用****斜体****表示，<strong> 用****粗体****表示。两个标签相比，目前国内前端程序员更喜欢使用<strong>表示强调

## 2.5 **<span>标签为文字设置单独样式/使用<br>标签分行显示文本**

<span>标签是没有语义的，它的作用就是为了设置单独的样式用的

## 2.6 **<q>标签，短文本引用 <blockquote>标签，长文本引用**

<q>引用文本</q>..注意要引用的文本不用加****双引号****，浏览器会对q标签自动添加双引号。

<blockquote>的作用也是引用别人的文本。但它是对****长文本****的引用，如在文章中引入大段某知名作家的文字，这时需要这个标签。

## 2.7 **空格 + <hr />标签，**

## **添加水平横线 &nbsp;**

## 2.8 **<address>标签，为网页加入地址信息**

<address>本文的作者：<a href="mailto:lilian@imooc.com">lilian</a></address>

## 2.9 **想加入一行代码吗？使用<code>标签**

## 使用<pre>标签为你的网页加入大段代码

## 2.10 **使用ul，ol,添加新闻信息列表**

ul-li是****没有前后顺序****的信息列表。

****语法：****

<ul>

<li>[信息](http://www.zuowen.com/e/20130805/51ff0eacd8d21.shtml" \o "精彩少年" \t "http://www.imooc.com/code/_blank)</li>

<li>[信息](http://www.zuowen.com/e/20130805/51ff0eacd8d21.shtml" \o "精彩少年" \t "http://www.imooc.com/code/_blank)</li>

......

</ul>

<ol>在网页中显示的默认样式一般为：每项<li>前都自带一个序号，序号默认从1开始

****语法：****

<ol>

<li>[信息](http://www.zuowen.com/e/20130805/51ff0eacd8d21.shtml" \o "精彩少年" \t "http://www.imooc.com/code/_blank)</li>

<li>[信息](http://www.zuowen.com/e/20130805/51ff0eacd8d21.shtml" \o "精彩少年" \t "http://www.imooc.com/code/_blank)</li>

......

</ol>

## 2.11 **认识div在排版中的作用**

在网页制作过程过中，可以把一些独立的逻辑部分划分出来，放在一个<div>标签中，这个<div>标签的作用就相当于一个容器。

****语法：****

<div>…</div>

****确定逻辑部分：****

什么是逻辑部分？它是页面上相互关联的一组元素。如网页中的独立的****栏目版块****，就是一个典型的逻辑部分。

## **给div命名，使逻辑更加清晰**

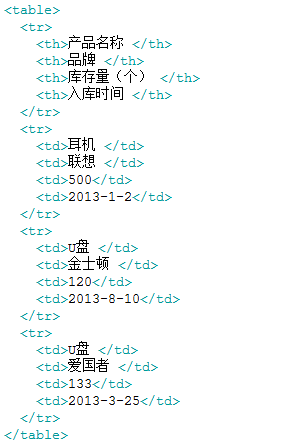
为了使逻辑更加清晰，我们可以为这一个独立的逻辑部分设置一个名称，用id属性来为<div>提供唯一的名称，这个就像我们每个人都有一个身份证号..****语法：****

<div  id="版块名称">…</div>

## 2.12 **table标签，认识网页上的表格**

[](http://img.mukewang.com/52d38f0b00017e4305060133.jpg)

想在网页上展示上述表格效果可以使用以下代码：

[](http://img.mukewang.com/52d38f83000160be02920445.jpg)

创建表格的四个元素：

table、tbody、tr、th、td

1、<table>…</table>：整个表格以<table>标记开始、</table>标记结束。

2、<tbody>…</tbody>：当表格内容非常多时，表格会下载一点显示一点，但如果加上<tbody>标签后，这个表格就要等表格内容全部下载完才会显示。如右侧代码编辑器中的代码。

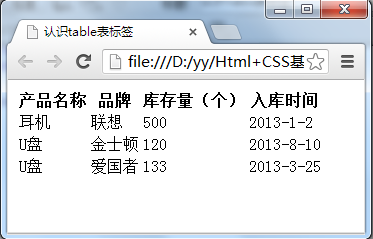
3、<tr>…</tr>：表格的一行，所以有几对tr 表格就有几行。

4、<td>…</td>：表格的一个单元格，一行中包含几对<td>...</td>，说明一行中就有几列。

5、<th>…</th>：表格的头部的一个单元格，****表格表头。****

6、表格中列的个数，取决于一行中数据单元格的个数。

上述代码在浏览器中显示的默认的样式为：

[](http://img.mukewang.com/52d3902b0001142703730239.jpg)

总结：

1、table表格在没有添加css样式之前，在浏览器中显示是没有表格线的

2、表头，也就是th标签中的文本默认为****粗体****并且****居中****显示

## 2.13 **用css样式，为表格加入边框**

Table 表格在没有添加 css 样式之前，是没有边框的。这样不便于我们后期****合并单元格****知识点的讲解，所以在这一节中我们为表格添加一些样式，为它添加边框。

在右侧代码编辑器中添加如下代码：

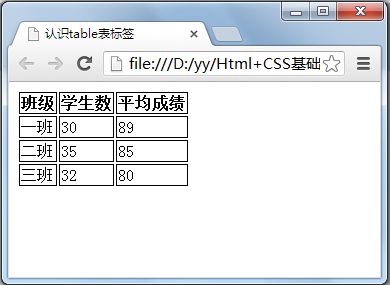
<style type="text/css">

table tr td,th{border:1px solid #000;}

</style>

上述代码是用 css 样式代码（****后面章节会详细讲解****），为th，td单元格添加粗细为一个像素的黑色边框。

结果窗口显示出结果样式：

[](http://img.mukewang.com/52d3993b00010d6203900285.jpg)

## 2.14 **caption标签，为表格添加标题和摘要**

摘要的内容是不会在浏览器中显示出来的。它的作用是增加表格的可读性(语义化)，使搜索引擎更好的读懂表格内容，还可以使屏幕阅读器更好的帮助特殊用户读取表格内容。

****语法：****<table summary="表格简介文本">

****标题****

用以描述表格内容，标题的显示位置：表格上方。

****语法：****

<table>

****<caption>标题文本</caption>****

<tr>

<td>…</td>

<td>…</td>

</tr>

</table>

## 2.15 **使用<a>标签，链接到另一个页面**

使用<a>标签可实现超链接，它在网页制作中可以说是无处不在，只要有链接的地方，就会有这个标签。

****语法：****

<a href="目标网址" title="鼠标滑过显示的文本">链接显示的文本</a>

## 2.16 **在新建浏览器窗口中打开链接**

<a>标签在默认情况下，链接的网页是在当前浏览器窗口中打开，有时我们需要在新的浏览器窗口中打开。

如下代码：

<a href="目标网址" ****target="\_blank"****>click here!</a>

## 2.17 **使用mailto在网页中链接Email地址**

<a>标签还有一个作用是可以链接Email地址，使用mailto能让访问者便捷向网站管理者发送电子邮件。我们还可以利用mailto做许多其它事情。下面一一进行讲解，请看详细图示：

[](http://img.mukewang.com/52da4f2a000150b714280550.jpg)

注意：如果mailto后面同时有多个参数的话，第一个参数必须以“?”开头，后面的参数每一个都以“&”分隔。

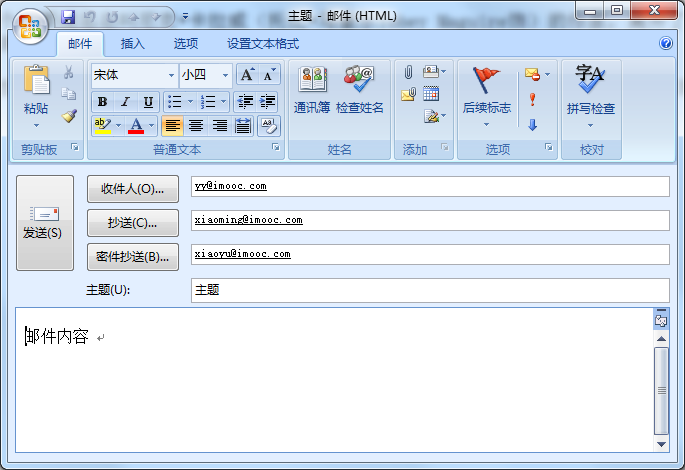
下面是一个完整的实例:



在浏览器中显示的结果：

发送

点击链接会打开电子邮件应用，并自动填写收件人等设置好的信息，如下图：

[](http://img.mukewang.com/52da5026000148a706850470.jpg)

## 2.18 **认识<img>标签，为网页插入图片**

在网页的制作中为使网页炫丽美观，肯定是缺少不了图片，可以使用<img>标签来插入图片。

****语法：****

<img src="图片地址" alt="下载失败时的替换文本" title = "提示文本">

****举例：****

<img src = "myimage.gif" alt = "My Image" title = "My Image" />

****讲解：****

****1、src：****标识图像的位置；

****2、alt：****指定图像的描述性文本，当图像不可见时（下载不成功时），可看到该属性指定的文本；

****3、title：****提供在图像可见时对图像的描述(鼠标滑过图片时显示的文本)；

****4、****图像可以是GIF，PNG，JPEG格式的图像文件。

# 3 使用表单标签，与用户交互

3.1 网站怎样与用户进行交互？答案是使用HTML表单(form)。表单是可以把浏览者输入的数据传送到服务器端，这样服务器端程序就可以处理表单传过来的数据。

语法：

<form method="传送方式" action="服务器文件">

****讲解：****

1.****<form> ：****<form>标签是成对出现的，以<form>开始，以</form>结束。

2.****action**** ****：****浏览者输入的数据被传送到的地方,比如一个PHP页面(save.php)。

3.****method**** ****：**** 数据传送的方式（get/post）。

<form ****method="post" action="save.php"****>

<label for="username">用户名:</label>

<input type="text" name="username" />

<label for="pass">密码:</label>

<input type="password" name="pass" />

</form>

****注意:****

1. ****所有表单控件（文本框、文本域、按钮、单选框、复选框等）都必须放在<form></form>标签之间（否则用户输入的信息可提交不到服务器上哦！）。****

3.2当用户要在表单中键入字母、数字等内容时，就会用到****文本输入框****。文本框也可以转化为****密码输入框****。

****语法****：

<form>

<input type="text/password" name="名称" value="文本" />

</form>

****1、type：****

   当type="****text****"时，输入框为****文本输入框;****

   当type="****password****"时, 输入框为****密码输入框。****

****2、name：****为文本框命名，以备后台程序ASP 、PHP使用。

****3、value：****为文本输入框设置默认值。(一般起到提示作用)

****举例：****

<form>

姓名：

<input type="text" name="myName">

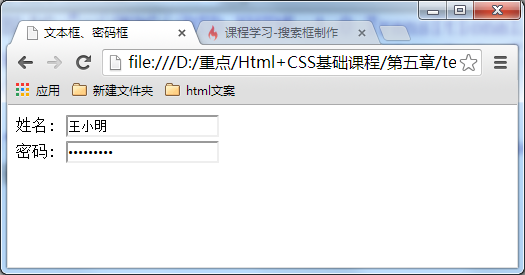
<br/>

密码：

<input type="password" name="pass">

</form>

在浏览器中显示的结果：

[](http://img.mukewang.com/52e4e9be000152ca05250275.jpg)

## 3.3 **文本域，支持多行文本输入**

当用户需要在表单中输入大段文字时，需要用到文本输入域。

****语法****：

<textarea ****rows="行数"**** ****cols="列数"****>文本</textarea>

1、<textarea>标签是成对出现的，以<textarea>开始，以</textarea>结束。

2、****cols ：****多行输入域的****列数****。

3、****rows ：****多行输入域的****行数****。

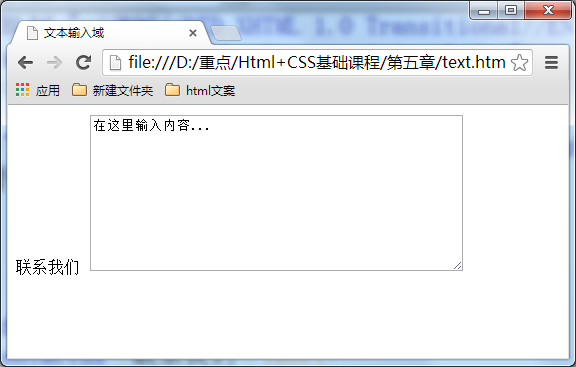
4、在<textarea></textarea>标签之间可以输入****默认值****。

****举例：****

<form method="post" action="save.php"> <label>联系我们</label> <textarea cols="50" rows="10" >在这里输入内容...</textarea></form>

注意：代码中的<label>标签在本章5-9中讲解。

在浏览器中显示结果：

[](http://img.mukewang.com/52e5b4040001f4af05760367.jpg)

注意这两个属性可用css样式的width和height来代替：col用width、row用height来代替

## 3.4 **使用单选框、复选框，让用户选择**

语法：

<input type="radio/checkbox" value="值" name="名称" checked="checked"/>

1、****type:****

   当 ****type="radio"****时，控件为****单选框****

   当 ****type="checkbox"**** 时，控件为****复选框****

2、****value：****提交数据到服务器的值（后台程序PHP使用）

3、****name：****为控件命名，以备后台程序 ASP、PHP 使用

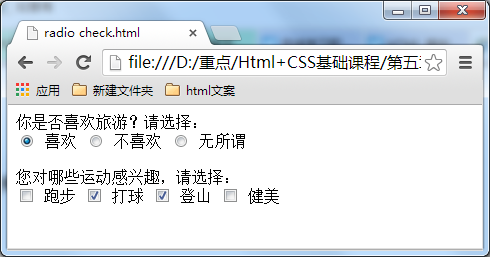
4、****checked：****当设置 checked="checked" 时，该选项被默认选中

如下面代码：

[](http://img.mukewang.com/52e5f7c60001a23f07360267.jpg)

注意：代码中的<label>标签在本章 5-9 中有讲解。

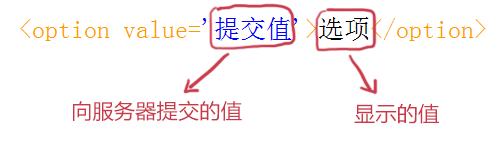
在浏览器中显示的结果：

[](http://img.mukewang.com/52e5f8010001159804900257.jpg)

注意****:同一组****的单选按钮，name 取值一定要一致，比如上面例子为同一个名称“radioLove”，这样同一组的单选按钮才可以起到单选的作用

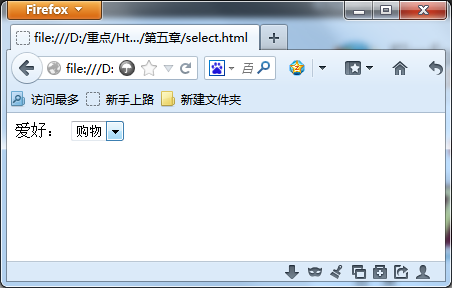
## 3.5 **使用下拉列表框，节省空间**

****1、value：****

[](http://img.mukewang.com/52e6037300015a9905030165.jpg)

2、****selected="selected"****：

设置****selected="selected"****属性，则该选项就被默认选中。

在浏览器中显示的结果：[](http://img.mukewang.com/52e605340001014804520288.jpg)

## **多选**

<select multiple="multiple">

## 3 **使用提交按钮，提交数据**

在表单中有两种按钮可以使用，分别为：提交按钮、重置。这一小节讲解提交按钮：当用户需要提交表单信息到服务器时，需要用到****提交按钮****。

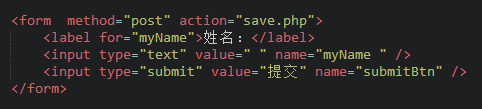
****语法****：

<input ****type="submit"**** value="提交">

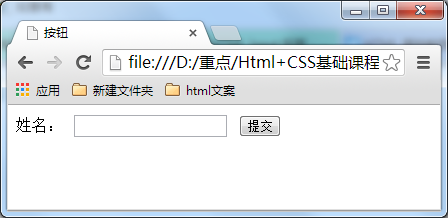
type****：只有当type值设置为submit时，按钮才有提交作用****

value****：****按钮上显示的文字

****举例：****

**[](http://img.mukewang.com/52e613350001461604820109.jpg)**

在浏览器中显示的结果：

[](http://img.mukewang.com/52e6126f0001496a04480218.jpg)

## **4 使用重置按钮，重置表单信息**

****语法****：

<input ****type="reset"**** value="重置">

type****：只有当type值设置为reset时，按钮才有重置作用****

value****：****按钮上显示的文字

## 5 **form表单中的label标签**

label标签不会向用户呈现任何特殊效果，它的作用是为鼠标用户改进了可用性。如果你在 label 标签内点击文本，就会触发此控件。就是说，当用户单击选中该label标签时，浏览器就会自动将焦点转到和标签相关的表单控件上（就自动选中和该label标签相关连的表单控件上）。

语法：

<label for="控件id名称">

注意：标签的 for 属性中的值应当与相关控件的 id 属性值一定要相同。

例子：

<form>

<label for="male">男</label>

<input type="radio" name="gender" id="male" />

<br />

<label for="female">女</label>

<input type="radio" name="gender" id="female" />

<label for="email">输入你的邮箱地址</label>

<input type="email" id="email" placeholder="Enter email">

</form>

### **CSS**

**1** CSS全称为“层叠样式表 (Cascading Style Sheets)”，它主要是用于定义HTML内容在浏览器内的显示样式，如文字大小、颜色、字体加粗等。如下：

p{

font-size:20px;/\*设置文字字号\*/

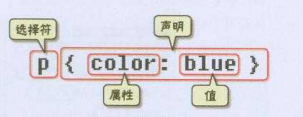
color:red;/\*设置文字颜色\*/

font-weight:bold;/\*设置字体加粗\*/

}

## 1.1 **CSS代码语法** /\*注释语句\*/

css 样式由****选择符****和****声明****组成，而****声明****又由****属性****和****值****组成，如下图所示：

[](http://img.mukewang.com/52fde5c30001b0fe03030117.jpg)

****选择符：****又称选择器，指明网页中要应用样式规则的元素，如本例中是网页中所有的段（p）的文字将变成蓝色，而其他的元素（如ol）不会受到影响。

****声明：****在英文大括号“｛｝”中的的就是声明，属性和值之间用英文冒号“：”分隔。当有多条声明时，中间可以英文分号“;”分隔，如下所示：

p{font-size:12px;color:red;}

注意：

1、最后一条声明可以没有分号，但是为了以后修改方便，一般也加上分号。

## 2 **内联式css样式，直接写在现有的HTML标签中**

内联式css样式表就是把css代码直接写在现有的HTML标签中，如下面代码：

<p style="color:red">这里文字是红色。</p>

## 2.1 **嵌入式css样式，写在当前的文件中**

<head>

<meta http-equiv="Content-Type" content="text/html; charset=utf-8">

<title>嵌入式css样式</title>

<style type="text/css">

span{

color:blue;

}

</style>

</head>

## 2.2 **外部式css样式，写在单独的一个文件中**

外部式css样式(也可称为外联式)就是把css代码写一个单独的外部文件中，这个css样式文件以“.css”为扩展名，在<head>内（不是在<style>标签内）使用<link>标签将css样式文件链接到HTML文件内，如下面代码：

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

注意：

1、css样式文件名称以有意义的英文字母命名，如 main.css。

2、rel="stylesheet" type="text/css" 是固定写法不可修改。

3、<link>标签位置一般写在<head>标签之内。

优先级：内联式 > 嵌入式 > 外部式

# 3 什么是选择器？

每一条css样式声明（定义）由两部分组成，形式如下：

选择器{

样式;

}

在{}之前的部分就是“选择器”，“选择器”指明了{}中的“样式”的作用对象，也就是“样式”作用于网页中的哪些元素。比如右侧代码编辑器中第7行代码中的“body”就是选择器。

## **3.1 标签选择器**

标签选择器其实就是html代码中的标签。如右侧代码编辑器中的<html>、<body>、<h1>、<p>、<img>。例如下面代码：

p{font-size:12px;line-height:1.6em;}

上面的css样式代码的作用：为p标签设置12px字号，行间距设置1.6em的样式。

## **3.2 类选择器 \*常用的**

语法：

****.****类选器名称{css样式代码;}

注意：

1、****英文圆点开头****

2、其中****类选器名称****可以任意起名（但不要起中文噢）

使用方法：

第一步：使用合适的标签把要修饰的内容标记起来，如下：

<span>胆小如鼠</span>

第二步：使用class="类选择器名称"为标签设置一个类，如下：

<span ****class="stress"****>胆小如鼠</span>

第三步：设置类选器css样式，如下：

.stress{color:red;}/\*****类前面要加入一个英文圆点****\*/

## **3.3 ID选择器**

在很多方面，ID选择器都类似于类选择符，但也有一些重要的区别：

1、为标签设置id="ID名称"，而不是class="类名称"。

2、ID选择符的前面是井号****（#）****号，而不是英文圆点****（.）****。

右侧代码编辑器中就是一个ID选择符的完整实例。

**#stress{**

**color:red;**

**}**

**<span id="stress">胆小如鼠</span>**

## **3.4 类和ID选择器的区别**

1. ****ID选择器只能在文档中使用一次****。与类选择器不同，在一个HTML文档中，ID选择器只能使用一次，而且仅一次。而**类选择器可以使用多次**。
2. ****可以使用类选择器词列表方法为一个元素同时设置多个样式。****我们可以为一个元素同时设多个样式，但只可以用类选择器的方法实现，ID选择器是不可以的（****不能使用 ID 词列表）。****

下面的代码是****正确****的(完整代码见右侧代码编辑器)

.stress{

    color:red;

}

.bigsize{

    font-size:25px;

}

<p>到了<span ****class="stress bigsize"****>三年级</span>下学期时，我们班上了一节公开课...</p>

上面代码的作用是为“三年级”三个文字设置文本颜色为红色并且字号为25px。

下面的代码是****不正确****的(完整代码见右侧代码编辑器)

#stressid{

    color:red;

}

#bigsizeid{

    font-size:25px;

}

<p>到了<span ****id="stressid bigsizeid"****>三年级</span>下学期时，我们班上了一节公开课...</p>

上面代码不可以实现为“三年级”三个文字设置文本颜色为红色并且字号为25px的作用。

## ****3.5** 子选择器**

还有一个比较有用的选择器****子选择器****，即大于符号(>),用于选择指定标签元素的****第一代子元素。****如右侧代码编辑器中的代码：

.food>li{border:1px solid red;}

这行代码会使class名为food下的子元素li（水果、蔬菜）加入红色实线边框

## 3.6 **包含(后代)选择器**

****包含选择器****，即加入空格,用于选择指定标签元素下的****后辈元素。****如右侧代码编辑器中的代码：

.first span{color:red;}

这行代码会使第一段文字内容中的“胆小如鼠”字体颜色变为红色。

请注意这个选择器与子选择器的区别，子选择器（child selector）仅是指它的直接后代，或者你可以理解为作用于子元素的第一代后代。而后代选择器是作用于所有子后代元素。后代选择器通过空格来进行选择，而子选择器是通过“>”进行选择。

总结：****>****作用于元素的第一代后代，****空格****作用于元素的所有后代。

## 3.7 **通用选择器**

通用选择器是功能最强大的选择器，它使用一个（\*）号指定，它的作用是匹配html中所有标签元素，如下使用下面代码使用html中任意标签元素字体颜色全部设置为红色：

## 3.8 **伪类选择符**

更有趣的是伪类选择符，为什么叫做伪类选择符，它允许给html不存在的标签（标签的某种状态）设置样式，比如说我们给html中一个标签元素的鼠标滑过的状态来设置字体颜色：

a:hover{color:red;}

## 3.9 **分组选择符**

当你想为html中多个标签元素设置同一个样式时，可以使用分组选择符（，），如下代码为右侧代码编辑器中的h1、span标签同时设置字体颜色为红色：

h1,span{color:red;}

它相当于下面两行代码：

h1{color:red;}

span{color:red;}

# 4 CSS的继承、层叠和特殊性

4.1 如某种颜色应用于p标签，这个颜色设置不仅应用p标签，还应用于p标签中的所有子元素文本，这里子元素为span标签。

p{color:red;}

<p>三年级时，我还是一个<span>胆小如鼠</span>的小女孩。</p>

可见右侧结果窗口中p中的文本与span中的文本都设置为了红色。但注意有一些css样式是不具有继承性的。如border:1px solid red;

p{border:1px solid red;}

<p>三年级时，我还是一个<span>胆小如鼠</span>的小女孩。</p>

在上面例子中它代码的作用只是给p标签设置了边框为1像素、红色、实心边框线，而对于子元素span是没用起到作用的。

## 4.2 **特殊性**

有的时候我们为同一个元素设置了不同的CSS样式代码，那么元素会启用哪一个CSS样式呢?我们来看一下面的代码：

p{color:red;}

.first{color:green;}

<p class="first">三年级时，我还是一个<span>胆小如鼠</span>的小女孩。</p>

p和.first都匹配到了p这个标签上，那么会显示哪种颜色呢？green是正确的颜色，那么为什么呢？是因为浏览器是根据权值来判断使用哪种css样式的，权值高的就使用哪种css样式。

下面是权值的规则：

****标签的权值为1，类选择符的权值为10，ID选择符的权值最高为100。****例如下面的代码：

p{color:red;} ****/\*权值为1\*/****

p span{color:green;} ****/\*权值为1+1=2\*/****

.warning{color:white;} ****/\*权值为10\*/****

p span.warning{color:purple;} ****/\*权值为1+1+10=12\*/****

#footer .note p{color:yellow;} ****/\*权值为100+10+1=111\*/****

****注意：还有一个权值比较特殊--继承也有权值但很低，有的文献提出它只有0.1，所以可以理解为继承的权值最低。****

## 4.3 **层叠**

****层叠****就是在html文件中对于同一个元素可以有多个css样式存在，当有相同权重的样式存在时，会根据这些css样式的前后顺序来决定，处于最后面的css样式会被应用

## 4.4 **重要性**

## 有些特殊的情况需要为某些样式设置具有最高权值，怎么办？这时候我们可以使用****!important****来解决。

如下代码：

p{color:red****!important****;}

p{color:green;}

<p class="first">三年级时，我还是一个<span>胆小如鼠</span>的小女孩。</p>

这时 p 段落中的文本会显示的red红色。

****注意：!important要写在分号的前面****

# 5 CSS格式化排版

## 5.1 **文字排版--字体**

body{font-family:"Microsoft Yahei";}

## **5.2字号、颜色**

可以使用下面代码设置网页中文字的字号为12像素，并把字体颜色设置为#666(灰色)：

body{font-size:12px;color:#666}

## **5.3粗体**

p span{font-weight:bold;}

## **5.4斜体**

p a{font-style:italic;}

## **5.5下划线**

p a{text-decoration:underline;}

<p>三年级时，我还是一个<a>胆小如鼠</a>的小女孩。</p>

## **5.6删除线**

<style type="text/css">

.oldPrice{text-decoration:line-through;}

</style>

</head>

<body>

<p>原价：<span class="oldPrice">300</span>元 现价：230 元</p>

## **5.7缩进**

中文文字中的段前习惯空两个文字的空白，这个特殊的样式可以用下面代码来实现：

p{text-indent:2em;}

<p>1922年的春天，一个想要成名名叫尼克卡拉威（托比?马奎尔Tobey Maguire 饰）的作家，离开了美国中西部，来到了纽约。那是一个道德感渐失，爵士乐流行，走私为王，股票飞涨的时代。为了追寻他的美国梦，他搬入纽约附近一海湾居住。</p>

注意：2em的意思就是文字的2倍大小。

## **5.8行间距（行高）**

p{line-height:2em;}

## **5.9中文字间距、字母间距**

****中文字间隔、字母间隔设置：****

如果想在网页排版中设置****文字间隔****或者****字母间隔****就可以使用    ****letter-spacing****来实现，如下面代码：

h1{

letter-spacing:50px;

}

...

<h1>了不起的盖茨比</h1>

## **5.10对齐**

text-align:center; 居中

text-align:left; 居左

text-align:right; 居右

# 6 CSS盒子模型

6.1 元素分类

在CSS中，html中的标签元素大体被分为三种不同的类型：****块状元素****、****内联元素(又叫行内元素)****和****内联块状元素****。

****常用的块状元素有：****

<div>、<p>、<h1>...<h6>、<ol>、<ul>、<dl>、<table>、<address>、<blockquote> 、<form>

****常用的内联元素有：****

<a>、<span>、<br>、<i>、<em>、<strong>、<label>、<q>、<var>、<cite>、<code>

****常用的内联块状元素有：****

<img>、<input>

 6.2 块级元素

什么是块级元素？在html中<div>、 <p>、<h1>、<form>、<ul> 和 <li>就是块级元素。设置display:block就是将元素显示为块级元素。如下代码就是将****内联元素a****转换为****块状元素****，从而使a元素具有****块状元素****特点。

a{display:block;}

****块级元素特点：****

1、每个块级元素都从新的一行开始，并且其后的元素也另起一行。（真霸道，一个块级元素独占一行）

2、元素的高度、宽度、行高以及顶和底边距都可设置。

3、元素宽度在不设置的情况下，是它本身父容器的100%（和父元素的宽度一致），除非设定一个宽度。

## 6.3 **内联元素**

在html中，<span>、<a>、<label>、 <strong> 和<em>就是典型的****内联元素****（****行内元素****）（inline）元素。当然****块状元素****也可以通过代码display:inline将元素设置为****内联元素****。如下代码就是将****块状元素div****转换为****内联元素****，从而使 div 元素具有****内联元素****特点。

div{

display:inline;

}

<div>我要变成内联元素</div>

## 6.4 **内联块状元素**

****内联块状元素（****inline-block****）****就是同时具备内联元素、块状元素的特点，代码display:inline-block就是将元素设置为内联块状元素。(css2.1新增)，<img>、<input>标签就是这种内联块状标签。

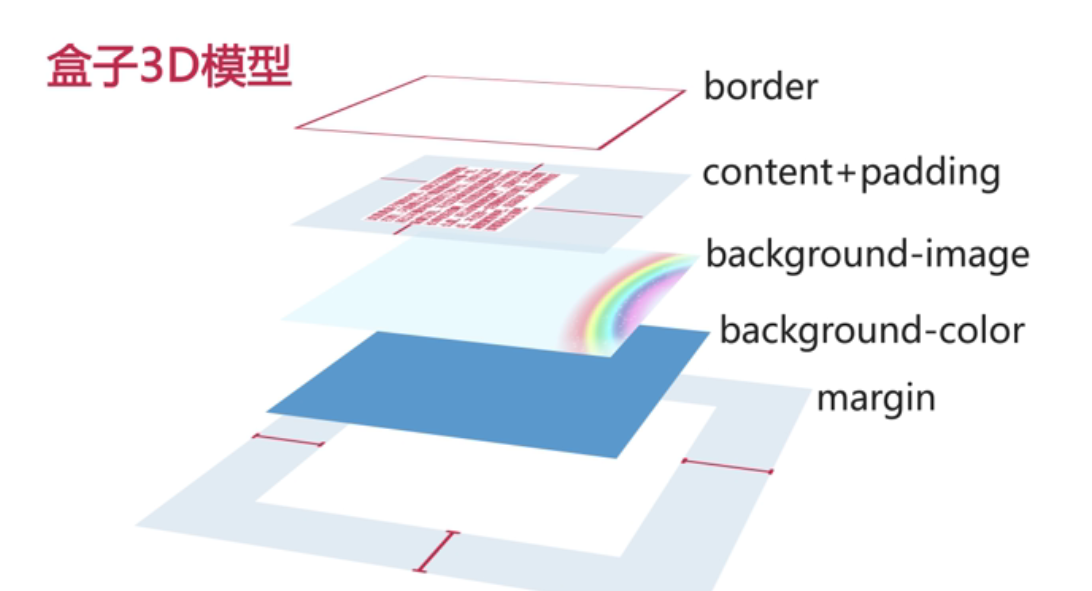
inline-block 元素特点：

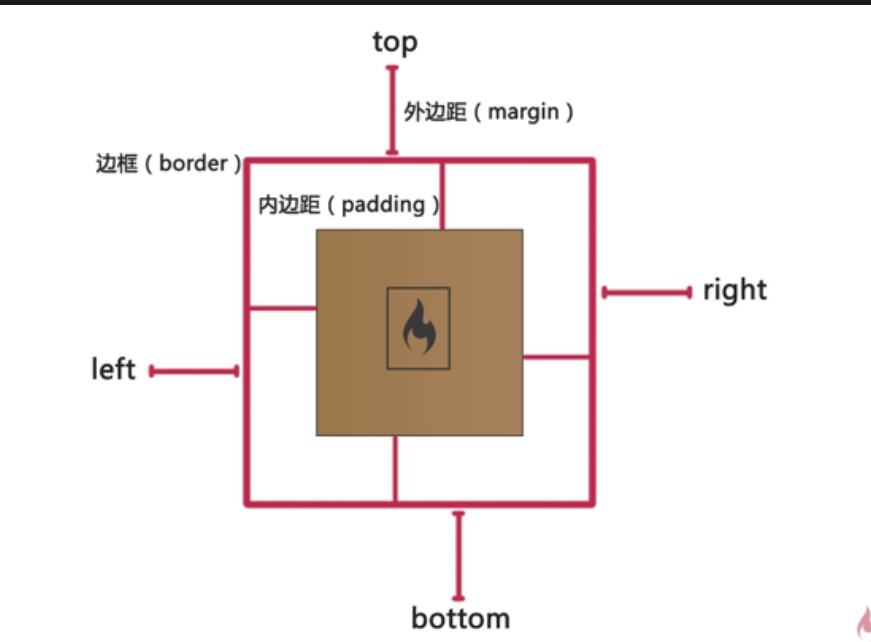
1、和其他元素都在一行上；

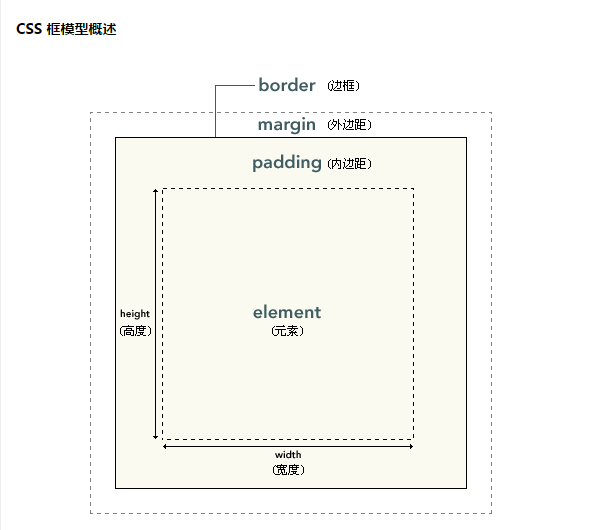
2、元素的高度、宽度、行高以及顶和底边距都可设置。



6.5 盒模型



**第一层到第五层依次为：border、content+padding、background-image、background-color、margin**。

<http://www.w3school.com.cn/css/css_boxmodel.asp> 

#box {

width: 70px;

margin: 10px;

padding: 5px;

}

## 6.5 盒子**边框**

如下面代码为 div 来设置边框粗细为 2px、样式为实心的、颜色为红色的边框：

div{

border:2px solid red;

}

上面是 border 代码的缩写形式，可以分开写：

div{

border-width:2px;

border-style:solid;

border-color:red;

}

****注意：****

1、border-style（边框样式）常见样式有：

dashed（虚线）| dotted（点线）| solid（实线）。

2、border-color（边框颜色）中的颜色可设置为十六进制颜色，如:

border-color:#888;//前面的井号不要忘掉。

3、border-width（边框宽度）中的宽度也可以设置为：

thin | medium | thick（但不是很常用），最常还是用象素（px）。

4、为一个方向的边框设置样式：

div{****border-bottom****:1px solid red;}

同样可以使用下面代码实现其它三边(上、右、左)边框的设置：

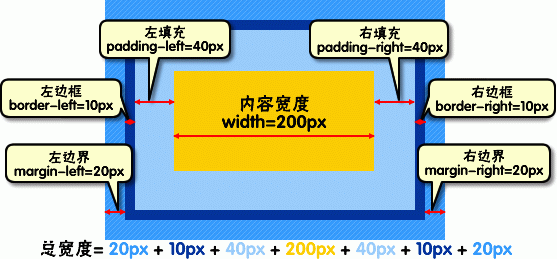
border-top:1px solid red;

border-right:1px solid red;

border-left:1px solid red;

## 6.6 **盒模型--宽度和高度**

一个元素实际宽度（盒子的宽度）=左边界+左边框+左填充+内容宽度+右填充+右边框+右边界。

[](http://img.mukewang.com/539fbb3a0001304305570259.jpg)

元素的高度也是同理。

比如：

css代码：

div{

width:200px;

padding:20px;

border:1px solid red;

margin:10px;

}

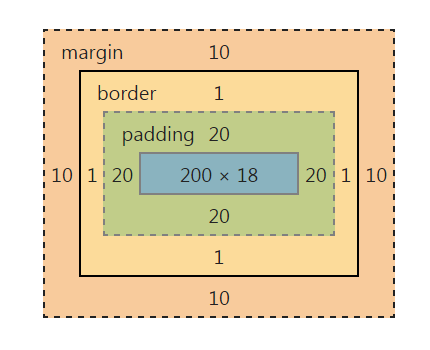
html代码：

<body>

<div>文本内容</div>

</body>

元素的实际长度为：10px+1px+20px+200px+20px+1px+10px=262px。在chrome浏览器下可查看元素盒模型，如下图：

[](http://img.mukewang.com/543b4cae0001b34304300350.jpg)

## 6.7 **盒模型--填充**

元素内容与边框之间是可以设置距离的，称之为“填充”。填充也可分为上、右、下、左(顺时针)。如下代码：

div{padding:20px 10px 15px 30px;}

顺序一定不要搞混。可以分开写上面代码：

div{

padding-top:20px;

padding-right:10px;

padding-bottom:15px;

padding-left:30px;

}

如果上、右、下、左的填充都为10px;可以这么写

div{padding:10px;}

如果上下填充一样为10px，左右一样为20px，可以这么写：

div{padding:10px 20px;}

# 7 CSS布局

## 7.1 **css布局模型**

在网页中，元素有三种布局模型：  
1、流动模型（Flow）  
2、浮动模型 (Float)  
3、层模型（Layer）

7.2 ****流动模型****，流动（Flow）是默认的网页布局模式。也就是说网页在默认状态下的 HTML 网页元素都是根据流动模型来分布网页内容的。

流动布局模型具有2个比较典型的特征：

第一点，****块状元素****都会在所处的****包含元素内****自上而下按顺序垂直延伸分布，因为在默认状态下，块状元素的宽度都为****100%****。

第二点，在流动模型下，****内联元素****都会在所处的包含元素内从左到右水平分布显示。（内联元素可不像块状元素这么霸道独占一行）

## 7.2 **浮动模型**

块状元素这么霸道都是独占一行，如果现在我们想让两个块状元素并排显示，怎么办呢？不要着急，设置元素浮动就可以实现这一愿望。

任何元素在默认情况下是不能浮动的，但可以用 CSS 定义为浮动，如 div、p、table、img 等元素都可以被定义为浮动。如下代码可以实现两个 div 元素一行显示。

<title>浮动模型</title>

<style type="text/css">

div{

border:2px red solid;

width:200px;

height:400px;

}

#div1{float:left;}

#div2{float:right;}

</style>

</head>

<body>

<div id="div1">栏目1</div>

<div id="div2">栏目2</div>

</body>

</html>

## 7.3 **什么是层模型？**

层布局。

如何让html元素在网页中精确定位，就像图像软件PhotoShop中的图层一样可以对每个图层能够精确定位操作。CSS定义了一组定位（positioning）属性来支持层布局模型。

层模型有三种形式：

1、****绝对定位****(position: absolute)

2、****相对定位****(position: relative)

3、****固定定位****(position: fixed)

## 7.4 **层模型--绝对定位**

如果想为元素设置层模型中的绝对定位，需要设置****position:absolute****(表示绝对定位)，这条语句的作用将元素从文档流中拖出来，然后使用left、right、top、bottom属性相对于其最接近的一个具有定位属性的父包含块进行绝对定位。如果不存在这样的包含块，则相对于body元素，即相对于****浏览器窗口****。

如下面代码可以实现div元素相对于浏览器窗口向右移动100px，向下移动50px。

div{

width:200px;

height:200px;

border:2px red solid;

position:absolute;

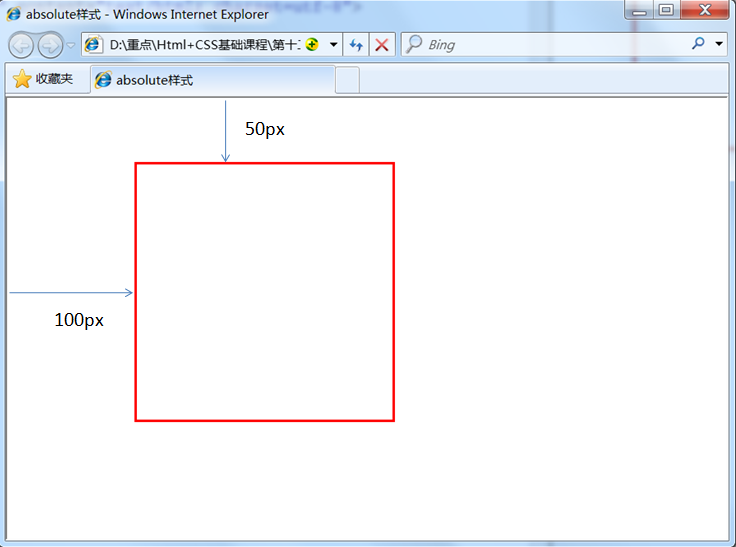
left:100px;

top:50px;

}

<div id="div1"></div>

效果如下：



## 7.5 **层模型--相对定位**

如果想为元素设置层模型中的相对定位，需要设置position:relative（表示相对定位），它通过left、right、top、bottom属性确定元素在****正常文档流中****的偏移位置。相对定位完成的过程是首先按static(float)方式生成一个元素(并且元素像层一样浮动了起来)，然后相对于****以前的位置移动，****移动的方向和幅度由left、right、top、bottom属性确定，偏移前的位置保留不动。

如下代码实现相对于以前位置向下移动50px，向右移动100px;

#div1{

width:200px;

height:200px;

border:2px red solid;

position:relative;

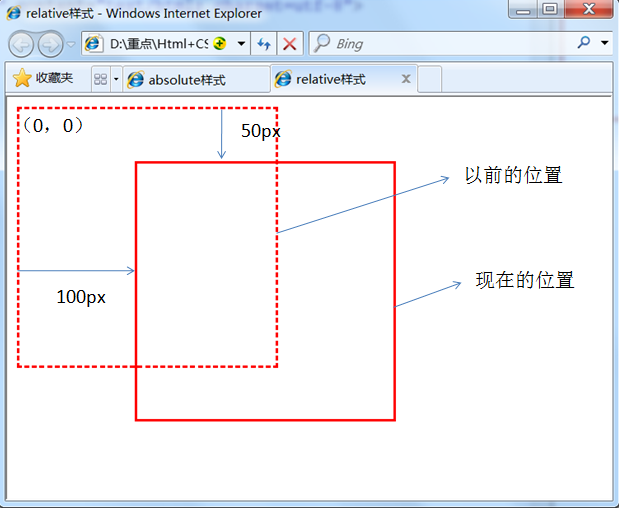
left:100px;

top:50px;

}

<div id="div1"></div>

效果图：



## 7.6 **层模型--固定定位**

fixed：表示固定定位，与absolute定位类型类似，但它的相对移动的坐标是视图（****屏幕内的网页窗口****）本身。由于视图本身是固定的，它不会随浏览器窗口的滚动条滚动而变化，除非你在屏幕中移动浏览器窗口的屏幕位置，或改变浏览器窗口的显示大小，因此固定定位的元素会始终位于浏览器窗口内视图的某个位置，不会受文档流动影响，这与background-attachment:fixed;属性功能相同。以下代码可以实现相对于****浏览器视图****向右移动100px，向下移动50px。并且拖动滚动条时位置固定不变。

#div1{

width:200px;

height:200px;

border:2px red solid;

position:fixed;

left:100px;

top:50px;

}

7.7**Relative与Absolute组合使用**

1、参照定位的元素必须是相对定位元素的前辈元素：

<div id="box1"><!--参照定位的元素-->

<div id="box2">相对参照元素进行定位</div><!--相对定位元素-->

</div>

从上面代码可以看出box1是box2的父元素（父元素当然也是前辈元素了）。

2、参照定位的元素必须加入position:relative;

#box1{

width:200px;

height:200px;

position:relative;

}

3、定位元素加入position:absolute，便可以使用top、bottom、left、right来进行偏移定位了。

#box2{

position:absolute;

top:20px;

left:30px;

}

这样box2就可以相对于父元素box1定位了（这里注意参照物就可以不是浏览器了，而可以自由设置了）。

# 8 CSS代码缩写

## **8.1 盒模型代码简写**

还记得在讲盒模型时外边距(margin)、内边距(padding)和边框(border)设置上下左右四个方向的边距是按照顺时针方向设置的：上右下左。具体应用在margin和padding的例子如下：

margin:10px 15px 12px 14px;/\*上设置为10px、右设置为15px、下设置为12px、左设置为14px\*/

通常有下面三种缩写方法:

1、如果top、right、bottom、left的值相同，如下面代码：

margin:10px 10px 10px 10px;

可缩写为：

margin:10px;

2、如果top和bottom值相同、left和 right的值相同，如下面代码：

margin:10px 20px 10px 20px;

可缩写为：

margin:10px 20px;

3、如果left和right的值相同，如下面代码：

margin:10px 20px 30px 20px;

可缩写为：

margin:10px 20px 30px;

注意：padding、border的缩写方法和margin是一致的

## 8.2 **颜色值缩写**

关于颜色的css样式也是可以缩写的，当你设置的颜色是16进制的色彩值时，如果每两位的值相同，可以缩写一半。

例子1：

p{color:#000000;}

可以缩写为：

p{color: #000;}

例子2：

p{color: #336699;}

可以缩写为：

p{color: #369;}

## **8.3 字体缩写**

网页中的字体css样式代码也有他自己的缩写方式，下面是给网页设置字体的代码：

body{

font-style:italic;

font-variant:small-caps;

font-weight:bold;

font-size:12px;

line-height:1.5em;

font-family:"宋体",sans-serif;

}

这么多行的代码其实可以缩写为一句：

body{

font:italic small-caps bold 12px/1.5em "宋体",sans-serif;

}

注意：

1、使用这一简写方式你至少要指定 font-size 和 font-family 属性，其他的属性(如 font-weight、font-style、font-varient、line-height)如未指定将自动使用默认值。

2、在缩写时 font-size 与 line-height 中间要加入“/”斜扛。

一般情况下因为对于中文网站，英文还是比较少的，所以下面缩写代码比较常用：

body{

font:12px/1.5em "宋体",sans-serif;

}

只是有字号、行间距、中文字体、英文字体设置。

## **8.4 颜色值**

在网页中的颜色设置是非常重要，有字体颜色（color）、背景颜色（background-color）、边框颜色（border）等，设置颜色的方法也有很多种：

1、英文命令颜色

前面几个小节中经常用到的就是这种设置方法：

p{color:red;}

2、RGB颜色

这个与 photoshop 中的 RGB 颜色是一致的，由 R(red)、G(green)、B(blue) 三种颜色的比例来配色。

p{color:rgb(133,45,200);}

每一项的值可以是 0~255 之间的整数，也可以是 0%~100% 的百分数。如：

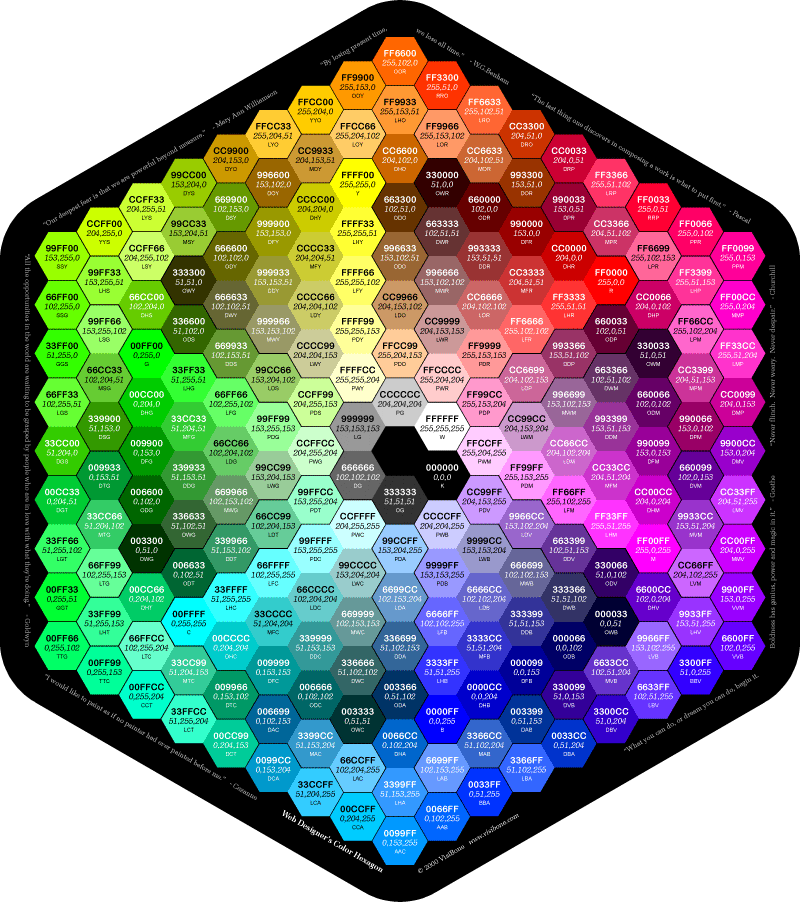
p{color:rgb(20%,33%,25%);}

3、十六进制颜色

这种颜色设置方法是现在比较普遍使用的方法，其原理其实也是 RGB 设置，但是其每一项的值由 0-255 变成了十六进制 00-ff。

p{color:#00ffff;}

配色表：

[](http://img.mukewang.com/54c5b4120001f20808000902.jpg)

## **8.5 长度值**

长度单位总结一下，目前比较常用到px（像素）、em、% 百分比，要注意其实这三种单位都是相对单位。

****1、像素****

像素为什么是相对单位呢？因为像素指的是显示器上的小点（CSS规范中假设“90像素=1英寸”）。实际情况是浏览器会使用显示器的实际像素值有关，在目前大多数的设计者都倾向于使用像素（px）作为单位。

****2、em****

就是本元素给定字体的 font-size 值，如果元素的 font-size 为 14px ，那么 1em = 14px；如果 font-size 为 18px，那么 1em = 18px。如下代码：

p{font-size:12px;text-indent:2em;}

上面代码就是可以实现段落首行缩进 24px（也就是两个字体大小的距离）。

****下面注意一个特殊情况：****

但当给 font-size 设置单位为 em 时，此时计算的标准以 p 的父元素的 font-size 为基础。如下代码：

html:

<p>以这个<span>例子</span>为例。</p>

css:

p{font-size:14px}

span{font-size:0.8em;}

结果 span 中的字体“例子”字体大小就为 11.2px（14 \* 0.8 = 11.2px）。

****3、百分比****

p{font-size:12px;line-height:130%}

设置行高（行间距）为字体的130%（12 \* 1.3 = 15.6px）。

# 9 CSS样式设置小技巧

## 9.1 **水平居中设置-行内元素**

水平居中是通过给****父元素****设置 text-align:center 来实现的。如下代码：

html代码：

<body>

<div class="txtCenter">我是文本，哈哈，我想要在父容器中水平居中显示。</div>

</body>

css代码：

<style>

div.txtCenter{

text-align:center;

}

</style>

9.2 **定宽块状元素**

当被设置元素为块状元素时用 text-align：center 就不起作用了，这时也分两种情况：****定宽****块状元素和****不定宽****块状元素。这一小节我们先来讲一讲定宽块状元素。

满足定宽和块状两个条件的元素是可以通过设置“左右margin”值为“auto”来实现居中的。我们来看个例子就是设置 div 这个块状元素水平居中：

html代码：

<body>

<div>我是定宽块状元素，哈哈，我要水平居中显示。</div>

</body>

css代码：

<style>

div{

border:1px solid red;/\*为了显示居中效果明显为 div 设置了边框\*/

width:500px;/\*定宽\*/

margin:20px auto;/\* margin-left 与 margin-right 设置为 auto \*/

}

</style>

## 9.3 **水平居中总结-不定宽块状元素方法（一）**

在实际工作中我们会遇到需要为“不定宽度的块状元素”设置居中，比如网页上的分页导航，因为分页的数量是不确定的，所以我们不能通过设置宽度来限制它的弹性。

不定宽度的块状元素有三种方法居中（这三种方法目前使用的都比多）：

1. 加入 table 标签
2. 设置 display;inline 方法
3. 设置 position:relative 和 left:50%;

这一小节我们来讲一下第一种方法：

第一步：为需要设置的居中的元素外面加入一个 table 标签 ( 包括 <tbody>、<tr>、<td> )。

第二步：为这个 table 设置“左右 margin 居中”（这个和定宽块状元素的方法一样）。

举例如下：

html代码：

<div><table>

<tbody>

<tr><td>

<ul>

<li><a href="#">1</a></li>

<li><a href="#">2</a></li>

<li><a href="#">3</a></li>

</ul>

</td></tr>

</tbody>

</table>

</div>

css代码：

<style>table{

margin:0 auto;

}

ul{list-style:none;margin:0;padding:0;}

li{float:left;display:inline;margin-right:8px;}

</style>

## 9.4 **水平居中总结-不定宽块状元素方法（二）**

第二种方法：改变块级元素的 display 为 inline 类型，然后使用 text-align:center 来实现居中效果。如下例子：

<!DOCTYPE HTML>

<html>

<head>

<meta charset="utf-8">

<title>不定宽块状元素水平居中</title>

<style>

.container{text-align:center;}

.container ul{list-style:none;margin:0;padding:0;display:inline;}

.container li{margin-right:8px;display:inline;}

</style>

</head>

<body>

<div class="container">

<ul>

<li><a href="#">1</a></li>

<li><a href="#">2</a></li>

<li><a href="#">3</a></li>

</ul>

</div>

</body>

</html>

## 9.5 **水平居中总结-不定宽块状元素方法（三）**

方法三：通过给父元素设置 float，然后给父元素设置 position:relative 和 left:50%，子元素设置 position:relative 和 left:-50% 来实现水平居中。

代码如下：

<body>

<div class="container">

<ul>

<li><a href="#">1</a></li>

<li><a href="#">2</a></li>

<li><a href="#">3</a></li>

</ul>

</div>

</body>

css代码：

<style>

.container{

float:left;

position:relative;

left:50%

}

.container ul{

list-style:none;

margin:0;

padding:0;

position:relative;

left:-50%;

}

.container li{float:left;display:inline;margin-right:8px;}

</style>

这种方法可以保留块状元素仍以 display:block 的形式显示，优点不添加无语议表标签，不增加嵌套深度，但它的缺点是设置了 position:relative，带来了一定的副作用。

## **9.6 垂直居中-父元素高度确定的单行文本**

****父元素高度确定的单行文本****的竖直居中的方法是通过设置父元素的 height 和 line-height 高度一致来实现的。如下代码：

<div class="container">

hi,imooc!

</div>

css代码：

<style>

.container{

height:100px;

line-height:100px;

background:#999;

}

</style>

## 9.7 **垂直居中-父元素高度确定的多行文本（方法一）**

父元素高度确定的多行文本、图片、块状元素的竖直居中的方法有两种：

方法一：使用插入 table (包括tbody、tr、td)标签，同时设置 vertical-align：middle。

说到竖直居中，css 中有一个用于竖直居中的属性 vertical-align，但这个样式只有在父元素为 td 或 th 时，才会生效。所以又要插入 table 标签了。下面看一下例子：

html代码：

<body><table><tbody><tr><td class="wrap">

<div>

<p>看我是否可以居中。</p>

<p>看我是否可以居中。</p>

<p>看我是否可以居中。</p>

<p>看我是否可以居中。</p>

<p>看我是否可以居中。</p>

</div></td></tr></tbody></table>

</body>

css代码：

table td{height:500px;background:#ccc}

因为 td 标签默认情况下就默认设置了 vertical-align 为 middle，所以我们不需要显式地设置了。

## 9.8 **垂直居中-父元素高度确定的多行文本（方法二）**

在 chrome、firefox 及 IE8 以上的浏览器下可以设置块级元素的 display 为 table-cell，激活 vertical-align 属性，但注意 IE6、7 并不支持这个样式。

html代码：

<div class="container">

<div>

<p>看我是否可以居中。</p>

<p>看我是否可以居中。</p>

<p>看我是否可以居中。</p>

<p>看我是否可以居中。</p>

<p>看我是否可以居中。</p>

</div>

</div>

css代码：

<style>

.container{

height:300px;

background:#ccc; display:table-cell;/\*IE8以上及Chrome、Firefox\*/

vertical-align:middle;/\*IE8以上及Chrome、Firefox\*/

}

</style>

这种方法的好处是不用添加多余的无意义的标签，但缺点也很明显，它的兼容性不是很好，不兼容 IE6、7。

## 9.9 **隐性改变display类型**

有一个有趣的现象就是当为元素（不论之前是什么类型元素，display:none 除外）设置以下 2 个句之一：

1 position : absolute 2 float : left 或 float:right

元素会自动变为以 display:inline-block 的方式显示，当然就可以设置元素的 width 和 height 了且默认宽度不占满父元素。

如下面的代码，小伙伴们都知道 a 标签是行内元素，所以设置它的 width 是 没有效果的，但是设置为 position:absolute 以后，就可以了。

<div class="container">

<a href="#" title="">进入课程请单击这里</a>

</div>

css代码

<style>

.container a{

position:absolute;

width:200px;

background:#ccc;

}

</style>

1 input需要清掉哪些默认样式，和兼容问题如何解决？

2 如何解决IE8一下的input框文字偏上问题？

<!doctype html>

<html>

<head>

<meta charset="utf-8">

<title>Busy - 电商</title>

<style>

/\*

任务：

1 input需要清掉哪些默认样式，和兼容问题如何解决？

2 如何解决IE8一下的input框文字偏上问题

\*/

.search\_box{

width:500px;

height:30px;

margin:100px auto;

}

.search\_input{

width:440px;

height:30px;

float:left;

}

.search\_btn{

width:60px;

height:30px;

line-height:30px;

float:left;

color:#fff;

font-family:"Microsoft YaHei","微软雅黑";

background-color:#ff8c00;

}

</style>

</head>

<body>

<div class="search\_box">

<input type="text" class="search\_input">

<input type="button" class="search\_btn" value="搜索">

</div>

</body>

</html>