**前端自学笔记**

**第一天**

<!DOCTYPE HTML>

<html>

<head>

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

<title>认识html标签</title>

</head>

<body>

<h1>勇气</h1> //<h1></h1>就是**标题标签**

<p>三年级时，我还是一个胆小如鼠的小女孩，上课从来不敢回答老师提出的问题，生怕回答错了老师会批评我。就一直没有这个勇气来回答老师提出的问题。学校举办的活动我也没勇气参加。</p>

//**<p></p>**是**段落标签 不同于<q></q> 后者是引用文本**

<p>到了三年级下学期时，我们班上了一节公开课，老师提出了一个很简单的问题，班里很多同学都举手了，甚至成绩比我差很多的，也举手了，还说着："我来，我来。"我环顾了四周，就我没有举手。</p>

<img src="http://img.mukewang.com/52b4113500018cf102000200.jpg" >

//img标签显示图片

</body>

</html>

**标签**由英文尖括号**<**和**>**括起来，如<html>就是一个标签。

 html中的标签一般都是成对出现的，分**开始标签**和**结束标签**。结束标签比开始标签多了一个/

HTML标签不区分大小写，<h1>和<H1>是一样的，但建议小写，因为大部分程序员都以小写为准。

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。

注释方法：

**<!--注释文字 -->**

头部标签：

<head>

<title>...</title>

<meta>

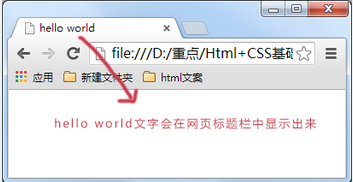
<link>

<style>...</style>

<script>...</script>

</head>

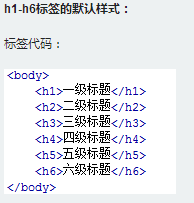
<title>标签：在<title>和</title>标签之间的文字内容是网页的标题信息，它会**出现在浏览器的标题栏**中。网页的title标签用于**告诉用户和搜索引擎这个网页的主要内容是什么**，**搜索引擎可以通过网页标题，迅速的判断出网页的主题**。每个网页的内容都是不同的，每个网页都应该有一个独一无二的title

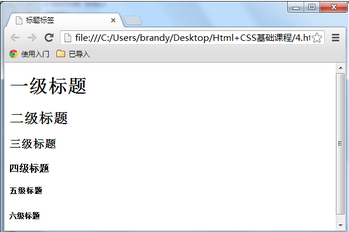


**标签的用途**：我们学习网页制作时，常常会听到一个词，**语义化**。那么什么叫做语义化呢，说的通俗点就是：明白每个标签的用途（在什么情况下使用此标签合理）比如，网页上的文章的**标题**就可以用标题标签，网页上的各个栏目的**栏目名称**也可以使用标题标签。文章中内容的段落就得放在**段落标签**中，在文章中有想强调的文本，就可以使用 em 标签表示强调等等。

<hx>标签来制作**文章的标题**

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。





**<em>或<strong>标签**

想在一段话中特别强调某几个文字，这时候就可以用到<em>或<strong>标签。

<em> 默认用**斜体**表示，<strong> 用**粗体**表示。

**语法：**

<em>需要强调的文本</em>

<strong>需要强调的文本</strong>

**<span>标签**

<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。比如改变颜色

**语法：**

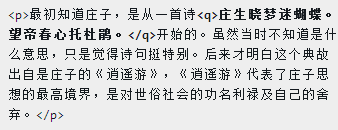
<span>文本</span>

**第二天**

**<q>标签,短文本引用**

**语法：**

<q>引用文本</q>

****

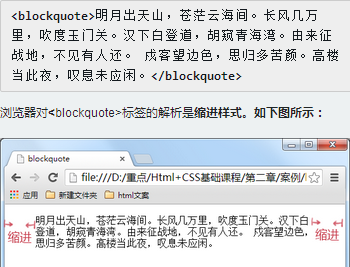
1. 在上面的例子中，“**庄生晓梦迷蝴蝶。望帝春心托杜鹃。**” 这是一句诗歌，出自晚唐诗人李商隐的《锦瑟》 。因为不是作者自己的文字，所以需要使用<q></q>实现引用。

2. 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>标签,长文本引用**

**语法:**

**<**blockquote>引用文本</blockquote> 浏览器对**<**blockquote>标签的解析是**缩进样式。**

****

**<br>标签,分行显示文本**

**语法：**

**xhtml1.0写法：（常用写法）**

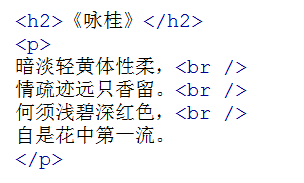
<br />

**html4.01写法：**

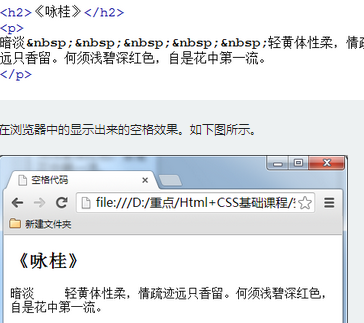
<br>

在 html 中是忽略**回车**和**空格**的，你输入的再多回车和空格也是显示不出来的。

写法如下：

****

**空格输入 : &nbsp;**



**<hr>标签，添加水平横线**

**语法：**

html4.01版本<hr>

xhtml1.0版本 <hr /> (常用写法)

<hr />标签和<br />标签一样也是一个**空标签**，所以只有一个开始标签，没有结束标签。

## <address>标签，为网页加入地址信息

**语法：**

<address>联系地址信息</address>

 在浏览器上显示的样式为**斜体**，如果不喜欢斜体，当然可以，可以在后面的课程中使用 css 样式来修改它

## <code>标签，加入一行代码

**语法：**

<code>代码语言</code>

注：如果是多行代码，可以使用<pre>标签。 <pre>语言代码段</pre>

<body>

<pre>

var message="欢迎";

for(var i=1;i<=10;i++)<br>

{<br>

alert(message); <br>

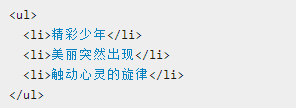
}<br>

</pre>

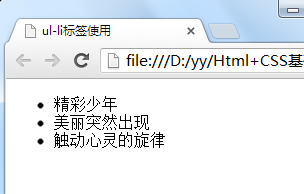
</body>

## 使用ul，添加新闻信息列表

实例：



ul-li在网页中显示的默认样式一般为：每项li前都自带一个圆点



## 使用ol，添加图书销售排行榜

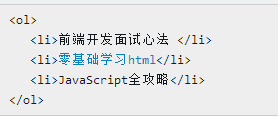
<ol>

<li>[信息](http://www.zuowen.com/e/20130805/51ff0eacd8d21.shtml)</li>

<li>[信息](http://www.zuowen.com/e/20130805/51ff0eacd8d21.shtml)</li>

......

</ol>



<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



## div在排版中的作用

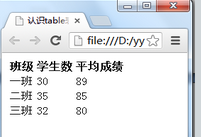
**语法：**

<div><id = “版块名字”></id>…描述.</div> id用于为div命名



**第三天**

**table标签，网页上的表格**



表格效果

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。（黑体）**

6、表格中列的个数，取决于一行中数据单元格的个数

<table>

<tbody>

<tr>

<th>班级</th>

<th>学生数</th>

<th>平均成绩</th>

</tr>

<tr>

<td>一班</td>

<td>30</td>

<td>89</td>

</tr>

<tr>

<td>二班</td>

<td>35</td>

<td>85</td>

</tr>

</tbody>

</table> //显示效果如上

**<style type="text/css">**

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

**</style> 可为表格添加边框**

**caption标签，为表格添加标题和摘要**

**语法：**<table summary="表格简介文本">

**语法：**

<table>

**<caption>标题文本</caption>**

<tr>

<td>…</td>

<td>…</td>

…

</tr>

…

</table>

**<a>标签，链接到另一个页面**

**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（就是那些蓝题字）</a>

**实例：**

<a href="http://www.imooc.com" title="点击进入慕课网">click here!</a>

只要为文本加入a标签后，文字的颜色就会自动变为蓝色（被点击过的文本颜色为紫色），

<a href="目标网址" **target="\_blank"**>click here!</a> 在新的浏览器中打开（创建一个新的页面，不覆盖之前页面）

<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

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。

**<img>标签，为网页插入图片**

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

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

**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

**表单标签，与用户交互**

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">

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）。

**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！）。**

**2、method:post/get的区别这一部分内容属于后端程序员考虑的问题。感兴趣的小伙伴可以查看本小节的wiki，里面有详细介绍。**

**实例：**<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>

<form>

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

</form>

   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。**

**name：**为文本框命名，以备后台程序ASP 、PHP使用。

**value：**为文本输入框设置默认值。(一般起到提示作用)

**文本域，支持多行文本输入**

**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**

**实例：**

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

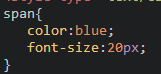
<label>联系我们</label>

<textarea cols="50" rows="10" >在这里输入内容...</textarea>

</form>

**CSS笔记**

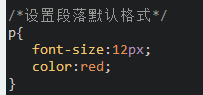
CSS全称为“**层叠样式表** (Cascading Style Sheets)”， 主要是用于定义HTML内容在浏览器内的显示样式





**选择符：**又称选择器，指明网页中要应用样式规则的元素，

**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，



**css样式：内联式、嵌入式和外部式**

**内联式：**css样式表就是把css代码直接写在现有的HTML标签中，

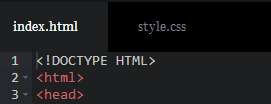
<p style="color:red;font-size:12px">这里文字是红色。</p>

**嵌入式css样式**：就是可以把css样式代码写在**<style type="text/css"></style>**标签之间。



**外部式css样式**：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，

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



注意：

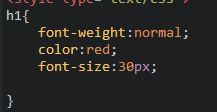
1、css样式文件名称以有意义的英文字母命名，如 main.css。

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**

**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>

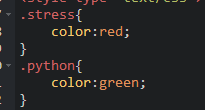
****

**类选择器**在css样式编码中是最常用到的，

**.**类选器名称{css样式代码;}

1、**英文圆点开头**

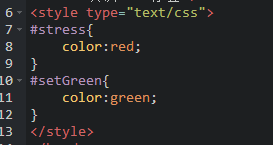
2、其中**类选器名称**可以任意起名

****

**ID选择器：**类似于类选择符

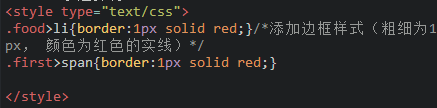
1、为标签设置id="ID名称"，而不是class="类名称"。

2、ID选择符的前面是井号**（#）**号，而不是英文圆点**（.）**。

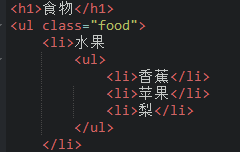


**ID选择器只能在文档中使用一次**。与类选择器不同，在一个HTML文档中，ID选择器只能使用一次，而且仅一次。而类选择器可以使用多次。

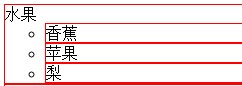
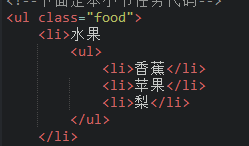
**子选择器**：即大于符号(>),用于选择指定标签元素的**第一代子元素。**

****

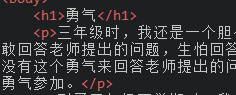
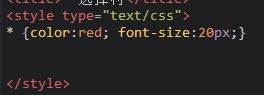
****

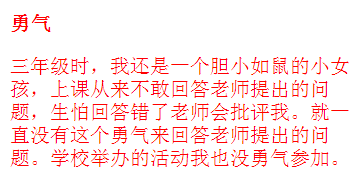
****

**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。**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。

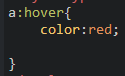
****

**通用选择器**：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

****

****

**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

**固定的**

**分组选择符：**为html中多个标签元素设置同一个样式。

h1,span{color:red;} 等价于 h1{color:red;}

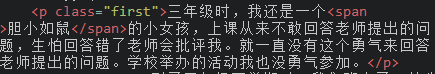
span{color:red;}

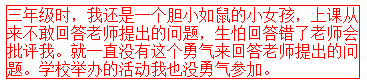
可以和类选择器组合使用，并匹配后者span eg：.class,#second span{color:green;}

**继承**

继承是一种规则，它允许样式不仅应用于某个特定html标签元素，而且应用于其后代。

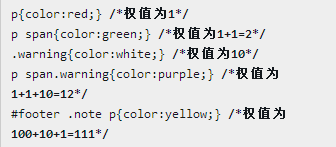
****

**结果如下：**

****

**特殊性**

**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。**

****

**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。**

**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。

同名时，执行后者。

p{color:red;}

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>

最后 p 中的文本会设置为green，这个层叠很好理解，理解为后面的样式会覆盖前面的样式。

**重要性**

**!important：**某些样式设置具有最高权值。

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**

**文字排版和段落排版**

**字体：**

为网页中的文字设置字体为宋体。body{font-family:"宋体";}

一般网页喜欢设置“微软雅黑”： body{font-family:"Microsoft Yahei";}

注意：第二种方法比第一种方法兼容性更好一些。

**字号和颜色：**

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}

**粗体：**

粗体、斜体、下划线、删除线，可以使用下面代码实现设置文字以粗体样式显示出来。

p span{font-weight:bold;}

**斜体：**

p a{font-style:italic;}

**下划线：**

p a{text-decoration:underline;}

**删除线：**

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

**缩进：**

p{text-indent:2em;}

注意：2em的意思就是文字的2倍大小。

**行间距（行高）：**

p{line-height:1.5em;}

**段落排版—中间字间距、字母间距**

置**文字间隔**或者**字母间隔**就可以使用    **letter-spacing**来实现

h1{

letter-spacing:50px;

}

注意：这个样式使用在英文单词时，是设置字母与字母之间的间距。

使用 **word-spacing**来实现 英文单词之间的间距。

h1{

word-spacing:50px;

}

**对齐：**

h1{

text-align:center（right、left）;居中、右对齐和左对齐

}

**总结**

文字样式有：**字体**font-family、**字号和颜色**font-size color、**粗体**font-weight、**斜体**font-style、**下划线**text-decoration、**删除线**text-decoration、**缩进**text-indent、**行间距（行高）**line-height

段落排版有：**文字、字母间隔**letter-spacing、**英文单词间隔**word-spacing、**对齐**text-align:center

**元素分类**

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>

**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：**

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。

**内联元素：块状元素**也可以通过代码display:inline将元素设置为**内联元素**

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;}

**内联元素特点：**

1、和其他元素都在一行上；

2、元素的高度、宽度及顶部和底部边距**不可**设置；

3、元素的宽度就是它包含的文字或图片的宽度，不可改变

**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，

display:inline-block就是将元素设置为内联块状元素。

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。

提示：下一小节是用视频动画来讲解css中的盒模型。

**盒模型：**

**边框：**盒子模型的边框就是围绕着内容及补白的线，这条线你可以设置它的粗细、样式和颜色(边框三个属性)

div{

border-width:2px;

border-style:solid;

border-color:red;

} //边框的三种属性

div{border:2px solid red;} 简写

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）。

标签单独设置下边框，而其它三边都不设置边框样式：

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

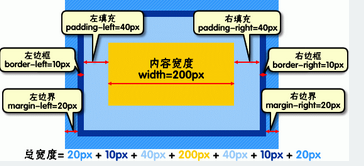
**其他方位的边框设置**

border-top:1px solid red;

border-right:1px solid red;

border-left:1px solid red;

**宽度和高度**

****

元素的高度也是同理。

**例如：**

div{

width:200px;

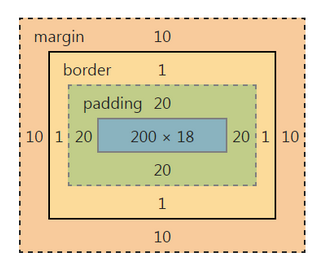
padding:20px;

border:1px solid red;

margin:10px;

}

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：

****

**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)。

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

div{

padding-top:20px;

padding-right:10px;

padding-bottom:15px;

padding-left:30px;

}

div{padding:10px;} 上下左右都为10px

div{padding:10px 20px;}上下为10px，左右为20px

**边界：**

元素与其它元素之间的距离可以使用边界（margin）来设置。边界也是可分为上、右、下、左。

div{margin:20px 10px 15px 30px;} 和填完全类似

padding和margin的区别，padding在边框里，margin在边框外。

**Css布局模型：**

CSS包含3种基本的布局模型，用英文概括为：Flow、Layer 和 Float。

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）

**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。

**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%**。实际上，块状元素都会以行的形式占据位置。如右侧代码编辑器中三个块状元素标签(div，h1，p)宽度显示为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）

**浮动模型：**设置元素浮动就可以实现让两个块状元素并排显示。

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

div{

width:200px;

height:200px;

border:2px red solid;

float:left;设置为浮动模式也可以float:right; #div1{float:left;}

#div2{float:right;}

}

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

<div id="div2"></div>

**层模型：**

层布局模型就像是图像软件PhotoShop中非常流行的图层编辑功能一样，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)

**绝对定位：**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>

距离浏览器的边框距离，貌似这点与margin不同

**相对定位：**需要设置position:relative（表示相对定位）

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>

**偏移前的位置保留不动**

**固定定位：**需要设置position:fixed（表示固定定位）

但它的相对移动的坐标是视图（**屏幕内的网页窗口**）本身。平时跟着滑动的广告窗口。使用和上面两种一样。

**颜色值：**有字体颜色（color）、背景颜色（background-color）、边框颜色（border）等

1、英文命令颜色p{color:red;}

2、RGB颜色p{color:rgb(133,45,200);}

1. 十六进制颜色p{color:#00ffff;}

**长度值：**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 即px**

**2、Em** 如果元素的 font-size 为 14px ，那么 1em = 14px；如果 font-size 为 18px，那么 1em = 18px。p{font-size:12px;text-indent:2em;}

**3、百分比** p{font-size:12px;line-height:130%}为前者的百分比

**水平居中设置-行内元素**

如果被设置元素为文本、图片等行内元素时，水平居中是通过给**父元素**设置 text-align:center 来实现的。

当被设置元素为 [块状元素](http://www.imooc.com/code/2048) 时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。

<style>

div{

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

width:200px;/\*定宽\*/

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

}

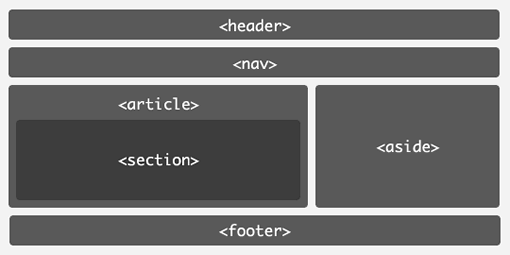
</style>

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很多）：

1. 加入 [table](http://www.imooc.com/code/292) 标签.。不定义其长度也不默认父元素body的长度（table其长度根据其内文本长度决定），因此可以看做一个定宽度块元素，然后再利用定宽度块状居中的margin的方法，使其水平居中
2. 设置 [display: inline](http://www.imooc.com/code/2049) 方法：与第一种类似，显示类型设为 行内元素，进行不定宽元素的属性设置
3. 设置 [position:relative](http://www.imooc.com/code/2074) 和 left:50%：利用 相对定位 的方式，将元素向左偏移 50% ，即达到居中的目的
4. **H5新特性**

对div的排版进行了优化：

<header>、<nav>、<aside>、<section>、<footer>、<article>



header: 一般用于定义页眉信息

nav: 用于定义导航栏

article：用于显示正文内容

section：用于显示正文中的某一部分

aside：定义侧边栏内容

footer：定义页脚以及相关编号等信息

这些都是可以使用div 来代替的，不过为了优化阅读性，特意定义了应的标签。

语义化标签：

Mark： 凸显标签 使内容显示更为突出

Progress：进度条标签

Details： 下拉选项标签， open属性设置默认打开

Datalist： 选项标签 ：和input连用 id属性值 与input的list 属性值必须相同 （优点在于智能匹配输入内容）

**Input新增输入类型：**

Email: 定义email地址字段 验证输入email的合法性

url：验证输入url的合法性

number：验证输入数字的合法性

range：精确值不重要的输入数字， 和number一样效果，但外观不一样

date pickers：定义日期输入(date，month，week，time，datetime-local等)

search：定义用于搜索字符串的的文本框

tel：定义电话字段，验证输入电话号码的合法性

color：定义拾色器

**Input新增属性：**

Autocomplete: 规定输入字段是否启动自动完成功能（即输入后，在第二次输入时是否保存上次输入信息 一般用户名“on” 密码“off”

Formtarget：“\_self”自身页面跳转 “\_blank”开启新的页面

Autofoucs：input自动获得焦点

Width和height：规定高度和宽度（只针对type = “image”）

Max、min和step：最大值、最小值、步进间隔

Multiple：可以选择多个文件上传（type = “file”）

Pattern：验证输入框中的值是否符合规定的正则式

Placeholder：规定 输入框提示消息的内容

Required：规定输入框提交前必须填写输入

**Canvas绘制图形标签**（只是一个容器，通过JavaScript在该区域绘制）

Js中通过getContext获取Context对象，以获得允许进行绘制的2D环境

context.beginPath()开始绘制新路径时使用， context.stroke()绘制效果显示

lineJion：设置线条的连接样式

绘制复杂路径时， 每条子路径的终点将作为新的上下文点， 通过lineTo()，arcTo()，quadraticCurveTo()，bezierCurveTo()创建新的子路径，

arc() : 绘制圆

rect() : 绘制矩形

还能绘制阴影和透明度

绘制线性渐变：

1、 使用createLinearGradient()创建canvasGradient对象

（四个参数，前两个为渐变的起点，后两个为渐变的终点）

2、 使用addColorStop()定义色标位置并上色

（两个参数，前一个为渐变中色标的偏移量， 后一个为颜色）

绘制径向渐变：

1、 使用createRadialGradient()创建canvasGradient对象

（六个参数，前三个为一个圆的圆心和半径， 后三个为另一个圆的圆心和半径）

2、 使用addColorStop()定义色标位置并上色

（两个参数，前一个为渐变中色标的偏移量， 后一个为颜色）

**Canvas绘制图像**

方法1：context.drawImage(imageObj, x, y) 一个图像对象和一个起始坐标相对于canvas的左上角

方法2：context.drawImage(imageObj, x, y， width， height)，增加宽高设定

方法3：context.drawImage(imageObj, sx, sy，sw，sh，dx，dy，dw，dh)对图像对象剪切放入新的位置

**Canvas绘制文本**

方法：context.fillText(Text, x, y)

Font属性设置字体，大小， 样式

fillStyle设置颜色

textAlign 设置对齐方式 left， right， start， end， center

measureText测量绘制的宽高

音频和视频：

对应的属性：

<http://www.devdo.net/html5-audio-video-attributes.html>

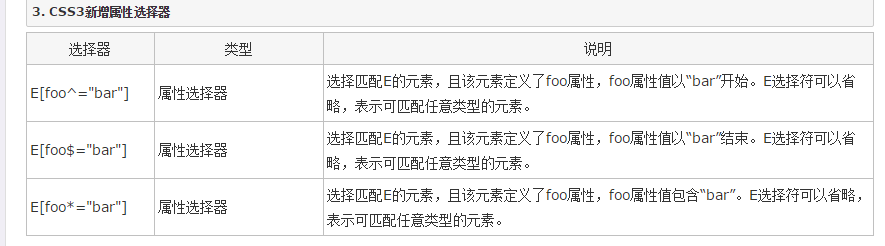
对应的事件：

<http://www.jb51.net/html5/432061.html>

方法

<http://www.open-open.com/lib/view/open1333958974139.html>

Css3

新增选择器：

常用于超链接选择。

新增结构性伪类选择器： 可以用于表格行列样式更改。

<http://www.jb51.net/css/43327.html>

新增ui伪类选择器：

E:enable  指定当元素处于可用状态时的样式；

E:disable 指定当元素处于不可用状态时的样式；

E:read-only  指定当元素处于只读状态时的样式；

E:read-write 指定当元素处于非只读状态时的样式；

E::selection  指定当元素处于选中状态时的样式；

E：out-of-range 用于标签的值在指定区间之外时显示的样式

E：in-range 用于标签的值在指定区间之内时显示的样式

E：valid和E：invalid 用于匹配输入值为合法、非法的元素

E：required 用于匹配设置了“required”属性的元素

E：hover鼠标指针浮动在其上的元素，并设置其样式

Css3 设置边框属性

border-radius 添加圆角边框

8个值， 分别为四个角的x、y轴的半径，可以不填写这么多个

border-image 添加边框背景图

5个值 source：图片url、slice：裁切方式、width：边框显示大小

Outset：图像偏移位置（不支持）、repeat：背景的重复性

box-shadow 设置一个或者多个下拉阴影的框

5个值， h-shadow：水平阴影位置可以为负（都为正在右下角）、v-shadow：垂直阴影位置可以为负， blur：模糊距离（可选） spread：阴影大小（可选） color：颜色（可选）

CSS3设置背景属性

Background-origin 定义背景图片的定位区域

有三个位置：padding-box，content-box，border-box，背景图片分别从内边界框、内容框、边界框开始

Background-clip 定义背景图片的裁剪区域

有三个位置：padding-box，content-box，border-box，从内边框、内容区域，边框区域开始向外裁剪

Background-size：定义背景图像的大小

Css3常用文本属性

Text-shadow 定义文本阴影或模糊效果

4个值：h-shadow：水平阴影位置可以为负（都为正在右下角）、v-shadow：垂直阴影位置可以为负， blur：模糊距离（可选） color：颜色（可选）

Text-overflow 定义省略文本处理方式 clip直接减去， ellipsis用省略号代替被修剪内容（同时定义white-space：nowrap<文本不会换行>，overflow：hidden<超过部分隐藏>才会有效果）

Word-wrap 定义文本超过指定容器边界是否断开 转行

Css3 2D变形

Transform：对元素实现变形（旋转、缩放、移动） 与hover伪类选择器使用

用法：transform：none/变形函数（参数）

函数如下： 旋转rotate（45deg）旋转45度

缩放scale（0.8, 0.8）缩写到0.8倍（前X轴，后Y轴）

移动translate（10px，10px）右下方移动10px（负值方向相反）

倾斜（矩形变平行四边形类似3D）skew（45deg 45deg）倾斜45度

多个效果可以叠加

Transform-origin：修改2D变形的中心点 ，改变中心点，2D变形将会变化

**Css3动画**

Transition（过渡）动画：从一种样式渐变成另一种效果

即：transition是通过设置开始点与结束点的样式进行平滑过渡

需要指定：1、添加效果的css属性2、效果持续时间

transition属性是一个速记属性有四个属性：

transition-property , transition-duration, transition-timing-function, transition-delay。



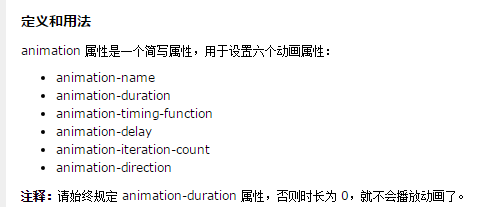
Animations动画

通过设置关键帧点的样式，来实现动画效果

首先需要了解@keyframes

<http://www.cnblogs.com/zuihongyan/p/5902541.html>

animations





Web storage 存储

http://blog.csdn.net/shenzhennba/article/details/8480756

Web Storage是HTML5引入的一个非常重要的功能，在前端开发中经常用到，可以在客户端本地存储数据，类似HTML4的cookie，但可实现功能要比cookie强大的多，cookie大小被限制在4KB，Web Storage官方**建议**为每个网站5MB。Web Storage又分为两种：

**sessionStorage**将数据保存在session中，浏览器关闭也就没了；

**localStorage**则一直将数据保存在客户端本地，除非主动删除数据，否则数据是永远不会过期的；

不管是sessionStorage，还是localStorage，可使用的API都相同，常用的有如下几个方法：

1. 保存数据：**localStorage.setItem( key,value );      sessionStorage.setItem( key, value );**
2. 读取数据：**localStorage.getItem( key );     sessionStorage.getItem( key );**
3. 删除单个数据：**localStorage.removeItem( key );     sessionStorage.removeItem( key );**
4. 删除所有数据：**localStorage.clear( );     sessionStorage.clear( );**
5. 得到某个索引的key：**localStorage.key( index );     sessionStorage.key( index );**

Web sql

<http://www.cnblogs.com/chuncn/archive/2010/11/22/1884783.html>

HTML5离线存储

http://www.360doc.com/content/12/0210/12/6128332\_185509811.shtml

需要使用的三功能：1、离线资源缓存。2、在线状态监测。3、本地数据存储

Line-height: