<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。

语法： <em>需要强调的文本</em>

<strong>需要强调的文本</strong>

<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。

<q>引用文本</q>

用<q>标签的真正关键点不是它的默认样式双引号（如果这样我们不如自己在键盘上输入双引号就行了），而是它的语义：引用别人的话

<blockquote>标签，长文本引用

<hr/>标签，添加水平横线

<address>标签，为网页加入地址信息

用<code>标签在网页上显示一行程序代码

用<pre>标签为你的网页加入大段代码

<pre> 标签不只是为显示计算机的源代码时用的，在你需要在网页中预显示格式时都可以使用它

<ul>是没有前后顺序的信息列表。

<ol>有前后顺序的信息列表

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。

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。

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">

<caption>标题文本</caption>

用以描述表格内容，标题的显示位置：表格上方。

<a>标签

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

在新建浏览器窗口中打开链接

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

使用mailto在网页中链接Email地址



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。

http://img.mukewang.com/52da52200001e00e07930061.jpg

貌似盗版用不了。。。

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>

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。

在使用表单设计调查表时，为了减少用户的操作，使用选择框是一个好主意，html中有两种选择框，即单选框和复选框，两者的区别是单选框中的选项用户只能选择一项，而复选框中用户可以任意选择多项，甚至全选。

请看下面的例子: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

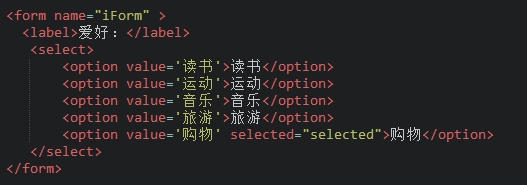
2、value：提交数据到服务器的值（后台程序PHP使用）

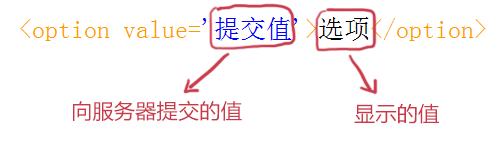
3、name：为控件命名，以备后台程序 ASP、PHP 使用

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

注意:同一组的单选按钮，name 取值一定要一致，这样同一组的单选按钮才可以起到单选的作用。

下拉列表在网页中也常会用到，它可以有效的节省网页空间。既可以单选、又可以多选。如下代码：

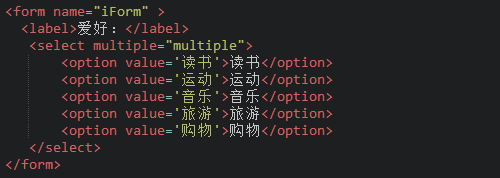


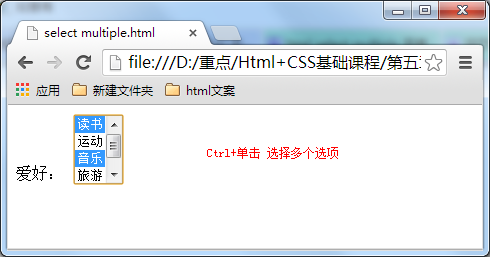


selected="selected"：设置selected="selected"属性，则该选项就被默认选中。

下拉列表也可以进行多选操作，在<select>标签中设置multiple="multiple"属性，就可以实现多选功能，在 widows 操作系统下，进行多选时按下Ctrl键同时进行单击（在 Mac下使用 Command +单击），可以选择多个选项。

如下代码：





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

当用户需要重置表单信息到初始时的状态时，比如用户输入“用户名”后，发现书写有误，可以使用重置按钮使输入框恢复到初始状态。只需要把type设置为"reset"就可以。

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

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。

CSS 样式代码插入的形式来看基本可以分为以下3种：内联式、嵌入式和外部式三种

1.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：<p style="color:red">这里文字是红色。</p>

2. 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。

如下面代码实现把三个<span>标签中的文字设置为红色：

<style type="text/css">

span{color:red;}

</style>

3. 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。

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（.），圆点是类选择符。

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。

如右侧代码：.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。

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：.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。

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：\* {color:red;}

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;}

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;}

样式优先级为：浏览器默认的样式 < 网页制作者样式 < 用户自己设置的样式，但记住!important优先级样式是个例外，权值高于用户自己设置的样式

p{color:red!important;}

注意：!important要写在分号的前面

文字排版--粗体

p span{font-weight:bold;}

文字排版--斜体

p a{font-style:italic;}

文字排版--下划线

p a{text-decoration:underline;}

文字排版--删除线

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

段落排版--缩进

p{text-indent:2em;}

2em的意思就是文字的2倍大小。

段落排版中起重要作用的行间距（行高）

属性（line-height）p{line-height:1.5em;}

段落排版--中文字间距、字母间距

设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; }

单词间距设置：使用 word-spacing 来实现。

h1{ word-spacing:50px; }

块状元素中的文本、图片设置居中样式可以使用text-align样式

例：text-align：center text-align：left text-align：right

设置display:block就是将元素显示为块级元素。

例：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。

块状元素也可以通过代码display:inline将元素设置为内联元素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。

4、内联元素之间有一个间距

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。

盒子模型一般是块级标签的代表部分，如div、ul、ol、p。

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)

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

3、边框方向

上 top

右 right

左 left

下 bottom

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)。

顺序一定不要搞混。

如下代码：div{padding:20px 10px 15px 30px;}

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;}

盒模型--边界：margin

padding在边框里，margin在边框外。

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）

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)

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。

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）。

1. 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）。

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）。

1. 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）。

行内元素 还是 块状元素 ，块状元素里面又分为定宽块状元素，以及不定宽块状元素。

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。

父元素和子元素：

div是“我想要在父容器中水平居中显示”这个文本的父元素。反之这个文本是div的子元素

(定宽块状元素：块状元素的宽度width为固定值。)

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。

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。(不定宽块状元素：块状元素的宽度width不固定。)

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
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" \o "忘了的小伙伴，让我们点击链接复习一下吧! ) 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

这一小节我们来讲一下第一种方法：

为什么选择方法一加入table标签? 是利用table标签的长度自适应性---即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。

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>我是第一行文本</li>

        <li>我是第二行文本</li>

        <li>我是第三行文本</li>

    </ul>

    </td></tr>

  </tbody>

</table>

</div>

css代码：

<style>

table{

border:1px solid;

margin:0 auto;

}

</style>

第二种方法：改变块级元素的 display 为 inline 类型（设置为 [行内元素](http://www.imooc.com/code/2049) 显示），然后使用 text-align:center 来实现居中效果。如下例子：

html代码：

<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{

text-align:center;

}

/\* margin:0;padding:0（消除文本与div边框之间的间隙）\*/

.container ul{

list-style:none;

margin:0;

padding:0;

display:inline;

}

/\* margin-right:8px（设置li文本之间的间隔）\*/

.container li{

margin-right:8px;

display:inline;

}

</style>

方法三：通过给父元素设置[float](http://www.imooc.com/code/2071" \o "忘了的小伙伴，让我们点击链接复习一下吧! )，然后给父元素设置 [position:relative](http://www.imooc.com/code/2074" \o "忘了的小伙伴，让我们点击链接复习一下吧! ) 和 left:50%，子元素设置 position:relative 和 left: -50% 来实现水平居中。

我们可以这样理解：假想ul层的父层（即下面例子中的div层）中间有条平分线将ul层的父层（div层）平均分为两份，ul层的css代码是将ul层的最左端与ul层的父层（div层）的平分线对齐；而li层的css代码则是将li层的平分线与ul层的最左端（也是div层的平分线）对齐，从而实现li层的居中。

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>

我们在实际工作中也会遇到需要设置垂直居中的场景，比如好多报纸的文章标题在左右一侧时，常常会设置为垂直居中，为了用户体验性好。

这里我们又得分两种情况：父元素高度确定的单行文本，以及父元素高度确定的多行文本。

本节我们先来看第一种父元素高度确定的单行文本, 怎么设置它为垂直居中呢？

**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](http://www.imooc.com/code/2083" \o "忘了的小伙伴，让我们点击链接复习一下吧!)高度一致来实现的。(height: 该元素的高度，line-height: 顾名思义，行高（行间距），指在文本中，行与行之间的 基线间的距离 )。

line-height 与 font-size 的计算值之差，在 CSS 中成为“行间距”。分为两半，分别加到一个文本行内容的顶部和底部。

这种文字行高与块高一致带来了一个弊端：当文字内容的长度大于块的宽时，就有内容脱离了块。

如下代码：

<div class="container">

hi,imooc!

</div>

css代码：

<style>

.container{

height:100px;

line-height:100px;

background:#999;

}

</style>

父元素高度确定的多行文本、图片等的竖直居中的方法有两种：

方法一：使用插入 [table](http://www.imooc.com/code/292)  (包括tbody、tr、td)标签，同时设置 vertical-align：middle。

css 中有一个用于竖直居中的属性 vertical-align，在父元素设置此样式时，会对inline-block类型的子元素都有用。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>

</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。

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（设置为表格单元显示），激活 vertical-align 属性，但注意 IE6、7 并不支持这个样式, 兼容性比较差。

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>

</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而且这样修改display的block变成了table-cell，破坏了原有的块状元素的性质。

**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](http://www.imooc.com/code/2073)

 2. float : left 或 [float:right](http://www.imooc.com/code/2071" \o "忘了的小伙伴，让我们点击链接复习一下吧! )

简单来说，只要html代码中出现以上两句之一，元素的display显示类型就会自动变为以 display:inline-block（[块状元素](http://www.imooc.com/code/2048)）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](http://www.imooc.com/code/2049) ，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>