**HTML**

**HTML详解**

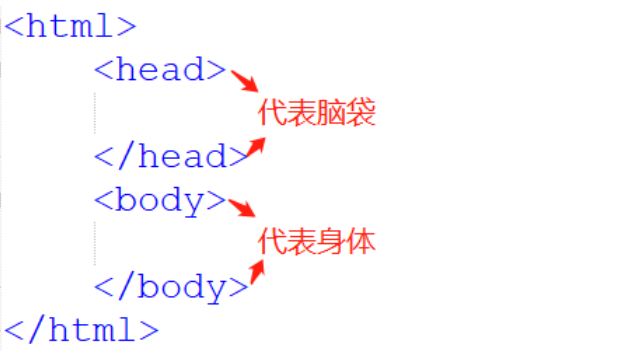
1. HTML文档 = 网页
2. 什么是HTML？--Hyper Text Markup Language--超文本标记语言
3. 超文本：超级文本
4. 标记语言：这门语言由**标签（元素）**堆砌而成
5. 标签：把需要显示在网页中的内容标记起来，并把这些内容显示成特殊的样式显示在网页上面
6. 网页文件格式：.html或.htm
7. 浏览器在解析一段HTML代码进行渲染的时候是通过逐行解析的
8. 当浏览器所接收到的一段HTML代码中包含着一些对外部文件的引用，那么服务器所发送给浏览器的响应报文中，其实是并不包含着这些所引用的外部文件的，只有当浏览器接收到服务端所返回的HTML代码并进行逐行解析渲染的时候，浏览器才会逐个对它们向服务端发送一次新的请求去索取，如下图：



**HTML中的标签的注意事项**

1. HTML标签中，**成对的标签**都是有始有终的，而**单独的标签**则是从自己开始从自己结束
2. HTML中的标签可以嵌套使用
3. HTML中的标签不区分大小写
4. HTML文件是**流式加载**的，也就是遵循由上至下的的规律，如果希望一段代码先被执行，那么可以写在后被执行代码的上面
5. HTML中指定文件地址都是用**“/”**来分割，区别于系统中的地址写法
6. HTML中所访问的文件，比如说：<img src = "文件路径">，都必须要在**当前所请求的网站文件的根目录下**（也可以放在其他服务器上，但这涉及到分部署存储的问题，不在这篇讲解的范围内），因为浏览器有一种安全风险保护机制，只能够访问当前请求网站文件根目录下的文件

**HTML的基本框架（所有网页都需要这种框架）**



**HTML标签中的转义符**

1. 空格：&nbsp 微信截图_20180117002624 微信截图_20180117002615
2. >：&gt 微信截图_20180117002824 微信截图_20180117002849
3. <：&lt 微信截图_20180117002930 微信截图_20180117002938
4. &：&amp 微信截图_20180117003004 微信截图_20180117003015

**HTML标签的属性简介**

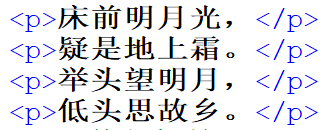
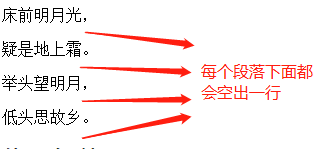
1. 表现各标签的特征
2. 如：<font size=xxx/>中的size即为属性
3. 属性都是以 属性名 = “值”的形式出现
4. 属性的值建议用引号括起来
5. 属性建议均以键值对的形式括起来
6. 一个标签可有多个属性，用空格分开

**HTML中的一些常用的标签**

------------------------------------------------------------------------------------------------------------------------------

1. **段落标签：<p></p> （成对的标签） 写在<body></body>标签中**

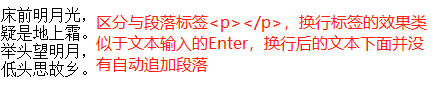
如果我们不添加代码实现HTML中的文本内容的换行，那么在网页中显示的内容是不会换行的

------------------------------------------------------------------------------------------------------------------------------

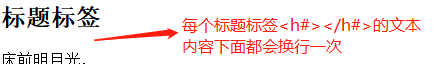
1. **换行标签：<br/> （单独的标签） 写在<body></body>标签中**

如果我们不添加代码实现HTML中的文本内容的换行，那么在网页中显示的内容是不会换行的

** **

------------------------------------------------------------------------------------------------------------------------------

1. **标题标签：<h#></h#>：#=控制标题的大小=1~6 （成对的标签） 写在<body></body>标签中**

**微信截图_20180116190917 **

------------------------------------------------------------------------------------------------------------------------------

1. **注释标签：<!-- （需要注释的文本）--> （单独的标签） 写在<body></body>标签中**



------------------------------------------------------------------------------------------------------------------------------

1. **图片标签：<img（设置图片标签的属性）/> （单独的标签） 写在<body></body>标签中**

**图片标签的属性：**

1. src：图片的路径（绝对路径/相对路径） 微信截图_20180116191644

-> 每当动态的修改了src属性，如果修改后的值于修改前的值不同，浏览器就会针对这个新的src发出一次HTTP请求来接收指定目录下的数据去改变img标签的图片显示（每当我们请求一个页面的时候，浏览器会把当前所请求的页面上的文件缓存在系统的一个临时目录下，图片也不例外，图片的保存方式对应着所请求图片的URL地址，所以当我们动态修改img标签的src属性的时候，浏览器会根据所改变的src的值去缓存目录上查找是否存在已缓存该url所对应的文件，如果有就直接拿出来用，没有的话就发出一次新的HTTP请求去获取该图片并缓存下来，由此可以推断出，浏览器在这一方面所做出的判断是只认src修改后和修改前是否相同再去考虑要不要发出HTTP请求去获取图片文件的。需要了解的是，旧版chrome浏览器是不存在这一问题的，就算动态改变了img标签src属性的值和改变前的值是相同的，chrome浏览器也是一样会发出HTTP请求去获取图片的，而新版chrome浏览器为了统一性已经解决了这一问题），需要注意的是，发出的这次HTTP请求所接收到的数据**仅仅是针对该img标签图片的获取和更换的**

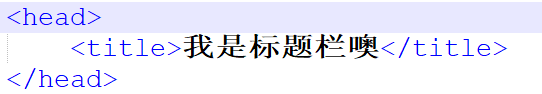
1. height：图片的高度（px为单位） 微信截图_20180116191650
2. width：图片的宽度（px为单位） 微信截图_20180116191658
3. title：当鼠标移动到图片的时候显示的文本内容 微信截图_20180116191705
4. alt：当图片无法加载的时候显示的文本内容 
5. border：在图片四周显示边框，单位为：px 

微信截图_20180116192024



------------------------------------------------------------------------------------------------------------------------------

1. **标签页内容标签：<title></title> （成对的标签） 写在<head></head>标签中**

 微信截图_20180116192424

------------------------------------------------------------------------------------------------------------------------------

1. **标尺线标签：<hr（设置标尺线标签的属性）/> （单独的标签） 写在<body></body>标签中**

**标尺线标签的属性：**



微信截图_20180117005932 

------------------------------------------------------------------------------------------------------------------------------

1. **字体标签：<font（设置字体标签属性）></font>（成对的标签）写在<body></body>标签中**

**字体标签的属性：**

1. size：设置字体的大小，范围为：1~7 微信截图_20180117010349
2. face：设置字体的样式 微信截图_20180117010355
3. color：设置字体的颜色 微信截图_20180117010343

微信截图_20180117010409 微信截图_20180117010422

------------------------------------------------------------------------------------------------------------------------------

1. **定义粗体字体标签：<b></b>（成对的标签）写在<body></body>标签中**

**微信截图_20180117012517 微信截图_20180117012524**

------------------------------------------------------------------------------------------------------------------------------

1. **定义斜体字体标签：<i></i>（成对的标签）写在<body></body>标签中**

**微信截图_20180117012532 微信截图_20180117012540**

------------------------------------------------------------------------------------------------------------------------------

1. **定义打字机文本标签：<tt></tt>（成对的标签）写在<body></body>标签中**

**微信截图_20180117012551 微信截图_20180117012558**

------------------------------------------------------------------------------------------------------------------------------

1. **定义下划线文本标签：<u></u>（成对的标签）写在<body></body>标签中**

**微信截图_20180117012605 微信截图_20180117012612**

------------------------------------------------------------------------------------------------------------------------------

1. **定义删除线文本标签：<s></s>（成对的标签）写在<body></body>标签中**

**微信截图_20180117012621 微信截图_20180117012627**

------------------------------------------------------------------------------------------------------------------------------

1. **定义文本上标标签：<sup></sup>（成对的标签）写在<body></body>标签中**

**微信截图_20180117013124 微信截图_20180117013134**

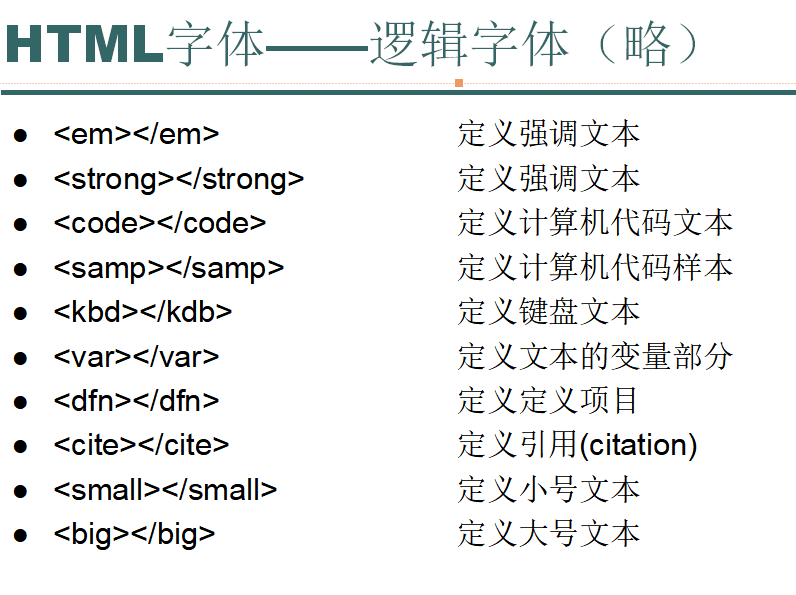
------------------------------------------------------------------------------------------------------------------------------

1. **定义文本下标标签：<sub></sub>（成对的标签）写在<body></body>标签中**

**微信截图_20180117013051 微信截图_20180117013100**

------------------------------------------------------------------------------------------------------------------------------

1. **定义逻辑字体的标签： 略 （成对的标签）写在<body></body>标签中**

****

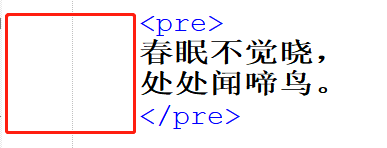
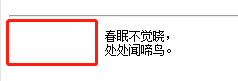
------------------------------------------------------------------------------------------------------------------------------

1. **预定义格式标签： <pre></pre> （成对的标签）写在<body></body>标签中**

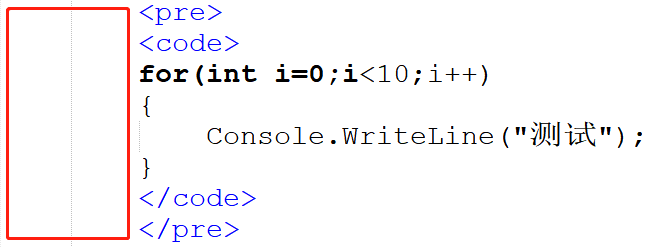
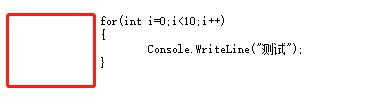
**预定义格式标签的详解：**

1. 输出在网页中的格式会保留在代码中的空格和换行
2. 与标签-十六中的<code></code>标签一起使用能够在网页中更精确地呈现代码的样式

**预定义格式标签的普通代码和显示结果**

**预定义格式标签与<code></code>标签配合使用的代码和显示结果**

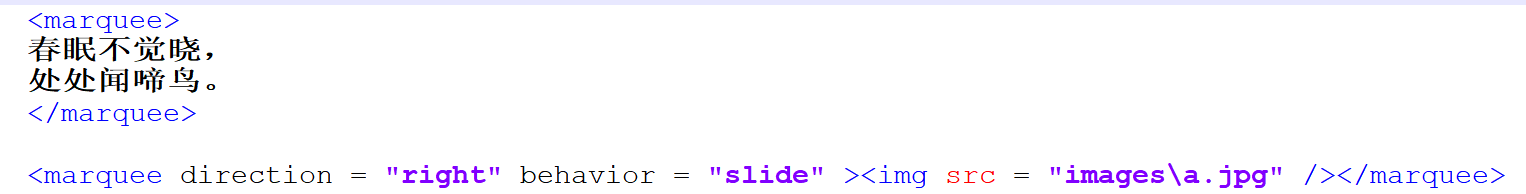
 

------------------------------------------------------------------------------------------------------------------------------

1. **能让标签移动的标签： <marquee></marquee> （成对的标签）写在<body></body>标签中**

**<marquee>标签的属性：**

1. direction：设置标签的移动方向（left、right、down、up） 微信截图_20180117022105
2. behavior：设置标签的移动模式（scroll、altermate、slide） 微信截图_20180117022847

------------------------------------------------------------------------------------------------------------------------------

1. **超链接标签： <a></a> （成对的标签）写在<body></body>标签中**

**链接标签的属性：**

1. name：设置该标签的一个标识 微信截图_20180118001243
2. href：设置提交请求的路径、标识 微信截图_20180118001150

-> a标签所提交的请求方式为GET请求，如果我们想要通过a标签把一些数据提交至服务器当中，我们可以通过在href所指定的路径的末尾处来书写提交的参数和数据来达到效果（GET请求所提交的参数实体会在HTTP请求报文中的请求行的URL中体现出来）

1. target：设置该标签打开新页面的方式（\_self=不打开新标签页、\_blank=打开新的标签页） 微信截图_20180118001316
2. 链接可以是一个字、一个词或是一句话，也可以是一副图片

**链接标签的几种使用方法：**

**用法一：跳转至某个网页**

微信截图_20180118001604



**用法二：跳转至另一个地方显示JPG图片**

微信截图_20180118022621 （相对路径）



**用法三：跳转至当前HTML文件中的某个指定位置（#加上A标签的标识(name)，就是能跳到指定位置）**

微信截图_20180118001810

微信截图_20180118001842



**用法四：跳转至另外一个HTML文件的某个指定的位置**

微信截图_20180118023727（相对路径）

微信截图_20180118023737（相对路径）



**用法五：跳转至发送邮件页面（系统默认邮件）**

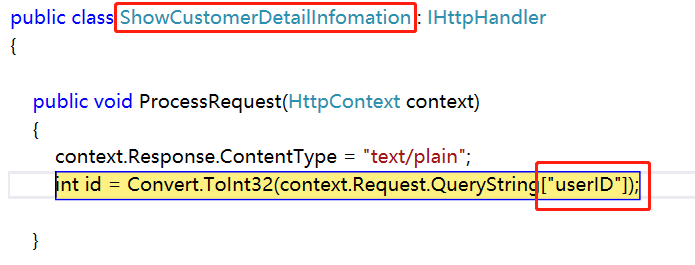
mailto:收件人的邮箱名?subject=主题名&body=邮件内容

微信截图_20180118211744



**用法六：通过在href数据的末尾处按照HTTP GET方式的参数书写格式把数据提交至服务器当中**

微信截图_20190206164341



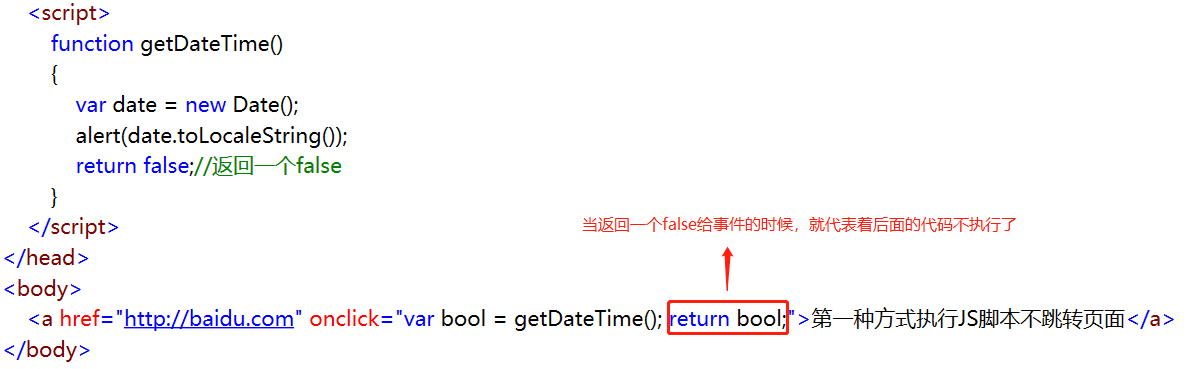
**用法七：、跳转至本HTML页面的顶端**





**用法八：两种方法设置<a></a>标签停止GET请求的发送(不跳转页面)，而是执行一段JavaScript脚本**

1. 使JavaScript定义的函数返回一个false值，并且<a></a>的标签也返回JavaScript函数的值，也就是返回false

****

1. 直接在<a>></a>标签的href指定只执行写在里面的JavaScript脚本不进行链接的跳转

****

------------------------------------------------------------------------------------------------------------------------------

1. **HTML文档主体的标签： <body></body> （成对的标签）**

**HTML文档主体标签的属性：**

1. text：设置该标签里面的所有的文本的颜色  微信截图_20180118214950
2. link：设置<a></a>链接标签的颜色 微信截图_20180118215600 微信截图_20180118215547
3. alink：设置<a></a>链接标签点击时的颜色 微信截图_20180118215913 
4. vlink：设置<a></a>链接标签点击后的颜色 微信截图_20180118220351 
5. bgcolor：设置网页的背景颜色

微信截图_20180118220501



1. background：设置网页的背景图片（平铺）

微信截图_20180118220703（相对路径）



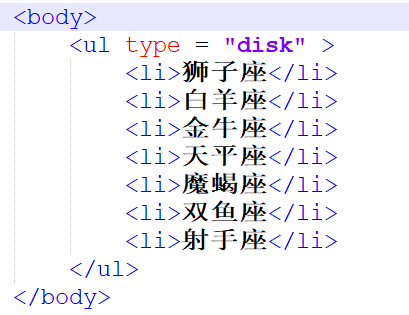
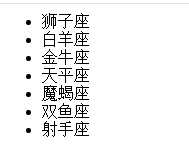
------------------------------------------------------------------------------------------------------------------------------

1. **HTML常用的三种列表标签（成对的标签）写在<body></body>标签中**
2. **无序列表：**
3. <ul></ul>：无序列表**（成对的标签）**
4. <li></li>：列表里面的内容（嵌套于<ul></ul>标签）**（成对的标签）**

**属性：**

1. type：设置列表的样式（square(方块)、circle(小圆圈)、 disk(小圆点)）微信截图_20180118224233 微信截图_20180118224218

**代码与结果：**

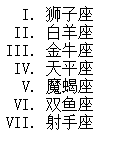
 

1. **有序列表：<ol><li>第一</li><li>第二</li></ol>**
2. <ol></ol>：有序列表**（成对的标签）**
3. <li></li>：列表里面的内容（嵌套于<ol></ol>标签）**（成对的标签）**

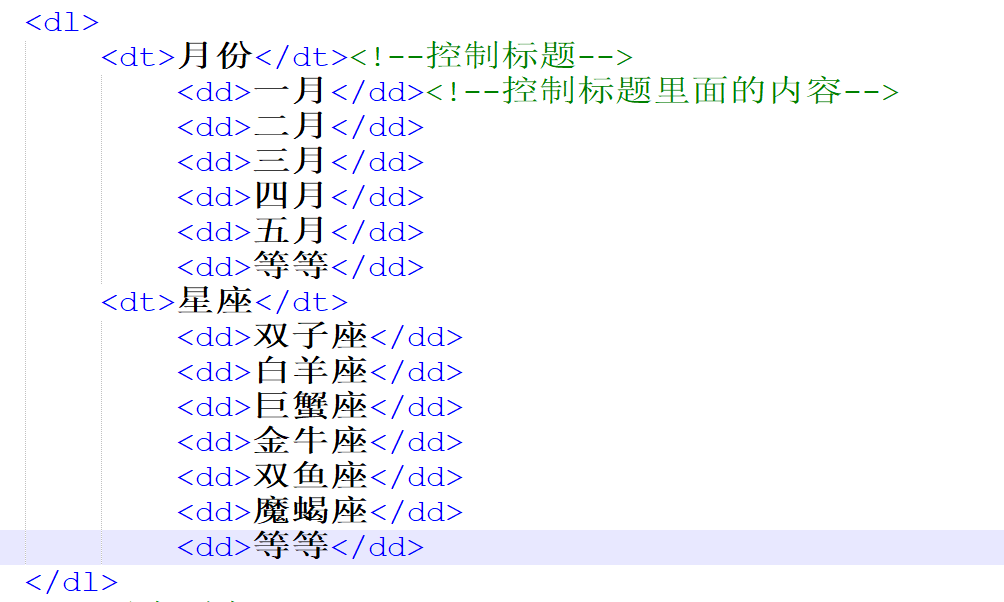
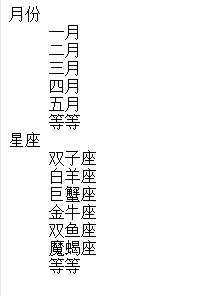
**属性：**

1. type：设置列表的样式（“A”、“1”、……） 微信截图_20180118224543 

**代码与结果：**

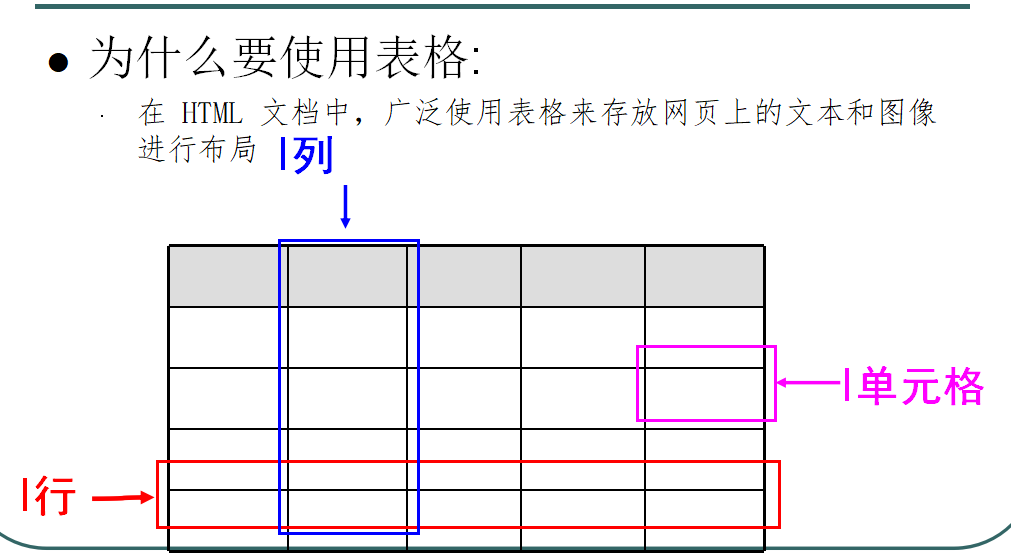
 

1. **自定义列表：<dl><dt>列表标题</dt><dd>列表标题里面的内容</dd></dl>**
2. <dl></dl>：自定义列表**（成对的标签）**
3. <dt></dt>：列表标题（嵌套于<dl></dl>标签）**（成对的标签）**
4. <dd></dd>：标题里面的内容（嵌套于<dl></dl>标签）**（成对的标签）**

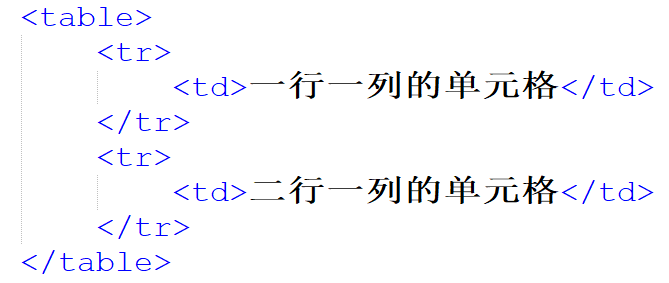
 

1. **表格标签 <table><tr><td></td></tr></table>写在<body></body>标签中**

**表格的概念：**



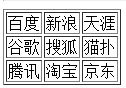
**表格标签由三个标签组成：**



1. <table></table>：控制表格标签**（成对的标签）**
2. <tr></tr>：控制表格中每一行的标签（嵌套于<table></table>标签）**（成对的标签）**
3. <td></td>：控制单元格的标签（嵌套于<tr></tr>标签）**（成对的标签）**

**<table></table>标签的属性：**

1. border：设置表格的边框大小（单位为px）

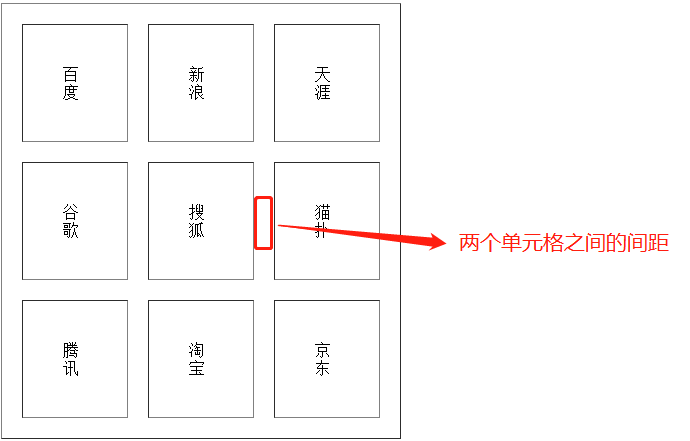
微信截图_20180119000212 

1. width/height：设置表格的宽和高（单位为px）

微信截图_20180119000648

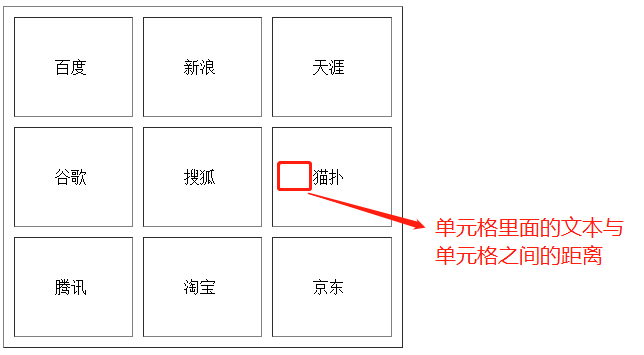
1. cellspacing：设置单元格之间的间距（单位为px）



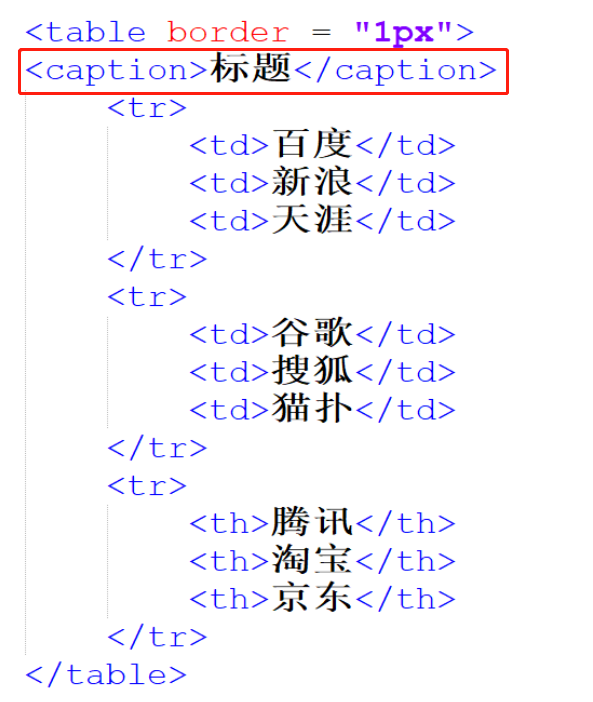


1. cellpadding：设置单元格里面的文本与单元格之间的距离（单位为px）

微信截图_20180119001304



1. <caption></caption>：设置表格的标题内容（嵌套于<table></table>标签中）



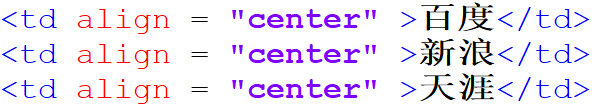


1. 还有很多不常用的如下图



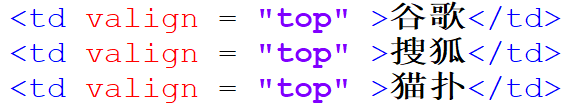
**<td></td>标签的属性：**

1. align：控制单元格文本内容横向的位置（left、center、right）



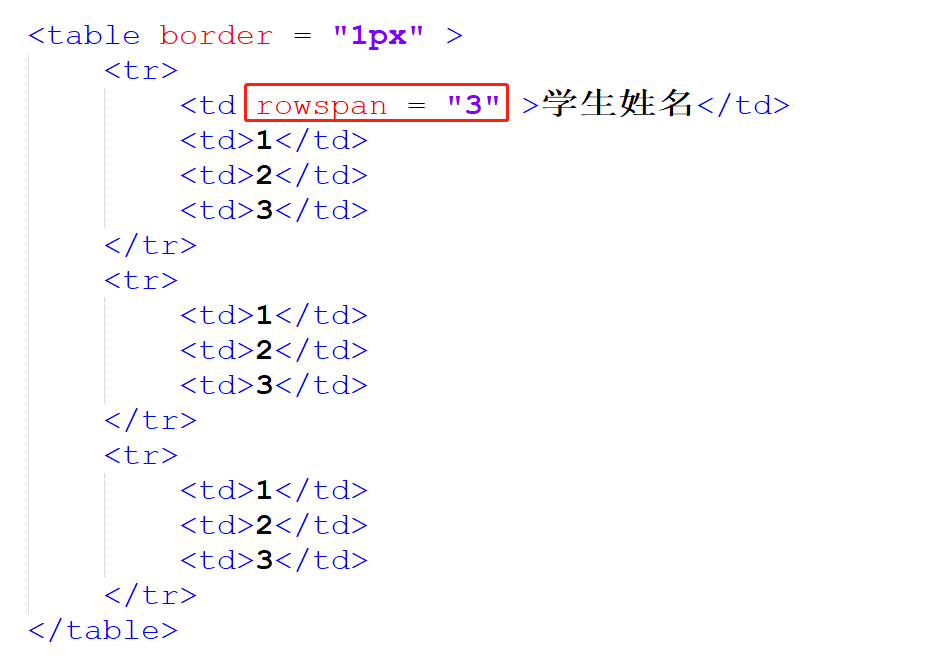


1. valign：控制单元格文本内容纵向的位置（top、middle、bottom）



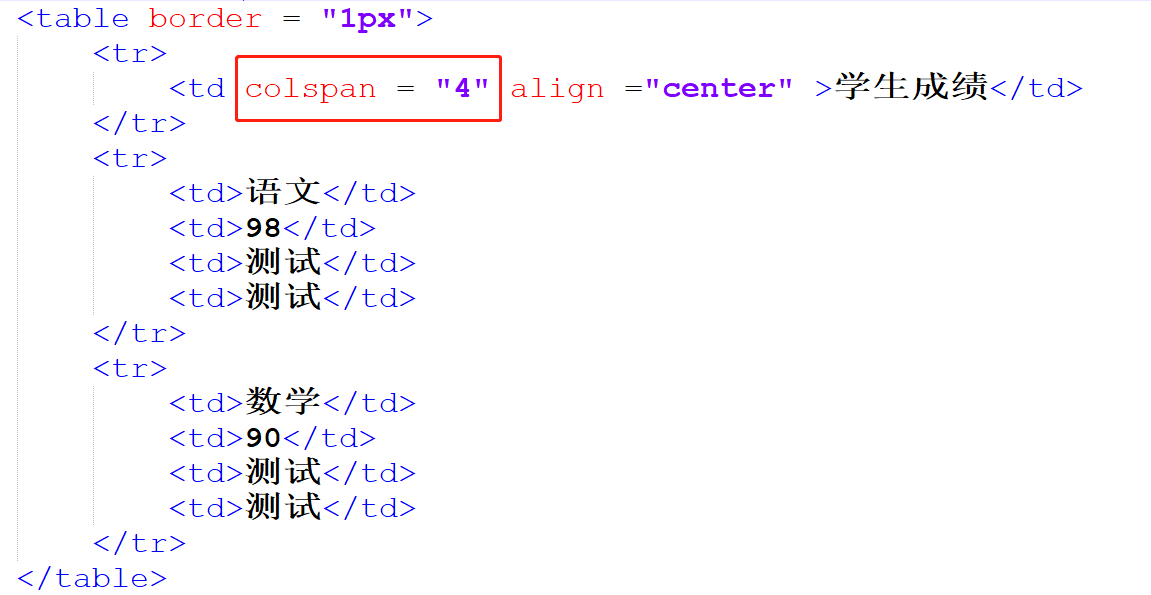


1. rowspan：设置该单元格所跨站的行数（数跨行跨列的表格里有多少行多少列，要数最多出现的行数和列数）



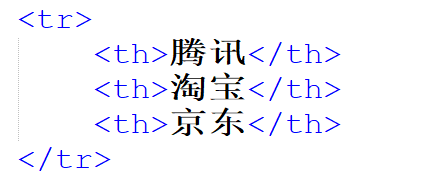
（3行4列）

1. colspan：设置该单元格所跨站的列数（数跨行跨列的表格里有多少行多少列，要数最多出现的行数和列数）



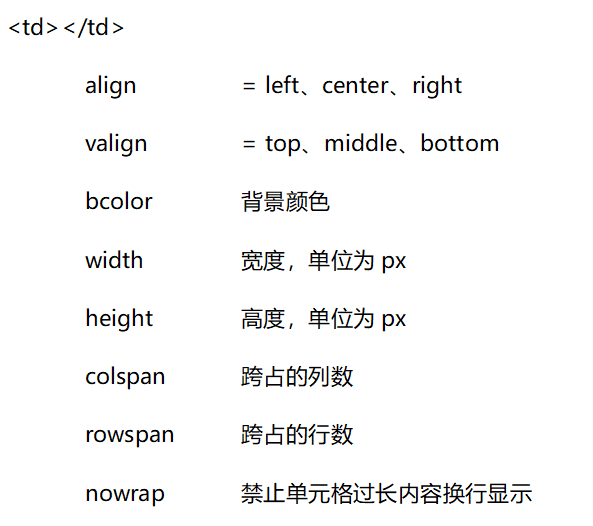
（3行4列）

1. <th></th>：粗体居中（替换掉<td></td>）



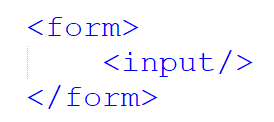


1. 还有很多不常用的如下图



------------------------------------------------------------------------------------------------------------------------------

1. **表单标签：<form><input/></form> 写在<body></body>标签中**

****

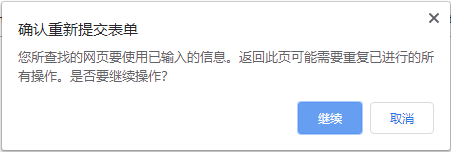
**表单标签的详解：**

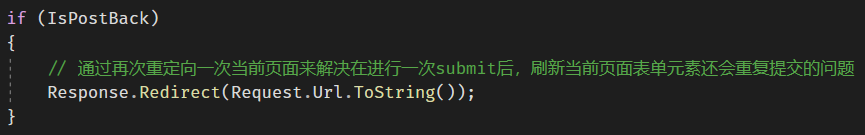
1. <form></form>：表单标签**（成对的标签）**
2. <input/>：表单控件(元素)标签，**type属性决定了该控件（元素）的样式（单独的标签）**
3. HTML表单标签的作用在于向服务器传输数据
4. HTML的表单由两各部分的标签构成，其中包含着表单中的控件标签的<form></form>作用在于向服务器发送数据，而嵌套在里面的<input/>标签则是表单中的一些控件（类似于winform中的控件），显示于网页上并作用于收集用户的输入等等
5. 表单所提交的数据只针对 [表单元素] ，也就是说当我们在一个form标签下写了一个不是表单元素的HTML标签，如：div、span，那么当我们在提交表单的时候，div和span就不会列入提交队列中

**<form></form>标签的属性：**

1. action：设置数据需要提交到服务器的地址 微信截图_20180120000750

-> 当我们设置了该属性的值为当前页面后，当我们进行了一次表单的submit，表单会跳转回该页面，但是这时候会有一种问题，虽然在submit后跳转回该页面是通过get请求来完成，也就是说当前界面上表单元素的数据都被清空了，但是其实它内部还会保留着我们在submit前所写入的数据的，这时候如果我们进行当前页面的刷新的话，就又会自动触发一次submit的动作，并把在第一次submit前所写入的表单元素的数据一并提交上去，我们可以通过在提交当前页面后，当前页面的后端的代码重定向一次当前页面的URL地址即可解决这一问题





1. method：设置提交数据到服务器的方式，get（提交url形式）、post（通过报文提交） 微信截图_20180120000757
2. enctype：设置提交数据的编码格式，他可以控制请求体中数据的组织格式

-> 如果需要提交文件，则必须要把编码格式设置成：multipart/form-data

1. application/x-www-form-urlencoded：默认的编码格式，是最常见的提交数据的编码格式，该编码格式下所提交的数据会以键值对的形式存在，并且参数以 [&] 来分割



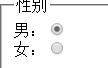
1. multipart/form-data：一种提交实体数据的编码格式，当以这种编码格式提交数据后，请求首部字段中会多出一个 [boundary] 的字段，请求体中的数据则会以该字段的值来作为分隔符去分隔开来，并且还会把表单元素中文件域所指定的文件文件名读取出来，转换成文件流放入请求体当中



1. text/plain：比较少用的编码格式，该编码格式下所提交的数据会以文本的形式体现出来，并且参数会以换行的结束作为分割



**<input/>标签的属性：**

1. value：设置该控件的初始值 微信截图_20180120194924 微信截图_20180120194937
2. size：设置该控件的初始宽度 微信截图_20180125165328
3. maxlength：设置该文本控件中输入的字符最大数 微信截图_20180120001741
4. checked：设置该单选、多选控件是否被选中 微信截图_20180120194805 

-> 在HTML标签中控制disabeld属性的值的时候，不管写什么值，只要设置了checked属性，控件都是被选中的，只有去掉这个属性，控件才不被选中

-> 在JS DOM 中就可以通过true来设置控件不可用，通过false或 "" 来设置控件可用**（jQuery也可以实现）**

1. 设置控件被选中



1. 设置控件不被选中



1. disabled：指示控件是否可用

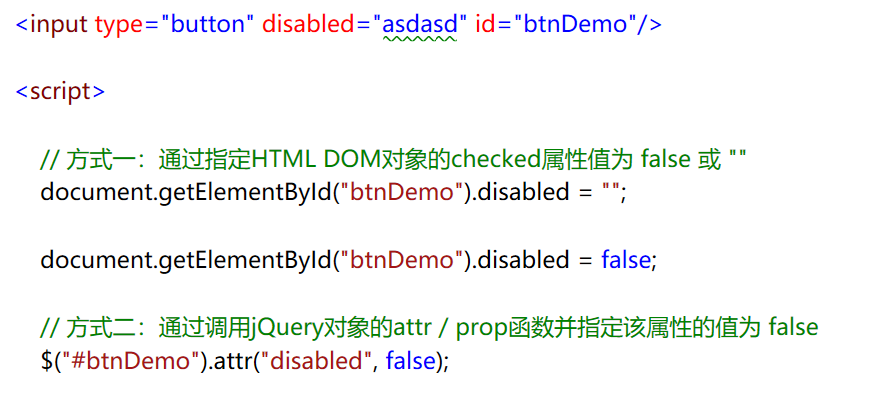
-> 在HTML标签中控制disabeld属性的值的时候，不管写什么值，只要设置了disabled属性，控件都是不可用的，只有去掉这个属性才为可用状态

-> 在JS DOM 中就可以通过true来设置控件不可用，通过false或 "" 来设置控件可用**（jQuery也可以实现）**

1. 设置控件不可用

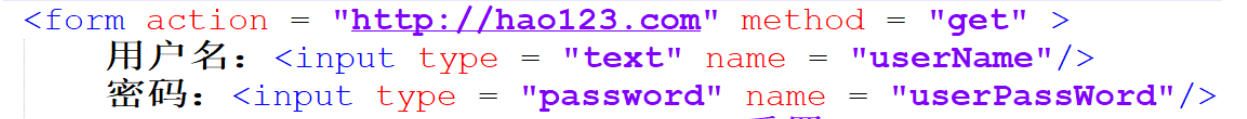


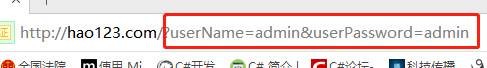
1. 设置控件可用



1. name：设置该控件的标识，有两个作用，作用如下：
2. 服务器接收数据的时候能够查找到

所提交的表单元素必须要有其name属性，否则当我们在客户端提交数据后，HTTP请求报文是不会带有表单元素的数据的

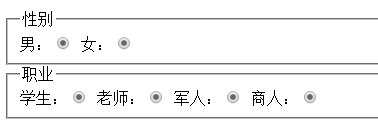




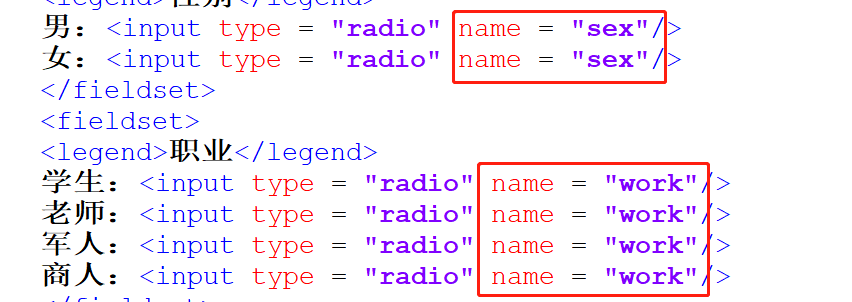
1. 控件之间的分组

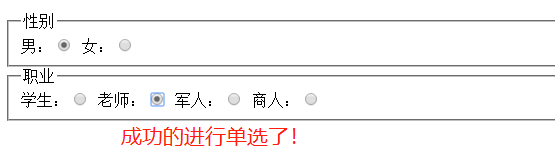
定义一个单选按钮的控件（type = radio），如果不对不同类型的<input/>标签的name属性进行赋值（控件之间的分组），我们就不能进行单选，这样就失去了单选的意义





当我们对不同类型的单选框进行分组后（对name属性进行赋值），我们就可以进行单选了





1. type：设置表单控件标签的具体类型，决定着该控件的具体样式
2. type = text：单行文本输入框

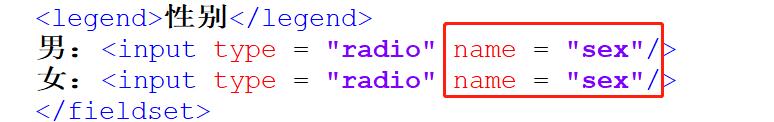
微信截图_20180120184629微信截图_20180120184726

1. type = password：密码框

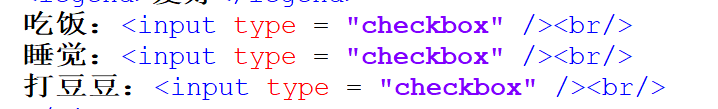
微信截图_20180120184715 微信截图_20180120184726

1. type = radio：单选按钮

**注意：需要对不同类型的单选框之间的name进行赋值（分组），否则不能单选**

微信截图_20180120193406

1. type = checkbox：复选按钮



1. type = reset：重置按钮

微信截图_20180120194309 微信截图_20180120194331

1. type = submit：提交数据按钮

微信截图_20180120194301 微信截图_20180120194323

1. type = file：文件上传域，如果想要正确的提交所选定的文件的数据，则需要设置 [form] 标签的 [enctype] 属性为：multipart/form-data4

微信截图_20180120213930 微信截图_20180120213920

1. type = hidden：隐藏域

微信截图_20190130205130

1. type = button：按钮控件

微信截图_20180120223838

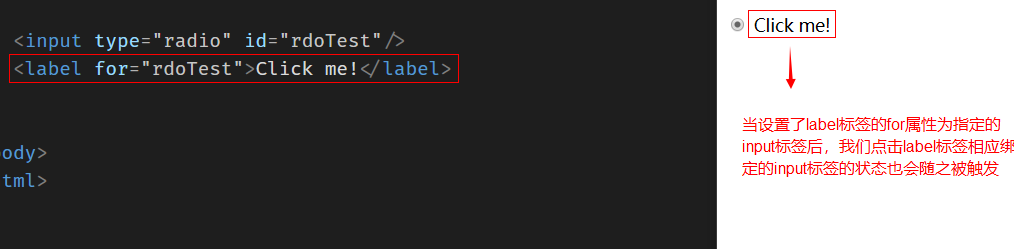


------------------------------------------------------------------------------------------------------------------------------

1. **标签：<label></label>**
2. 该标签主要用于与一些input标签进行绑定关系，当我们设置了该标签的绑定关系为某一个input标签的时候，这个标签的响应信息也会为所绑定的input标签所触发

**<label></label>标签的属性**

1. for：用于指定该label所绑定的input标签的ID属性的值



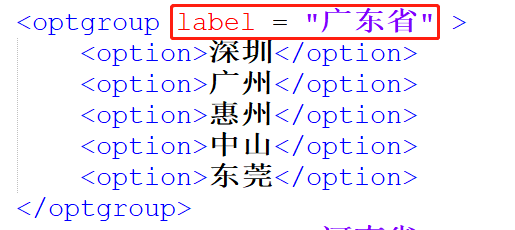
------------------------------------------------------------------------------------------------------------------------------

1. **预选下拉列表标签：<select><optgroup><option></option></optgroup></select> 写在<body></body标签中>，配合<form></form>标签使用**
2. <select></select>：预选下拉列表标签（成对的标签）
3. <option></option>：预选下拉列表里面的元素标签（成对的标签）
4. <optgroup></optgroup>：预选下拉列表里面的元素的标题标签（成对的标签）

**<select></select>标签的属性：**

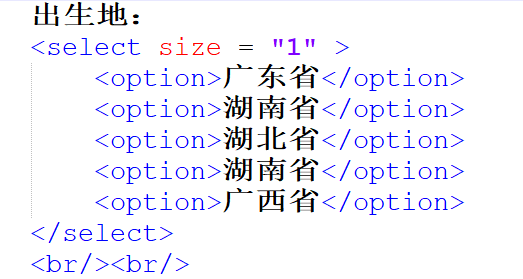
1. size：定义该下拉列表控件的大小，单位为number 微信截图_20180120220312 
2. value：该属性需要通过DOM或获取该标签的对象，然后通过value属性可以当前下拉列表标签select选择项的值
3. multiple：把下拉框设置成表框的样式，并且可以实现多选 

**<optgroup></optgroup>标签的属性：**

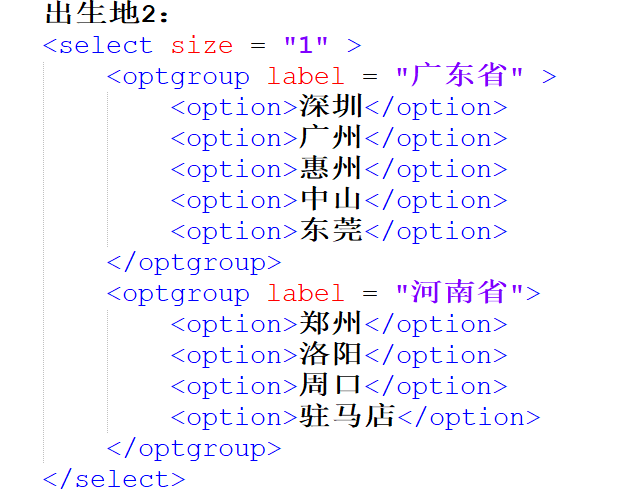
1. label：设置标题的名称 

**预选下拉列表标签的两种用法，用法如下：**

**用法一：没写<optgroup></optgroup>的标题**

****

**用法二：写了<optgroup></optgroup>的标题**

****

**总结：**

****

------------------------------------------------------------------------------------------------------------------------------

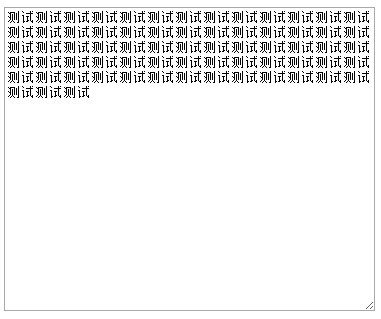
1. **多行文本编辑标签：<textarera></textarera> 写在<body></body标签中>，配合<form></form>标签使用**

**多行文本编辑标签的属性：**

1. rows：设置文本行（高）的大小，单位为百分比 微信截图_20180120222511
2. cols：设置文本列（宽）的大小，单位为百分比 微信截图_20180120222523
3. disabled：设置该标签是否为可用状态 微信截图_20180120222552
4. value：该属性需要通过DOM获取该标签的对象，然后通过value属性可以获取文本域编辑控件的值

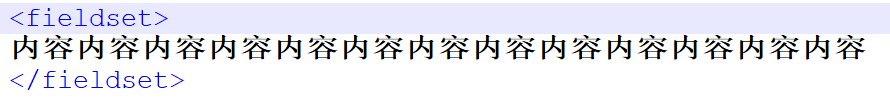
**多行文本编辑标签的代码和实现图：**

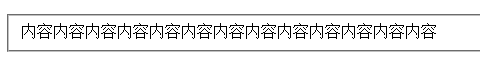
**微信截图_20180120222905**

****

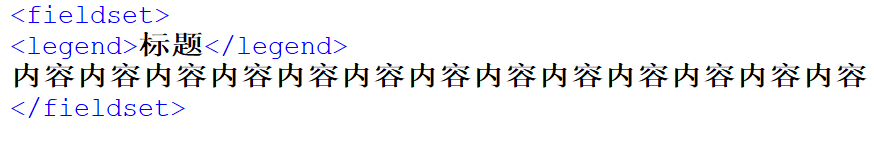
------------------------------------------------------------------------------------------------------------------------------

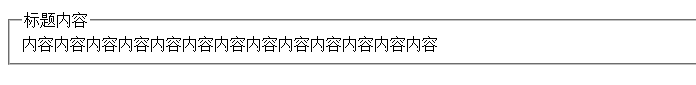
1. **内容分层处理的标签：<fieldset><legend></legend></fieldset> 写在<body></body>标签中**
2. <fieldset></fieldset>：给标签里面的内容分层





1. <legend></legend>：分层内容的标题





------------------------------------------------------------------------------------------------------------------------------

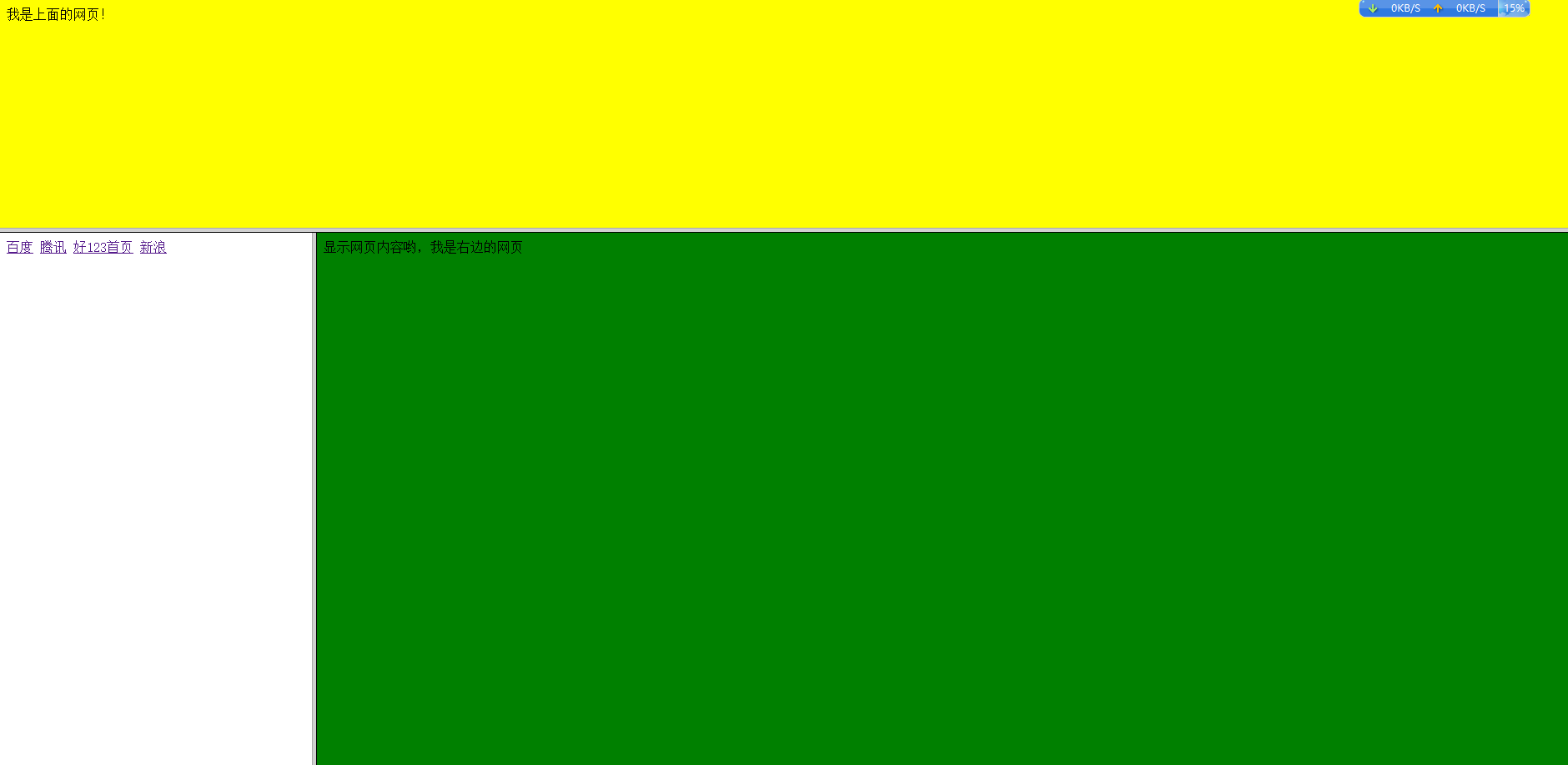
1. **网页分层框架标签： <frameset><frame/></frameset> 写在承载页的代码中，需替换掉<body></body>标签(不用写<body></body>标签)**

**分层框架标签的拆解：**

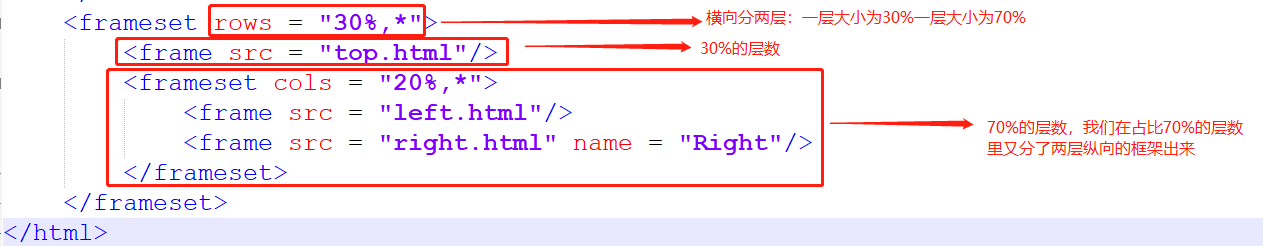
1. <frameset></frameset>标签：控制网页间分层的格式
2. <frame/>标签：设置分层的网页文件

**分层框架标签的注意事项：**

1. 我们在做基于互联网的WEB开发的时候，应当尽量避免使用网页分层框架标签。这其实和各大搜索引擎内置的 [蜘蛛爬虫] 的工作有关，我们都知道现在的用户在上网想浏览一个页面的时候，都是通过搜索引擎来查找想要浏览网页的关键字然后继续浏览的，而挂在搜索引擎关键字所显示出来的结果栏目其实都是根据 [蜘蛛爬虫] 预先爬好的网页然后再根据用户输入的搜索关键字去筛选呈现出来的，这时候就造成了一个问题，因为 [蜘蛛爬虫] 的工作原理是查找当前页面的HTML源代码里面相关的内容，并且会智能的爬取当前HTML源代码中相关<a></a>标签所指定src网页地址中的相关内容，层层进行爬取，而当我们使用了网页分层框架标签的时候，在分层标签(<frame/>或<iframe></iframe>)内显示网页的HTML源代码并不会出现在承载页的HTML源代码里面，这时候 [蜘蛛爬虫] 就不会再进下一层查找相关的内容，进而然的降低了我们的网站在发布至互联网后的SEO(网页搜索引擎优化)、降低了访客量
2. <frameset></frameset>标签和<frame/>标签必须要脱离了<body></body>标签才能使用，且<frame/>标签必须要承载在<frameset></frameset>标签之内
3. 除了<frame/>标签外，我们还能够使用<iframe></iframe>标签，其功能是等同于<frame/>标签的，但需要注意的是，<iframe></iframe>标签的使用是不需要脱离<body></body>标签的，并且<iframe></iframe>标签是不需要承载在<frameset></frameset>标签之内的，以下只讲解<frame/>标签的一些属性和功能
4. 网页框架分层，并不是由一个网页分成需要的部分，而是如以下网页，需要由三个分层页面文件、一个承载页面文件来组成：



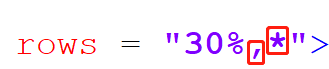
1. 网页的分层只能够纵向分层和横向分层，当我们需要多次分层的时候（如上图所示的网页），逻辑如下图：



（先横向分两层，一层为30%一层为70%，然后我们再70%的分层页中再进行一次分层）

**<frameset></frameset>标签的属性：**

1. rows：设置承载页由横向分层，单位为百分比，当需要横向分多层的时候需要“，”来分割，表示不同分层的大小，“\*”代表网页剩下的分层大小



（横向分两层，第一层为30%的大小，第二层为70%的大小）

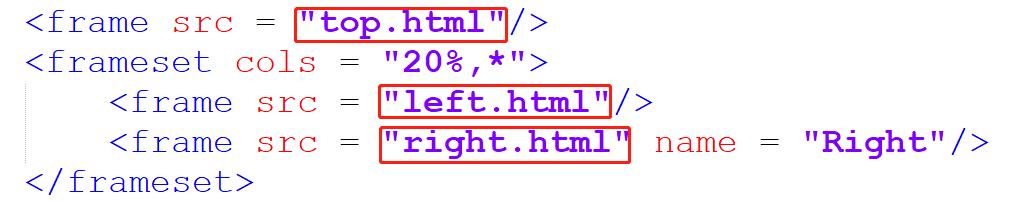
1. cols：设置承载页由纵向分层，单位为百分比，当需要纵向分多层的时候需要“，”分割，表示不同分层的大小，“\*”代表网页剩下的分层大小

微信截图_20180122023253

（纵向分两层，第一层为20%的大小，第二层为80%的大小）

**<frame/>标签的属性：**

1. src：设置分层网页的路径（设置哪一个网页文件作为承载页中的分层页）

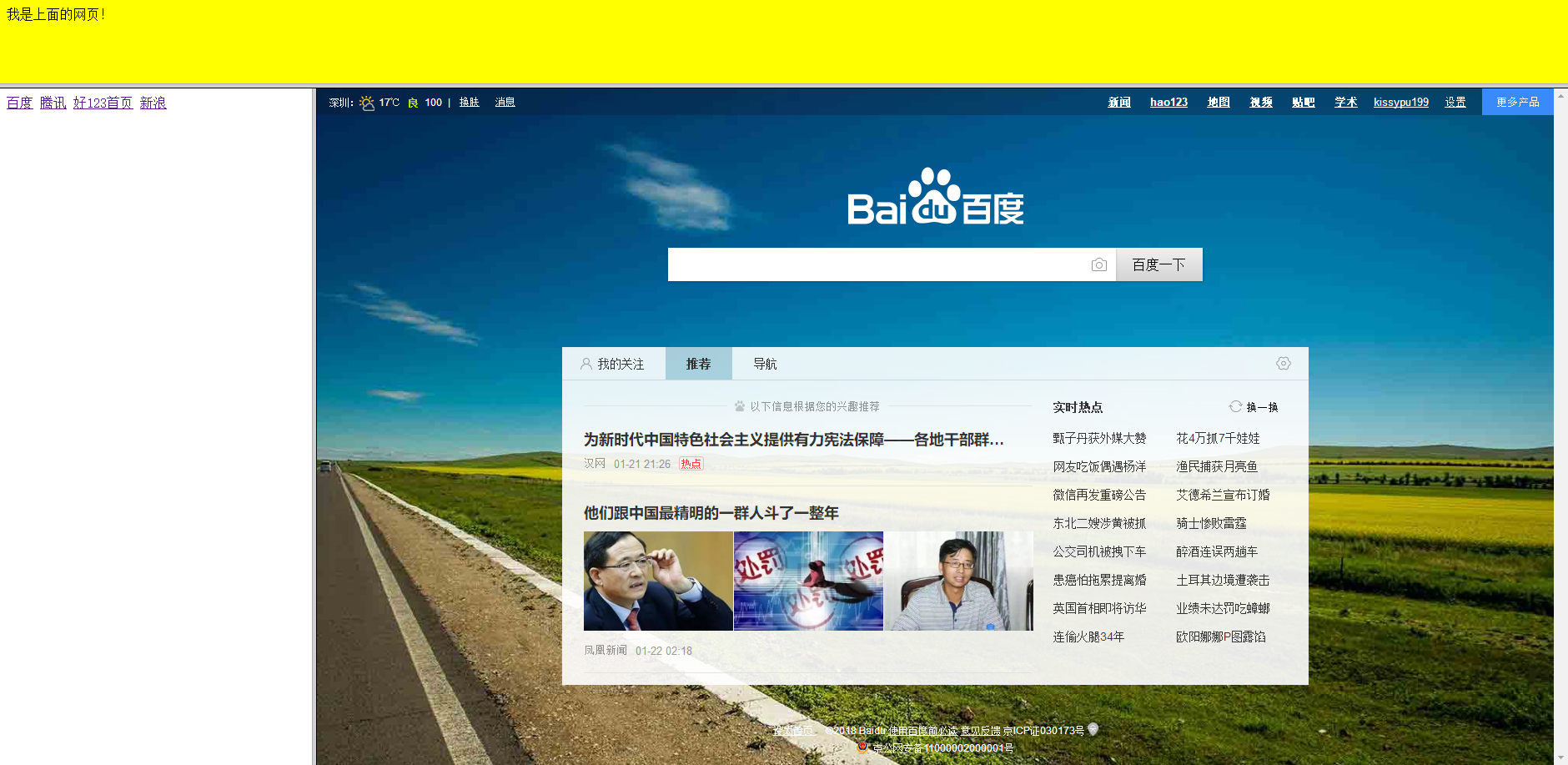


1. noresize：设置是否能够在网页中更改该分层页的大小，单位为“noresize”

微信截图_20180122024506

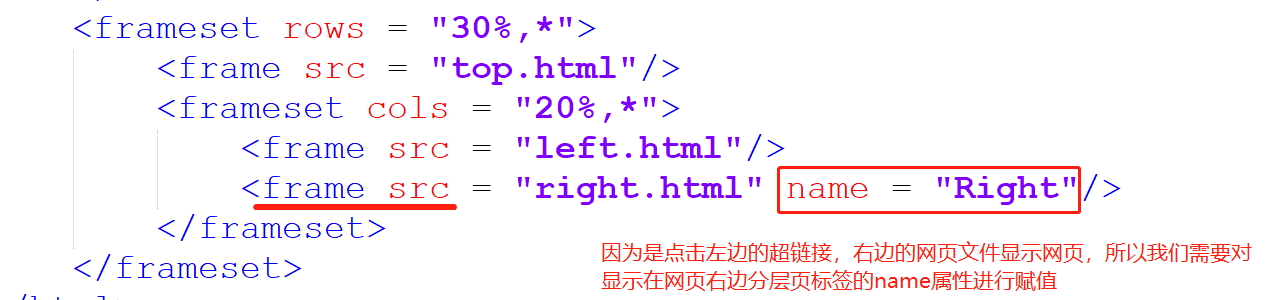
1. name：设置该分层页的一个标识

**当我们想在网页分层页的左边放一堆超链接，当点击超链接的时候在网页分层页的右边弹出网页我们需要怎么做？如下图：**

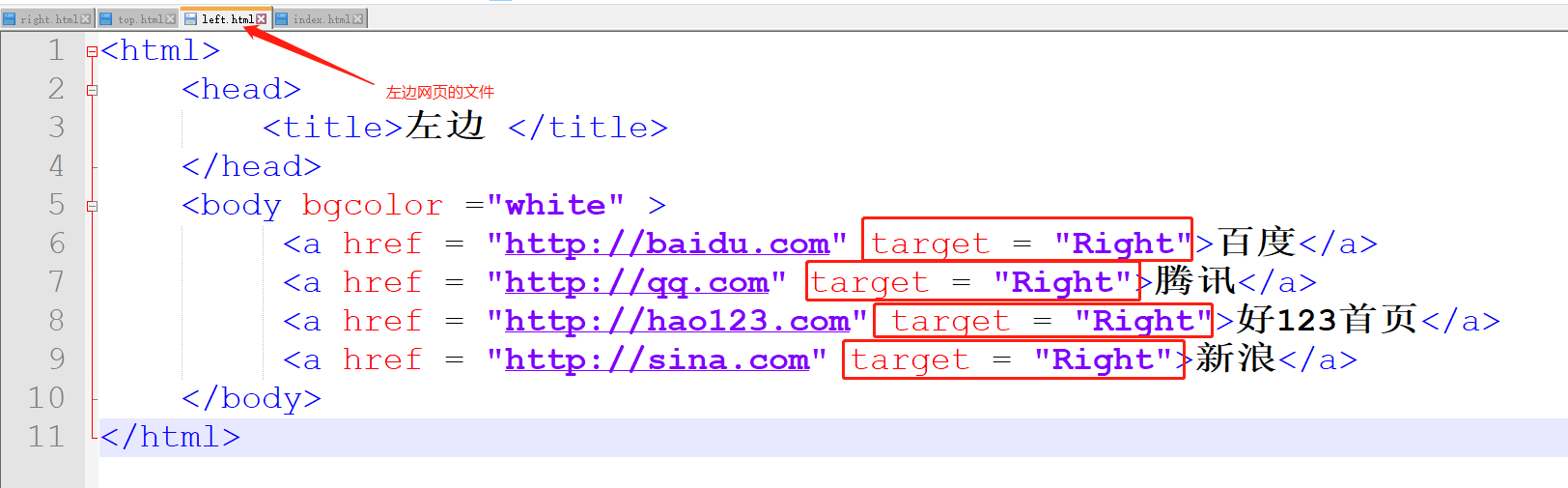


（该网页由三个分层网页文件和一个承载网页文件所组成）

**那么我们就需要在显示网页的分层标签对它的name属性设置标识**



**然后我们在拥有超链接网页文件的链接标签<a></a>的target属性中，设置成显示网页的分层网页标签的name属性的值，这样当我们点击的时候，name属性就会起到一个标识的作用，告诉浏览器，这个链接标签的超链接需要跳转到该网页文件中**

****

------------------------------------------------------------------------------------------------------------------------------

1. **区块标签：<div></div> 写在<body></body>标签中**

<div></div>标签定义HTML文档中的一个分隔区块或者一个区域一部分，不允许其他元素（标签）显示在这一层，但是可以通过css代码来改变这一状况，<div></div>标签经常配合css一起使用

微信截图_20180123182928



------------------------------------------------------------------------------------------------------------------------------

**二十九、 文本标签：<span></span> 写在<body></body>标签中**

<span></span>标签，能够把一些文本中的单独的一些内容分离出来，改变其样式，如下代码：我们想改变“眼睛”的样式

微信截图_20180123183426

但是我们没办法指定改变“眼睛这两个字”，因为他都存在于<p></p>标签当中，这时候我们在眼睛写上<span></span>标签，并用css内联嵌入代码，改变其样式

微信截图_20180123183654

微信截图_20180123183702

总而言之，span标签提供了一种能够将文本、文字的一部分独立出来的作用，当我们想改变一段文本中的某一句话或某一个字的时候，span标签发挥了很大作用

------------------------------------------------------------------------------------------------------------------------------