**·互联网/因特网/万维网**

互联网包含因特网，因特网包含万维网

互联网：internet。凡是能彼此通信的设备组成的网络就是互联网，互联网是一个广泛的概念。

因特网：Internet。是互联网的一种，由成千上万台机器组成的互联网，是互联网中最大的一个。因特网提供的服务一般包括有：www（万维网）服务、电子邮件服务（outlook）、远程登录（QQ）服务、文件传输（FTP）服务、网络电话等等。

万维网：是因特网提供的服务之一，

只要应用层使用的是Http协议，就是万维网

**·统一资源定位符**（Uniform Resource Locator，URL）也被称为网页地址，如同在网络上的门牌，是因特网上标准的资源的地址（Address)。

|  |  |  |
| --- | --- | --- |
| 序号 | 服务（协议）类型 | 含义 |
| 1 | http | 超文本传输协议资源 |
| 2 | https | 用加密传送的超文本传输协议 |
| 3 | ftp | 文件传输协议 |
| 4 | mailto | 电子邮件地址 |
| 5 | ldap | 轻型目录访问协议搜索 |
| 6 | news | Usenet新闻组 |
| 7 | file | 当地电脑或网上分享的文件 |
| 8 | gopher | Gopher协议 |

**·超链接（Hyper Link）**是指从一个网页指向另一个目标的连接关系，这个目标可以是另一个网页，也可以是相同网页上的不同位置，还可以是一个图片、一个电子邮件地址、一个文件，甚至是一个应用程序。超链接在本质上属于一个网页的一部分，它是一种允许我们同其他网页或站点之间进行连接的元素。

**·Web特点**

* **1.Web是易导航和图形化**
* **2.Web是具有平台无关性**
* **3.Web是支持分布式结构**
* **4.Web是具有动态性**
* **5.Web是具有交互性**

**·HTML（Hypertext Text Markup Language）超文本标记语言**是标准通用标记语言SGML（Standard　Generalized Markup Language是一种定义电子文档结构和描述其内容的国际标准语言，是所有电子文档标记语言的起源）下的一个应用，也是一种标准规范，它通过标记符号来标记要显示的网页中的各个部分。HTML是构成Web页面(Page)的基础。

**层叠样式表CSS（Cascading Style Sheet）级联样式表**

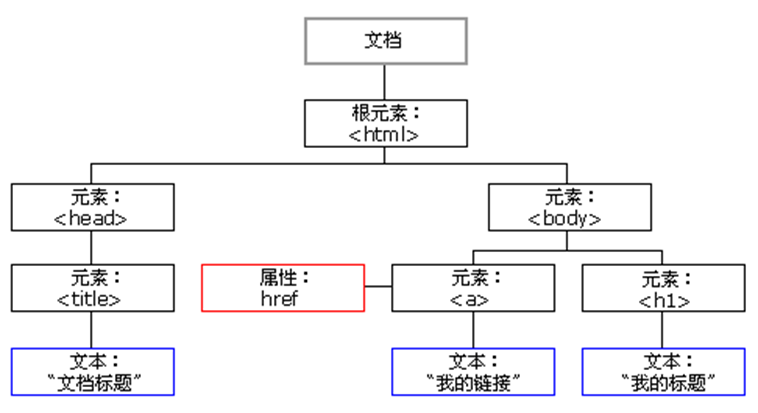
**·作用**：可以有效地对页面的布局、字体、颜色、背景和其它效果实现更加精确的控制 。

**JavaScript：**

·JavaScript最初由网景公司（Netscape）的Brendan Eich设计，**是一种由Netscape的LiveScript发展而来的客户端脚本语言，**主要**目的**是为了解决服务器端语言，提供数据验证的基本功能。

·一个完整的JavaScript实现是由3个不同部分组成的：核心（ECMAScript） 、文档对象模型（DOM） 、浏览器对象模型（BOM）。

**·DOM结构**



·Ajax的核心是JavaScript对象XMLHttpRequest。该对象在Internet Explorer 5中首次引入，它是一种支持异步请求的技术。简而言之，XMLHttpRequest使您可以使用JavaScript向服务器提出请求并处理响应，而不阻塞用户。

·**AJAX优点：**

①最大的一点是页面无刷新，在页面内与服务器通信，给用户的体验非常好；

②使用异步方式与服务器通信，不需要打断用户的操作，具有更加迅速的响应能力；

③可以把服务器负担的工作转嫁到客户端，利用客户端闲置的能力来处理，减轻服务器和带宽的负担，节约空间和宽带租用成本。AJAX的原则是“按需取数据”，可以最大程度的减少冗余请求、响应对服务器造成的负担。

④基于标准化的并被广泛支持的技术，不需要下载插件或者小程序。

·**AJAX的缺点**

①AJAX破坏浏览器的前进与后退功能，使得用户的习惯得不到延续；

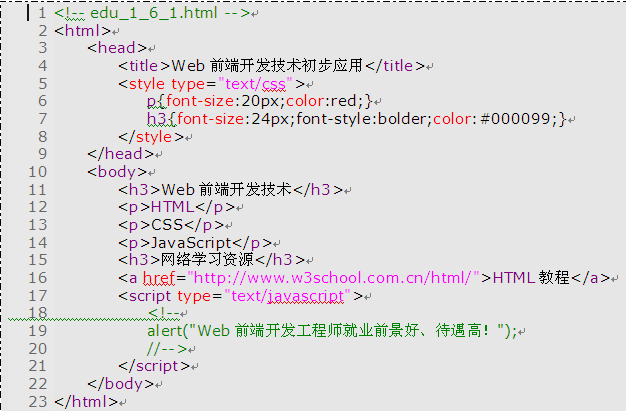
②安全问题；

③对搜索引擎的支持比较弱；

④破坏了程序的异常机制；

⑤违背了URL和资源定位的初衷；

⑥一些手持设备（如手机、PDA等）现在还不能很好的支持AJAX。



**· HTML文档结构：**一个完整的HTML文档由**头部head**和**主体body**两个部分组成。在头部<head>标记中，可**定义标题、样式**等；在主体<body>标记中，可**定义段落、标题字、超链接、脚本、表格、表单等元素，主体内容是网页要显示的信息。**

**头部标记所包含的信息一般不会显示在网页上。**

META标记用来描述一个HTML网页文档的属性，也称为元信息（meta-information），这些信息并不会显示在浏览器的页面中。

**·**设置body标记属性可以改变Web页面显示效果。body标记主要属性有text、bgcolor、background、link、alink、vlink、topmargin、leftmargin。

|  |  |  |
| --- | --- | --- |
| Body标记属性 | 值 | 描述 |
| text | * rgb(R,G,B) grb(R%,G%,B%) * *#RRGGBB* * *colorname* | 规定文档中所有文本的颜色。  **不赞成使用。请使用样式取代它。** |
| bgcolor | * *同上* | 规定文档的背景颜色。**不赞成使用。** |
| alink | * *同上* | 规定文档中活动链接的颜色。 |
| link | * *同上* | 规定文档中未访问链接的默认颜色。 |
| vlink | * *同上* | 规定文档中已被访问链接的颜色。 |
| background | *URL* | 规定文档的背景图像。 |
| topmargin | * *pixel* | 规定文档中上边距的大小 |
| leftmargin | * *pixel* | 规定文档中左边距的大小 |

**最常用的单标记有<br>、<hr>。<br>、<br/>表示换行，<hr>、<hr/>表示水平分隔线。**

属性应在开始标记（首标记）内定义，并且和标记名之间有一个空格分隔。属性值可以直接书写，也可以使用双引号“”括起来。

**添加注释：**

* **<!-- 注释信息 -->**
* **<comment>注释信息</comment>**

**·HTML页面编码基本规范**

1.所有标记均以“<”开始、以“>”结束。

2.根据标记类型，正确输入标记，单标记最好在右尖括号前加1个斜杠“/”，如换行标记是单标记<br/>，双标记最好同时输入起始标记和结束标记，以免忘记。

3.标记可以嵌套使用，但不能交叉使用。

4.在HTML代码中不区分大小写。

6.标记中可以设置各种属性，属性值建议用双引号标注起来

7.书写开始与结束标记时，在左尖括号与标记名或与斜杠“/”之间不能留有多余空格，否则浏览器标记不能识别，导致错误标记直接显示在页面上，影响页面美观效果。

8.编写HTML代码时，应该使用锯齿结构，即采用缩进风格，使代码结构清晰，便于理解和分析页面的结构，便于代码后期阅读和维护。

**·HTML文件命名规则**

1.文件的**扩展名为html或者htm**，建议统一用html作为文件名的后缀。

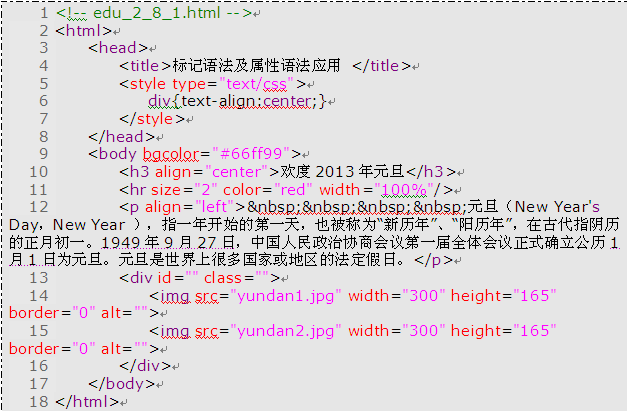
2.文件名中只可由英文字母、数字或下划线组成，建议以字母或下划线开始。

3.**文件名中不能包含特殊符号，如空格、$、&等。**

4.文件名区分大小写，特别在Unix、Linux系统中大小写表示的文件是不同。

5.网站首页文件名一般是index.html或default.html。

·三种DTD类型：严格Strict、过渡Transitional以及框架Frameset。

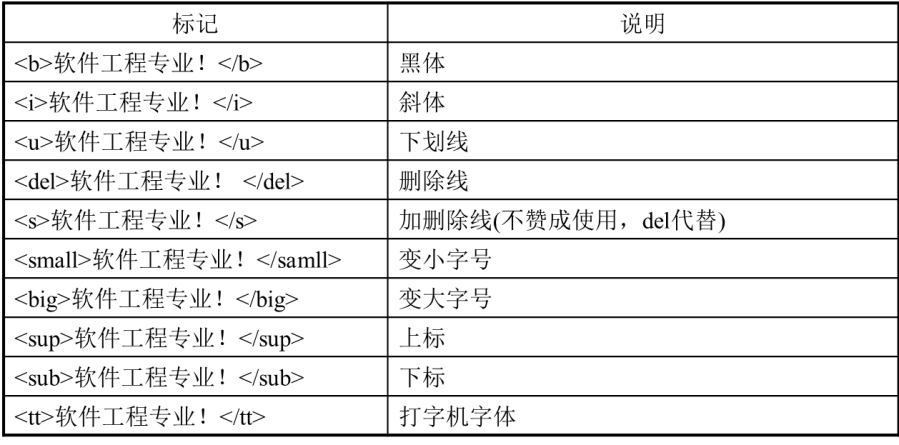


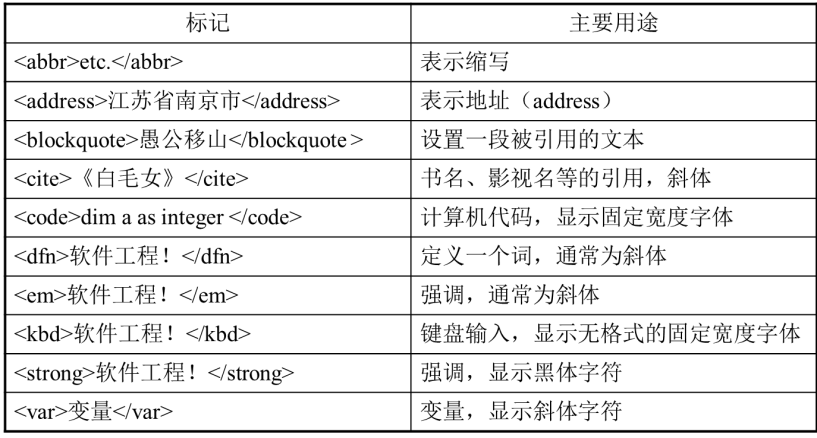
**Web页面设计需要遵循设计的3C原则 ：简洁 、一致性 、好的对比度**

标题是通过 <h1> - <h6> 等标记进行定义的。

·在HTML文件中，**添加空格的方式**与其他文档添加空格的方式不同，网页中通过代码控制来添加空格，而在其他编辑器中通过键盘空格键来输入空格。

|  |  |  |  |
| --- | --- | --- | --- |
| 显示结果 | 说明 | Entity Name | Entity Number |
|  | 显示一个空格 | **&nbsp;** | &#160; |
| < | 小于 | **&lt;** | &#60; |
| > | 大于 | **&gt;** | &#62; |
| & | &符号 | **&amp;** | &#38; |
| " | 双引号 | **&quot;** | &#34; |
| © | 版权 | **&copy;** | &#169; |
| ® | 注册商标 | **&reg;** | &#174; |
| × | 乘号 | **&times;** | &#215; |
| ÷ | 除号 | **&divide;** | &#247; |

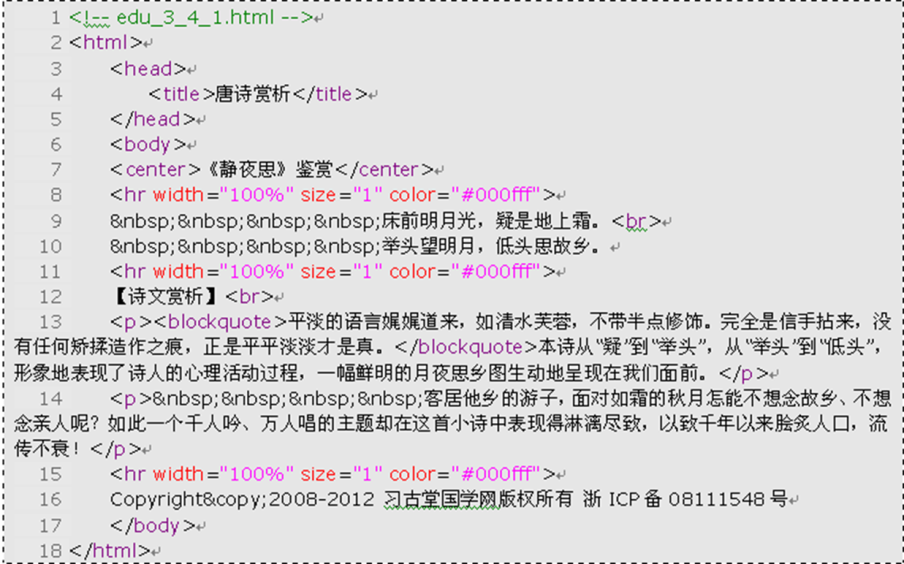




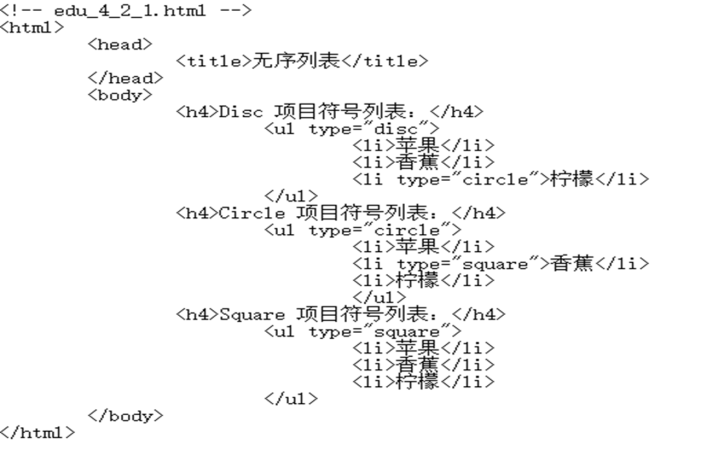
**·<font>元素**用来改变默认的字体、颜色、大小等属性，这些更改分别通过不同的属性定义完成。

**①段落标记<p></p>；②换行标记<br>单独出现，表示换行符；③水平分割线标记<hr>，显示一条细线以分割两个区域。**

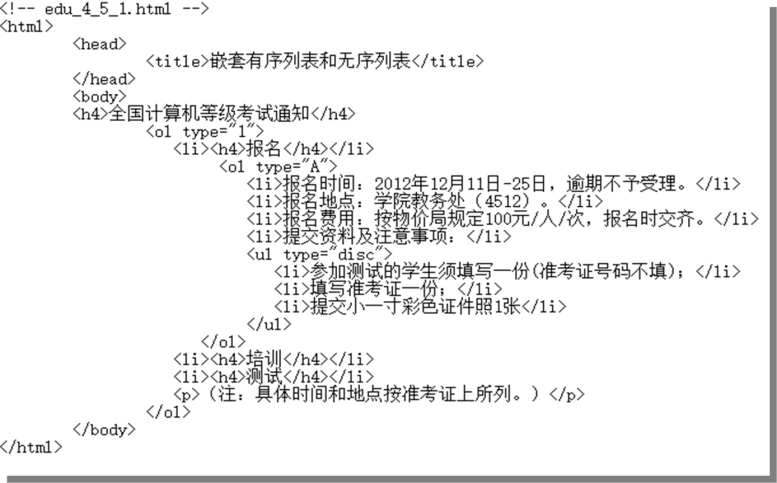
**段落缩进标记<blockquote>**

****

|  |  |  |
| --- | --- | --- |
| **列表类型** | **标记符号** | **备注** |
| **无序列表** | **<ul>…</ul>** | **常用** |
| **菜单列表** | **<menu>…</menu>** | **不常用** |
| **目录列表** | **<dir>…</dir>** | **不常用** |
| **有序列表** | **<ol>…</ol>** | **常用** |
| **定义列表** | **<dl>…</dl>** | **常用** |

****

**嵌套有序列表和无序列表**

****

****

·超链接在本质上属于一个网页的一部分，它是一种允许我们同其他网页或站点之间进行连接的元素。各个网页链接在一起后，才能真正构成一个网站。

**·超链接由目的地址、链接标签、打开位置等三部分组成：**

1.href：链接指向的目标文件

2. name：创建文档内的标签。

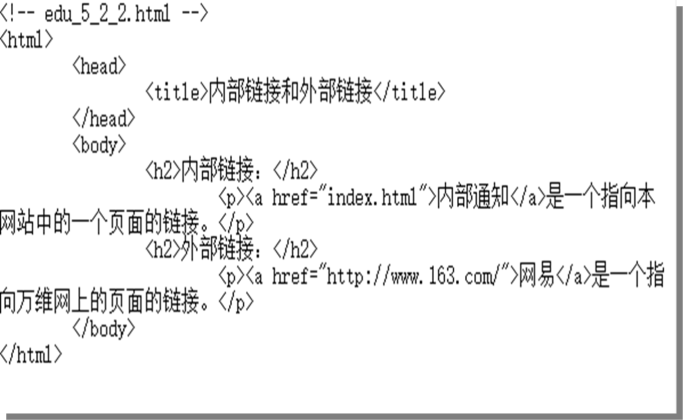
3.title：指向链接的提示信息

4.target：指定打开的目标窗口

**·在HTML文件中提供了三种路径：绝对路径、相对路径、根路径。**

**①绝对路径**指文件的完整路径，包括文件传输的协议http、ftp等，一般用于网站的外部链接，例如：http://www.edu.cn。

**②相对路径**是指相对于当前文件的路径，它包含了从当前文件指向目地文件的路径。



FTP服务器链接和网页链接区别在于所用协议不同，浏览网页采用http协议，而FTP服务器采用FTP协议连接。

**图片链接< img src=“url” width=“” >：**

alt ：添加图片的提示文字；

width/height：设置图片的宽度和高度(px,%)

border：设置图片边框(px)

align ：设置图片对齐方式（水平/垂直两个方向）

hspase/vspase ：设置图片的间距设置(px)



·Img标记的alt属性用来添加图片的提示文字，提示文字有两个作用。首先当浏览网页时，如果图片下载完成，鼠标放在图片上，鼠标旁边会出现提示文字。也就是说，当鼠标指向图片上方的时候，稍等片刻，可以出现图片的提示性文字，用于说明或者描述图片。其次，图片没有被下载的时候，在图片的位置上会显示提示文字，起到提醒的作用。



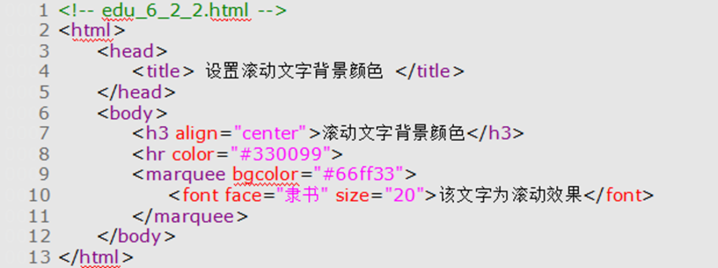
**图片和文字之间的对齐方式通过img标记中的align属性来设置**。图片的绝对对齐方式和文字的对齐方式一样，有居左、居右和居中；而相对文字的对齐方式则是指图片与同行文字的相对位置。

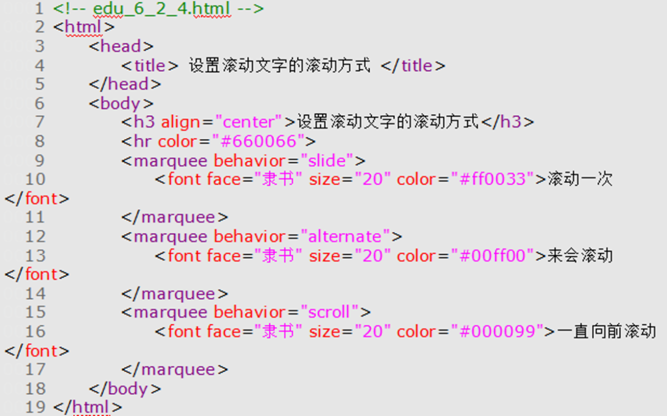
**<img src=”图片地址” hspace=”水平间距”>**

**<img src=”图片地址” vspace=”垂直间距”>**

**<marquee bgcolor=”颜色代码”>滚动文字</marquee>**

**文字背景颜色采用十六进制颜色代码。Direction属性表示滚动方向。属性behavior用来设置滚动方式。**

****

****

**·<marquee scrollamount=”滚动速度” >滚动文字 </marquee>**

滚动速度实际上就是滚动文字每次移动的长度，这个长度用数字表示，单位为像素。

**·<marquee scrolldelay=”延迟时间” >滚动文字 </marquee>**

延迟时间以毫秒为单位，该延迟时间设置得越小滚动速度越快，延迟时间设置得越大即会出现走走停停的效果。

**·<marquee hspace=”水平范围” vspace=”垂直范围” >滚动文字</marquee>**

**·<marquee loop=”循环次数” >滚动文字</marquee>**

·<bgsound src=”背景音乐地址” loop=”播放次数”>

① src属性用来指定背景音乐文件的地址或文件名称，而且音乐文件要加上后缀；

② loop属性用来指定背景音乐播放的次数，用数字表示。Infinite 和-1 表示播放无限次，直到关闭浏览器为止。

·<embed src=”多媒体文件地址” width=”播放界面的宽度” height=”播放界面的高度” autostart=”true|false” loop=”循环次数”></embed>

①width和height一定要设置，单位为像素，否则无法正确显示播放多媒体文件的软件；

②autostart的取值有两个：一个是true，表示自动播放；另一个是false，表示不自动播放；

③loop表示循环的次数，用数字表示。值为true时，循环播放；值为false时，不循环播放。



**CSS（Cascading Style Sheet）层叠样式表，也称为级联样式表，用来进行网页风格设计。**在网页制作时采用CSS技术，可以有效地对页面的布局、字体、颜色、背景和其他效果实现更加精确的控制。只要对相应的代码做一些简单的修改，就可以改变同一页面的不同部分，或者不同网页的外观和格式。

**·传统HTML的特点：**

**1、维护困难：**为了修改某个特殊标记的格式，需要花费很多时间，尤其对于整个网站而言，后期修改和维护的成本很高。

**2、标记不足：**HTML自身的标记并不丰富，很多标记都是为网页内容服务的，而关于美工的标记，比如文字间距、段落缩进等，这些HTML中都很难找到。

**3、网页过“胖”：**由于对风格样式没有统一进行控制，用HTML编写的页面往往是体积过大，占用了很多宝贵的带宽。

**4、定位困难：**在整个页面布局时，HTML对于各个模块的位置调整显得捉襟见肘。

·CSS通过定义标签如何显示，对页面结构风格进行控制，分离文档的内容和显示，**克服了传统HTML的缺点。**CSS放在页面中，通过浏览器的解释执行，是完全的文本，只要懂得HTML都可以掌握。

**·CSS特点：**

**1.页面的字体变得更漂亮，更容易编排。**

**2.可以轻松控制页面的布局。**

**3.可以在几乎所有的浏览器上使用。**

**4.以前一些必须通过图片转换实现的功能，现在只要用CSS就可以轻松实现，从而更快地下载页面。**

**5.可以将站点上所有网页的风格都使用一个CSS文件进行控制，只要修改这个CSS文件中相应的行，那么整个站点的所有页面都会随之发生变动。**

**·CSS优点**

**1.表现和内容相分离**

CSS通过定义HTML标签如何显示控制网页的格式，使得页面内容和显示相分离，简化了网页格式设计，也使得对网页格式的修改更方便。

**2.加强了网页的表现力**

CSS样式属性提供了比HTML更多的格式设计功能。例如，可以通过CSS样式去掉网页中超级链接的下划线，甚至可以为文字添加阴影，翻转效果等。

**3.增强了网站格式的一致性**

将CSS样式定义到样式表文件中，然后在多个网页中同时应用样式表文件中的样式，就确保了多个网页具有一致的格式，并且可以随时更新样式表文件，以达到可以自动更新多个网页的格式设置，从而大大降低了网站的开发与维护工作。

**·CSS编辑方式**

**1、写在HTML文件中的CSS规则**

根据其位置又可以分为两种形式：一种是写在某个元素的属性部分，作为style属性值；另一种是写在head标记中，通过Style标记包含。

**2、将CSS规则写在单独的文件中**

该文件成为CSS文件，它是纯文本文件，可以使用任何编辑器编辑，文件后缀名为.css。在需要应用CSS规则的多个HTML文件里面引用该CSS文件，从而实现内容和表现的分离，也提高了网站可维护性。

* **CSS规则由两个主要的部分构成：选择器（Selector或称选择符）和声明（Declaration）。**
  + 选择器通常是需要改变样式的HTML元素。
  + 声明由一个或多个属性名称与属性值对组成。属性名称是CSS的关键字，如font-family（字体）、color（颜色）和border（边框）等。属性名称用于指定选择器某一方面的特性，而属性值则用于指定选择器特性的具体特征。

/\* css的注释方法\*/

**CSS选择器：标记选择器（或称元素选择器）、类选择器、id选择器、伪类选择器。**

**①标记选择器**= 元素选择器。直接使用HTML标记作为选择器，它定义的样式作用于页面中所有与选择器同名的标记。

**②类选择器**：**任何合法的HTML标记都可以使用class属性**。由点号**“.”**和**类名称**直接相连构成。

标记选择器和类选择器可以联合使用，使用方式是标记选择器与类选择器直接项链，成为联合选择器，可以用来设置特定类中的特定标记。  
**③ID 选择器**前面有一个 # 号 - 也称为棋盘号或井号。

**类选择器还是 ID 选择器？**

区别 1：只能在文档中使用一次：与类不同，在一个 HTML 文档中，ID 选择器会使用一次，而且仅一次。

区别 2：不能使用 ID 词列表：不同于类选择器，ID 选择器不能结合使用，因为 ID 属性不允许有以空格分隔的词列表。

**④伪类选择器：**标记：伪类名{/\*CSS规则\*/}

·在声明各种CSS选择器时，如果某些选择器的风格是完全相同的，或者部分相同，都可以使用集体声明用“，”将多个选择器隔开，对风格相同的CSS选择器同时声明。

**·通配选择器（universal selector），显示为一个星号（\*）。该选择器可以与任何元素匹配，就像是一个通配符。**

**·CSS按其位置可以分为4种：**

**①内联样式表（Inline Style Sheet）**

<标记 style=“属性1：属性值1；属性2：属性值2；…….”>几乎任何一个html标记都可以设置style属性。**慎用！**

**②内部样式表（Internal Style Sheet）**

内部样式表写在HTML的<head></head>中，只对所在的网页有效。使用<style></style>标记对来设置CSS规则。<!--…-->隐藏标记，如果浏览器不支持CSS样式，则会自动跳过此段代码。

**③链接外部样式表（Link External Style Sheet）④导入外部样式表（Import External Style Sheet）**

1·外部样式表是将CSS规则写在.css为后缀的CSS文件中，需要用到此样式的网页里引用该CSS文件。

2·一个CSS文件可以供多个网页引用，从而实现整体页面风格统一设置。

3·根据引用方式不同可以分为链接外部样式表和导入外部样式表，他们形式上的区别在于链接外部样式表通过链接link标记，导入外部样式表必须在内部样式表内首行通过“@import url（’外部样式文件’）”来定义。

**·外部样式表与内联，内部样式表相比，具有如下优点：**

（**1）便于重用。**一个外部CSS文件所定义的样式，可以被多个网页共用。

**（2）便于修改。**修改样式只需要修改CSS文件，无须修改每个网页。

**（3）提高显示速度。**样式写在网页里，网页文件变“胖”，增加网页传输的负担，降低网页显示速度。如果某CSS文件已被某网页引用并加载，则其他需要引用该CSS文件的网页时可以从缓存中直接读取该CSS文件，提高网页显示速度。

**·CSS继承即子标记会继承父标记的所有样式风格，并且可以在父标记样式风格的基础上再加以修改，产生新的样式，而子标记的样式风格完全不影响父标记。**

**行内样式 > id样式 > class样式 > 元素样式**

**·<div>是一个块级元素，其前后均有换行符，可定义文档中的分区或节。**

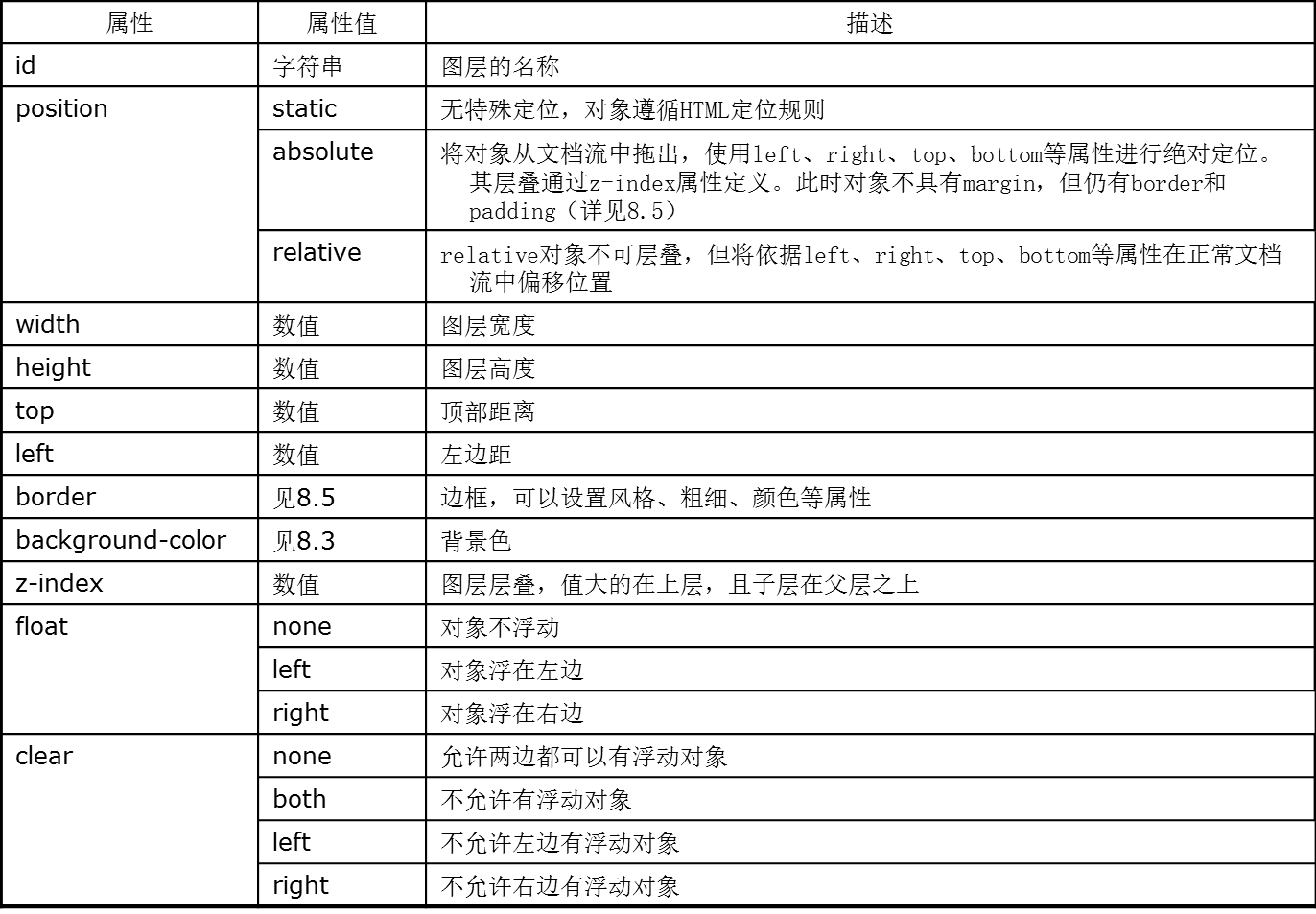
Style属性：设置层 的样式，如果不定义层的样式的话，在浏览器里什么都看不到。

•<div>通常和id或class配合使用，把文档分割为独立的、不同的部分，对文档进行布局。<div>是一个块级元素。这意味着它的内容自动地开始一个新行。实际上，换行是<div> 固有的唯一格式表现。可以通过<div>的class或id 应用额外的样式。

•display属性可以改变元素的常规CSS显示类型，取值包括block、inline和none，即按块元素显示、以内联方式显示和隐藏元素。

•使用多个<div>时，一方面它们可以将页面分割成不同的块，块与块之间没有包含关系；另一方面又可以把功能相近的块组织到一个更大的块中，便于整体控制，这便是图层嵌套。

•多个<div>还可以层叠。<div>层叠必须首先将position属性设置为absolute，然后利用z-index属性控制层叠关系。



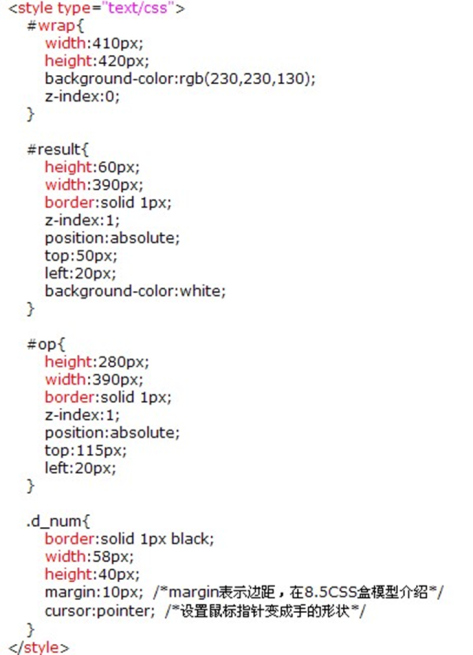
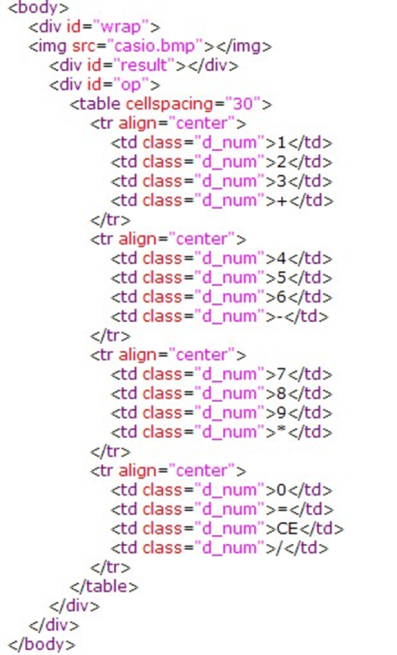
**<div>和<span>**

①<div>和<span>共同的特点是默认都没有对标记内的对象进行任何格式化渲染，主要用于应用CSS样式，对页面进行分割、布局。

②不同的是，<div>是块级元素，其前后均有换行符；而<span>是行内元素，没有固定的格式表现。

③但是块元素和行元素不是绝对的，通过定义CSS的display属性可以相互转化。





|  |  |
| --- | --- |
| **属性** | **描述** |
| **font-style** | **设置字体的风格** |
| **font-variant** | **设置小型的大写字母字体** |
| **font-weight** | **设置字体的粗细** |
| **font-size** | **设置字体的大小** |
| **font-family** | **设置字体名** |

**·font-style属性：**

normal：表示不使用斜体，是font-style属性的默认值

Italic：表示使用斜体显示文字

oblique：表示使用倾斜字体显示

**·font-variant属性：**

normal：表示正常的字体，是font-variant属性的默认值

small-caps：表示使用小型的大写字母字体

**·font-weight属性**

normal：表示正常的字体，是font-weight属性的默认值

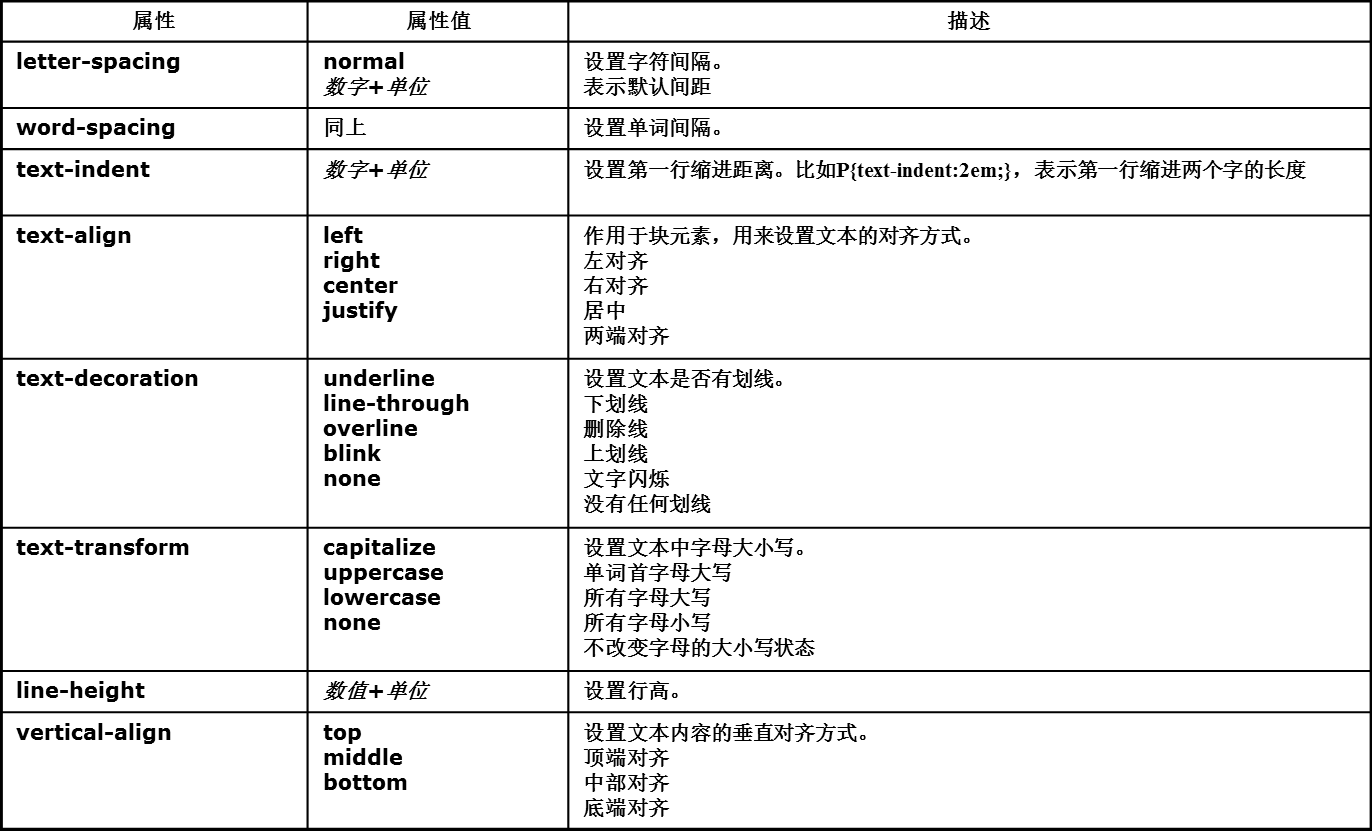
bold：表示标准的粗体

bolder：表示特粗体（为相对参数）

lighter：表示细体（为相对参数）

整数：取值为100、200、300、400、500、600、700、800、900来表示粗细程度，100表示最细、400等价于normal、700等价于bold

**·font-family专门用于设置字体名称。**

****

**·**在HTML中段落的首行往往需要通过插入4个“&nbsp；”才能实现首行空2个字符的排版格式，而在CSS中可以使用**text-indent**属性来设置首行缩进量。

**·字符装饰属性text-decoration主要用来完成文字加下划线、顶线、删除线及文字闪烁等效果。**

**none：**表示文字无装饰。

**blink：**表示文字闪烁。

**underline：**表示文字加下划线。

**line-through：**表示文字加贯穿线。

**overline :** 表示文字加上划线。

**·利用text-transform属性以转换英文大小写。**

**capitalize：**将每个单词的第一个字母转换成大写，其余无转换发生。

**uppercase：**转换成大写。

**lowercase：**转换成小写。

**none：**无转换发生。

**·text-align属性规定元素中的文本的水平对齐方式。**

left：表示左对齐，默认值。

right：表示右对齐。

enter：表示居中。

justify：表示两端对齐。

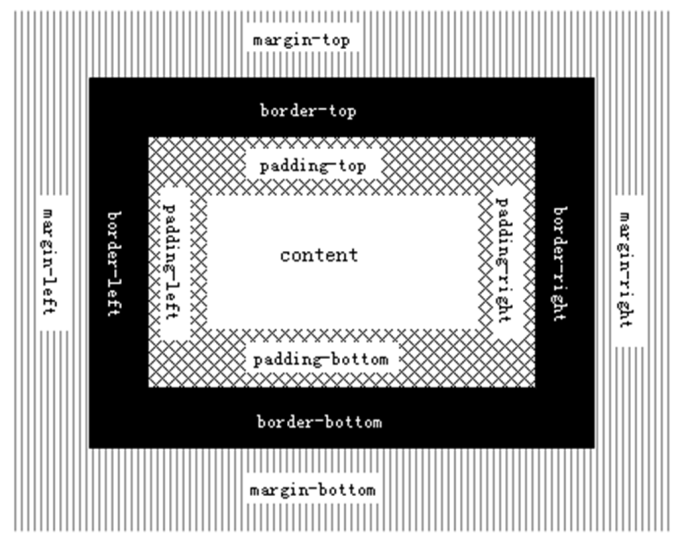
**·vertical-align属性以设置元素的垂直对齐方式。**

|  |  |
| --- | --- |
| **值** | **说 明** |
| **baseline** | **默认。元素放置在父元素的基线上。** |
| **sub** | **垂直对齐文本的下标。** |
| **super** | **垂直对齐文本的上标** |
| **top** | **把元素的顶端与行中最高元素的顶端对齐** |
| **text-top** | **把元素的顶端与父元素字体的顶端对齐** |
| **middle** | **把此元素放置在父元素的中部。** |
| **bottom** | **把元素的顶端与行中最低的元素的顶端对齐。** |
| **text-bottom** | **把元素的底端与父元素字体的底端对齐。** |
| **length** | **指定长度值** |
| **%** | **使用 "line-height" 属性的百分比值来排列此元素。允许使用负值。** |
| **inherit** | **规定应该从父元素继承 vertical-align 属性的值。** |

**·line-height用于设置行与行之间的距离。**

|  |  |  |  |
| --- | --- | --- | --- |
| **Black** | **#000000** | **Gray** | **#808080** |
| **White** | **#FFFFFF** | **Silver** | **#c0c0c0** |
| **Red** | **#FF0000** | **Maroon** | **#800000** |
| **Yellow** | **#FFFF00** | **Olive** | **#808000** |
| **Lime** | **#00FF00** | **Green** | **#008000** |
| **Aqua** | **#00FFFF** | **Teal** | **#008080** |
| **Blue** | **#0000FF** | **Navy** | **#000080** |
| **Fuchsia** | **#FF00FF** | **Purple** | **#800080** |

**·盒子：内容（content）、填充（padding）、边框（border）、边界（margin）**

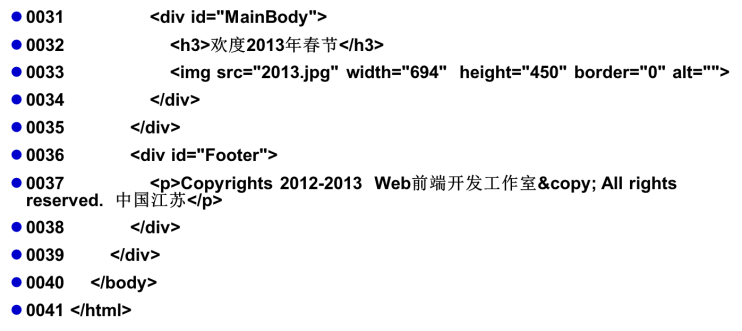
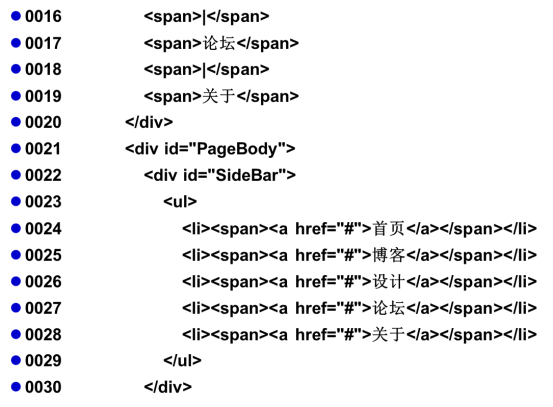
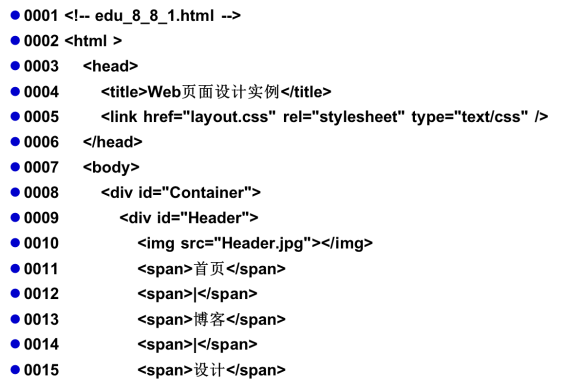


**·边界属性是margin，**也称为外边距，表示盒子边框与页面边界或其他盒子之间的距离，属性值为长度值、百分数或auto，属性效果是围绕元素边框的“空白” 。

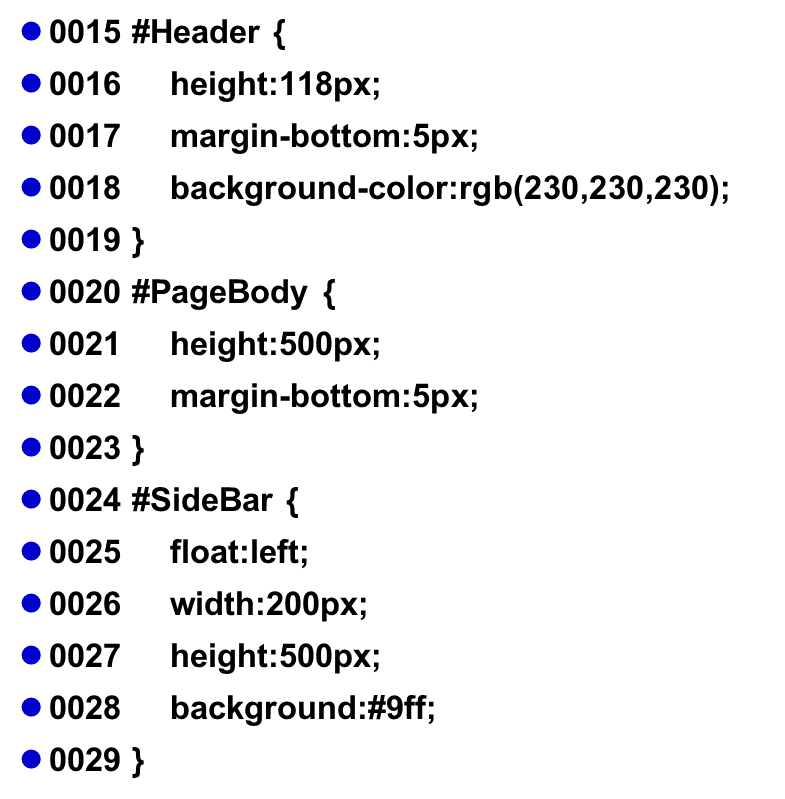
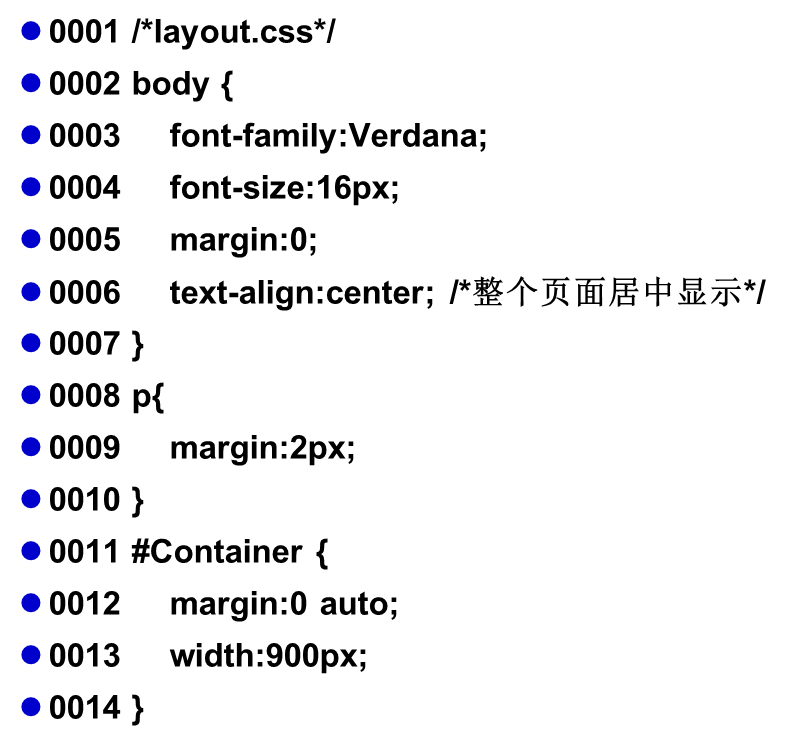
·**边框属性是border，**用于设置边框：风格（border-style）、宽度（border-width）、颜色（border-color）

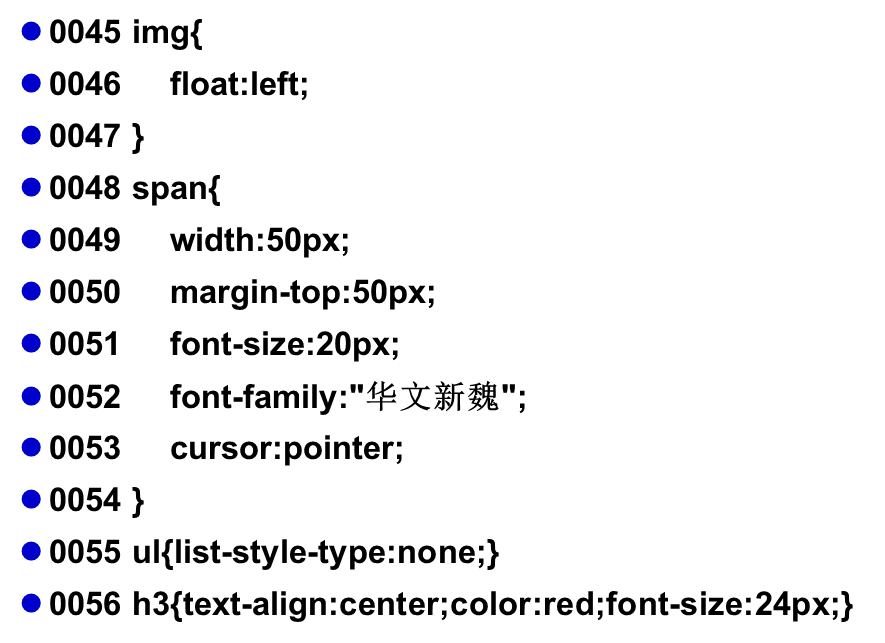
**·填充属性是padding，**也称为内边距，表示元素内容与边框之间的距离，属性值为长度值、百分数，属性效果是包含在元素边框里面并围绕着元素内容的“元素背景”。





**layout.css**





**·在HTML中，表格主要通过3个标记构成：table、tr、td标记。**

表格标记：<table></table>

表格标题标记：<caption></caption>

表格表头标记：<th></th>

表格行标记：<tr></tr>

表格列标记：<td></td>

定义表格的表头：<thead></thead>

定义表格的主体：<tbody></tbody>

定义表格的页脚：<tfoot></tfoot>

**<tr>表示插入一行； <td >表示插入一列；<caption>插入表格标题</caption>**

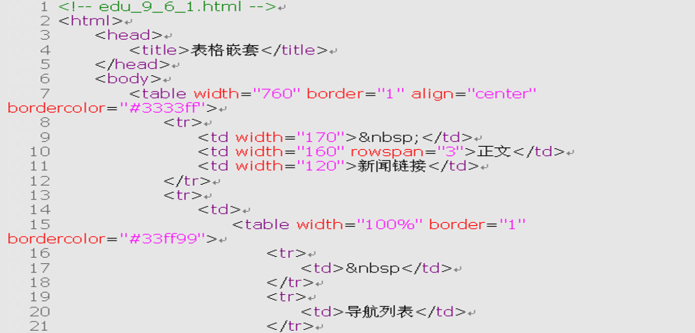
**·**制作表格时，常常需要制作表头将表格中的元素属性分类，在网页文件中插入表格需要给表格定义表头内容时，使用**成对<th>标记可以实现**，**表头内容使用的是粗体样式显示，默认对齐方式是居中对齐。**

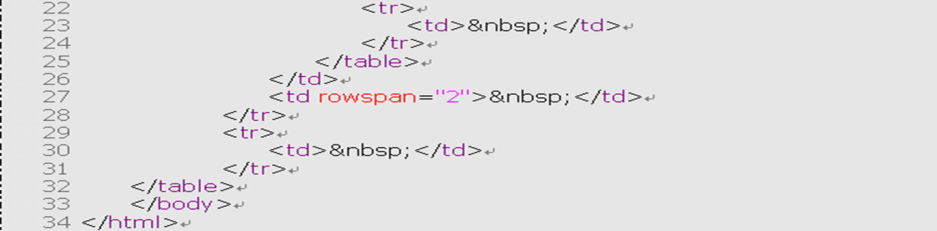
**·表格水平对齐属性<table align=”value”> </table>**

****

****

**·在网页制作过程中，有时需要对网页中的表格进行单元格的横向合并，这在网页中叫做设置跨列。<td colspan=”2”></td>**

****

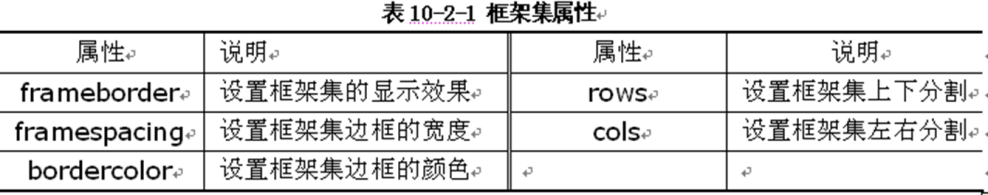
****

**一个简单的网站布局**

****

**·框架技术**是一种在**一个页面中显示多个网页的技术**，通过超链接可以使**框架之间建立内容之间的联系**，从而实现页面导航的功能。通过使用框架，可以在同一个浏览器窗口中显示不止一个页面。每个HTML文档称为一个框架，并且每个框架都独立于其他的框架。**框架的基础结构分为框架集和框架，框架集中可包含许多框架。**

**·**在网页文件中，**使用框架集的页面的<body>标记将被<frameset>标记替代，然后再利用<frame>标记去定义框架结构**，常见的分割框架方式有：左右分割、上下分割、嵌套分割。

****

**·表单**是动态网页中非常重要的元素，**是用户与网页间进行动态交互的接口**，主要用于**数据采集**，一个表单由如下部分组成：

**①表单标记：**包含了处理表单数据所用CGI程序的URL，以及将数据提交到服务器所采取的方法。

**②表单域：**包含了单行文本框、密码框、隐藏域、多行文本框、复选框、单选按钮、下拉列表框和文件选择框等。

**③表单按钮：**包括提交按钮、重置按钮和普通按钮；用于将表单数据传送到服务器上的CGI脚本或者取消输入，还可以用表单按钮来控制其他定义了处理脚本的处理工作。

**·表单form标记为双标记，用于声明一个表单，定义采集数据的范围，以<form>开始和</form>结束。**

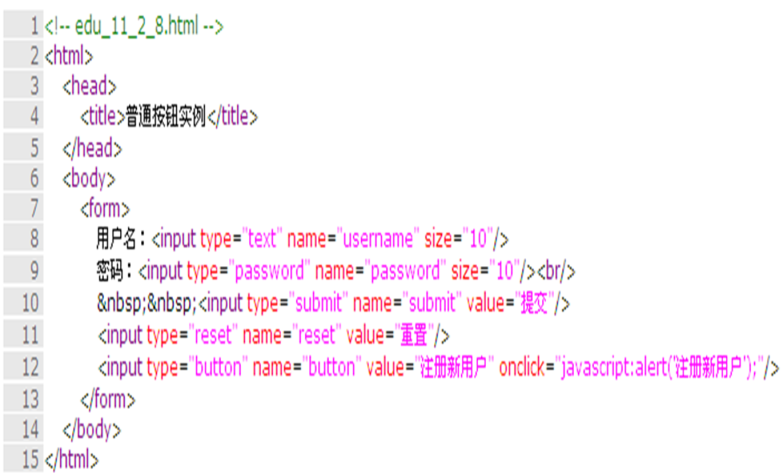
**·<input>是表单中用于输入信息的标记**；<input>标记中type属性值text用于向表单中插入单行文本框，单行文本框中可以输入任意类型的数据，但是输入的数据只能单行显示，不能换行。

**·设置input标记的type属性值为“password”，可以实现向表单中插入密码输入框。**密码输入框中可以输入任意类型的数据，和单行文本输入框有所不同，这些数据不是显式地显示在页面上，而是显示字符“·”替代输入内容，这样设计可以保障用户输入的密码不被泄露。

**·设置input标记的type属性值为“checkbox”，** 可以实现向表单中插入复选框，用户可利用复选框在网页上设置多项选择。**设置input标记的type属性值为“radio”，**可以实现向表单中插入单选按钮，用户可利用单选按钮在网页上为某一选择设置多个单选项。**设置input标记的type属性值为“image”，**可以实现向表单中插入图像按钮，用户可利用图像按钮在网页中插入一张图像，通过src属性加载图像。

·**设置input标记的type属性值为“submit”，**可以实现向表单中插入提交按钮，提交按钮用于将表单的信息提交至服务器进行处理。在表单中插入提交按钮时，如果不设置属性value的值，则它的初始值是“提交查询按钮”。**所以一定要给value属性赋值。设置input标记的type属性值为“reset”，**可以实现向表单中插入重置按钮，重置按钮用于将表单中所有的信息清空，然后让用户可以重新填写。

**·设置input标记的type属性值为“button”，**可以实现向表单中插入普通按钮。



·设置input标记的type属性值为“hidden”，可以实现向表单中插入隐藏框

**·多行文本输入框textarea标记是双标记！<textarea name=”” rows=“” cols=“” wrap=““/>初始信息内容</textarea>**

**·select和option标记用于在表单中插入下拉菜单和列表项。**

·fieldset标记将表单内容的一部分打包，生成一组相关表单的字段。当一组表单元素放到fieldset标记内时，浏览器会以特殊方式来显示它们，它们可能有特殊的边界、3D效果，或者可创建一个子表单来处理这些元素。<legend>标记为<fieldset>元素定义标题。

**JavaScript：**

**JavaScript是一种基于对象和事件驱动并具有相对安全性的客户端脚本语言**，被广泛应用于各种客户端Web程序尤其是HTML开发中，**能给HTML网页添加动态功能**，响应用户各种操作，实现诸如欢迎信息、数字日历、跑马灯，显示浏览器停留时间等特殊效果，提高网页的可观性。

**特点：**

**1.简单性：**Javescript 是一种脚本编程语言，采用小程序段的方式实现编程，是一种解释性语言，无须编译，而是在程序运行过程中被逐行解释。

**2.安全性：**JS 不允许程序访问本地的硬盘资源，不能将数据存入服务器上，不允许对网络文档进行修改和删除，只能通过浏览器实现信息浏览或动态交互。

**3.动态性：**JS可以直接对用户的输入信息进行简单处理和响应，而无须向Web服务程序发送请求再等待响应。

**4.跨平台性：**JS程序运行只依赖于浏览器，与操作系统和机器硬件无关。只要是能够运行浏览器，并是支持javascript的浏览器，就可以正确执行。

**·一个脚本通常是解释执行而非编译。**

**·脚本语言可以分为两类：服务器脚本和客户端脚本**

**·**大多数脚本语言一样，Javascript在浏览器中是解释执行的。应用程序的执行通常有**解释和编译**两种。**编译**是将程序源代码翻译成可执行的**二进制代码文件**；**解释**的执行方式类似与我们日常生活中的“同声翻译”，应用程序源代码**一边由相应语言的解释器“翻译”成目标代码（机器语言），一边执行，即翻译一句就执行一句。**

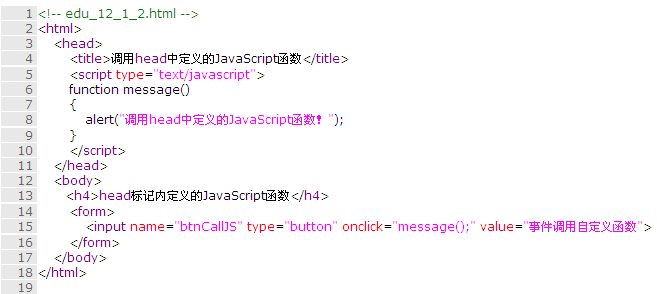
·与Java，C++等强类型语言不同，Javascript是一种弱类型语言。强类型语言要求变量的使用要严格符合定义，**要求用户在程序中使用的所有变量都必须先定义后使用**。也就是说，一旦一个变量被指定了某个数据类型，**如果不经过强制转换，那么它就永远是这个数据类型了**。相比较而言，Javascript则非常灵活，**在使用一个变量时，可以先声明再使用，也可以直接使用**，是数据类型可以被忽略的语言。此外，大多数情况下，**Javascript将根据需要自动进行类型的转换**。

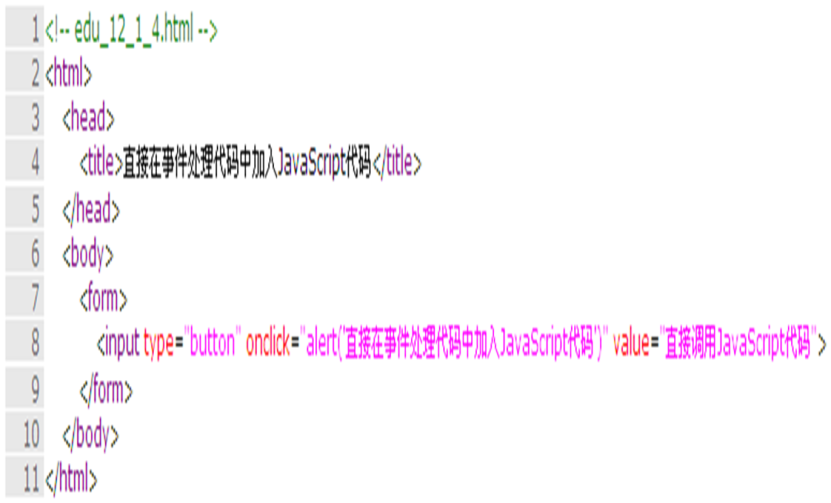
**·JavaScript的放置：**

* 放置在页面的head或body部分。当页面载入时，会执行位于body 部分的JavaScript；而位于head部分的JavaScript只有被显式调用时才会被执行。
* 可将JavaScript函数单独写成一个js文件，在HTML文档中引用该js文件。
* 直接写在事件处理代码中。

**语法：function functionname(参数1，参数2，…，参数n){ 函数体语句；}**

**javascript自定义函数必须以function关键字开始，然后给自定义函数命名，函数命名时一定遵守标识符命名规范。函数名称后面一定要有一对（），括号内可以有参数，也可以无参数，多个参数之间用“，”分隔。函数体语句必须放在大括号{ }中。**

****

****

**·JavaScript程序由语句、函数、对象、方法、属性等构成，程序结构包括顺序、分支和循环三种基本结构。**

·JavaScript代码是由若干条语句或语句块构成的执行体。

**JavaScript中的消息对话框分为警告框、确认框和提示框。**

**1.警告框**

**alert (message)；**

alert（）函数用于显示带有1个图标、1条制定消息和1个确定按钮的告警框。

message 是显示在弹出对话框窗口上的纯文本（非html文本）

**2.确认框**

**confirm (message)**

Confirm() 方法用于显示带有1个图标、指定消息和“确定”，“取消”按钮的对话框。

如果用户单击“确定”按钮，则confirm（）返回true。如果单击“取消”按钮，则返回false。

**3.提示框**

**prompt (text, defaultValue)**

如果用户单击提示框的取消按钮，则返回null。如果用户单击确认按钮，则返回文本输入框中输入的值。

text用于设置提示信息。defaultValue用于设置默认的输入值。





·单行注释使用“//”作为注释标记，多行注释能包含任意行数的注释文本，以“/\*”标记开始，以“\*/”标记结束，两个标记之间所有的内容都是注释文本。

**1.标识符：**标识符是计算机语言中用来表示变量名、函数名等的有效字符序列，简单来说，标识符就是一个名字，JavaScript关于标识符的规定如下：

(1)必须使用字母或者下划线开始。

(2)必须使用英文字母、数字、下划线组成，不能出现空格、制表符、标点符号和运算符等其他符号。

(3)不能使用JavaScript关键字与JavaScript保留字。

(4)不能使用JavaScript语言内部的单词，比如Infinity，NaN，undefined等。

(5)大小写敏感，如name和Name是不同的两个标识符。

**2.关键字：**是JavaScript中已经被赋予特定意义的一些单词，关键字不能作为标识符来使用。

**3.保留字：**JavaScript中除了关键字以外，还有一些用于未来扩展时使用的保留字，保留字同样不能用于标识符的定义。

**数据类型是每一种计算机语言中的重要基础，JavaScript中的数据类型可分为字符型、数值型、布尔型、Null、Undefined和对象六种类型。**

**1.字符型**

* 字符型数据又称为字符串，由若干个字符组成，**并且应该用单引号或双引号封装起来**。
* 在使用字符串的过程中，有时会遇到一种情况：在一个字符串中需要使用单引号或双引号。正确的方法是在由双引号标记的字符串中加入引用字符时使用单引号，在由单引号标记的字符串中加入引用字符时使用双引号，即**保证一个字符串的开头和结尾使用同一种引号**，而字符串内使用另一种引号。

**2.数值型**：JavaScript中的数值型分为**整型、浮点型、内部常量以及特殊值。**整型数值即整数的取值范围是-253～253，例如100、-3500、0等都是整数。浮点数的取值范围为-1.7976931348623157×10308～-5×10-324和1.7976931348623157×10308～5×-324，例如3.53、-534.87等都是浮点型数值。

****

**3.布尔型：**JavaScript中，通常采用true和false表示布尔型数据，但也可将他们转换为其他类型的数据，例如可将值为true的布尔型数据转换为整数1，而将值为false的布尔型数据转换为整数0。

**4.Null：**在JavaScript中，Null是一种特殊的数据类型，它表示“无值”。Null类型数据唯一的、合法的值是null。null除了表示Null类型的数据外，也可用在表示其他类型的数据中，比如对象、数组和字符串等。当变量不再使用时，将它赋值为null，以释放存储空间。

**5.Undefined：**JavaScript中，Undefined也是一类特殊的值，如果使用的变量不存在或者虽存在但没有赋值，则返回值就是Undefined。它与null值的不同之处在于：**null值表示已经对变量赋值，只不过赋的值是“无值”；而Undefined表示变量不存在或者没有赋值。**

**6.对象：**在JavaScript中除了数值型、字符型和布尔型这些基本的数据类型以外，还有一种**复合的数据类型称为对象，对象是属性和方法的集合**。对象的属性可以是任何类型的数据，包括数值、字符、布尔型，甚至是另一种类型的对象，而方法是一个定义在对象中的函数，用于实现特定的功能。

·**JavaScript是一种无数据类型的计算机语言，在定义变量时不需要指定变量的数据类型，统一使用关键字var声明，**JavaScript会在需要的时候自动对不同的数据类型进行转型。

·变量声明的基本语法为  **var 变量名[,变量名…] ;**

·变量赋值的基本语法为 **变量名=初值;**

**·**由**操作数**（变量、常量、函数调用等）和**运算符结合在一起构成的式子称为“表达式”，**最简单的表达式可以是常量名称。

·JavaScript算术运算符负责算术运算，用算术运算符和运算对象（操作数）连接起来符合规则的式子，称为算术表达式。

**1.加法运算符（+）**

（1）对数值型的操作数进行加法运算：11+34；

（2）如果两个数都是字符型，或者一个是字符型，一个是数

值型则加法运算可将数值转换成字符串，然后执行两个字符

串的**连接操作**。

**2.减法运算符（-）**

（1）对两个数值型操作数进行减法操作

（2）取反运算，一元运算符，操作数必须为数字，且运算符位于操作数前

（3）将字符串转化成数值类型

**3.乘法运算符（\*）**

（1）完成数值型的乘法操作

（2）如果操作数不是数值型，但是可以转化成数值型，乘法运算符会自动将其转换成数字，再进行乘法操作

（3）如果操作数无法转换成数值型，则运算结果为NaN

**4.除法运算符（/）**

（1）18/6 // 结果为3

（2）18 /”6” // 先将”6” 转换成数字，然后再执行操作，结果为3

（3）18/”A” // ”A” 无法转换为数字，结果为NaN

（4）20/0 // 被除数为20，除数为0，结果为Infinity

（5）-20/0 // 被除数为-20，除数为0，结果为-Infinity

**5.模运算符（%）：模运算又叫做取余数运算符。**

**6.自增运算符（++）**

（1）自增运算符为一元运算符，可对操作数执行自增运算，增量为1.

（2）操作数必须为变量，不能使常量。

（3）自增运算有两种形式：前置和后置

**7.自减运算符（--）**

（1）自增运算符为一元运算符，可对操作数执行自增运算，增量为1.

（2）操作数必须为变量，不能使常量。

（3）自增运算有两种形式：前置和后置

·关系运算符用于比较运算符两端的表达式的值，确定二者的关系，根据运算结果返回一个布尔值。用关系运算符和操作数连接起来符合规则的式子，称为关系表达式。

**1.等于运算符（==） 2.不等于运算符（!=） 3.小于运算符（<）**

**4.小于等于运算符（<=）5.大于运算符（>）6.大于等于运算符（>=）**

·逻辑运算符用于执行逻辑运算，其操作数都应该是布尔型数值和表达式或者是可以转换为布尔型的数值和表达式，其运算结果返回true或false。用逻辑运算符和操作数连接起来符合规则的式子，称为逻辑表达式。

**1.逻辑与运算符（&&） 2.逻辑或运算符（||）**

**3.逻辑非运算符（!）先计算操作数的布尔值，然后取反**

·赋值运算符要求其左操作数是一个变量、数组元素或对象属性，右操作数是一个任意类型的值，可以为常量、变量、数组元素或对象属性。**赋值运算符的作用就是将右操作数的值赋给左操作数。**

**·位运算符是对二进制表示的整数进行按位操作的运算符。**如果操作数是十进制或者其他进制表示的整数，**运算前先将这些整数转换成32位的二进制数字**，如果操作数无法转换成32位的二进制数表示，位运算的结果为NaN。

**1.& 按位与运算符 2.~ 按位非运算符**

**3.| 按位或运算符 4.^ 按位异或运算符**

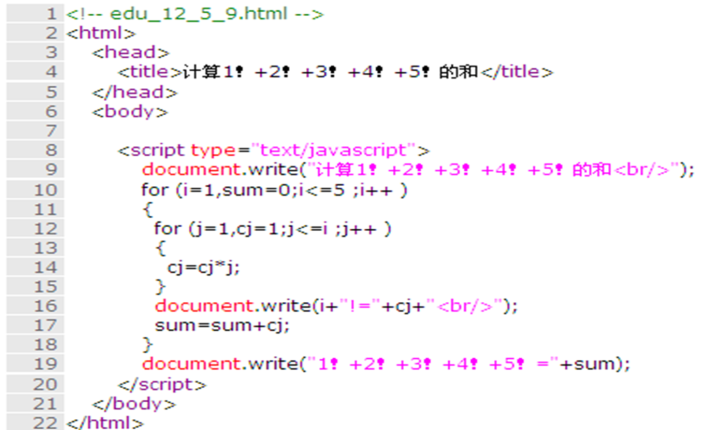
**·条件运算符是一个3元运算符，也就是该运算涉及3个操作数。**

**基本语法： <变量>=表达式1 ? 表达式2 : <表达式3>**

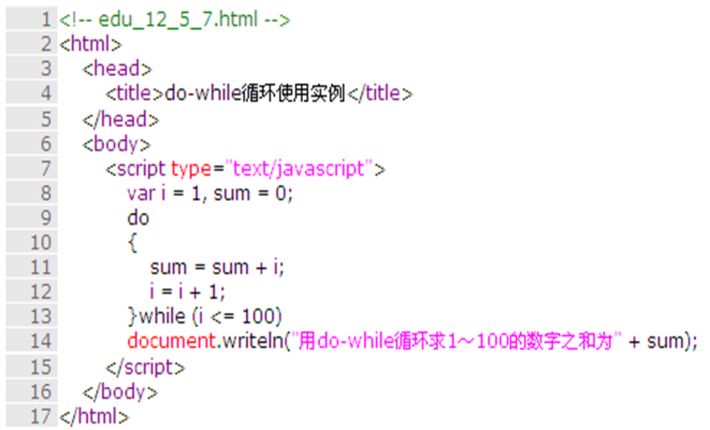
如果表达式1的结果为真，则将表达式2的值赋给变量，否则将表达式3的值赋给变量。

·JavaScript的出现使得网页和用户之间**实现了一种实时性的、动态的、交互性的关系**，使网页包含更多活跃元素和更加精彩的内容。

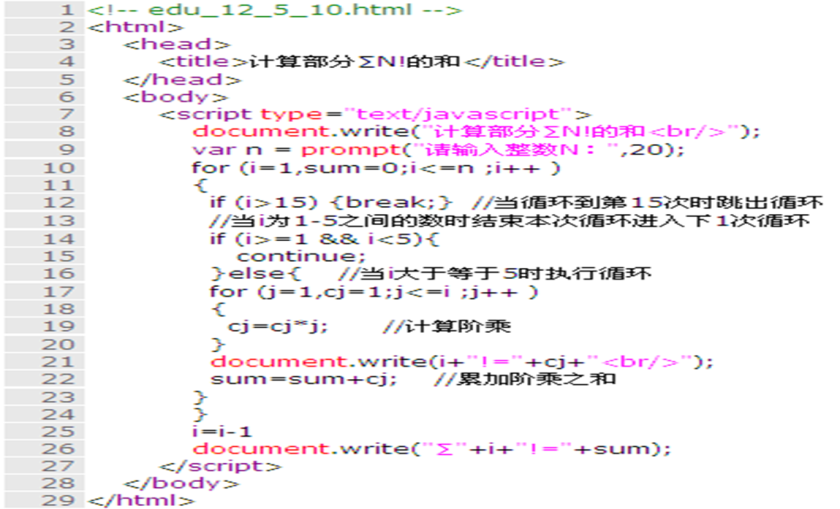
**【例12-5-9】计算1！+2！+3！+4！+5！的和。**



**【例12-5-7】使用do-while语句计算1～100之和**



**【例12-5-10】计算5！+6！+…+n！的和(5≤n≤15)。**



**函数定义**

**(1)有参函数的定义：**

function 函数名(var1,var2,...,varX)

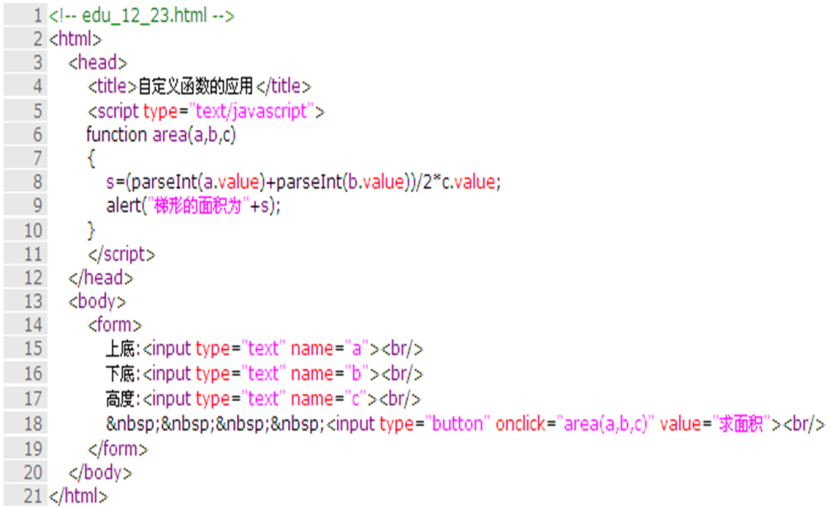
{ 代码... }

**(2)无参函数的定义：**

function 函数名()

{ 代码... }

**【例12-6-7】使用自定义函数求梯形面积**



·**事件**是一些可以通过脚本响应的页面动作。当用户按下鼠标键或者提交一个表单，甚至在页面上移动鼠标时，事件就会出现。**鼠标事件 、键盘事件 、HTML事件 、突变事件。**

**·**可以通过设置页面元素的事件处理句柄可以将页面元素的特定事件和一段事件处理代码关联起来。通常，**事件句柄的命名原则是在事件名称前加上前缀on**。**如鼠标移动MouseOver事件，其事件句柄为onMouseOver。<HTML标签 事件句柄= “JavaScript代码”> </HTML标签>**



**·事件处理方式**有以下3种：**在HTML标记中静态指定的处理方式、在JavaScript中的动态指定处理方式及特定对象的特定事件处理方式。**

**①在HTML标记中的静态指定**

<HTML标记 ... 事件句柄1="事件处理程序" [事件句柄2 = "事件处理程序" ...]></HTML标记>

**②在JavaScript 中动态指定**

<事件主角-对象>.<事件>=<事件处理程序>;

**③编写特定对象特定事件的JavaScript**

<script type=“text/javascript” for=“对象” event=“事件”> //事件处理程序代码</script>

**·事件处理程序的返回值：**在JavaScript中通常事件处理程序不需要有返回值，这时浏览器会按默认方式进行处理。很多情况下需要使用返回值，来判断事件处理程序是否正确进行处理，或者通过这个返回值来判断是否进行了下一步操作。在这种情况下，事件处理程序返回值都为布尔型值，如果为false则阻止浏览器的下一步操作，如果为true则进行默认的操作。

**·JavaScript脚本所要做的主要工作就是表单验证，如验证用户是否有未填信息，输入的数据格式是否正确等等。**

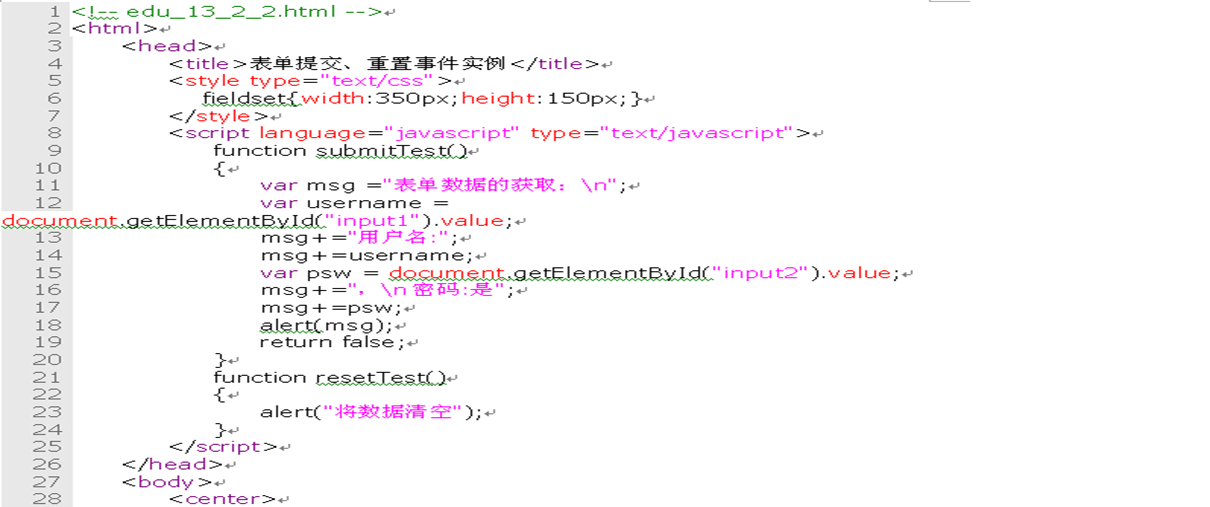
**·**当表单中的元素获得焦点时会触发Focus获得事件，当表单中的元素失去焦点时会触发Blur失去焦点事件。

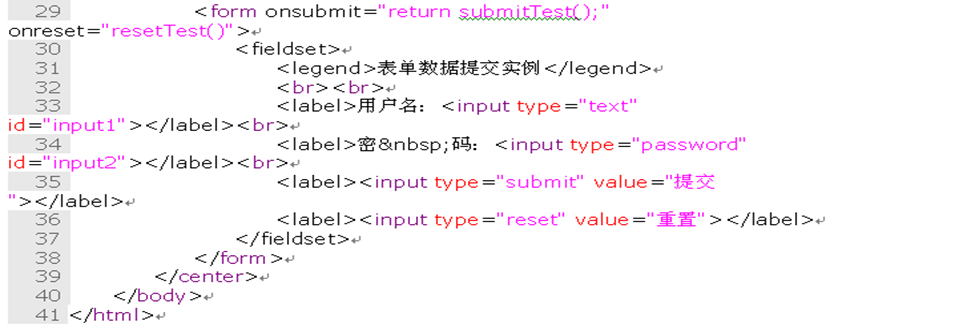
****

①表单中的Submit事件触发后会将表单中的数据提交到服务器端，Reset事件触发后会将表单中的数据重置为初始值。

②在表单中，可以通过将<input>元素的type属性值设置为”submit”来得到一个表单提交按钮，当点击该按钮时会触发表单的Submit事件 。

③通过将<input>元素的type属性值设置为”reset”来得到一个表单重置按钮，当点击该按钮时会触发表单的Reset事件。





【例13-2-3】下拉框Change元素改变事件的应用



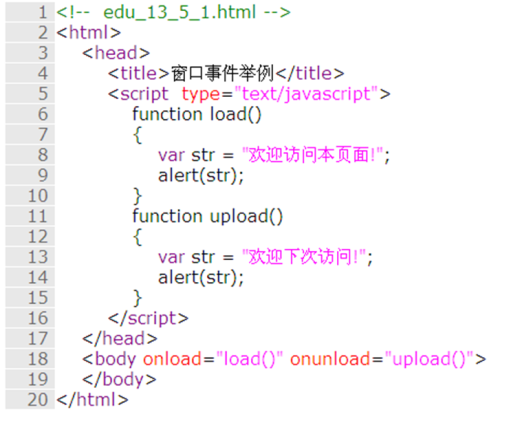
·**鼠标点击事件主要指用鼠标对页面中的元素进行单击及双击操作时触发的事件，它们也是网页开发中运用最多的事件，当鼠标单击页面中的按钮时可以触发鼠标单击事件**，例如<input type="button" name="click" value="鼠标单击" onClick="alert('你单击了我 !')">

·**当鼠标双击页面中的按钮时可以触发鼠标单击事件**，例如<input type="button" name="click" value="鼠标双击" onDblClick="alert('你双击了我 !')">

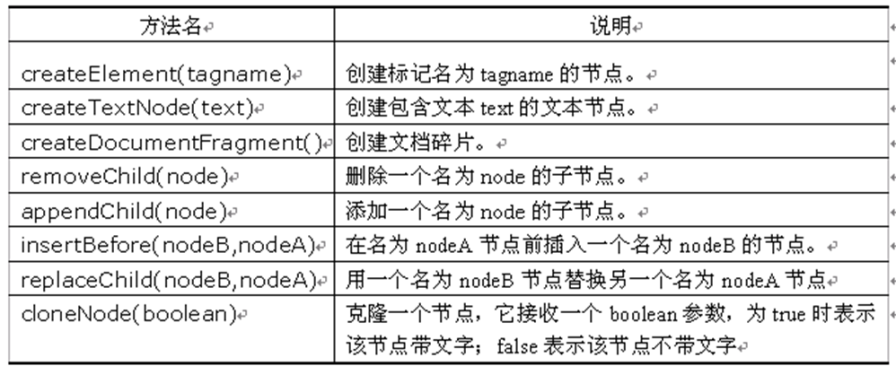
**【例13-3-2】通过鼠标移动替换图片**

****

**·窗口事件是指浏览器窗口在加载页面或卸载页面时触发的事件。加载页面时会触发Load事件，卸载页面时会触发UnLoad事件，这两个事件和<body>及<frameset>两个页面元素有关。**

****

****

****