## 网页常用单词代码：(书p23)

Left设置标题文字左对齐 （书p28）

Right设置标题文字右对齐

Center设置标题文字居中对齐

如：<h2 align=”right”>2级标签右对齐</h2>

水平线标签<hr />（书p29）

|  |  |
| --- | --- |
| 属性名 | 含义 |
| align | 设置对齐方式 |
| size | 设置水平线粗细 |
| color | 设置颜色 |
| width | 设置宽度 |

如：<hr color=”red”>

换行标签<br / >（强制换行）

文本样式标签（书p31）

<font 属性=”属性值”>文本内容</font>

|  |  |
| --- | --- |
| 属性名 | 含义 |
| face | 设置文字字体 |
| size | 设置字号大小，1~7 |
| color | 设置文字颜色 |

如：<p><font face=”宋体” size=”7” color=”red”>

文本</font></p>

文本格式化标签（书p32）

|  |  |
| --- | --- |
| 标签 | 效果 |
| <b></b>  <strong></strong> | 文字以粗体方式显示 |
| <u></u>  <ins></ins> | 文字加下画线显示 |
| <i></i>  <em></em> | 文字斜体显示 |
| <s></s>  <del></del> | 文字加删除线显示 |

文本语义标签（书p33）

|  |  |
| --- | --- |
| 标签 | 含义 |
| time | 定义时间或日期  可以为24小时某一时间 |
| datetime | 定义相应的时间日期  不定义时由文本定义 |
| pubdate | 定义发布日期  一般取值为pubdate |

如：<p>我们早上<time>9.00</time>上班</p>

我们<time datetime=”2015-1-1”>1号</time>去旅游

<time datetime=”2015-1-1” pubdate =” pubdate”>

本消息发布于2015年1月1日

<mark>标签——文本显示高亮

（书p35）

|  |  |  |
| --- | --- | --- |
| 特殊字符 | 描述 | 字符对应代码 |
|  | 空格符 | &nbsp; |
| < | 小于号 | &lt; |
| > | 大于号 | &gt; |
| & | 和号 | &amp; |
| ￥ | 人民币 | &yen; |
| © | 版权 | &copy; |
| ® | 注册商标 | &reg; |
| ° | 摄氏度 | &deg; |
| ± | 正负号 | &plusmn; |
| x | 乘号 | &times; |
| ÷ | 除号 | &divide; |
| A2 | 平方2（上标） | &sup2; |
| A3 | 立方3（上标） | &sup3; |

图像标签<img/>（书p36）

|  |  |  |
| --- | --- | --- |
| 属性 | 属性值 | 描述 |
| src | url | 图像的路径; |
| alt | 文本 | 图像不能显示时替换文本 |
| title | 文本 | 指针悬浮显示时显示文本 |
| width | px | 图像宽度 |
| height | px | 图像高度; |
| border | 数字 | 图像边框宽度 |
| vspace | px | 图像顶部和底部空白（垂直边距） |
| hspace | px | 图像左侧和右侧空白（水平边距） |
| align | left | 图像左对齐 |
| right | 图像右对齐 |
| top | 图像顶部和文本第一行文字对齐，  其他文字居图像下方 |
| middle | 图像水平中线和文本第一行文字对齐，其他文字居图像下方 |
| bottom | 图像底部和文本第一行文字对齐，  其他文字居图像下方 |

<img src=”对应的路径” title=”百搭，时尚” alt=”不能显示” />

<img src=”” height=”600” width=”300” />

<img src=”” height=”40%” width=”30%” />

<img src=”” border=”2”/>实际中不建议直接使用img标签

CSS样式（书p44）

1.内嵌式

<style type=”text/css”></style>

2.外链式

<head>

<link href=”CSS文件路径” type=”text/css” rel=”stylesheet” />

</head>注意位于头部标签中

CSS基础选择器（书p49）

1.标签选择器（优先级第三）(为1)

P{属性1：属性值1； 属性2：属性值2；}

2.类选择器（优先级第二）(为10)

.类名{属性1：属性值1； 属性2：属性值2；}

3.id选择器（优先级第一）（唯一性）(为100)

#id名{属性1：属性值1； 属性2：属性值2；}

4.通配符选择器（优先级第四）

\*{属性1：属性值1； 属性2：属性值2；}

文本样式

CSS字体样式属性（书p52）

font-size字号大小

|  |  |
| --- | --- |
| 相对长度单位 | 说明 |
| em | 倍率，相对于当前对象文本的字体尺寸 |
| px | 像素 |
| 绝对长度单位 | 说明 |
| in | 英寸 |
| cm | 厘米 |
| mm | 毫米 |
| pt | 点 |

Font-family字体

注：1.各种字体之间必须用英语逗号隔开

2.中文字体需要加英文的引号

Font-weight字体粗细

|  |  |
| --- | --- |
| 值 | 描述 |
| normal | 默认值 |
| bold | 定义粗体字符 |
| bolder | 定义更粗字符 |
| lighter | 定义更细字符 |
| 100~900 | 定义由细到粗字符  400为normal，700为bold  值越大字体越粗 |

Font-style字体风格

|  |  |
| --- | --- |
| normal | 默认值 |
| oblique | 倾斜 |
| italic | 斜体（更常使用） |

设置综合字体样式

顺序： p{

1. 字体风格 font-style:normal;
2. 字体粗细 font-weight:bold;
3. 字体大小/行高 font-size:30px; line-height:40px;
4. 字体 font-family:Arial,”宋体”;}

P{ font: normal bold 30px/40px Arial,”宋体” }

@font-face定义服务器字体名称

@font-face{ font-family：自己定义；

Src：url字体路径；}

P { font-family：定义的字体名称；}

CSS文本外观属性（书p55）

Color文本颜色

|  |  |  |
| --- | --- | --- |
| 颜色 | 十六进制 | Rgb |
| 红色red | #FF0000 | 255，0，0 |
| 绿色green | #008000 | 0，128，0 |
| 白色white | #FFFFFF | 255，255，255 |
| 黑色black | #000000 | 0，0，0 |
| 黄色yellow | #FFFF00 | 255，255，0 |
| 灰色gray | #808080 | 128，128，128 |
| 粉红色pink | #FFC0CB | 255，192，203 |
| 银色silver | #C0C0C0 | 192，192，192 |
| 褐色brown | #A52A2A | 139，35，35 |
| 蓝色blue | #0000FF | 0，0，255 |
| 紫色purple | #800080 | 128，0，128 |
| 金色gold | #FFD700 | 255，215，0 |
| 橙色orange | #FFA500 | 255，165，0 |
| 水绿色aqua | #00FFFF | 212，242，231 |

Letter-spacing：字间距

Word-spacing：单词间距

Line-height：行间距

Text-transform：文本转换

默认值none

首字母大写·capitalize

全部字符转换为大写uppercase

全部字符转换为小写lowercase

Text-decoration：文本装饰

没有装饰（默认值）none

下画线underline

上画线overline

删除线line-through

Text-align：水平对齐方式

左对齐（默认值）left

右对齐right

居中对齐center

Text-indent：首行缩进

（em/px/% 允许使用负值）

仅适用于块级元素，行内元素无效

White-space：空白符处理

默认值normal

保留空格，空行，原样显示pre

强制文本不能换行nowrap

Text-shadow：阴影效果

（水平和垂直距离可以为负值）

选择器{text-shadow：水平阴影距离 垂直阴影距离

模糊半径 阴影颜色； }

（可以多个阴影，中间用逗号隔开）

{ text-shadow:10px 10px 10px red,20px 20px 20px blue;}

Text-overflow：表示对象溢出文本

clip修剪溢出文本，不显示省略标签…

ellipsis用省略标签代替被修剪文本

（overflow：hidden 修剪溢出文本）

Word-wrap

Normal默认允许断字点换行

Break-word在长单词或url地址内部换行

CSS复合选择器（书p64）

1.标签指定式选择器（两个选择器之间不能有空格）

P.head{color：red}

则只会影响<p class=”head”>标签

不会影响其他head类标签

2.后代选择器（中间必须用空格隔开）

P strong{ color：red}

当p标签嵌套strong标签时，就可以对p中的strong进行控制

P strong em{ color：red }（不限于2个元素）

控制strong中的em

3.并集选择器（通过逗号连接）

如h2和h3想要蓝色，h2和h4加下画线

h2,h3{color：blue;}

h2,h4{ text-decoration: underline;}

应用style的元素拥有巨大的优先级

P{ h4 style=“color：red” }

Important最大优先级

#header{ color=red !important }

CSS属性选择器（书p74）

1.E[att^=value]选择器

P[id=”one”] p元素中id属性以one开头就会被选中

2. E[att$=value]选择器

P[id=”one”] p元素中id属性以one结尾就会被选中

3. E[att\*=value]选择器

P[id=”one”] p元素中id属性包含one就会被选中

CSS关系选择器（书p77）

1.子元素选择器（用于选择某一个元素的第一级子元素）

h1>strong h1后面的strong元素生效

2.临近兄弟选择器（前后两个元素，必须为紧跟元素）

h1+strong

h1后面的strong元素生效，第二个即使相同也不会生效

3.普通兄弟选择器（两个元素有同一个父亲，不用紧跟）

h1~strong h1后面的strong都会生效

结构化伪类选择器（书p80）

1.：root选择器（对页面所有标签都生效）

：root{ background：red； }

不需要该样式时可以单独覆盖

2.：not选择器

P \*：not（h3）对p中的h3不生效

3.：only-child选择器

匹配某父元素的唯一一个子元素

Strong：only-child{color：red；}

当p标签中只有一个元素且为strong时，strong生效

4.：first-child / ：last-child 选择器

用于选择父元素中第一个/最后一个元素

5.:nth-child（n）/:nth-last-child（n）选择器（n一般为2）

用于选择某一个父元素中第n个/最后n个子元素

6.:nth-of-type(n) / :nth-last-of type选择器

用于选择特定类型的第n个子元素和倒数第n个元素

奇数行odd

偶数行even

7:empty选择器

用于选择没有子元素或者文本内容为空的所有元素

伪元素选择器（书p86）

：before伪元素选择器（在某元素之前插入内容）

：after伪元素选择器（在某元素之后插入内容）

（必须配合content元素使用）

P：before{content：url图片路径；}

盒子模型（书p92）

1.边框样式设置border-style（必要）

|  |  |
| --- | --- |
| none | 没有边框 |
| solid | 单实线边框 |
| dashed | 虚线边框 |
| dotted | 点线边框 |
| double | 双实线边框 |

2.边框宽度border-width（默认为1px）

3.边框颜色border-color

注：如使用百分比颜色时，0也要加%

4.综合设置边框（顺时针顺序）

（4个属性时：上，右，下，左）

（3个属性时：上，左右，下）

（2个属性时：上下，左右）

（1个属性时：全生效）

Border-top-style上边框样式

Border-right-color右边框颜色

Border-bottom-width下边框宽度

Border-left-style左边框样式

P{ border：5px solid red}

5.内边距padding（可以%数值）（同下）

Padding-top上内边距

Padding-bottom下内边距

Padding-left左内边距

Padding-right右内边距

6.外边距margin（可以为负值）（同上）

margin-left：50px；margin-rifht：40px；等价于

margin{ 0 40px 0 50px；}（顺时针顺序）

为方便控制网页

\*{margin：0； padding：0； border：0；}

可用height width控制盒子的宽和高

.box{ height：200px； width：200px；}

背景属性（书p101）

1.背景颜色background-color

2.背景图像background-image：url

3.背景图像平铺background-repeat

|  |  |
| --- | --- |
| repeat（默认值） | 沿水平和垂直平铺 |
| no-repeat | 不平铺(位于左上角，仅显示一次) |
| repeat-x | 水平方向平铺 |
| repeat-y | 垂直方向平铺 |

4.背景图像位置background-position（可以px或%）

水平方向：left center right

垂直方向：top center bottom

5.背景图像固定background-attachment

scroll（默认）图像随页面一起滚动

fixed图像固定于屏幕上，不随页面滚动

6.综合设置（中间空格隔开，实际无顺序要求）

background:颜色 图像位置 平铺 水平位置 垂直位置 固定

新增盒子属性（书p106）

透明度

1.Rgb模式（透明度0~1）

P{background-color：rgb（255，0，0，0.5）}

0.5指的为透明度（0完全透明）（1完成不透明）

2.opacity属性

opacity：0.5；

圆角border-radius（px或%）

{水平半径参数1 水平半径参数2 水平半径参数3 水平半径参数4 /

垂直半径参数1 垂直半径参数2 垂直半径参数3 垂直半径参数4；}

（一个数值时为四个角半径相同）

（两个数值时为 左上角右下角 右上角左下角）

（三个数值时为 左上角 右上角 左下角 右下角）

（四个数值时为 左上角 右上角 右下角 左下角）

图片边框

|  |  |
| --- | --- |
| 属性 | 描述 |
| border-image-source | 图片路径 |
| border-image-slice | 图片上右下左偏移量（裁切位置） |
| border-image-width | 边框宽度 |
| border-image-outset | 边框背景向盒子外部延伸距离 |
| border-image-repeat | 图片平铺方式 |

border-image-repeat：stretch图片填充方式为拉伸填充

阴影box-shadow（书p111）

|  |  |
| --- | --- |
| 参数 | 描述 |
| h-shadow | 水平阴影距离 |
| v-shadow | 垂直阴影距离 |
| blur | 阴影模糊半径 |
| spread | 阴影拓展半径 |
| color | 阴影颜色 |
| outset/inset | 外阴影/内阴影 |

{ 5px 5px 10px 2px #999 inset; }

渐变（书p112）

1.线性渐变

(沿着一条直线过渡到结束颜色)

background-image:linear-gradient

(渐变角度，颜色值1，颜色值2，);

注：可在颜色值后面标注百分比，

用于标示颜色的位置(空格隔开)

（30deg,red, 50%,blue 80%）;

重复线性渐变Background-image:repeating-linear-gradient

|  |  |  |
| --- | --- | --- |
| 渐变角度是水平线和渐变线之间的夹角（deg为单位）  或在后面加to | | |
| to bottom | 180deg | 下方 |
| to left | 270deg | 左方 |
| to right | 90deg | 右方 |
| to top | 0deg | 上方 |

径向渐变background-image:radient-gradient

（渐变形状 圆心位置,颜色值1,颜色值2,……）;

重复径向渐变background-image:repeating-radient-gradient

1. 渐变形状

circle:指定圆形的径向渐变

ellipse:指定椭圆形的径向渐变

50px 80px:水平80px，垂直半径50px的椭圆形

（2）圆心位置（确定元素渐变中心的位置）

使用at加上关键词或参数来确定渐变中心位置

参数为像素或百分比（可以为负值）

（ellipse at center，red，blue）

（ellipse at 50% 90%，red，blue）

多背景图像（书p116）

background-image:url(图片1), url(图片2), url(图片3),

修剪背景图像background-size

（属性值1宽度，属性值2高度）

px，%：只设置一个则第二个默认为auto

cover：把背景图像扩展足够大

contain：扩展到最大尺寸，完成适合内容区域

设置背景的显示区域background-origin

padding-box:背景图像相对于内边距区域来定位

border-box: 背景图像相对于边框来定位

content-box: 背景图像相对于内容框来定位

设置背景图像的裁剪区域background-clip

padding-box: 从内边距向外裁剪背景

border-box: 默认，从边框区域向外裁剪背景

content-box: 从内容区域向外裁剪背景

元素类型（书p120）

块元素（独自占1行或多行，可设置宽度，高度，对齐等）

<h1>~<h6>，<p>，<div>，<ul>，<ol>，<li>等

行内元素（一般不占独立的区域）

<strong><b><em><i><del><s><ins><u><a><span>等

<span>为范围，可控制内容

<div><span class=”one”>内容<span><div>

元素类型的转换display

（行内元素需要设置宽度和高度

或块元素需要拥有某些行内元素的特性）

|  |  |
| --- | --- |
| inline | 此元素显示为行内元素 |
| block | 此元素显示为块元素 |
| Inline-block | 此元素显示为行内快元素，  可以设置宽高对齐属性，不会独占一行 |
| none | 此元素将被隐藏，不显示页不占页面空间 |

列表标签（书p131）

无序列表<ul>

（是不分顺序的列表，没有顺序级别之分，内部可以嵌套多个<li>标签）

每队<ul 至少嵌套一对<li>标签

type属性用于指定项目符号

|  |  |
| --- | --- |
| type属性值 | 显示效果 |
| dise默认值 | ⚫ |
| circle | ⭘ |
| square | ◼ |

有序列表<ol>

每对<ol>标签至少嵌套一对<li>标签

|  |  |  |
| --- | --- | --- |
| 属性· | 属性值（类型） | 描述 |
| type | 1默认 | 项目符号显示为数字1,2,3 |
| A或a | 项目符号显示为ABC abc |
| I或i | 项目符号显示为I ii iii |
| start | 数字 | 规定项目符号的起始值 |
| value | 数字 | 规定项目符号的数字 |

定义列表<dl>

包含三个列表<dl><dt><dd>

三个标签之间不允许出现其他标签

<dl>标签必须和<dt>标签相连

<dl>

<dt>内容</dt>

<dd>内容1</dd>

<dd>内容2</dd>

<dd>内容3</dd></dl>

CSS控制列表样式

list-style-type属性 ul{List-style-type：disc}

|  |  |  |  |
| --- | --- | --- | --- |
| 属性值 | 描述 | 属性值 | 描述 |
| disc | 实心圆（无序） | none | 不使用 |
| circle | 空心圆（无序） | Cjk-ideographic | 简单的表意数字 |
| square | 实心方块（无序） | georgian | 传统的乔治尼亚编号 |
| decimal | 阿拉伯数字 | decimal-leading-zero | 以0开头的阿拉伯数字 |
| lower-roman | 小写罗马数字 | upper-roman | 大写罗马数字 |
| lower-alpha | 小写英文字母 | upper-alpha | 大写英文字母 |
| lower-latin | 小写拉丁字母 | upper-latin | 大写拉丁字母 |
| hebrew | 传统希伯来编号方式 | armenian | 传统的亚美尼亚编号 |

list-style-image属性（引入图片项目符号）

<ul><li></li></ul>则为ul引入，li使用

list-style-position属性（控制项目符号位置）

inside：列表项目符号位于文本以内

outside：列表项目符号位于文本以外（默认值）

list-style属性（综合设置列表样式属性）

list-style:列表项目符号 列表项目符号的位置 列表项目图像；

超链接（书p139）

<a href=”跳转目标” target=”目标窗口的弹出方式”>文本或图像</a>(可设置border=“0”,清除图像边框效果)

href：链接目标的地址

Target：指定页面的打开方式

\_self：默认值，在原窗口打开

\_blank：在新窗口中打开

锚点链接：直接跳转到指定位置

<a href=”#one” >美美</a>

<h3 id=”one”>美美</h3>

链接伪类控制超链接（按表格从上到下顺序设置）

|  |  |
| --- | --- |
| 超链接标签a的伪类 | 描述 |
| a:link{CSS样式规则;} | 超链接默认样式 |
| a:visited{CSS样式规则;} | 超链接被访问后样式 |
| a:hover{CSS样式规则;} | 鼠标指针经过，悬停样式 |
| a:active{CSS样式规则;} | 鼠标点击不放时样式 |

a:link ，a:visited{color:red;}

设置默认和被访问后的颜色为红色

表格和表单（书p148）

<table><tr><td></></></>

<table>用于定义表格的开始和结束

<tr>代表行，有几个tr就有几行

<td>定义单元格tr中有几个就有几列

，<th>头部标签

<table>标签属性 <table border=”20’’ height=”400’’>

|  |  |  |
| --- | --- | --- |
| 属性 | 描述 | 常用属性值(默认) |
| border | 表格边框，默认为0 | px |
| cellspacing | 单元格与单元格之间空间 | px（2） |
| cellpadding | 单元格内容与单元格边缘空间 | px（1） |
| width | 表格宽度 | px |
| height | 表格高度 | px |
| align | 表格网页中水平对齐方式 | left,center,right |
| bgcolor | 表格背景颜色 | rgb |
| background | 表格背景图像 | url地址 |

<tr>标签属性

|  |  |  |
| --- | --- | --- |
| 属性 | 描述 | 常用属性值 |
| height | 设置行高度 | px |
| align | 一行内容水平对齐方式 | left,center,right |
| valign | 一行内容垂直对齐方式 | top,middle,bottom |
| bgcolor | 设置行背景颜色 | rgb |
| background | 设置行背景图像 | url地址 |

<td>标签属性

|  |  |  |
| --- | --- | --- |
| 属性名 | 含义 | 常用属性值 |
| width | 单元格宽度 | px |
| height | 单元格高度 | px |
| align | 单元格内容水平对齐方式 | left,center,right |
| valign | 单元格内容垂直对齐方式 | top,middle,bottom |
| bgcolor | 单元格背景颜色 | rgb |
| background | 单元格背景图像 | url地址 |
| colspan | 单元格横跨的列数  用于合并水平方向单元格 | 正整数 |
| rowspan | 单元格竖跨的行数  用于合并垂直方向单元格 | 正整数 |

表格的名称<caption></>（书p157）

<thead></>定义表格的头部

（包含网页logo，导航头部信息）

<tfoot></>定义表格的尾部

（位于thead之后）

<tbody></>定义表格主体内容（位于thead,tfoot之后）

CSS样式表格控制（书p158）

bordr-collapse：collapse去除单元格与边框之间的空间

<th><td>标签（tr标签无内外边距属性）

可以设置内边距padding属性，外边距margin属性无效

可为单元格单独设置宽高（同一行/列取决于其中较大者）

表单（书p163）

创建表单

<form action=”url地址’’ method=”提交方式” name=“表单名称”></.form>

url地址：可以为相对路径或绝对路径，或邮箱地址

method：get（默认）提交少量数据，保密性差，浏览器与表单服务器建立链接

post：分段发送，可以提交大量数据

表单控件<input type=”控件类型” />

|  |  |  |
| --- | --- | --- |
| 属性 | 属性值 | 描述 |
| type | text | 单行文本输入框 |
| password | 密码输入框 |
| radio | 单选按钮 |
| checkbox | 复选框 |
| button | 普通按钮 |
| submit | 提交按钮 |
| reset | 重置按钮 |
| image | 图像形式的提交按钮 |
| hidden | 隐藏域 |
| file | 文件域 |
| name | 自定义 | 控件的名称 |
| value | 自定义 | Input控件中默认文本值 |
| size | 正整数 | Input控件在页面显示宽度 |
| readonly | readonly | 该控件内容为只读  （不能编辑修改） |
| disabled | disabled | 第一次加载时禁用该控件  （默认显示为灰色） |
| checked | checked | 定义选择控件默认被选中的项 |
| maxlength | 正整数 | 控件允许输入的最多字符数 |

注：单选按钮必须有相同的name值才会生效

提交通常配合value使用\*

textarea控件，创建多行文本（书p167）

<Textarea cols=”每行中的字符数” rows=“显示的行数”>

文本内容</>

当type属性为text时可用

|  |  |  |
| --- | --- | --- |
| 其他属性 | 属性值 | 描述 |
| name | 自定义 | 控件的名称 |
| readonly | readonly | 该控件内容为只读  （不能编辑修改） |
| disabled | disabled | 第一次加载时禁用该控件  （默认显示为灰色） |

select控件，包含多个下拉菜单

<select>

<option>选项1</>

<option>选项2</>

<option>选项3</></>

|  |  |  |
| --- | --- | --- |
| 标签名 | 常用属性值 | 描述 |
| <select> | size | 下拉菜单可见数（正整数） |
| multiple | 定义Multiple=“Multiple”时  下拉菜单具有多项选择功能  (按住ctrl键的同时选择多项) |
| <option> | selected | 定义selected=“selected”时  该项为默认选中项 |

全新的input控制类型（书p173）

1.email类型<input type=”email” />

不符合Email格式，将提示错误信息

2.url类型<input type=”url” />

输入值不符合url地址格式，不允许提交

3.tel类型<input type=”tel” />

提供输入电话号码的文本框，与pattem属性配合使用

4.search类型<input type=”search” />

用户输入内容后，会记录，出现一个可以删除的图标

5.color类型<input type=”color” />

输入颜色，可通过value属性设置默认颜色

6.number类型<input type=”number” />

会检查输入数值是否正确

|  |  |
| --- | --- |
| Value | 指定输入框默认值 |
| Max | 指定输入框可以接受最大输入值 |
| Min | 指定输入框可以接受最小输入值 |
| Step | 输入域合法间隔，默认为1 |

7.range类型<input type=”range” />

提供一定范围内的输入值

|  |  |
| --- | --- |
| max | 最大值 |
| min | 最小值 |
| step | 滑动的步幅 |

8.date pickers类型<input type=” Date pickers” />

提供时间日期类型输入

|  |  |
| --- | --- |
| 时间/日期类型 | 说明 |
| date | 选取日，月，年 |
| month | 选取月，年 |
| week | 选取周，年 |
| time | 选择时间（小时和分钟） |
| datetime | 选取时间日，月，年（utc时间） |
| datetime-local | 选取时间日，月，年（本地时间） |

全新input属性（书p177页）

autofocus属性

指定页面加载后是否自动获取焦点

form属性

可以把表单内的子元素写在页面的任意位置

form属性适合所有input输入类型，

使用时只需引用表单所属id即可

<form action=”#” method=”post” id=”one”>

</form>

<input type=”text” form=”one”/>

list属性

datalist属性实现数据列表的下拉效果，则list属性能指定输入框所绑定的datalist元素，其值为datalist的某个id

<input type=”url” list=”one”/>

<datalist id=”one”>

<option label=”百度” value=“http”></>

<option label=”搜狗” value=“http”></>

<option label=”京东” value=“http”></>

</></>

multiple属性

可向文本框输入多个邮箱等，用逗号隔开，表示可以选择多个文件

min max step属性

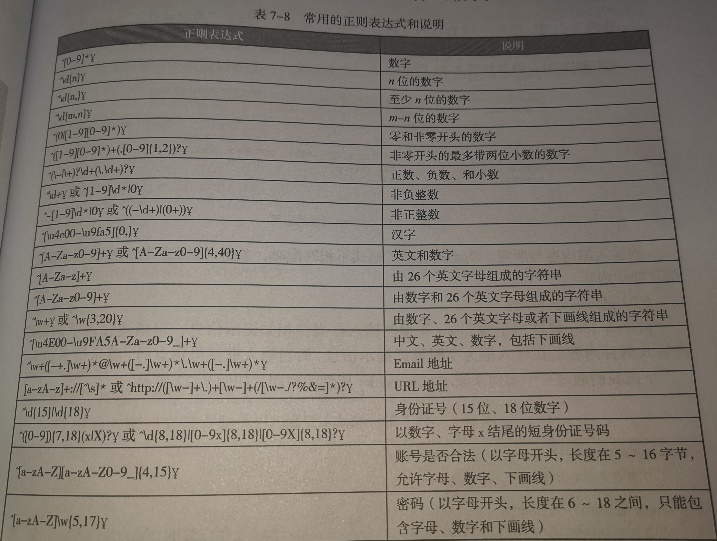
适用于date pickers number range标签

pattern属性（书p181）

适合于text ，search，url，tel Email，password

的<input>标签

常用表达式如图：



placeholder属性

为input的输入框提示相关信息

<input type=”text” Placeholder=”请输入你的名字” />

常适合于type属性为text，search，url，tel，email，password的<input>标签

required属性

<input Required=” Required” />

用于判断用户是否在表单中输入内容，当表单内容为空时不允许提交表单

常用布局属性（p193）

标签的浮动属性

选择器{float：属性值；}

|  |  |
| --- | --- |
| 属性值 | 描述 |
| left | 标签左浮动 |
| right | 标签右浮动 |
| none | 标签不浮动（默认值） |

清除浮动（p195）

选择器{cLear：属性值；}

|  |  |
| --- | --- |
| 属性值 | 描述 |
| left | 不允许左侧有浮动标签  (清除左侧浮动的影响) |
| right | 不允许右侧有浮动标签  (清除右侧浮动的影响) |
| both | 同时清除左右两侧浮动影响 |

1.使用空标签清除浮动(p197)

清除浮动对父盒子的影响（实际中不推荐使用）

.box04{clear：both；}

<div class=”box04”></>

2.使用overflow标签清除浮动

标签名{overflow:hidden;}

可以清除浮动对标签的影响，还可以弥补空标签清除浮动的不足

3.使用after伪对象清除浮动

设置时必须要对标签伪对象设置height：0样式，否则标签会比实际高度高出若干高度

必须在伪对象中设置content属性，属性值可以为空

.father：after{ display：block；Clear：both；

Content：“”Visibility：hidden；Height：0；}

<div class=father></div>

标签的定位属性（p199）

1.定位模式

选择器{position：属性值；}

|  |  |
| --- | --- |
| 值 | 描述 |
| Static | 自动定位（默认） |
| Relative | 相对定位，相对于原文档流的位置进行定位 |
| Absolute | 绝对定位  相对于上一个已经定位的父标签进行定位 |
| fixed | 固定定位，相对于浏览器窗口进行定位 |

2.边偏移（p200）

可以通过边偏移属性对标签位置进行定义

{top：200px；}

|  |  |
| --- | --- |
| 边偏移属性 | 描述 |
| top | 顶端偏移量  定义标签相对于父标签上边线的距离 |
| bottom | 底部偏移量  定义标签相对于父标签下边线的距离 |
| left | 左侧偏移量  定义标签相对于父标签左边线的距离 |
| right | 右侧偏移量  定义标签相对于父标签右边线的距离 |

定位类型

1.静态定位（标签默认定位方式）

选择器{position：static；}

静态定位下无法通过边偏移属性来改变标签的位置

2.相对定位（p202）

选择器{position：relative；}

可以通过边偏移属性改变标签的位置

3.绝对定位

选择器{position：absolute；}

会依据浏览器窗口进行定位

不论父标签位置如何偏移，

子标签相对于父标签的位置都不会改变

当left和right标签冲突时，以left为准

当top和bottom标签冲突时，以top为准

4.固定定位

选择器{position：fixed；}

不管窗口如何变化，

标签始终会显示在浏览器窗口的固定位置

布局的其他属性（p203）

1.overflow属性

选择器{overflow：属性值；}

用于规定溢出内容的显示状态

|  |  |
| --- | --- |
| 属性值 | 描述 |
| visible | 内容不会修剪，会呈现标签框之外(默认) |
| hidden | 溢出内容修剪，修剪内容不可见 |
| auto | 需要时出现滚动条，自适应显示内容 |
| scroll | 溢出内容修剪，滚动条始终可见 |

2.z-index属性

标签重叠时可以设置

属性取值为正整数，负整数，和0（默认）

属性取值越大，设置属性标签在层叠标签中越居上

布局类型（p206）

1.单列布局

<div id=”top”>头部</div>

<div id=”nav”>导航栏</div>

<div id=”banner”>焦点图</div>

<div id=”content”>内容</div>

<div id=”footer”>页面底部</div>

margin：5px auto标签上下外边距均为5px；

且位于水平居中位置

2.两列布局

<div id=”top”>头部</div>

<div id=”nav”>导航栏</div>

<div id=”banner”>焦点图</div>

<div id=”content”>

<div class=”content\_left”>内容左部分</div>

<div class=”content\_right”>内容右部分</div>

</div>

<div id=”footer”>页面底部</div>

3.三列布局（p209）

<div id=”top”>头部</div>

<div id=”nav”>导航栏</div>

<div id=”banner”>焦点图</div>

<div id=”content”>

<div class=”content\_left”>内容左部分</div>

<div class=”content\_middle”>内容中间部分</div>

<div class=”content\_right”>内容右部分</div>

</div>

<div id=”footer”>页面底部</div>

全新HTML5元素（p212）

1.header标签<header></>

用于页面头部内容，不限制header个数

2.nav标签<nav></>

用于定义导航链接

3.footer标签<footer></>

定义一个页面或区域的底部

4.article标签<article></>

代表文档，页面或程序与上下文不相关的独立部分

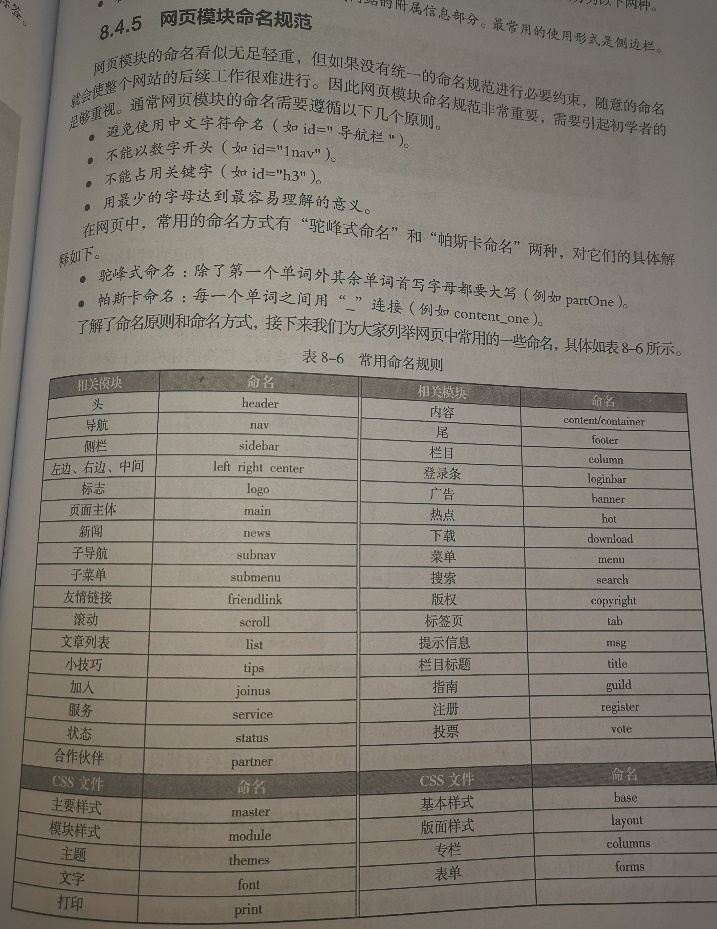
5.section标签<section></>

表示一段专题性内容，主要应用文章的章节中

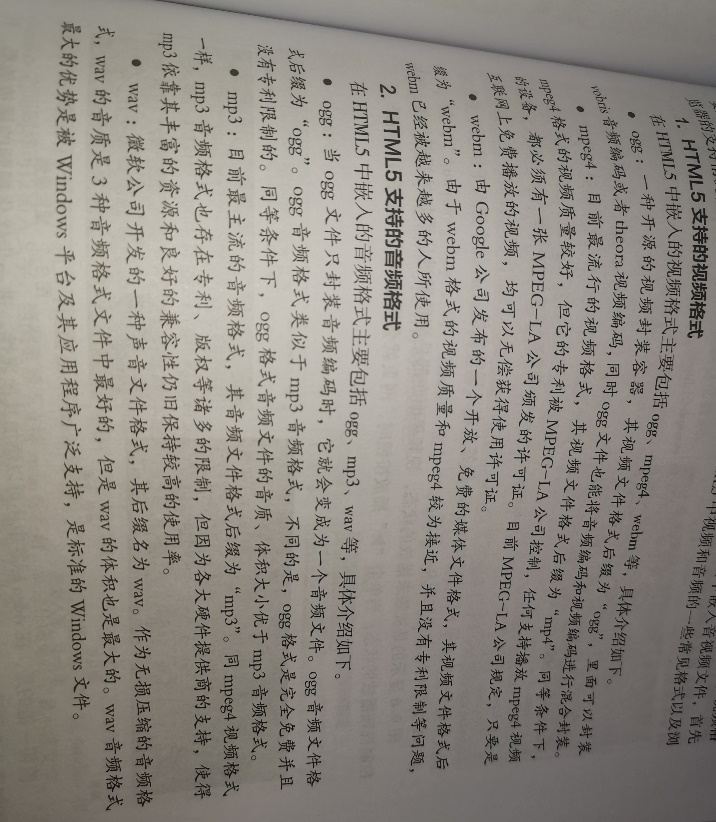
6.aside标签<aside></>

定义当前页面或主要内容相关的引用等有别于主要内容的部分

网页命名规范（p215）



多媒体嵌入（p223）



视频Video标签（p224）

<video url=”视频路径” controls=”controls” >

</video>

用于定义视频文件支持ogg，webm，mpeg4

|  |  |  |
| --- | --- | --- |
| 属性 | 值 | 描述 |
| autoplay | autoplay | 自动播放视频 |
| loop | loop | 自动重新播放 |
| preload | auto/meta/none | 预备播放 |
| poster | url | 视频播放失败时图像 |

可以直接使用height width为视频设置宽高

音频audio标签(p225)

<audio url=”音频路径” controls=”controls” >

</audio>

|  |  |  |
| --- | --- | --- |
| 属性 | 值 | 描述 |
| autoplay | autoplay | 自动播放视频 |
| loop | loop | 自动重新播放 |
| preload | auto/meta/none | 预备播放 |

url可以直接引用相应文件网址

过渡，变形，动画（p235）

过渡

1.transition-property属性

定义属性是否有过渡效果

|  |  |
| --- | --- |
| 属性值 | 描述 |
| none | 没有属性获得过渡效果 |
| all | 所有属性都获得过渡效果 |
| property  （名称） | 定义应用过渡属性css属性名称  多个名称用逗号分隔 |

transition-property:border-radius变为圆形

2.transition-duration属性

定义过渡效果持续时间

常用值为秒（s）

3.transition-timing-function属性

规定过渡效果的速度曲线

|  |  |
| --- | --- |
| 属性值 | 描述 |
| linear | 相同速度开始至结束的过渡效果  等同于cubic-bezier(0, 0, 1, 1) |
| ease(默认) | 慢速开始，然后加快，慢慢结束的过渡效果  等同于cubic-bezier(0.25, 0.1, 0.25, 1) |
| ease-in | 慢速开始，逐渐加快 ( 0.42, 0, 1, 1) |
| ease-out | 慢速结束 ( 0 , 0 , 0.58 ,1) |
| ease-in-out | 慢速开始和结束 (0.42, 0.1, 0.58 ,1) |
| cubic-bezier(n,n,n,n) | 自定义，值在0~1之间 |

4.transition-delay属性

规定过渡效果的开始时间

5. transition属性（复合属性）

transition：property duration

timing-function delay；

变形transform（p240）

1.平移

transform: translate(x-value,y-value）;

(1)x定义水平轴，y定义垂直轴坐标

(2)常用px和%为单位

(3)当参数为负数时则向反方向移动（向左和向上）

如果省略第二个坐标轴，则默认值为0，不移动

(4)单位不可省略，否则不起作用

2.缩放

transform: scale(x-value,y-value）;

(1)参数值可以为正值或负值和小数

不需要加单位

(2)x定义水平轴，y定义垂直轴坐标

(3)正数为放大元素，负数翻转缩放元素

小于1的元素用于缩小元素

3.倾斜

transform: skew(x-value,y-value）;

(1)x定义水平轴，y定义垂直轴坐标

(2)参数值为角度值，单位为deg

可以为正值，可以为负值

4.翻转

transform: rotate(angle）;

(1)angle表示要旋转的角度，单位为deg

(2)角度为正值，则按顺时针旋转；

角度为负值，则按逆时针旋转

注：如果一个元素需要设置多种变形效果，

中间用空格隔开。

5.更改变换中心点

默认情况下元素的中心点在x轴和y轴的50%位置

transform-origin：x-axis y-axis z-axis；

|  |  |
| --- | --- |
| 参数 | 描述 |
| x-axis | 定义视图被置于x轴的何处。属性值可以为em，%，px  也可以为top，right，bottom，left，center |
| y-axis | 定义视图被置于y轴的何处。属性值可以为em，%，px  也可以为top，right，bottom，left，center |
| z-axis | 定义视图被置于z轴的何处。注意该值不能是一个百分比值，否则会被示为无效值，一般单位为px |

3D变形

1.rotateX（）

transform：rotateX（a）

CSS3中，用于指定元素围绕x轴旋转

(1)a用于定义旋转的角度值，单位为deg

(2)角度为正值，则围绕x轴顺时针旋转；

角度为负值，则围绕x轴按逆时针旋转

2.rotateY（）

transform：rotateY（a）

CSS3中，用于指定元素围绕y轴旋转

（同上）

3.rotated3d（）

当想要同时设置x轴，y轴旋转时可用

rotated3d（x，y，z，angle）

xyz的取值可以取值0或1，当要沿着某轴移动则该轴设置为1，否则为0，angle为移动角度

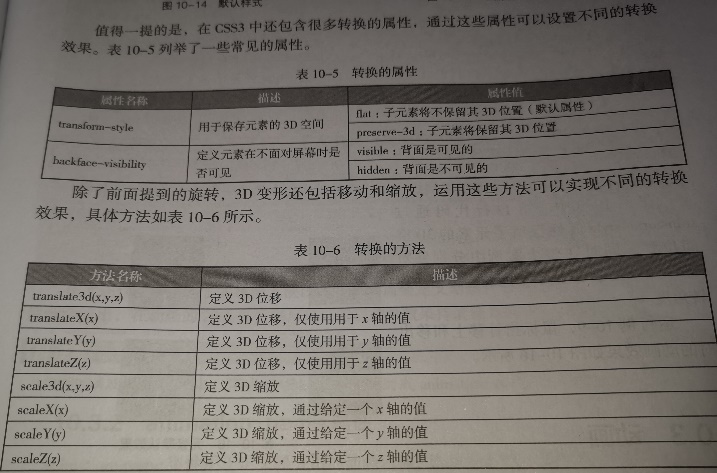
4.perspective属性

可以理解为视距，用于呈现良好的3D效果

perspective：参数值；

参数值可以为none或数值，（一般为单位px）

参数值越小，透视效果越突出



动画(p249)

1.@keyframes规则，用于创建规则

@keyframes animationname

{ keyframes-selector{css-styles;} }

(1)animationname:表示当前动画名称，不能为空

(2)keyframes-selector:关键帧选择器，即当前关键帧要应用到整个动画过程的位置，可以为%，form，to

(3)from和0%效果相同，表示动画开始

to和100%效果相同，表示动画结束

(4) css-styles:定义执行到当前关键帧的动画状态

多个属性之间用分号分隔，不能为空

@keyframes appear（动画名称）

{ 0%（开始状态）{opacity：0;} （完全透明）

100%（结束状态）{opacity：1;} （完全不透明）}

2.animation-name属性

定义动画名称，该名称会被@keyframes规则引用

animation-name：keyframesname / none；

3. animation-duration属性

定义整个动画效果完成时所需要的时间

animation-duration：time；

参数为（s）或（ms）

当设置为0时，则没有任何动画效果

当设置为负数时，则被视为0

4. animation-timing-function属性

规定动画的速度曲线，可以定义用那种方式来执行动画速率

animation-timing-function：属性值；

|  |  |
| --- | --- |
| 属性值 | 描述 |
| linear | 从头到尾的速度是相同的 |
| ease(默认) | 低速开始，然后加快，结束变慢 |
| ease-in | 低速开始 |
| ease-out | 低速结束 |
| ease-in-out | 低速开始，低速结束 |
| cubic-bezier(n,n,n,n) | 自定义，值在0~1之间 |

5.animation-delay属性

执行动画效果延迟的时间，即规定动画什么时候开始

animation-delay：time；

6. animation-iteration-count属性

用于定义动画的播放次数

animation-iteration-count：number；

当为number时，为播放次数

当为infinite时，动画循环播放

7. animation-direction属性

定义当前动画播放的方向，即动画播放完成后是否逆向交替循环

normal为默认值，动画正常播放

alternate值，奇数（1，3，5）正常播放

偶数（2，4，6）逆向播放

注：只有动画播放次数大于等于2时此属性才生效，所有应先定义animation-iteration-count属性

8. animation属性（综合）

animation-name：my；（动画名称）

animation-duration：5s；（动画时间）

animation-timing-function：linear；（动画速率）

animation-delay：2s；（动画延迟时间）

animation-iteration-count：3；（动画播放次数）

animation-direction：alternate；（动画逆向播放）

可简写为:

animation: my 5s linear 2s 3 alternate;

绘图（p257）

1.JavaScript的引入

(1)行内式（书p258，不推荐使用）

(2)嵌入式

<script type=”text/JavaScript”>

JavaScript语句；</ script >

(3)外联式(文件后缀为.js)

<script type=”text/JavaScript” src=“文件路径”>

</ script>

也可以省略type属性

2.变量

在JavaScript中，可以使用字母，数字，符号来命名变量

(1)必须以字母或下滑线开头，中间可以为数字，字母或下划线

(2)变量名不能包括空格，加，减号等

(3)不能使用JavaScript中的关键字，如var int等

(4)严格区分大小写

3.声明var变量与赋值

var unit,room; 声明

var unit=3;赋值

var room=1001;赋值

var name=’tom’,age=12;声明同时赋值

注：如果声明对象已有初始值，则声明为重新赋值

4.document对象

想要在JavaScript中操作某个标签，首选要获得该标签的属性，可通过document获取属性

|  |
| --- |
| document.getElementById( )  返回对拥有指定id名的第一个对象的引用  （简单理解为获取指定id名的标签） |
| document.getElementByName( )  返回带有指定name属性名的对象集合  （简单理解为获取指定name名的标签） |
| document.getElementByTagName( )  返回带有指定标签名的对象集合  （简单理解为获取标签名） |
| document.getElementByClassName( )  返回带有指定类名的对象集合  （简单理解为获取指定class名的标签） |

HTML5画布（p262）

1.创建画布

<canvas id=”画布名称” width=“数值”

height=“数值”></canvas>

2.获取画布

想要通过JavaScript控制画布，则必须要获取画布

var canvas=document.getElementById(‘cavs’)

获取id名为cavs的画布

3.准备画笔

通过content对象，在画布中绘制图形

canvas.getContent(‘2d’)

2d代表画布的种类，表示二维图像画图

webgl代表三维图形

注：在JavaScript中，通常会定义一个变量来保存获取的content对象

var content= canvas.getContent(‘2d’)；

绘制线

1.初始位置

moveTo(x,y)来定义初始位置

单位默认px，可省略单位

2.连接端点

content.lineTo（x，y）

绘制一个端点，并和初始位置相连接

3.描边

content.stroke（）；（括号不需要加入任何内容）

初始位置和端点的连线并不可见，需要添加描边

线的样式

1.宽度context.lineWidth=‘数值’；

2.描边颜色context.strokeStyle=‘颜色值’；

3.端点形状lineCap=’属性值’

|  |  |
| --- | --- |
| 属性值 | 显示效果 |
| butt（默认） | 无端点，显示直线方形边缘 |
| round | 显示圆形端点 |
| square | 显示方形端点 |

线的路径

1.重置端点

想要开始新的线，必须要重置端点

context.beginPath（）；

2.闭合路径

context.closePath（）；

3.填充路径

context.fill（）；

4.填充路径图形改变颜色

context.fillStyle=’颜色值’

绘制圆

arc（x，y，r，开始角，结束角，方向）

(1)x，y表示圆心的位置

(2)r表示圆的半径，用于表示圆的大小

(3)方向false为顺时针，true为逆时针（可省略）

(4)开始角，表示初始弧点位置

用Math.PI（圆周率，理解为180度）

如270度为1.5Math.PI

(5)结束角，结束时弧点的位置

