## CSS

在浏览器里面的computed下，有浏览器的默认样式表；

所以要做样式初始化：

1. 各个标签包括body,添加margin:0;padding:0;
2. a标签：text-description:none;
3. 列表标签：list-style-type:none;

书写：

1、在css样式表里书写方式：属性：属性值

2、在标签里书写方式：属性=‘属性值’

3、属性值写#表示空值

属性书写顺序：（建议）

1. 布局定位属性：display list-style position(对应的top right bottom left) float clear visibility overflow
2. 自身属性：width height margin padding border background
3. 文本属性：color font text-decoration text-align vertical-align white-space break-word
4. 其他（css3属性）:content cursor border-radius box-shadow text-shadow background:linear-gradient

如果span{width:100px;display:block}中先会读取width ,然后发现没用，再读display，读完发现width 有用，再回去读width；这是浏览器的回流机制；影响性能；

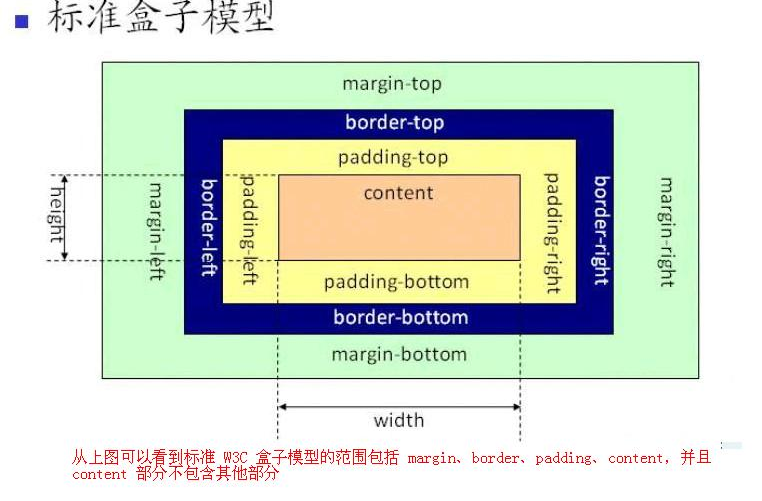
默认最小的文字大小为12PX（不能修改），火狐浏览器除外

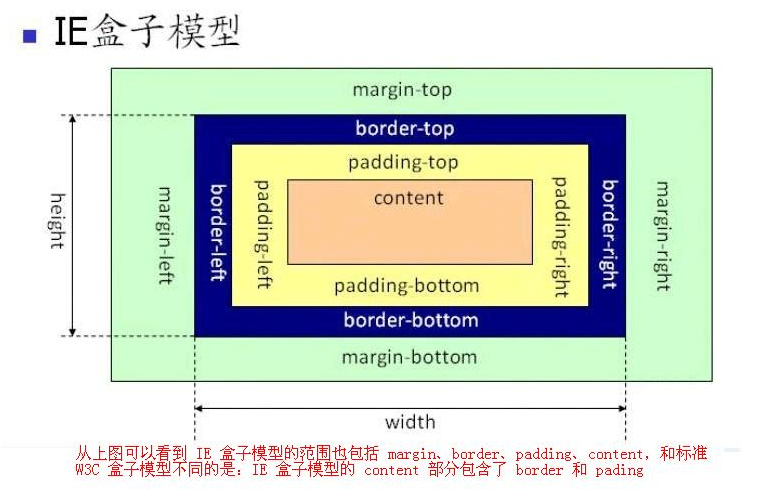
颜色值：不占位置

1. 十六进制：#000000或者#000；其中#aa0022===#a02
2. 英文颜色值：red/green
3. 三原色：rgb(0,0,0)其中的值为0~255；
4. 添加了透明度的三原色：rgba(0,0,0,0)前三个值范围为0~255，后一个为透明度：0~1；
5. Hsl:根据色盘颜色；hsl(120,50%,50%)
   1. H:颜色值（0~360）120为绿，240为蓝
   2. S:饱和度（纯色度）越大越纯0%-100%；
   3. L:亮度（关灯和开灯的区别）0~100%；低于50%为暗；高于50%为亮；

文档流：当前网页上元素的空间;css加载顺序从上到下；同等级时后面会覆盖前面；

在styles中盒子模型：





Display:规定元素类型，表示一个元素如何显示；

1. 属性值：
   1. 块级：block；
      1. 常见标签：div，h1~h6,p,ul,ol,dl,dt,dd，html,body
      2. 特征：
         1. 前后都有换行符；即独占一行；
         2. 默认宽度为父级内容宽度的100%；
         3. 能设置宽度，高度；
         4. 支持外边距；
         5. 支持内边距；
         6. 能嵌套内联和块级元素；
         7. 文档流内，两个块级垂直方向的外边距相遇，即上下都没有内边距、边框阻挡时，上下的外边距会合并，保留最大的外边距：
            1. 解决：

加上内边距或者边框阻挡；

父子级之间添加添加文本

换成行内块

设置浮动；分离层

定位：分离层

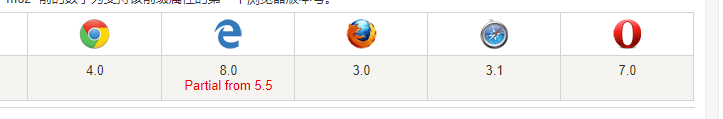
* + 1. 特殊的块级标签：P
       1. p标签不能嵌套任何块级，包括自己；
    2. 水平居中方式：
       1. Margin:0 auto;
          1. 居中条件：

自身设置有宽度

父级宽度比自身宽度大

自身有左右移动空间

* + - 1. 定位居中：
         1. 父级添加定位，自己添加相同的margin居中；
      2. 垂直居中：
         1. Line-height:父级高度；
  1. 行内：inline;
     1. 常见标签：span,img,em,b,i,strong,sup,sub,a
     2. 特征：
        1. 不能设置宽高；其宽高由内容长短决定；
        2. 能和行内/内联元素在一行；
        3. 不支持上下的外边距；
        4. 支持左右内边距；
        5. 上下内边距不占据空间，但是会对自身有效果；
        6. 不能嵌套块级，能嵌套内联、行内元素；
     3. 特殊的行内标签：img，具备行内块的所有特性
        1. 能设置宽高（是由于底层决定）；
        2. 支持内外边距
     4. 居中方式：
        1. 一排行内标签排列：（自身居中）
           1. 添加左右相同的padding;使每个左右添加空白居中；
           2. 添加左右margin;是每个左右添加空白居中；
        2. 在盒子里文字居中：
           1. 在父级中使用text-align:center;水平居中
           2. 在父级使用line-height:父级高度；设置自身文字占据高度为父级高度居中；垂直居中；
  2. 列表项：list-item
     1. 常见标签：li
     2. 特性：和block类似；但是list-item前面有标识标志；
  3. 行内块：inline-block
     1. 特性：
        1. 支持宽高；
        2. 可以和内联元素待在一行；
        3. 默认由内容撑开；
        4. 不支持margin:0 auto居中方式；
        5. 支持内外边距；
     2. 居中方式：
  4. 表格：table,作为块级表格显示；类似<table>
     1. 设置为块级表格，对外是块级，对内是表格；
     2. 其border属性为5，则会使最外边框宽度为5；如果是1，会使外边框、内边框都为1，如果为0，都是0；
  5. 表格：inline-table,作为内联表格显示;类似<table>
     1. 设置为行内表格
  6. 一个或多个行的分组表格：table-row-group,类似<tbody>
  7. 一个或多个行的分组表格：table-header-group,类似<thead>
  8. 一个或多个行的分组表格：table-footer-group,类似<tfoot>
  9. 一个表格行：table-row,类似<tr>
     1. 设置为一行；
  10. 一个或多个列的分组：table-column-group,类似<colgroup>
  11. 一个单元格列显示：table-column,类似<colgroup>
  12. 一个表格单元格：table-cell,类似<td><th>
      1. 设置为一个单元格
  13. 一个表格标题：table-caption,类似<caption>
  14. 弹性盒模型：flex;(又名弹性布局、flex布局）
      1. 在后面的的弹性盒模型中有详解；
      2. 多用于自适应，响应式；
  15. 无：none;
      1. 可以隐藏元素，并且不占据空间；使用该元素，会使原本该元素占据的空间释放，整个页面重新摆放；并且鼠标不能选中；
      2. 另一种隐藏元素方式：visibility:hidden;还会占据原本该元素的空间；
         1. 两个的差别在于，display会影响页面布局；visibility不会影响布局；

1. 兼容性：
   1. 
2. 注意：
   1. 浏览器会压缩代码中的所有距离，把所有空格距离压缩成一个空格的大小
   2. 浏览器只会解析文字间的空格；
   3. 行内、行内块本质为文字，被文档当成文字解析；所以行内块、行内元素之间的空格、换行都会被解析为空格，在浏览器中表现出一个间隔（为一个空格的大小）
      1. 去掉行内、行内块标签之间的空格：
         1. 排成一行（即标签之间不留空格）
         2. 在两个标签之间添加注释
         3. 设置父级文字大小为0；单独定义文字大小；（因为标签之间的空格产生是父级文字大小影响的空格；）
         4. 不同浏览器的默认空格不同（是因为不同浏览器默认的文字大小不一致）

Width:宽度；怪异盒模型（内容+padding+border）普通盒模型（内容）

1. 默认值：
   * 1. 块级：自动匹配父元素；
        1. 其默认值为父级内容的100%；
     2. 行内块：和块级一样；
     3. 行内：由内容撑开；（能设置，但没用）
2. 值：怪异盒模型（内容+padding+border）普通盒模型（内容）
   * 1. 百分比：
        1. 根据父级内容宽度的百分比；
     2. Auto:（怪异盒模型和普通盒模型一样）
        1. 为其父级内容的100% - padding -margin - border;
     3. 数值：带单位
        1. 为设置的大小；

Min-width:最小宽度；如果设置的是100%之类可以变动的值，当小于min-width时，会变成min-width；

Max-width:最大宽度；如果设置的是可以变动的数值，当大于max-width时，会变成max-width;

Height:高度；怪异盒模型（内容+padding+border）普通盒模型（内容）

1. 默认值：
   * 1. 行内：由内容撑开；（能设置，但是没用）
     2. 块级：没有内容为0；有内容，由内容撑开；
     3. 行内：和块级一样；
   1. 值：怪异盒模型（内容+padding+border） 普通盒模型（内容）
      1. 百分比：
         1. 父级内容的100%；
      2. Auto:(怪异盒模型和普通盒模型一样)（没有作用）
         1. 由内容撑开
      3. 数值：带单位
         1. 为设置的大小

Margin：外边距；盒子与盒子之间的距离

位置参考值：

1. 文档流外边距参考值为浏览器左上；
2. 定位的外边距参考值为定位父级左上角；
3. 盒子与盒子之间的距离，以左上的盒子为基准；

单个写法：

1.margin-top:上外边距

2.margin-right:右外边距

3.margin-bottom:下外边距

4.margin-left:左外边距

复合写法：

1. Margin : 上 右 下 左；
2. Margin: 上下 右左；
3. Margin: 上 右左 下；
4. Margin:上下左右；

值：

1. 百分比：
   1. 值为父级的width值的百分数；
      1. 数值：要加单位
         1. 值为与另一盒子（父级或同级）的间距；
      2. Auto:
         1. 行内元素：没有效果；
         2. 自动分配等值的左右margin；前提是有空间分配；
         3. 在定位中使用，会居中；

注意：

1. 行内元素的上下margin是没有效果的；左右margin有效果；
2. 浏览器上每个标签基本上都有默认边距；
3. 所有的距离都是由边距产生的（外边距或者内边距）
4. 如果子级和父级都是块级元素，并且两个内容之间没有边框（border）、内边距（padding）阻隔的话，子级和父级的上下边会合并，共享同一个margin-top/margin-bottom;

块级元素居中写法：margin:0 auto;(要确保盒子父级的空间足够)

Border：边框；只有默认值为颜色和大小；其默认为字体颜色（如果当前没有设置字体颜色，往父级查找；）；默认大小为3px;边框值不能有百分比；

单个属性写法：

1、border-width: 变框厚度；（可不写，默认为3px）none和0一样；

* + - 1. 上 右 下 左；
      2. 上下 右左；
      3. 上 右左 下；
      4. 上下左右
      5. 单边写法：
         1. Border-left-width : 大小
         2. Border-right-width: 大小；
         3. Border-bottom-width:大小
         4. Border-top-width:大小
      6. 复合写法：
         1. Border-width:大小

2、border-style:边框样式；（必须写）

1) Solid：实线

2) Dashed:虚线

3) Double:双边框

4) Dotted:点线

5) hidden:和none类似，不过在用于表单结构时，可以解决边框冲突；

6） groove:3D凹槽效果；每条边框有里外两条线，靠外边的线亮；

7） ridge:3D垄状效果；每条边框有里外两条线，靠里边的线亮；

8） inset:左、上较暗

9） outset:右、下较暗

5) 单边写法：

A. Border-left-style:样式

B. Border-right-style:样式

C. Border-top-style:样式

D. Border-bottom-style:样式

3、border-image:(css3的属性) 设置边框图片；默认值为none 100% 1 0 stretch;

1) 默认值：

* + - 1. **Border-image-source**:none;
      2. **Border-image-slice**:100%;
      3. **Border-image-width**:1
      4. **Border-image-outset**:0
      5. **Border-image-repeat**:stretch;
    1. 复合写法：
       1. Border-image: border-image-source border-image-slice border-image-width border-image-outset border-image-repeat;
    2. 单个写法：
       1. Border-image-source: 用在边框的图片的路径。
       2. Border-image-slice:图片边框向内偏移。从0到100%，从放得最大的8个图片往外收缩
          1. 百分比：

50%:完整的图和分散图的分界点；

100%：变成四个角的完整图片

0：放的最大的四个角和上下左右

* + - * 1. 数值：

图片的像素宽：变成四个角的完整图案

图片一半的像素宽：四个角分散的图案；

0：放的最大的四个角和上下左右

* + - 1. Border-image-width：图片边框的宽度。
         1. 百分百：

50%：全部占据了内容的背景；边框背景往中心缩，但是四个角是在放大；

0：完全没有边框背景；

* + - * 1. 数值：

盒子宽度的一半：占据了内容的背景；边框背景往中心缩，但是四个角是在放大；

边框背景往中心缩，但是四个角是在放大；

* + - 1. Border-image-outset : 边框图像区域超出边框的距离其值；不能为百分比；不占据空间
         1. 数值：

不带符号（倍数）

N:距离盒子N倍边框的距离绘制边框

带符号

N px:距离和在N px的距离绘制边框

* + - 1. Border-image-repeat:图像边框是否应平铺(repeat)、铺满(round)或拉伸(stretch)。
         1. Repeat:平铺；基本在上下左右平铺，四个角不会变化

图像填充区域

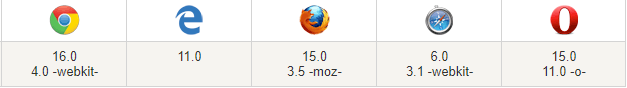
* + - * 1. Round:铺满；(感觉用处不大，herepeat一样）

如果无法完整平铺所有图像，进行缩放，以适应区域；

* + - * 1. Stretch:拉伸

只做拉伸，一张图；

* + 1. 兼容性：



1. border-radius:（css3的属性）圆角
   1. 单个写法：
      * + 1. Border-left-bottom-radius:左下角的圆角
          2. Border-left-top-radius:左上角
          3. Border-right-top-radius:右上角
          4. Border-right-bottom-radius:右下角
        1. 复合写法：
           1. Border-radius:

左上 右上 右下 左下；

左上 右上左下 右下

左上/右下 右上/左下

左上右上右下左下

* + - 1. 其值：
         1. 百分比：

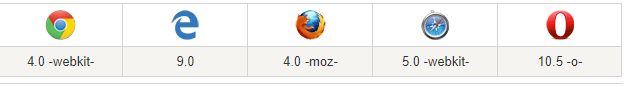
50%：正方形为圆，长方形为椭圆；

0：直角

* + - * 1. 数值：必须带单位

0：直角

宽高的一半加上边框的厚度：正方形为圆，长方形为球场形状；

* + - 1. 兼容性：
      2. 
      3. 注意：
         1. 其值具有最大值，再加大时会没效果；
         2. 设置了border-radius，画边框时，画的是圆形的边框；
      4. 算法：
         1. 数值:会以当前位置想x,y延伸至数值位置，并以其为原点画圆，显示其值长宽的范围；即先找到对应角，根据他的值延伸的元素内的点，根据点画个圆；裁剪圆外的地方；
         2. 百分比：还是获取实际的宽高，根据点延伸，之后画两个圆，从长边开始，到最后的短边长度；按比例来：0~长，90°~宽；

### Padding:内边距，盒子边框与内容之间的距离；

1. 复合写法：
   * 1. Padding:上 右 下 左
     2. Padding:上 右左 下；
     3. Padding:上下 左右；
     4. Padding:上下左右；
   1. 单个写法：
      1. Padding-left:左
      2. Padding-right:右内边距
      3. Padding-top:上内边距
      4. Padding-bottom:下内边距
      5. Padding-inline-start:元素在行开始时的边距,类似padding-left;
      6. Padding-inline-end:元素后面跟着的边距；类似padding-right;
   2. 值：
      1. 数值：带单位
         1. 按照数值大小；
      2. 百分比：
         1. 按照父级的内容宽度的百分数；（不管是怪异盒模型还是普通盒模型）
   3. 注意：
      1. 正常的盒子模型：内边距会增加盒子的大小；
         1. 盒子实际大小：内容+padding+border
         2. 盒子实际占用空间大小：内容+padding+border+margin
      2. 怪异盒模型（box-sizing:border-box）：内边距不会增加盒子的大小；增加内边距会压缩内容大小；
      3. 想要加内边距和边框，但是又不想增加盒子的实际大小，可以使用怪异盒模型；

### Box-sizing:盒子模型

1. 默认值：
   * 1. Content-box: 指定设置的宽高为盒子的宽度和高度；
2. 值：
   1. Content-box: 指定设置的宽高为内容的宽度和高度；和内边距和边框无关；
      1. 即盒子大小：内容+内边距+边框
   2. Border-box: 指定设置的宽高为盒子实际宽高；包含了内边距和边框；
      1. 盒子大小：宽高；
   3. 兼容性：
   4. 

## 背景属性：

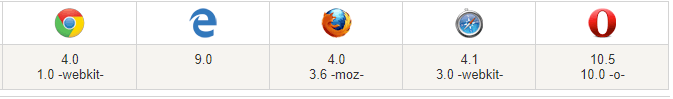
### Background:设置背景,可以设置多个，只要在其中间添加‘，’即可设置多个值；

1、单属性：如果设置多个单属性，使用‘，’隔开；并且每个位置会自动对应

1. background-color:背景颜色
   1. 默认值：
      1. Transparent:透明；
      2. 背景颜色的范围默认是：边框+padding+内容；
   2. 值：
      1. Transparent:透明,
      2. Color:
         1. Rgb(r,g,b) 值都是0~255；255是满颜色；
         2. Rgba(r,g,b,a) 和rgb相比，多了a透明度
         3. 颜色英文：red等
         4. 十六进制颜色：#000000，#000
2. Background-image:设置背景图像；
   1. 默认:
      1. 默认值：None：
      2. 图片显示位置，从padding左上角放置；
      3. 默认以图片原始尺寸放置
   2. 值：
      1. Url(‘路径’):图像的地址；
      2. None：无图片；
      3. Linear-gradient:线性渐变
      4. Radial-grdient:径向渐变；
   3. 注意：
      1. 超出部分不会占据空间；
3. Background-origin:指定background-position属性的相对位置（css3）
   1. 默认值：
      1. Padding-box:相对padding定位；
   2. 值：（以左上位置为准）
      1. Padding-box ：相对padding定位
      2. Border-box:相对border定位
      3. Content-box:相对content定位
   3. 注意：
      1. Background-attachment是固定的时候，这个属性没有作用
   4. 兼容性：



1. Background-position:指定背景图片的起始位置；背景定位；使用雪碧图/精灵图；参考位置为background-origin;
   1. 默认值：
      1. 0% 0%
      2. 只写一个值，另一个默认为center/50%;
      3. 使用百分比时，它的计算方式是根据图片的百分比盒子和盒子的百分比位置重合；
   2. 值：
      1. 关键字：
         1. Left top:background-origin的左上角对齐；
         2. Left bottom:对齐background-origin的左下角；
         3. Left center:对齐background-origin的左边
         4. Left:和left center一样
         5. Right top:对齐background-origin右上角
         6. Right bottom:对齐background-origin右下角
         7. Right center：对齐background-origin右边
         8. Right:和right center一样
         9. Center:拉伸
         10. Center center：和center一样
         11. Center top:对齐background-origin上面
         12. Center bottom:对齐background-origin下面
      2. 百分比：
         1. 0% 0% :左上角；正值：左上；
         2. 100% 100%：右下角；
      3. 数值：
         1. 0 0：左上角，正值：右下；
   3. 单个方向：
      1. Background-position-x:只往x轴移动；
         1. 百分比：正值往左(不推荐）
         2. 数值：正值往右
         3. Left/right;左/右
      2. Background-position-y:只往Y轴移动
         1. 百分比：正值往上（不推荐)
         2. 数值：正值往下
         3. Top/bottom:上/下
2. Background-repeat:设置图像的平铺方式；
   1. 默认值：
      1. Repeat;水平、垂直平铺；
   2. 值：
      1. Repeat:水平、垂直平铺；
      2. No-repeat:不平铺；
      3. Repeat-x:只水平平铺
      4. Repeat-y:只垂直平铺
3. Background-size:设置图像的大小；（css3）
   1. 默认值：
      1. Auto;
   2. 值：只写一个，另一个为Auto
      1. 百分比：
         1. 根据background-origin的指定的位置的百分比；
         2. 0 0为宽高为0；100% 100%为参考位置的大小；
      2. 数值：
         1. 直接设置图片大小；
         2. 0 0 为宽高0；
      3. Cover：根据短边缩放；有可能缺少一部分；会完全覆盖；（保持比例）
      4. Contain:根据长边缩放：完整一张图，有可能多出一部分；（保持比例）
   3. 兼容性：



1. Background-attachment:设置背景固定，还是随滚动条滚动而滚动
   1. 默认值：
      1. Scroll
   2. 属性值：
      1. Scroll:背景图片随着页面的其余部分滚动
         1. 背景图片贴在元素中；滚动滚动条显示的是同一个画面
      2. Fixed:背景图片是固定的；
         1. 背景图片帖子网页中；但是只有快到的地方才显示；块在哪显示哪部分；滚动滚动条时显示的部分不一样，
2. Background-clip:背景的裁切方式：即背景显示的部分；
   1. Border-box:默认，border以外的背景裁切；在border以内的内容能显示背景
   2. Content-box;内容以外的背景裁切；在content以内的内容能显示背景
   3. Padding-box:内边距以外的背景裁切；在padding以内的内容能显示背景
3. 复合写法：
   1. Background:颜色 图像 位置/大小 平铺 固定模式

background-color background-image background-position/background-size background-repeat background-attachment

1. 注意：
   1. 复合属性，包含了所有的背景样式的值；
   2. 设置background会显示的部分：内容+padding+border;
   3. 能设置多个背景图，使用‘，’分开；

### 线性渐变：linear-gradient（）

1. 说明：线性渐变属于有渐变颜色的背景图片；所以其是background-image里的属性值；
   1. 颜色延规定方向由初始值渐变到最终值；
2. 其括号内值：
   1. 第一个可以写方位值：to bottom/to bottom left；默认方位为to bottom；即从上往下渐变；
   2. 第一个也可以写角度值：n deg;表示，渐变方向为角度n;从上往下是0，右上到左下，为45deg ,以此类推；
   3. 可以不写方向，就会以默认方向从上到下渐变；
   4. 后面下渐变颜色；可以写多个；自动平方空间；
   5. 可以在颜色后面加空格数值，表示当前颜色要从前一个数值到当前数值的渐变；如果前面没有其他颜色或者后面没有其他颜色，则都是最近的数值的颜色；
3. 兼容性：
   1. 要在linear-gradient()前面加前缀；
   2. 并且方位值不再是to bottom了，变成了top；要写的是开始渐变的值；
   3. 如果写角度值，要比兼容之前的角度值-90deg;
4. 重复线性渐变：设置了颜色数值后能平铺渐变；
   1. repeating-linear-gradient();
   2. 数值和线性渐变一致；

径向渐变：radial-gradient()

1. 说明：径向渐变属性也是有渐变颜色的背景图片；所以其是background-image里的属性值；
   1. 颜色延四周由初始位置初始值渐变到最终值；
2. 其参数：
   1. 第一个可以写形状：渐变的形状根据元素的形状改变；
      1. 如果元素形状为长宽一致，则椭圆和圆一样；
      2. 如果元素形状长宽不一样；则渐变形状根据设置的来；
      3. Circle:圆形；
      4. Ellipse:椭圆；（默认）
      5. 如果第一个不写形状；则会以椭圆为默认形状；
      6. 在形状后面加入参数，作为渐变起始位；
         1. At center:默认；默认以中心开始；
         2. At left top:左上；
      7. 在形状后面加入数值；作为渐变宽高；
      8. 在形状后面加入参数：即渐变的半径；两个颜色之间的距离；
         1. Farthest-side:从圆心到最远的边为半径的渐变；
         2. Farthest-corner:从圆心到最远的角为半径的渐变；
         3. Closest-corner:从圆心到最近的角为半径的渐变；
         4. Closest-side:从圆心到最近的边为半径的渐变；
      9. 第一个参数写
         1. X Y:从中心开始的渐变宽高；
         2. 百分比：以宽高的百分比为X，Y；两个值则分别对应渐变宽高；
         3. 方位值：渐变开始的方位；
   2. 颜色：
      1. 必须要写两个以上；
      2. 颜色后面跟上数值、百分比：表示从当前数值、百分比值开始渐变；
3. 书写顺序：形状 大小 at 方位，初始颜色，...，最后颜色；
4. 重复径向渐变：
   1. -repeating-radial-gradient();
   2. 相当于允许平铺；

## 单位：

1. **px:**像素格子；1px就是1像素；
2. **Em:**相对单位（倍数），相对于文字大小；
   1. 先参考自己的font-size，如果自己没有，参考父级的；
3. **Rem:**相对单位（倍数），相对于跟标签（html）的文字大小；
   1. 常用于手机端响应式；

## 文字属性：都有继承性

Font:字体样式；

1. 单个写法：
2. Font-size:设置文字大小；
   * 1. 默认值：
        1. Medium:16px;
     2. 属性值：
        1. 字母：感觉用处不大；
        2. 数字:（px/em/rem）
           1. 直接设置固定像素值；
           2. 设置父级文字倍数；
           3. 设置跟标签文字倍数；
        3. 百分比：
           1. 根据父级文字大小的百分比；
   1. Font-size-adjust:文字大小写缩放比例；（自己理解的，没用过，不确定）
   2. Font-style:字体样式；
      1. 默认值：
         1. Normal；标准样式
      2. 属性值：
         1. Italic:斜体字；没有斜体字时，不操作
         2. Oblique：倾斜的字体；没有斜体字时，强制倾斜
   3. Font-variant:小型大写字母；
      1. 默认值：
         1. Namal:标准；
      2. 属性值：
         1. Small-caps:显示小型的大写字母；把所有小写字母变成小型大写字母；
   4. Font-weight:文字粗细；
      1. 默认值：normal
      2. 属性值：
         1. 标志：
            1. Bold:加粗
            2. Bolder:更加粗
            3. Lighter:更细
         2. 数字：
            1. 100-900；400相当于normal，700相当于bold；
   5. Font-family:设置文字字体；多个字体用逗号隔开
      1. 注意：
         1. 多个字体会从左到右依次判断，前面字体优先于后面的字体；
         2. 每个浏览器的默认字体不一样；
         3. 建议在body中定义字体；
         4. 属性值为字体名；（要显示出来要自己电脑当中存在）
   6. Line-height:行高字体高度有包裹字体的行高拉高
      1. 默认：normal;
      2. 值：
         1. 父级盒子高度（带单位）：显现居中效果；
         2. 百分比：太小，看不明显；是自身字体的百分比；
         3. 数值（没单位）：倍数，字体大小的倍数；
         4. 数值（带单位）：数值大小的行高；
      3. 注意：
         1. 行高没有了，但是字体还在；不会影响布局；
         2. 在行内元素中添加行高，不会撑开自己的盒子，但是会撑开他人；
         3. 行高对行内元素的盒子没有影响，因为行内没有高度
3. 复合写法：
   1. Font:font-style font-variant font-weight font-stretch font-size/line-height font-family ;

### Direction:规定子级、元素放置方式（文本对齐方式）；对元素、文字都有效果（只是摆放位置发生改变）；字符顺序没有改变

1. 默认值：（会影响文字的排列顺序，从ltr->rtl,原本是1 2 3三个子级，变成3 2 1在右边显示）
   1. Ltr:从左到右；
      1. 子级解析方式从上到下，对应渲染页面是从左到右
      2. 文字对齐方式为左对齐；
   2. Rtl:从右到左
      1. 子级解析方式从上到下，先解析的从右往左摆放；
      2. 文字对齐方式为右对齐；
2. 注意：
   1. 在父级设置，对子级有效果
   2. 子级排列方式、文字的书写方式；
   3. 如果父级宽度和子级宽度一致，将看不出来效果；
   4. 如果是两个行内元素，第一个行内先放最右边，第二个紧挨着放置；

### Text-transform:设置文本大小写

1. 默认值：none;无样式；
2. 值：
   1. Uppercase:全部转成大写字母；
   2. Lowercase：全部转成小写字母；
   3. Capitalize:单词首字母大写；

**Text-indent:**设置首行文本缩进；

1. 默认值：0；
2. 值：
   1. 数值：（单位）
      1. 固定的缩进；
   2. 百分比：
      1. 根据父级元素的宽度来计算缩进值；

### Text-align:设置子级、或者元素内的文字的水平对齐方式（只对文字有效果）

1. 默认值： （原本是1 2 3三个子级，left->right ,变成在右边显示 1 2 3）
   1. 根据direction的对齐方式排列；
   2. 不会影响原本文字的排列顺序
2. 值：（对父级）
   1. Left:文本排列到左边；
   2. Right:文本排列到右边；
   3. Center:文本居中；
   4. Justify:两端对齐；用于段落对齐
      1. 应用场景：段落对齐；多个对齐
3. 注意：
   1. 在父级设置，子级及其孙级均会受影响；
   2. 只对文字有效果；

Vertical-align:基线对齐；设置一个元素的垂直对齐；

1. 说明：
   1. 所有的行内、行内块都是按照基线对齐方式对齐（即小写X底边为基线）
   2. 直接放一个img时，下方会又间距，那是图片的基线；
   3. 所有有内联特点的元素，都具有一个默认的上下对齐方式；
   4. 会占据空间；改变空间；
2. 属性：
   1. Baseline:默认值，元素放置在父元素的基线上；会受文字大小影响
   2. Sub:垂直对齐文本的下标；比默认值往下一点点；
   3. Sup:垂直对齐文本的上班；比默认值往上一点；
   4. Top:
      1. 元素的定边与一行中最高元素的顶边对齐；
      2. 最高为父级；
      3. 谁添加该属性，谁对齐；其他不做改变；
      4. 给img加top,会和父元素基线对齐；
   5. Bottom:
      1. 底端对齐本行中最低元素的顶端对齐；
      2. 会受行高影响，各种小的动；
   6. Text-top:把元素的顶端和父元素字体的顶端对齐；
   7. Text-bottom:把元素的顶端和父元素字体的底端对齐；
   8. Middle:
      1. 把此元素放置在父元素的中部；
      2. 视觉上对齐，但是实际有误差；
      3. 要使两个人对齐，要两个人都加；
      4. 块级，对当前元素没有用；
   9. 像素值：
      1. 直接根据值对基线移动；
      2. 正值：往上，下方会添加对应空白；
      3. 负值：往下，上分会添加对应空白；
      4. 复选框与文字对齐，要加上-2px;
   10. 百分数：
       1. 根据当前元素的line-height的百分数计算；
3. 使用img下方会出现空白：是因为基线对齐造成的；
   1. 解决方式：
      1. 转成块；
      2. 设置vertical-align;只要不是baseline都可以；
4. 注意：
   * 1. 当盒子没有高度时，img的高度为100%时图片下方还是会有空格；
     2. 当盒子有高度时，img的高度为100%时，还是有空格，但是会溢出；
     3. 只给一个元素设置，会对其他元素产生影响；
     4. 会受文字大小影响；

Letter-spacing:字间距；

1. 默认：normal:字符间无间距
2. 属性值：
   1. 数值：
      1. 固定字间距；
3. 注意；
   1. 数值能设置负数；
   2. 当到一定负值时，能呈现类似翻转的效果；

Word-spacing:词间距;

1. 默认：normal:默认单词间的标准空间
2. 属性值：数值：固定词间距；
3. 注意：
   1. 数值能设置负值；
   2. 数值是指上一个词的最后一个字符到当前词的第一个字符
   3. 中文要加空格才能有效果；
   4. 如果到达了宽度的边界，中文的话会自动换行；数字和字母要添加空格才会换行，否则会溢出，不换行；

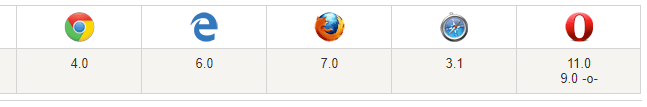
Word-break:设置自动换行方式；设置中日韩的断行规则

1. 默认值：
   1. Normal:浏览器默认规则；（两种属性选一个）
2. 属性值：
   1. Break-all：允许浏览器在单词中换行；
      1. 会出现单词中断；
   2. Keep-all:只有在空格时换行、或者在连字符换行；对中文字符有特殊效果
      1. 完整词语在一起；

White-space:指定元素内空白部分处理方式；

1. 默认：normal:空白会忽略；
2. 属性值：
   1. Pre:空白会保留;写有几个空白，就留几个；且不换行；
   2. Nowrap:永远不换行；
   3. Pre-wrap:保留空白，但会换行
   4. Pre-line:合并空白符序列，保留换行符；几个空白合并成一个空白

Text-overflow:处理文章溢出的方式（css3）配合overflow使用

1. 默认：clip:裁切超出文本；
2. 属性值：
   1. Elipsis:超出文本以省略号显示；
3. 兼容性：
   1. 

Overflow:设置盒子内容超出时的处理方式；

1. 默认：
   1. Visible:显示，超出继续显示；呈现在元素框之外
2. 数值：
   1. Hidden:超出内容会被裁剪；其余内容是不显示的
   2. Scroll:超出内容会被裁剪，但是可以通过滚动条滚动查看；
   3. Auto:如果内容被裁剪，浏览器呈现滚动条，以便查看其余内容；
3. 单个值：只给一个值，另一边为auto；
   1. Overflow-x:只限制x轴方向的
      1. 值：visible，hidden，scroll
   2. Overflow-y:限制y轴方向的
      1. 值：visible，hidden，scroll
4. 注意：
   1. 只对块级元素有效；
   2. 对行内元素无效；因为行内没有固定宽度；
   3. 要想对行内元素设置超出隐藏：
      1. 设置为行内块；
      2. 给其加个父级盒子；行内相当于文字；
   4. 使用overflow：hidden时，有可能会导致文章只显示一半；解决方法：
      1. 加一个右边的padding-left;
      2. 加宽度；

List-style-type:列表样式类型；只有有序/无序列表才有的属性；有继承性：列表项继承列表值；

1. 默认值：
   * 1. Ul/li : disc;小实心圆；；
     2. Ol: decimal;数字
     3. 默认颜色和字体颜色一致
2. 值：
   * 1. None:无标记 ；
     2. Disc:实心圆；
     3. Circle:空心圆；
     4. Square:实心方块；
     5. Decimal：数字；
     6. Decimal-leading-zero: 0开头的数字，超过十和decimal一样；
     7. Lower-roman:小写罗马数字（i,ii,iii）
     8. Upper-roman:大写罗马数字（I,II,III）
     9. Lower-alpha:小写字母
     10. Upper-alpha:大写字母
     11. Lower-greek:小写希腊字母
     12. Lower-latin:小写拉丁字母
     13. Upper-lation:大写拉丁字母
     14. Hebrew：希伯来编号
     15. Armeniam:亚美尼亚编号
     16. Georgian：乔治亚编号
     17. cjk-ideographic：表意数字（大写数字）
     18. Hiragana：日文编号；
     19. Katakana：日文编号
     20. hiragana-iroha：日文编号
     21. katakana-iroha：日文编号
   1. 注意：
      1. 如果是ul/ol/和li设置的值不一致；会应用li里的list-style-type;

Text-decoration:添加到文本的修饰。只能用在文本中；

1. 默认值：
   * 1. None solid；默认颜色为父级字符颜色；
     2. a标签为*underline*
     3. 只写一个style也可以，其他两个样式会为默认样式；
2. 复合写法值：顺序随意；text-decoration-line text-decoration-style text-decoration-color
   * 1. None：没有修饰
     2. Underline:下划线
     3. Overline:上划线
     4. Line-through:删除线
     5. Blink:闪烁的文本；不知道怎么用；没见过效果；
3. 单个写法：
   * 1. text-decoration-line；设置修饰线；
     2. text-decoration-style：设置修饰线的样式

1) Solid：实线

2) Dashed:虚线

3) Double:双边框

4) Dotted:点线

* + 1. text-decoration-color：设置修饰线的颜色

1. 注意：
   1. 只能对文字有效果；

Text-shadow:文字阴影；

1. 说明：
   1. 只有外部阴影，逗号可以隔开多个阴影；没有拓展半径；
2. 属性：按顺序：x位置 Y位置 模糊半径 颜色；
   1. 水平、垂直阴影的位置：正则向右下，负值向左上；
   2. 拓展半径：向四周延展；其值时向外拓展值；
   3. 颜色：阴影的颜色；默认为父级字体颜色；

box-shadow：盒子阴影一个或多个下拉阴影的框；（css3）

1. 默认：none；
2. 分类方式：要想写为内部阴影，在后面添加inset;
   1. 外部阴影：在盒子四周产生；
      * 1. 属性：按顺序写；x位置 y位置 模糊半径 拓展半径 颜色；
           1. 水平阴影的位置；正值向右，负值向左 默认为0
           2. 垂直阴影的位置；正值往下，负值往上；默认为0
           3. 模糊半径，羽化，从最外面开始模糊；默认为0
           4. 扩展半径：向四周延展，（放大或者缩小）其值是向外拓展值；默认为0；
           5. 颜色：阴影的颜色；默认父级字体颜色；
   2. 内部阴影：在盒子内部产生；
      1. 属性：按照顺序写：x位置 y位置 模糊半径 拓展半径 颜色 inset；
      2. 注意:
         1. 内部阴影会挡住背景颜色，但是不会挡住内容；不挡img图片；
         2. 内部阴影的移动会带动盒子背景移动；相当于占领了原本的背景位置；
3. 注意：
   1. 阴影能设置多个，以逗号隔开，在前面的阴影在最上面，后面的阴影在下面；
   2. 阴影的移动都是相对于当前自己的阴影；

Float:浮动；

1. 特点：
   1. 脱离文档流；不占据文档流内的空间
   2. 使用display：flex弹性盒模型时，会使浮动失效；
   3. 撑不开父级；会导致父级塌陷；
   4. 由内容撑开宽度；
   5. 所有浮动元素都在同一个层面；定位比浮动层级高
   6. 浮动会改变自身的类型；他能把元素转换成块级，但是不会独占一行，会使元素并排排列；
   7. 宽高默认为0；
   8. 只会影响后面的元素
   9. 块级之所以有默认父级100%，是因为其有独占一行的特性；使用float后，其独占一行的特性没有了，将不再默认父级100%；
2. 属性值：设置水平排列的方式
   1. Left:左浮动
   2. Right:右浮动
3. 清除父级塌陷：
   1. 给盒子父级设置宽高；设置为块级，设置宽高；
   2. 给父级加float:left;
   3. 给兄弟加个clear:both;
   4. 给父级加个overflow：hidden，display：block；
      1. 因为如果想要超出隐藏必须要知道元素的宽高，会强行计算由内容撑开盒子的高度；
   5. 给父级加个伪元素，并添加clear属性和display:block属性；
4. 注意：
   1. 不能用行内清除浮动，行内本身不受影响；
   2. 父级空间不够的情况下，浮动元素会往下掉，会出现卡住的情况；先从第一行最右边到最左边，满了直接往下一行最右到最左；
   3. 浮动会在遇到父级边缘是停下来
   4. 浮动会在遇到别的浮动时停下来；
   5. 浮动之后还是能支持margin;
   6. 加了浮动之后就不能使用margin：0 auto居中；
   7. 浮动没有层级；

Position:指定一个元素定位的类型；

1. 主属性值：
   1. Absolute:绝对定位
      1. 相对于父级定位元素的定位；如果父级没有设置定位，会往上查找
      2. 会脱离文档流；
      3. 不占空间，会影响布局；
      4. 可以很多个叠加在一起；
      5. 会改变元素类型为block;
   2. Relative:相对定位
      1. 相对于自身的定位；
      2. 占空间，不会影响布局；
      3. 脱离文档流；自身移动不会影响布局；
      4. 其移动，其本身所占空间不会变；
      5. 不会改变元素类型；
   3. Fixed:固定定位；
      1. 相对于浏览器窗口的定位；
      2. 即使浏览器滚动也不会移动；
      3. 脱离文档流；
      4. 不占空间 ；
   4. Static:没有定位；
      1. 默认效果；
      2. 占空间；
      3. 没有定位；
   5. Sticky:粘性定位；
      1. 相对于定位父级，如果父级盒子滚过去了，也会被隐藏；
      2. 相对于滚动位置变化；设置的值为最小值，如果小于该值，会一直为该值；
      3. 占空间；不会影响布局；
      4. 和relative类似；相当于relative和fixed合并；
      5. 脱离文档流；自身移动不会影响布局
2. 副属性：依赖于position定位；
   1. 方位：
      1. Top:相对于相对位置顶部的值；
      2. Left:相对于相对位置左边的值；
      3. Right:相对于相对位置右边的值；
      4. Bottom:相对于相对位置下边的值；
      5. 默认：0;
      6. 属性值：
         1. 数值：相对于相对父级顶部的值；
         2. 百分比：其值为其父级宽高的百分数；
            1. 注意：

块级：

如果父级没有设置高度，则为0；

如果父级没有设置宽度，则默认为父级的100%；

行内：（适用于relative定位，因为其他定位会改变元素类型）

百分比是根据body的宽度；设置高度的百分比没用；

* 1. 层级：
     1. Z-index:设置定位的层数；
        1. 同级在后面写的在上面；
        2. 设置的数值越大，越在上面；

1. 居中方式：（前提是当前元素有宽度；）
   1. left：0;right:0;top:0;bottom:0;margin:auto;垂直、水平居中；
      1. 当left:0;right:0;时，会拉扯其宽度为父级的宽度，如果当前元素有了宽度就不会拉扯；如果添加数值时，会以当前位置移动；
      2. 如果left和right相等，margin：auto也是居中；
   2. Left：50%；会往右移动50%，但是不居中；要在添加margin-left:-50%才行；

### box-sizing:盒子模型;

1. 分类：
   1. Border-box:怪异盒模型：宽高定义整个盒子大小；
   2. Content-box:正常盒模型；宽高定义内容大小；
2. 怪异盒模型和正常盒模型的对比：
   1. 正常盒模型占位：width/height+padding+border+margin
   2. 怪异盒模型占位：width/height+margin
   3. 正常盒模型大小：width/height+padding+border
   4. 怪异盒模型大小：width/height;
3. 怪异盒模型：
   1. Padding和边框在盒子内部产生，会挤压内部区域；
   2. 当padding很大，大到最大宽高时，会挤压内容区域为0；但是文字会超出内容区域显示；

Cursor:鼠标指针；

1、说明：鼠标移动到元素上的显示；光标形状；

2、属性值：

1)pointer：小手

2)move:四个方向的箭头，可以移动的显示

3)wait:根据你电脑中设定的等待标识

4)help:箭头+下面的问号

5)url:

（1）自定义标识，要cur文件或者ani文件，要加备用指针才能使用；

（2）写法：url(‘’),pointer;

6)default:默认；

7)auto：自动匹配浏览器的；

8)crosshair:十字

9)row-resize:上下，可以拉高的标识；

10）e-resize:左右箭头；

11）ne-resize:左下右上箭头（北、东）

12）nw-resize:左上右下箭头；（北、西）

13）n-resize:上下箭头；

14）se-resize:左上右下箭头；（南、东）

15）sw-resize:左下右上箭头；（南、西）

16）s-resize:上下箭头；

17）w-resize:左右箭头；

18）text:插入文字标识

3、注意：

1）鼠标的标识，根据的是你电脑中设置的标识；

Opacity:透明度；又被称为滤镜；

1. 说明：
   1. 数值：0~1（透明~不透明）
2. 兼容性：
   1. 在ie8以下：filter:alpha（opacity:数值）；数值为0~100；

### Form:表单：

1. 主要作用：收集用户信息，提交给后台
2. 写法：
   1. <form action= ‘后台地址’name></form>
3. 标签属性：
   1. Action:提交地址
   2. Name:标识；
4. 表单控件：Input
   1. 单标签;
   2. 规定了用户可以在其输入数据的输入字段；
   3. 输入字段可以通过type属性来改变；
   4. 属性：
      1. Type:设置表单控件的种类
         1. Text:文本输入框.
         2. Password:密码输入框（默认会隐藏输入数据）
         3. Radio:单选框；
            1. 相同的name绑定在一组；
            2. Checked:设置页面加载时预先选中的项目；
         4. Checkbox:多选框；
            1. Name绑定；
            2. 添加disabled，标识默认不能选中；
            3. Checked:设置页面加载时预先选中的项目；
         5. Submit:提交按钮；
            1. Formenctype:规定当表单数据提交到服务器的编码方式；覆盖form的entype;(h5)
            2. Formmethod：规定表单数据发送到服务器的http方式；(分get/post)覆盖了form的method属性(h5)
            3. Formtarget:规定响应表单请求的方式(h5)覆盖form的target;

\_blank:在新窗口打开

\_self:当前窗口

\_top:在整个窗口显示；

* + - 1. Reset:重置按钮；
      2. File:文件上传；
         1. Accept:规定可通过文件上传提交到服务器接受的文件类型；使用多个时，用逗号隔开；

Audio/\*;

Video/\*;

Image/\*;

* + - 1. Image:插入图片
         1. 说明：

当用户单击图像时，浏览器将以像素为单位，将鼠标相对于图像边界的偏移量发送到服务器，其中包括从图像左边界开始的水平偏移量，以及从图像上边界开始的垂直偏移量。

* + - * 1. alt:用在网络不佳，加载不出图片时替代文本；
        2. Formenctype:规定当表单数据提交到服务器的编码方式；覆盖form的entype;(h5)
        3. Formmethod:规定表单数据发送到服务器的http方式（get/post）覆盖了form的method(h5)
        4. Formtarget:规定响应表单请求的方式(h5)覆盖form的target;

\_blank:在新窗口打开

\_self:当前窗口

\_top:在整个窗口显示；

* + - * 1. Height:规定input标签的高度；(h5)
        2. width：规定input标签的宽度；(h5)
        3. Src:显示提交按钮的图片的url;(必写项）
    1. List;包含input元素的预定义选项；和datalist标签一起使用
       1. 用法：
          1. 在input中设置list属性；其属性值为对应的datalist标签的id值；
          2. 在datalist标签中设置id;其子级为option标签；每个option的value为对应的预选项；
    2. 副属性：
       1. Autofocus：设置的字段在页面加载时自动获取焦点；（h5）
       2. disabled：设置禁用的input元素；禁用后无法使用、获取点击，不会提交；
       3. Form:规定label字段所属的一个或多个表单；（h5的新属性）
       4. Formaction:规定提交表单时处理输入控件的文件的url;即相当于覆盖了form的action;(h5）
       5. Max/min:规定input元素的最大最小值；适用于要输入数值有关的；（h5）
       6. Maxlength:规定input元素的最大长度；
       7. Multiple:允许用户输入到input的元素为多个值；适用于：email/file;(h5)
       8. Name:input表单的名称；多个input可以使用同一个name,会把同一个name分为一组；
       9. Pattern；规定用于验证input输入的正则表达式；适用于text/search/url/tel/email/password;(h5)
       10. Placeholder:规定未输入时，输入框中的文字；适用于text/search/url/tel/email/password;（h5）
       11. Readonly:规定元素为只读状态；可复制；
       12. Required:规定必须在提交表单之前必须填写该字段；（h5）
       13. Size:设置以字符数计算的input元素的可见宽度；
       14. Step:规定input元素的合法数字间隔；设置了step=3,则以3的倍数的数字才是合法的；适用于number、range、date、datetime、datetime-local、month、time 和 week。(h5)
       15. Value:设置input上的值；不同的input显示不同；不适用与file;
           1. Button,reset,submit\_\_\_\_\_\_\_定义按钮上的文本
           2. Text、password\_\_\_\_\_定义输入字段的初始值；为传入后台的数据；
           3. Checkbox、radio、image\_\_\_\_\_\_定义与input有关的值，当表单提交时当成键传给action的url；
       16. Outline:元素当前的外轮廓，和边框一样的值；默认为无，写的时候出现；只有能获取焦点的才有该属性；

1. 标注标签：label
   1. 当用户选择该标签时，浏览器就会自动将焦点转到和标签相关的表单控件上；
   2. 属性
      1. For:指定指向的表单控件ID；相当于绑定到对应的表单控件中；
      2. Form:规定label字段所属的一个或多个表单；（h5的新属性）
2. 分组标签：fieldset
   1. 相关元素的分组；
   2. 相当于在外面加了边框
   3. 默认属性：
      1. 块级；
      2. Margin-inline-start:2px;
      3. Margin-inline-end:2px;
      4. Padding-block-start:0.35em;
      5. Padding-block-end:0.625em;
      6. Padding-inline-end:0.75em;
      7. Padding-inline-start:0.75em;
      8. Border-width:2px;
      9. Border-style:groove;
   4. 分组标题标签：legend
      1. 默认情况：
         1. 是一个块级;
         2. padding-inline-start:2px;
         3. Padding-inline-end:2px;
   5. 属性：
      1. Disabled:整个分组都不能使用；（h5的新属性）
      2. Form:规定fieldset所属的表单；（h5的新属性）
      3. Name:规定fieldset的名称；（h5的新属性）
3. 下拉选择框标签：select;
   1. 子标签：option:配套的，中间不能穿插其他的；
      1. Disabled:首次加载时禁用；
      2. Label:选项的值；优先级label>标签包裹的文字>行内的value；
      3. selected：规定当前选项为首选项；
      4. Value:发送至服务器的选项值；
   2. 属性：
      1. Size:下拉框的可见属性个数；
      2. Autofocus:页面加载时自动获得焦点；（h5的新属性）
      3. Disabled:当设置了true或者写入内联中，会禁用下拉列表；
      4. Form:定义其所属的表单；（h5的新属性）
      5. Multiple:设置其可以选择多个；
      6. Name:设置下拉列表的名字；
      7. Required:设置下拉列表必须选择一个；（h5的新属性）
4. 文本框标签：textarea;
   1. 是一个多行的文本输入空间；
   2. 属性：
      1. Autofocus:页面加载时自动获得焦点；
      2. Disabled:当设置了true或者写入内联中，会禁用下拉列表；
      3. Form:定义其所属的表单；（h5的新属性）
      4. Required:设置必须选择要输入；（h5的新属性）
      5. Readonly:设置页面为只读；
      6. Maxlength:设置文本框允许的最大字数；（h5的新属性）
      7. Placeholder:设置提示信息
      8. Name:设置文本框的名称
      9. Rows:设置文本框可见行数；

Iframe:框架，在页面里生成内部框架，相当于页面中的页面；有的网页不支持内嵌；

1. 自身有默认宽高：300\*150；行内元素；border-style:inset;border-color:RGB(238,238,238);
2. 属性：
   1. Frameborder:边框显示情况；0为不显示边框，1为显示边框；（h5不支持）
   2. Scrolling:是否显示滚动条；（h5不支持）
      1. Yes:显示；
      2. No:不显示；
      3. Auto:根据内容显示；
   3. Name :名称，可以在a中设置target打开方式时，指向iframe的name，在iframe中打开；使用target和iframe的name绑定；
   4. Marginheight:规定顶部到底部的距离；（h5不支持）
   5. Marginwidht:规定左边到右边的距离；（h5不支持）
   6. Sandbox:规定框架内是否允许使用脚本；
      1. Allow-scripts:允许脚本执行；
      2. Allow-forms:允许表单提交；
      3. Allow-top-navigation:框架内的上下文可以加载内容到顶部浏览上下文环境；
      4. Allow-same-origin:允许内容当做普通来源；
   7. Src:显示在框架内的url；
   8. Srcdoc:规定页面的html内容显示在框架内；

继承：跟文本有关的都继承；（font/text）

1. a标签、line-height/color除外；
2. 标题标签比较特殊，相对于其他，不完成安装继承的文字大小显示；

Table:表格（自带border属性）

1. 下级标签：
   1. Thead:表格的头部；（标题）
      1. 有border,没有margin,没有padding;
   2. Tbody:表格的主体内容；
      1. 有border,没有margin/padding;
   3. Tfoot:表格的脚部；（页脚备注）
      1. 有border,没有margin/padding;
   4. Tr:行标签;
      1. 只有border,没有margin/padding;
      2. Td:单元格：
         1. 有border/padding,没有margin；默认padding:1px;
         2. 垂直居中
      3. Th:单元格：
         1. 只有border/padding,没margin；默认padding:1px;
         2. 在头部的单元格
         3. 默认居中，加粗；
2. 其属性
   1. border：如果为5，则表示外边框为5,；如果是1，内外边框都是1；如果是0，则内外边框都是0；默认为0；
   2. Cellspacing:内外边框的间距；如果为0，则为合并，相当于把两个之间的距离为0；
   3. Border-collapse:设置内外边框是合并还是分离；只能在样式表中添加；在行内样式中添加无效；
      1. Collapse:内外边框合并；
      2. Separate:内外边框分开；
3. 合并单元格：只能在td/th中的行内样式中设置；
   1. Rowspan：跨行合并；
      1. 在要合并的第一个添加，并在后面的对应的列中删除；如果不删除后面的，把最后一个挤出去；影响布局；
      2. 设置的数值不用带单位；跨多少行；
   2. Colspan:跨列合并；
      1. 在要合并的第一个添加，并在后面对应的行中删除；如果不擅长后面的，会把最后一个挤出去；
      2. 设置的数值不带单位，跨多少列；
4. 特点：
   1. 给表格写高度，没有用，默认内容撑开；tbody为最大的高；
   2. 每一行的高度一致，每一列的宽度一致；
   3. 设置固定宽高时，第一行和第二行件差1px
   4. 每一行中，以最高的单元格高度为整行的高度
   5. 每一列中，以宽最大的单元格宽度为整列的宽度
   6. 表格table定义的宽度，表格内部所有元素都会根据内部调整宽度；
   7. 如果设置表格内部的宽度综合超过或在小于table宽度，table不会变换
   8. 如果内部元素加起来超过表格的宽度，那么会按照比例缩小；
   9. 如果内部元素加起来小于表格宽度，那么会安装比例放大；
   10. 表格有最小高度，有thead/tbody/tfoot撑开；
5. 缺点；
   1. 改动一个地方，会导致整体布局乱；
   2. 加载顺序慢，其他加载完了再加载表格；
6. 优点：
   1. 能优先显示主体部分；
   2. 适用于后台；
   3. 可以等宽高；
7. 注意：
   1. Thead/tfoot可以不写，但是要写，都写；

阿里图标：（www.iconfont.cn）

1. 说明：
   1. 是一种文章，因为图片放大时容易失真，把图片转成一种文字，放大了就不会失真；
2. 操作步骤：
   1. 在网站中搜索想要的图标，
   2. 创建项目，添加图片入项目；
   3. 下载到本地；
   4. 使用demo查看其使用情况；
3. 三种方式：
   1. Symbol:支持彩色；兼容性差，消耗性能
   2. Fontclass:兼容性良好，ie8以上都能兼容，语义明确；只支持纯色
   3. Unicode:兼容性最好，语义不明，只支持纯色；

Transition:过渡（css3）

1. 说明：
   1. 从一种样式到另一种样式的变化过程；
   2. 想要在谁有过渡效果，就在其身上加；
2. 单属性：
   1. Transition-property：需要过渡的样式名称；如果有多个，用逗号隔开；写all或者不写默认全部能生效的样式；
   2. Transition-duration:过渡时间，从一个样式到另一个样式要花费的时间，多个用逗号隔开；
   3. Transition-timing-function:过渡的速度曲线（默认ease,慢-块-慢）
      1. Linear:匀速
      2. Ease-in:先慢后快
      3. Ease-out:先块后慢；
      4. Ease-in-out:先慢再快最后慢
      5. Cubic-bezier(n,n,n,n)：贝塞尔曲线；自定义运动曲线；设置网站：<http://cubic-bezier.com/#.17>
   4. Transition-delay：延时时间；设置好之后，过了多少秒之后才会开始执行过渡；
3. 复合属性：
   1. 名称 过渡时间 速度曲线 延时
4. 注意：
   1. 不能自己动，需要一个媒介；
   2. display不能过渡，因为display：none时，相当于消失，不存在了；
   3. 想要显示隐藏的过渡：
      1. 使用visibility:hidden/visible;
         1. 优点:不占空间、有过渡效果，hover不到;存在与页面，不影响其他元素；
      2. 使用opacity:0-1;
5. 能过渡的样式：
   1. Width/height/background-color/font-size/opacity/overflow;

Animation:动画；

1. 说明：自动执行；
   1. 动画帧、动画的播放器:
      1. @keyframes run{
         1. 0%{}100%{}
      2. }
      3. 如果100%没定义，动画会按照声音时间回到原位（不是0%）
      4. 0%相当于动画帧0状态；100%动画帧100%状态；
      5. 运动过程：元素原状态、动画帧0状态、动画帧100%状态、元素原状态；
2. 属性：
   1. Animation-name:run;（动画播放器名字）
   2. Animation-duration:运动时间；
   3. Animation-timing-function:运动曲线（和过渡的运动曲线一致）
   4. Animation-delay:延时时间；
   5. Animation-iteration-count：运动次数
      1. 数值：数值次数；
      2. Infinite:无限循环；
   6. Animation-direction:动画执行方向；
      1. Normal:默认；（正向）
      2. Reverse:反向；100%-->10%；
      3. Alternate:一正一反（奇数正，偶数反）
      4. Alternate-reverse:一反一正（奇反偶正）
   7. Animation-fill-mode:播放结束后的位置
      1. none：会原始位置；
      2. Forwards:停留在100%；
      3. Backwards:忽略原始状态（停留在0%）
      4. Both:忽略原始状态，动画结束停在100%；
   8. Animation-play-state:运动状态
      1. Paused:暂停；
      2. Running:运动（默认）
3. 复合写法：
   1. Animation:name 时间 次数 延迟 其他

弹性盒模型：容器属性；

1. 说明：写法为display:flex；
   1. 使当前元素为弹性容器，简称容器；
   2. 只有直接子级会受影响，这样的子级叫项目（弹性）
   3. 设置为弹性容器之后，项目会压缩；
2. 作用：均等分；
   1. 其对齐方式：
      1. 容器内部的对齐方式：在容器中设置参数；
         1. Flex-direction：定义主轴方向；
            1. Row：（默认）横向从左向右
            2. Row-reverse:从右向左
            3. Column:纵向从上到下
            4. Column-reverse:纵向从下到上；
         2. Flex-wrap:设置是否换行（定义交叉轴的方向）
            1. Nowrap:不换行；（如果设置不换行，则会压缩项目）
            2. Wrap:换行，从上到下；
            3. Wrap-reverse:换行从下到上；（相当于，前面的写在下面）
         3. Align-content：定义项目在交叉轴的对齐方式；（多根轴线的情况）
            1. Flex-start：沿着交叉轴的起点对齐；
            2. Flex-end:沿着交叉轴的终点对齐；
            3. Center:居中对齐；
            4. Space-between:两端对齐；

每行间隔相等，并且和主轴前后没有间隙；

* + - * 1. Space-around:两端对齐；

每行间隔相等，与主轴前后有间隙；相当于加了左右的margin；

* + - * 1. Stretch:拉伸平方空间（默认）
      1. Align-items:只能定义（单根轴线）在交叉轴上的对齐；
         1. Flex-start:延交叉轴起点对齐；
         2. Flex-end:沿交叉轴终点对齐；
         3. Center:居中对齐；
         4. Stretch:默认，拉伸平分空间；
         5. Baseline:根据项目的基线对齐；项目的内容基线对齐；
      2. Justify-content:主轴上的对齐方式；
         1. Flex-start:以主轴起点对齐（默认方式）
         2. Flex-end:以主轴终点对齐
         3. Center:以主轴居中对齐
         4. Space-between：两端对齐；两边贴边
         5. Space-round:两端对齐：两边不贴边
      3. Flex-direction和flex-wrap的组合：
         1. Flew-flow:flex-direction flex-wrap;
    1. 项目之间的对齐方式；在项目中设置参数；
       1. Flex-order/order:项目排序，排序方式；
          1. 默认为0；
          2. 序号越小，越靠前；
          3. 序号相同，按先后顺序
       2. Flex-shrik:设置项目的缩小比例（默认为1）
          1. 1：当容器空间不够时自动缩小
          2. 0：不管容器空间如何，就是不缩小
       3. Flex-grow:设置项目的放大比例（默认为0）
          1. 1：按剩余空间比重放大

剩余空间\*项目比例/总比例和=增加的大小

* + - * 1. 0：不放大
      1. Flex-basis:分配主轴剩余空间：没有剩余空间时不分配
         1. 数值：（有单位）

直接分别主轴多少空间；

* + - * 1. 百分数：安装剩余空间的百分数
      1. 复合写法：
         1. Flex:flex-shrik flex-grow flex-basis;
      2. 常用：
         1. Flex : 1 1 auto:自动缩小放大；
         2. Flex: 0 0 auto:不放大缩小；

1. 注意：
   1. 主轴和交叉轴一定是垂直的；
2. 影响：会使float、vertical-align、clear无效（或者会影响布局）

Transform:变换；

1. 方式：
   1. Rotate()：以基点旋转；
      1. 值为度数值：deg为单位；加为顺时针，减为逆时针
      2. RotateX()：以X轴旋转;
      3. RotateY():以Y轴旋转；
   2. Translate()：位移：类似相对定位；
      1. 一个值：为x轴；两个值为x,y轴，以逗号隔开（单位为像素值）
      2. TranslateX():以水平轴平移；
      3. TranslateY():以竖直轴平移；
      4. 设置基点对位移没作用；
   3. Skew():斜切；默认以基点变化；
      1. 两个值：一个以水平方向斜切，一个以垂直方向斜切；单位为度数值；（以逗号隔开）
      2. 可以由基点改变斜切中心点；
      3. 水平斜切相当于水平向外拉伸；但是宽度不变；
      4. 垂直斜切相当于垂直向外拉伸，但是高度不变；
      5. SkewX():水平斜切
      6. SkewY():垂直斜切
   4. Scale():缩放；根据基点缩放
      1. 值为倍数，没有单位；
      2. 只写一个宽高都变动一样的倍数；写两个会根据两个倍数变动；（以逗号隔开）
      3. 大于1是放大，小于1 是缩小；负数为反向；
      4. ScaleX():只缩放水平方向；
      5. ScaleY():只缩放垂直方向；
2. Transform-origin:基点；
   1. 默认：center/50%;中心点；
   2. 要写两个值,基于当前元素的x/y坐标；不写，会为auto(默认50%);
3. 复合写法：
   1. Transform:旋转 平移 斜切 缩放；
   2. 如果顺序写错了，解析不会报错，但是由于是从左往右解析，会导致出现异常；

### 特殊符号：

1. 空格：&nbsp;占位符，一个代表一个空格；
   1. 如果不加，浏览器在解析时，会把多个空格、换行均当成一个空格处理
2. 小于：&lt;
3. 大于：&gt;
4. 版权：©:&copy
5. 注册商标: ®:&reg

## 选择器：对要修改的样式进行选择；写在css表里；在不影响别人的情况下，准确的选择到元素;能设置组合写法；

1. 通配符选择器：选择所有的标签；（包括html,head,body标签）
   1. 符号：\*
   2. 应用：\*{}；选择所有的标签
2. 群组选择器：选择多个标签或者符合指定选择器的标签；选择所有设置的标签
   1. 符合：，
   2. 应用：
      1. Ul,ol{}：匹配ul和ol的标签
      2. .nav,#head{}: 匹配class为nav或者id为head的标签
3. Id选择器：根据id选择单个标签
   1. 符号:#id
   2. 应用：
      1. 在标签中添加id属性及属性值；
      2. 在选择器中，使用符号#id值{}处理；
4. Class选择器：根据class选择标签，由于多个标签能设置同一个class,所以设置是要注意
   1. 符号：.class
   2. 应用：
      1. 在标签中添加class属性及值
      2. 在选择器中，使用符号.class{}处理
5. Tag(标签)选择器：根据标签选择标签，由于一个网页会有多个相同的标签，所以使用的时候要注意，最好带上父级
   1. 符号：标签
   2. 应用：
      1. 直接在选择器中，使用 标签{}选择
6. 后代选择器：选择所有后代中匹配的标签，精确选择；
   1. 符号： 父级 自己
   2. 应用：
      1. 使用 父级 自己{ }选择；
7. 子代选择器：选择儿子辈中匹配的标签；
   1. 写法：父级 > 子级
   2. 注意：只会匹配直接子级
8. 兄弟选择器：选择相邻的后面一个标签（匹配的）
   1. 写法：兄+ 第
   2. 注意：
      1. 只会选择后面的一个相邻的匹配的标签（后一个、相邻、匹配）
9. 同代选择器：选择同一父级下的所有兄弟元素
   1. 写法：兄 ~ 第
   2. 注意：会匹配当前元素后面的所有匹配的标签；
10. 伪类选择器：
    1. 分类：
       1. 状态
          1. ：hover:鼠标悬停，即鼠标停止该与元素上的时候
          2. ：active:鼠标点击不放手；
          3. ：focus:选择获取焦点的元素；
       2. a链接标签
          1. A:link:a标签专属链接，匹配没有点击过的链接
          2. A:visited:a标签专属伪类，匹配访问过的链接
          3. A:target：给锚链接对象添加样式；指向了我们给了标记的元素；
    2. E：disabled:匹配E元素里的不可点击的表单控件
    3. E：enabled:匹配E元素里的可以点击的表单控件
    4. E：checked:匹配E元素里被选中的表单控件
    5. E:first-line:匹配E元素中的第一行文字；
    6. E：first-letter：匹配E元素中的第一个文字；
    7. E：：selection:匹配E元素被选中的文字
    8. E：not(.on)：匹配没有class为on的值的E元素；
    9. E~F：匹配E元素后面的所有兄弟元素F；
11. 结构性伪类选择器：E:代表元素
    1. E:nth-child(n):匹配E元素父级的所有子元素的第n个子节点（并且该节点还必须和E一样的选择器）
    2. E:nth-last-child(n):匹配E元素父级的所有子元素中从后往前第n个字节点（必须是和E一样的选择器）
    3. E：nth-of-child(n):相同E元素所有集合中的第n个子节点（必须是和E一样的选择器）
    4. E:nth-last-of-type(n):相同E元素所有集合中从后往前的第n个节点（必须要和E元素一样）；
    5. E：nth-child(odd):odd匹配的是奇数，也可以写成2n-1;匹配所有E元素父级的所有的子元素的奇数个；
    6. E：nth-child(even):even匹配的是偶数，也可以写成2n;匹配所有E元素父级的所有的子元素的偶数个；
    7. E :empty:表示这个元素中没有子节点；（包括回车空格都不能有）
12. 标签选择器：
    1. [class]:匹配所有带有class属性的标签.和class的优先级一样；
    2. [class=’a’]:匹配所有class=’a’的元素；
       1. Class内容完全等于才行
    3. [class ~= ‘box’]:模糊选择器：扩大选择范围，选择具有class值为box的元素，（可以有多个class,但必须其中一个为box）
    4. [class |=’box’]:选择box,box-开头的class元素的标签，并且要在第一位才能判断；如果标签中有’box’时，必完全等于box才行;
13. 属性选择器：根据属性、属性值来进行选择；其中attr代表属性，val代表属性值；
    1. [attr]：匹配所有具有attr属性的标签；
    2. [attr=val]：匹配所有具有attr属性，并且值为val的标签；
    3. [attr~=val]:匹配所有attr属性包含或等于val的标签；（当是等于时，val必须是独立的）
    4. [attr|=val]:匹配所有attr属性以val-开头或等于val的标签（如果是val时，要全等于才行；）
    5. [attr\*=val]:匹配所有的attr属性包含val的；只要其值有val的词就行；
    6. [attr^=val]:匹配所有attr属性以val开头的；
    7. [attr&=val]:匹配所有attr属性以val结尾的；
14. 伪元素选择器：假的元素；不能被选中；不能被js操作；不占结构；是一个行内元素；
    1. 写法：h5规范，使用双冒号；
       1. .box::before：在class=’box’的标签之前添加一个元素
       2. .box::after:在class=’box’的标签之后添加一个元素；
    2. 注意：
       1. 必须要写content:’属性值’：文字内容，不能被选中；
    3. 用作清除浮动的父级塌陷:
       1. 在浮动元素的父级中添加伪元素选择器；
       2. 添加属性display:block;content=’’;clear=both/ overflow:hidden;content=’’,clear=both

Calc():运算规则；

1. 写法：calc(表达式)
2. 要求：运算符前后必须留空格；
3. 注意：使用\*、/时，要前面带单位

选择器优先级：当相同样式、元素相同在不同选择器中有不同值时，会根据选择器优先级来选择；

1. 注意：
   1. 优先级，每个等级都有等级优势；
   2. 加冒号的选择器，优先级等于一个class
   3. 添加！Important属性，可以使元素优先级为最高；写在样式后面
2. 同等级选择器：后面的优先级更高（后面的覆盖前面，不同属性则会添加）
3. 样式优先级顺序：
   1. 行内>内部样式>外部样式；（不准确）
   2. 最大优先级为！Important>行内>导入样式（在内部样式后面）>内部样式>外部样式
4. 外部样式：
   1. 要在新的css文件中加@charset ‘utf-8’;
   2. 引入外部文件：<link rel=’stylesheet’ href=’路径’/>
   3. 在css中引入其他的css,使用@import url(‘路径’);（不推荐性能差并且是在最后才加载）

### CSS3兼容性：

1. css3的属性要在使用要添加前缀；由于每个浏览器的内核不一致，所以写法不一致；
   1. IE：-ms-
   2. 谷歌：-webkit-
   3. 欧朋（opera）:-o-
   4. 火狐：-moz-
2. 建议在最后再加上标准的写法
3. 需要添加兼容的css3属性：
   1. Transition;
   2. Animation;
   3. @keyframes
   4. Transform
   5. Box-shadow
   6. Text-shadow
   7. Hsla
   8. Calc

Contenteditable:设置可编辑文本属性；

1. 属性值：
   1. False:不可编辑；
   2. True:可编辑；

### 响应式：根据不同页面做不同的样式；

1. 媒体类型：
   1. All：所有类型；
   2. Screen:彩屏设备；
   3. Print:用于打印机上的浏览；
2. 写法：
   1. 在css中写@media 媒体查询{ 选择器{样式}}
   2. 例如：@media print{选择器{样式}}
   3. 其媒体查询中能写判定依据：max-width:600px即最大宽度为600px时
   4. 其关键字有：
      1. Only :仅仅
      2. Not :除当前外；
      3. And：同时满足；
3. 第二种写法：
   1. 在link引入css中添加media属性，其值为媒体查询的结果；

### 视口：viewport

作用：在移动浏览器中，当页面宽度超过设备，浏览器内部虚拟的一个页面容器，将页面容器缩放到设备这么大，然后展示；

在移动端中要添加视口：用于移动端适配；

1. 在<head>中添加<meta name=’viewport’ Content=’width=device-width,minnum-scale=1.0,maximum-scale=1.0,user-scalable=’no’/>
2. 移动端使用rem布局；只参考html标签（font-size）,相对于跟标签，跟标签文章大小倍数，一般在根标签写宽度/10的font-size;
3. 10rem-->10 \* font-size;
4. Em:相当于自身文字大小的计算，是自身文字大小倍数；