# 元素类型

* 块元素会独占一行，无论内容多少。
* 例子：<h1>, <p>, <ul>, <table>，<div>
* 内联元素不会以新行开始
* 例子：<b>, <td>, <a>, <img>
* span元素没有任何语义，span标签专门用来选中文本设置样式
* a元素可以包含任意元素，除了他本身
* p元素不可以包含任何的块元素
* 行内元素：span,a,var ,em,input,img,img,textarea,var,em,strong,select,lable
* 块级标签：div,p,h1-h3,ul,ol,dl,li,dd,dt,table,td,tr

# css选择器

* id选择器 #xx{}
* class选择器 .xx{}
* 分组选择器（并集） #xx , .xx , xx{}
* 通配选择器（选择所有） \*{}
* 复合选择器 （交集）xx.xx{}
* 后代元素选择器 xx xx{}
* 子元素选择器（只找儿子） xx >xx{}
* 相邻选择器 (只找紧接着下一个的) xx + xx{}
* 左右选择器（找后边的全部） xx ~ xx{}
* 正数元素选择器 nth-child(){}
* 倒数元素选择器 nth-last-child(){}
* 属性选择器1 ： xx[属性]{}
* 属性选择器2 ： xx[属性=”xx”]{}
* 属性选择器3 ： xx[属性^=”xx”]{} 选择以xx开头的属性的元素
* 属性选择器4 ： xx[属性$=”xx”]{} 选择以xx结尾的属性的元素
* 属性选择器5 ： xx[属性\*=”xx”]{} 选择包含xx的属性的元素

# Css伪类，伪元素

* 伪类 为特殊元素设置样式 比如链接
* a:link 表示普通的链接（没访问过的）
* a:visited 表示访问过的链接 （浏览器是通过历史记录来访问是否访问过）
* 由于涉及到用户的隐私，visited只能设置字体的颜色
* :hover 表示伪类进入鼠标移入状态
* a:active 为被点击的超链接增加样式
* :focus{} 为获取焦点后的元素添加样式
* ::selection{} 为鼠标选中的部分添加样式
* ::moz-selection()为火狐浏览器的鼠标选中的部分添加样式
* 第一个元素选择器 :first-child{}
* 唯一元素选择器 :only-child{}
* 最后一个元素选择器 :last-child{}
* 靠类型选第一个元素选择器 xx:first-of-type
* 靠类型选第几个元素选择器 xx:nth-of-type(odd) (3n+2)表示从2开始的每3个元素
* 靠类型选唯一元素选择器 xx:only-of-type
* 靠类型选最后一个元素选择器 xx:last-of-type
* 找内容为空的元素 :empty
* 找不包含x的元素 :not(xx)
* 伪元素（在特殊的位置进行样式修改）
* :first-letter 第一个字
* :first-line 第一行
* :before 添加在元素的最前面，一般配合content属性一起使用
* :after 添加在元素的最后面
* Css的继承性：并不是所有子元素都会继承父元素 ，背景相关的，边框相关的都不会被继承。
* Css的优先级：当使用不同的选择器选中同一个元素时，并且设置相同的样式时，由选择器优先级决定显示
* 优先级规则：
* 内联样式 ：优先级 1000
* ID选择器 ：优先级100
* 类和伪类： 优先级 10
* 元素选择器 优先级 1
* 通配选择器 优先级0
* 继承的样式 优先级 没有优先级
* 当选择器包含多种选择器时，优先级会相加然后再比较，
* 但是，选择器优先级计算不会超过他的最大的数量级（99，999）
* 如果选择器的优先级一样，使用后声明的选择器
* 可以在样式的最后，添加一个 !important， 则此时该样式会超过所有优先级显示，超过内联样式，开发中尽量不要使用
* Css的伪类的顺序：涉及到a的伪类有四个(优先级相同)：
* Link
* Visited （link和visited不能写在hover和active的下面）
* Hover （hover不能写在avtive下面）
* Active

# html的文本标签：

* <em>
* <strong>
* <i> 斜体，无语义
* <b> 粗体，无语义
* <small>
* <cite> 表示参考的内容，如书名，歌名，电影名啥的
* <q> 表示短的引用的内容
* <blockquote> 表示长引用（块级） 必须放在div里面
* <sup> 设置一个上标
* <sub> 设置一个下标
* <ins> 表示插入的内容 （）
* <del> 表示删除的内容（删除线）
* <pre> 是一个预格式标签，会将代码中的格式保存，不会忽略多个空格
* <code> 语义上表示代码的文本标签，并没有任何效果，一般套在pre标签里面一起使用来引用代码。

# html中的列表

* 无序列表
* 通过ul和li标签 一个li就是一个列表项
* 通过type属性可以修改无序列表的而项目符号
* Disc
* Square
* Circle
* 一般不用默认，因为各个浏览器的默认样式不相同
* Ul和li都是块级元素，因为都会霸占一行
* 有序列表
* 通过ol和li标签来表示
* 通过type属性指定序号的类型
* 1，默认值，使用阿拉伯数字
* a/A 采用小写或者大写作为序号
* i/I 采用小写或者带小额的罗马数字作为序号
* 定义列表
* 对于一些词汇或者内容进行定义
* 使用dl 和dt和dd标签来创建
* Dt表示被定义的内容
* Dd表示定义内容的描述
* 无序列表与有序列表与定义列表三者可以互相嵌套

# 网页中的单位：

* 长度单位
* 像素px：我们在网页中使用的最多的一个单位
* 一个像素就是我们屏幕中的一个小点1920乘以1080就是1920个像素
* 不同显示器一个像素的大小也不相同
* 百分比%：浏览器将根据其父元素的样式来计算该值
* 子元素使用百分比长度时，当父元素的属性值发生变化，子元素也会跟着发生变化
* 创建一个自适应的而页面时，经常使用百分比作为单位
* Em：与百分比类似，时相当于当前元素的字体大小进行计算的
* 1em = 1font-size
* 使用em时，当字体大小发生改变时，em也会发生改变。
* 颜色单位
* css中可以直接使用颜色的单词来表示不同的颜色，如：red,blue,green
* 也可以使用rgb
* 颜色的浓度需要一个0-255之间的值，255表示最大，0表示没有
* 颜色的浓度也可以用百分数来设置，需要设置0%-100%，0%表示0，100%表示255
* 也可以使用16进制来代表颜色
* 使用三组两位的16进制 语法：#红绿蓝
* 第一组为红色，第二组为绿色，第三组为蓝色
* 一组的值为00-ff 00表示0 ，ff表示255
* 两位重复的颜色，可以简写，如#ff00aa 可以写成#f0a

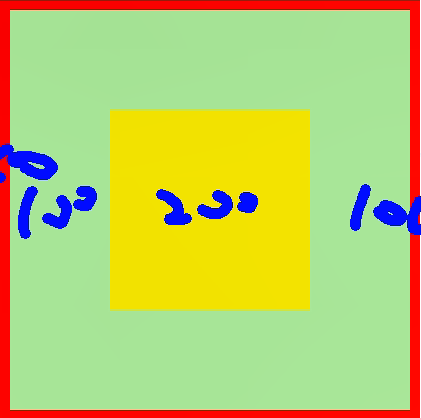
# css中的字体样式：

* Color:xx
* Font-size:xx
* 默认的字体大小时16px
* 网页中的文字都是在看不见的框中的，字体大小设置的就是这个框的大小
* Font-family：xx
* 当采用某种字体时，如果浏览器支持，则使用该字体，如果不支持，则使用默认字体
* 该样式，可以同时指定多个字体，多个字体使用都好分开
* 当采用多个字体时，浏览器会优先使用前边的字体，如果前边没有则采用下一个
* 浏览器使用的字体是使用的本地计算机中储存的字体，如果有就用
* 所以在开发中，如果字体太奇怪，用的太少，尽量不用。有可能用户的电脑中没有，达不到效果。
* 字体5大分类：
* 种类：
* Serif（衬线字体）
* Sans-serif（非衬线字体）
* Monospace（等宽字体）
* Cursive(草书字体)
* Fantasy(虚幻字体)
* 一般在设置font-family的最后设置5大分类，给浏览器识别
* Font-style 可以用来设置文字的斜体
* Normal 默认值
* Italic
* Oblique
* Font-weight 可以用来设置文字的粗体
* Normal
* Bold
* Bolder
* 100-900之间的9个值来设置加粗的程度
* 由于用户的计算机往往没有这么多级别的字体，所以有可能200跟100可能一样粗
* Font-variant
* Normal
* Small-caps 小写的字母文本以大写字母显示，但是是小型的（比正常的大写字母小），
* Font
* 可以将以上的所有属性都写在font中
* Font设置字体样式时，没有任何顺序要求，甚至可写可不写，不写按照默认值
* 但是要求文字的大小和字体必须写，而且字体必须放在最后一个样式，大小必须放在倒数第二个样式
* 如果写了大小没有写行高的话，会用默认行高覆盖掉父元素写过的行高。
* Line-height
* 在cssz中并没有为我们提供一个直接设置行间距的方式，我们只能通过设置行高来设置行间距。
* 行间距=行高-字体大小
* Line-height：
* 可用用px设置
* 也可以用百分数设置 （不常用）
* 也可以直接传一个数值，则行高会设置字体大小相应的倍数（常用 ）
* 对于单行文本，可以设置与父元素一样的高度的行高来保持文本垂直居中
* 在字体大小样式后可以用/来添加行高，如： font: 30px/50px “微软雅黑”
* 如果不写行高，会使用默认值，所以如果行高在字体大小之前设定的话，在字体大小后如果不指定行高的话，会用默认值覆盖前面指定的行高，

# Css中的文本样式

* **Text-transform**
* 可以用来设置文本的大小写
* None 默认值
* Capitalize 单词的首字母大写，通过空格来识别一个单词
* Uppercase 所有字母都大写
* Lowercase 所有的字母都小写
* **Text-decoration**
* 可以修饰文本
* None
* Underline 为文本添加下划线
* Overline 为文本添加上划线
* Line-through 为文本添加删除线
* 因为超链接会默认添加下划线，所以a 标签的text-decoration的默认值是underline，所以要去掉a标签的下划线的话可以修改text-decoration
* Letter-spacing
* 可以用来指定字符间距（可以用来设置中文）
* 用像素指定
* Word-spacing
* 可以用来指定单词之间的距离（对于中文没意义）
* Text-align
* 用于设置文本的对齐方式
* Left 默认值，文本靠坐对齐
* Right 文本靠右对齐
* Center 文本居中对齐
* Justify 两端都对齐
* 通过调整空格的大小，来达到两端对齐
* 适合段落用，单行文本没啥效果
* Text-indent
* 设置首单词缩进
* 指定一个正值时，会自动向右侧缩进，负值则相反，负值可以用来隐藏文本
* 一般用2em的单位来指定首单词缩进两个字

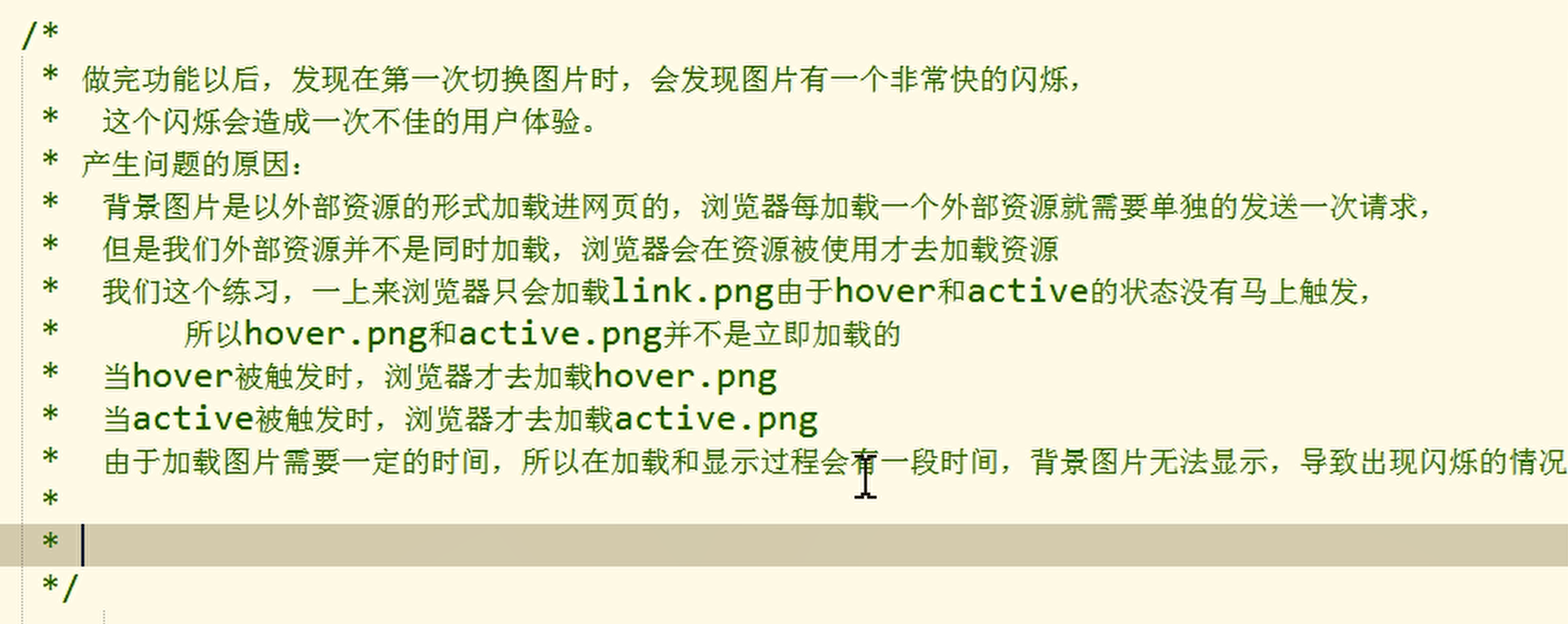
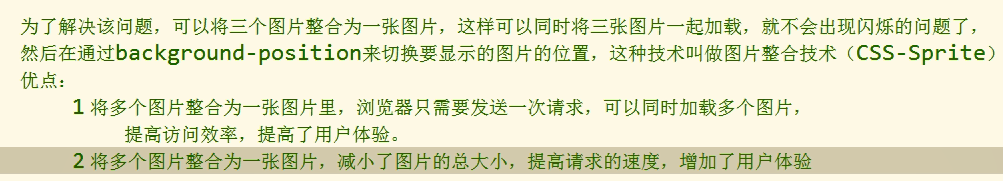
# 盒子模型

* 盒子模型一般分为四个部分
* 外边距（margin）
* 内边距 (padding )
* 边框 (border)
* 内容区 (content)
* Css样式的width和height只是设置此区域的大小
* 子元素也是存在于父元素的内容区中
* 边框
* 边框的三个样式缺一不可
* Border-width
* 可以同时指定四边的宽度,四个分别按照 上 右 下 左的顺时针的顺序设置
* 可以同时指定三个值， 分别按照上，左右，下的顺序设置
* 可以同时指定两个值 分别按照上下，左右顺序设置
* 可以指定一个值 ，全部指定
* 除了border-width外，css中还提供了border-xxx-width 来指定一侧的宽度，xxx可以为top right bottom left
* Border-color
* Color与width的方式完全一样，上面的方式全部适用
* Border-style
* 可选值
* None
* Solid
* Dotted 点状边框
* Dashed 虚线
* Doble 双线
* Style也可以分别指定四边的样式，规则也和width一样
* 大部分的浏览器中，变宽和宽度和颜色都是有默认值的，边框的样式默认值都是None
* Border
* 边框的简写样式，可以同时指定四个边框的样式，宽度，颜色，
* 没有任何顺序要求
* Border一指定就是同时指定四个边，不能分别同时指定
* 可以使用border-xxx属性来指定单独一边的style ,color,width
* 内边距
* 盒子的内容区与边框之间的距离 padding （只有长度，无颜色color与样式style）
* 一共有四个四个方向的内边距，可以用padding-xxx来设置四个方向的内边距
*  绿色部分为内边距，黄色部分为内容区，红色部分为边框
* 盒子的大小就是由，内边距+内容区+边框决定的
* 改变盒子的大小，可以通过修改这三个值去修改
* 内边距可以用单独的padding简写同时设定4个值，三个值，两个值，一个值。
* Padding:xxpx xxpx xxpx xxpx 上右下左
* Padding: xxpx xxpx xxpx 上 左右 下
* Padding :xxpx xxpx 上下 左右
* Padding xxpx 四边
* 外边距
* 当前盒子与其他盒子的距离
* 不会影响可见框的大小，只会影响盒子的位置
* 四个方向的外边距 margin-top, margin-right, margin-bottom, margin-left
* 由于页面中的元素都是靠左靠上摆放的，
* 所以注意当我们设置上和左外边距时，盒子自身的位置会发生变化
* 如果设置右和下外边距时，会改变其他盒子的位置
* 外边距也可以指定为负值，盒子会往反方向移动，如果设置右和下外边距的负值时，其他盒子会移动，产生重叠
* Margin还可以设置为auto
* 一般只设置水平方向为auto
* 如果只指定左外边距或者右外边距为auto时，会将外边距设置为最大值
* 如果只指定上外边距和下外边距为auto时，会将外边距设置为0
* 如果指定left和right同时设置为auto，会将两侧的margin设置为相同的值，会使元素自动在父元素中水平居中（常用），（必须保证宽度不是百分百）
* 外边距同样可以使用简写属性margin，用法为padding一样
* 垂直外边距的重叠
* 兄弟元素之间相邻的垂直外边距会发生外边距的重叠，取最大值而不是像水平外边距一样取和
* 父子元素之间的垂直外边距相邻时，则子元素的外边距会设置给父元素（单独设置子元素的上外边距时，父元素也会跟着下移）
* 如果不希望父元素下移的话，需要通过父元素的padding或者border来改变父子的垂直外边距相邻状态，从而可以只移动子元素的盒子（需注意，增加padding和Border会改变盒子的可见大小）
* 也可以不设置子元素的外边距，只增加父元素的内边距来使子元素下移
* 默认样式
* 浏览器中有默认的外边距，内边距的默认样式，一般开发中不适用默认的样式，所以会统统去掉
* \*{margin:0 ; padding : 0}
* 内联元素的盒子模型
* 内联元素不能设置width和height
* 可以设置水平的内边距 会影响页面的布局
* 可以设置垂直的内边距 但是不会影响页面的布局
* 可以设置水平的边框，会影响页面的布局
* 可以设置垂直的边框 ，不会影响页面的布局
* 可以设置水平的外边距，会影响页面的布局，相邻不会重叠
* 不可以设置垂直的外边距
* Display样式
* 可以修改元素的类型
* 可选项
* Inline 内联元素
* Block 块元素
* Inline-block 行内块元素
* 既有内联元素的特点，又有块元素的特点
* 既可以设置宽高，又不会独占一行
* None 不会显示元素，并且元素不会再页面中占有位置
* 需要与visibility区别
* Visibility可选项
* Visible 显示元素
* Hidden 隐藏元素，但是位置还在
* Overflow样式
* 子元素默认是存在在父元素的内容区中，超出父元素的子元素的内容称为溢出的内容
* 通过overflow可以处理溢出的内容
* 可选项
* Visible 默认值，显示溢出的内容
* Hidden 溢出的内容被裁减
* Scroll 会为父元素添加滚动条，不论内容是否溢出，都会添加水平和垂直的滚动条
* Auto 会根据需求自动添加滚动条
* 文档流
* 文档流处在网页的最底层，它表示的事一个页面中的位置，我们创建的元素都默认在文档流中
* 块元素
* 在文档中会独占一行，块元素会自上向下排列
* 块元素在文档流中默认宽度是父元素的100%
* 块元素在文档流中默认高度是被内容（子元素）撑开的
* 当元素的宽度的值为auto，此时指定的内边距不会影响可见框的大小，而是会自动修改宽度来适应内边距
* 内联元素
* 在文档流中只占自身的大小，会默认从左向右排列，如果一行中不足以容纳所有的内联元素，则换到下一行，继续自左向右
* 在文档流中，内联元素的宽度高度都会被内容撑开
* 浮动
* 块元素在文档流中是默认垂直排列的，所以div中会自上而下的依次排开
* 如果希望快元素在页面中水平排列的话，可以使块元素脱离文档流
* 使用float来使元素浮动，从而脱离文档流
* None 默认值
* Left 元素立即脱离文档流，向页面的左侧浮动
* Right元素立即脱离文档流，向页面的右侧浮动
* 当一个元素设置浮动以后（float属性是一个非none的值），元素立即脱离文档流，它下边的元素就会往上移动，浮动的元素会尽量往页面的左上或者右上浮动，直到遇到父元素边框，或者遇到其他浮动元素
* 如果需要浮动的元素上边是一个块元素，那么浮动后的元素会在块元素的盒子下面浮动
* 浮动的元素不会超过他上边的兄弟浮动元素，最多一边齐
* 浮动的元素不会盖住文件，文字会自动环绕在浮动元素的周围，可以通过浮动设置文字环绕图片的效果
* 当元素设置浮动以后，会完全脱离文档流，块元素的高度和宽度都会自动跟块元素中内容自动撑开或者缩小。
* 内联元素脱离文档流以后会变成块元素，块元素的高度和宽度都会自动跟块元素中内容自动撑开
* Clear样式
* 可以清楚其他浮动元素对该元素的影响
* 可选项
* None
* Left : 清楚左侧浮动对当前元素的影响
* Right 清楚右侧浮动元素对当前元素的影响
* Both 清楚两侧浮动元素对当前元素的影响（清楚对他影响最大的浮动元素的影响）
* 高度坍塌问题
* 高度塌陷
* 在父元素的高度没有声明的情况下，如果子元素浮动，脱离文档流的话，高度将塌陷，下边元素将上移，导致布局混乱
* 解决高度塌陷问题1
* 根据W3C的标准，在页面元素都有一个隐含的属性叫做，Block Formatting Context 简称BFC 该属性可以设置打开或者关闭，默认是关闭。
* 开启BFC时，元素将会具有如下的特性
* 父元素的垂直外边距不会和子元素重叠 （相邻父子元素的外边距会传递）
* 开启BFC的元素不会被浮动元素所覆盖
* 开启BFC的元素可以包含浮动的子元素
* 开启BFC方式4种方式
* 设置元素浮动：虽然可以被子元素撑开，但是会导致宽度丢失，且下边元素还是会上移
* 设置元素绝对定位（副作用与上面一样）
* 设置元素为inline-block：宽度会丢失
* 将元素的overflow设置为一个非visible的值 （设置为hidden是副作用最小的开启BFC的最佳方式）
* 在IE6及以下的浏览器中没有BFC但是可以通过 ZOOM:1 来实现同样的样式
* 解决高度塌陷问题2
* 可以直接在高度坍塌的父元素里添加一个空白的div，由于这个div并没有浮动，所以他是可以撑开父元素的高度的，然后该div进行清楚其他元素浮动的影响，这样就可以通过这个空白div来撑开父元素的高度了。
* 此方法基本没副作用
* 代码<div class=’box’> <浮动div模块><清楚浮动的div><div/>
* 该方法是用添加结构(HTML)的方式来改变样式（CSS）
* 解决高度坍塌问题3
* 通过after伪类，进行box1的后边
* 可以通过after伪类向元素的最后添加按一个空白的块元素，然后对其清除浮动
* 代码 clear:after{content: ””; displayed: block ; clear: both }

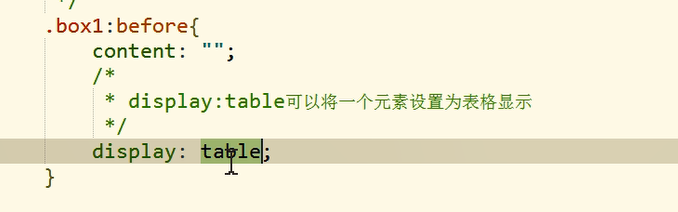
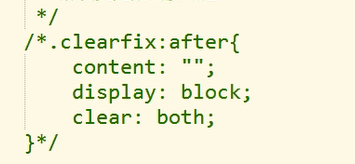
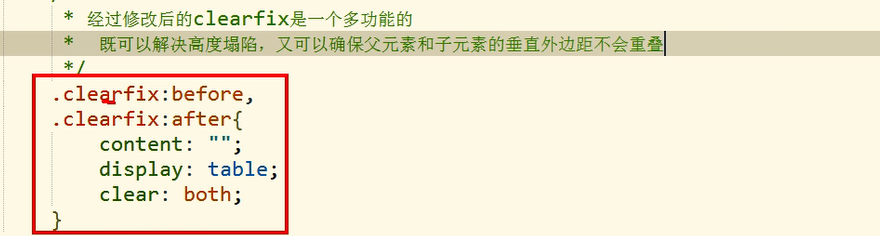
# 定位样式

* 将指定的元素摆放到页面的任意位置
* 可以任意的摆放我们的元素
* 可选值
* Static 默认值，没有开启定位
* Relative 开启元素的相对定位
* Absolute 开启元素的绝对定位
* Fixed 开启元素的固定定位（绝对定位的一种）
* Relative
* 当开启相对定位而不设置偏移量时，元素不会发生任何变化
* 开启定位后，可以通过left ,right top bottom四个属性来设置元素的偏移量（相对于元素原来在文档流中的定位）
* 使用了相对定位的元素不会脱离文档流（有点像灵魂出窍）
* 相对定位会使元素提升一个层级(z-index)
* 相对定位不会改变元素的性质，块还是块，内联还是内联
* Absolute
* 开启绝对定位，会使元素脱离文档流
* 开启定位后，如果不设置偏移量，则元素的位置不发生变化
* 绝对定位时相对于离他最近的最近的开启了定位的祖先元素进行定位的（一般开启了子元素的绝对定位，都会开启父元素的相对定位）
* 如果所有祖先的都没有开启定位的话，会相对于浏览器窗口定位
* 绝对定位也会使元素提升一个层级
* 绝对定位会改变元素的性质，内联元素会变成块元素。
* Fixed
* 固定定位也时一种绝对定位，大部分特点都和绝对定位一样
* 不同的是：固定定位永远都会相对于浏览器窗口进行定位的。跟祖先没毛线关系
* 固定定位会永远固定在浏览器窗口的某个位置，不会跟随着滚动条滚动消失
* IE6不支持固定定位
* z-index
* 如果定位元素的层级是一样的话，下边元素会盖住上边元素
* 通关z-index属性可以用来设置元素的层级
* 没有开启定位的元素，无法使用z-index
* 父元素的层级再高，也不会盖住子元素
* Opacity
* 可以用0-1来设置透明度，0表示完全透明，1表示完全不透明
* IE8及以下浏览器不支持
* IE8及以下属性需要使用如下属性 alpha(opacity=0-100)

# 背景样式

* Background-color
* Background-image
* Background-image: url(xxxx.xxx)
* 如果背景图片大于元素大小，默认会显示图片的左上角
* 如果背景图片等于元素大小，显示全部图片
* 如果背景图片小于元素大小，则会默认将背景图片平铺满元素
* 同时为元素指定背景颜色和背景图片时，会将颜色作为底色
* Background-repeat
* 可选值
* Repeat 默认值，背景图片会双方向重复（平铺）
* No-repeat 背景图片不会重复，有多大显示多大(默认左上角显示)
* Repeat-x
* Repeat-y
* Background-position
* 可以调整背景图片在元素中的位置
* 可选值：top left right center 中的一个值或者两个不冲突的值随意组合。（九宫格）
* 可选值2：可以直接用x,y进行偏移设置（可指定负值）
* Background-attachment
* 可选值：
* Scoll 背景图片随滚动条滚动
* Fixed 不随滚动条滚动 （背景图片永远相对于浏览器来定位的），一般设置给body元素
* Background
* 通关该属性可以同时设置所有背景相关的样式
* 没有顺序的要求，也没有数量的要求，不写的样式会使用默认值
* 按钮练习：
* 
* 

# Html中的表格

* <table><tr><td>
* 长表格 <thead> <tbody> <tfoot>
* 如果在表格中没有写tbody，浏览器会自动添加tbody，所以如果使用table > tr 无法选中tr ，因为tr并不是table的子元素。
* 在古老的年代，会有人用表格布局。因为表格可以一直互相嵌套
* 使用空的<table>标签可以隔离父子元素的外边距重叠问题
  + 但是如果直接在html中解决的话，会出现无用的table结构
  + 用before的伪类添加一个table标签
  + （外边距重叠问题， 无法用block，只能用table）
  + 与高度塌陷可以配合使用
  + （高度塌陷解决方法，可用block可用table）
  + 最终解决方法：
  + 

# 表单

1. 在表单中可以用fieldset和legend标签给表单进行分组
2. Select标签下可以用optiongroup标签(lable属性)给select的值分组
3. Input可以用lable标签配合for属性指定input的id来连接两者关系。
4. 表单的input一般都是inline-block元素

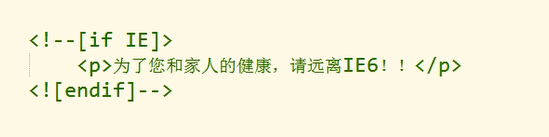
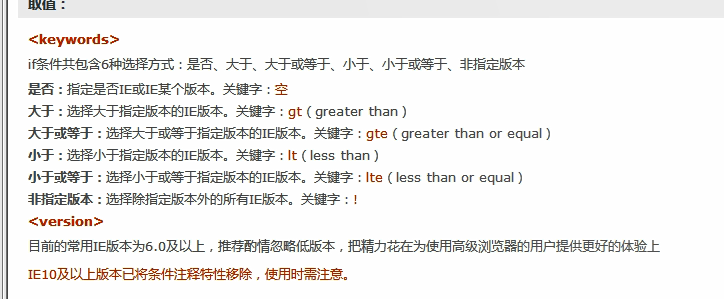
# 框架集

1. 框架集和内联框架类似，但是html5中推荐使用框架集
2. 使用frameset创建一个框架集，然后用<frame src=””>来引入页面，注意frameset不能与body标签出现在同一个页面
3. frameset属性：
   1. rows 指定框架集所有的框架一行一行的排列
   2. cols指定框架集中的所有的额框架一列一列的排列
4. frameset中还能嵌套frameset
5. frameset和ifame一样，它里面的内容都不会被搜索引擎所检索
6. 引入几个网页，浏览器就要发几个请求，所以用户体验比较差。

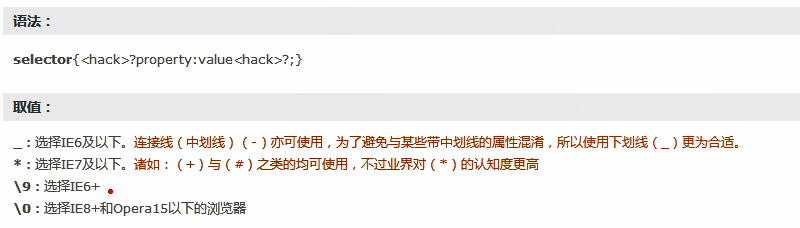
# CSS-Hack

1. 有些情况，有一些特殊的代码只需要在某些特殊的浏览器中执行。此时需要css-hack
2. CSS hack不到万不得已的情况下尽量不使用。Hack有风险，使用需谨慎。

条件hack

1. 
2. 只对ie有效，其他的浏览器会把此段代码当成注释
3. 也可以使用[if IE 6] 或者[if gt IE 6]或者[if gte IE 6] 之类的判断语句
4. 具体用法
   1. 

属性级Hack

1. 
2. 比如\_background-color: yellow 只能给ie6及其以下使用