在外层无父级开启定位的情况下，元素是会基于初始包含块来定位的，初始包含块默认为一个视窗的大小

定位的默认值为auto，而非0

Width值默认为auto，而当其为auto时，该元素默认为一个border-box

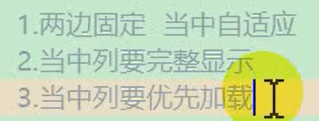
Margin、padding的百分比属性值参照的是包含块Width的属性值

盒模型可以分为两层，一层为盒模型底层，即盒模型本身，基于其上面的一层为内容层，float浮动只会使盒模型底层浮动起来，而不会对内容层造成影响(0.5级)

如改图 元素B的盒模型底层已经进入浮动A元素的下面了，而其的内容层，也就是“我是B啊”这段文字不受影响，所以没有跑到A元素下面

(PS:position定位的浮动为1级，即会将盒模型连带其内容一起浮动起来)

三列布局的三点要求



块级元素浮动起来就是浮动元素，会丢失块级元素的属性，比如margin

伪等高布局：利用大值padding将填充色铺开，再将底部边缘收上来，最后切割父级元素块，营造出一种表面等高效果

双飞翼布局：和圣杯布局同样是为了解决三列布局的问题，它和圣杯布局的区别是，圣杯是通过CSS父级padding加定位，而双飞翼仅仅是通过HTML标签嵌套，将内容放入内部标签，在对内部这个元素块添加padding

黏连布局

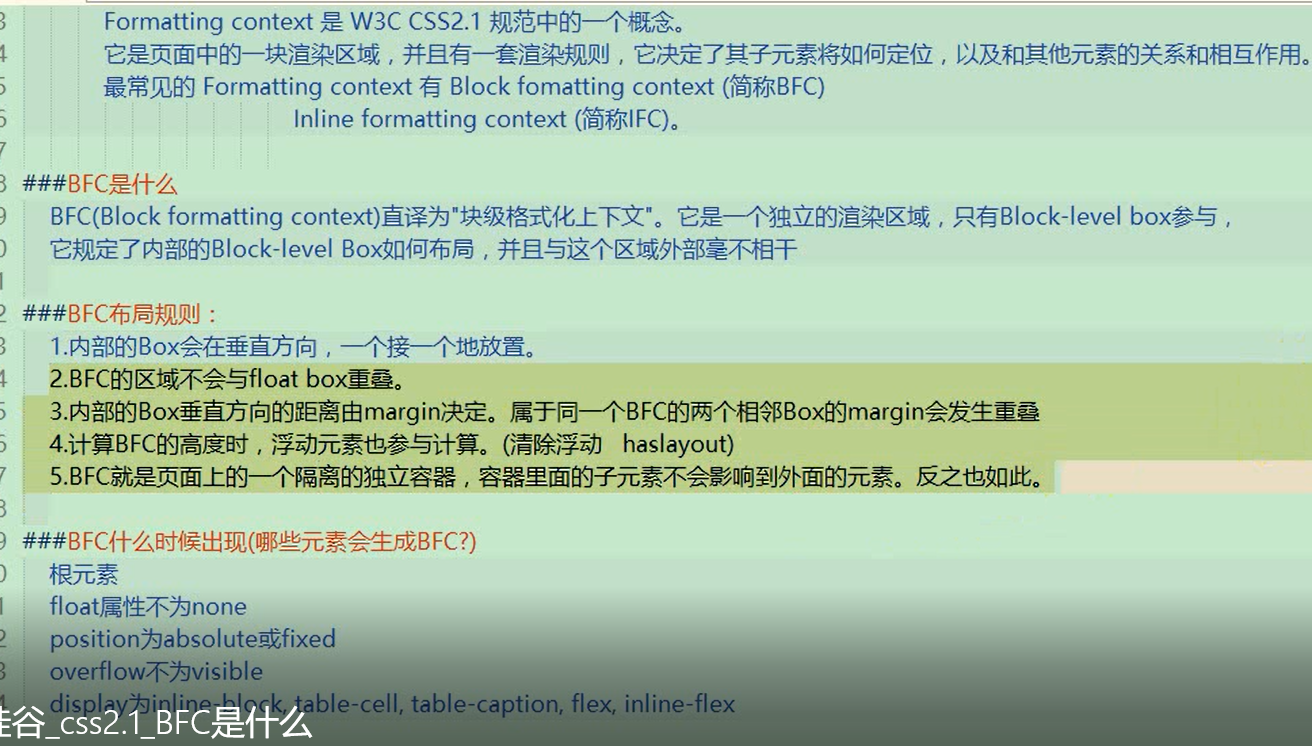
//内容不足以撑大父级，footer会默认待在可视区底部，如果内容足够多，footer会自动跟着内容走

绝对定位模仿固定定位：

让body通过html将document的高度嫁接下来，将html的overflow设为hidden，body的overflow设为auto(html和body只有一个有overflow属性，这个属性会默认转到document身上，两个都添加的话，html的overflow回转移到document身上，body的保留下来)，然后滚动栏就会跑到body身上，

然而默认的绝对定位参照是body之外的初始包含块，所以body身上的滚动栏，不会对默认绝对定位的元素造成滑动效果，就形成了看似固定定位的效果

BFC



**BFC盒模型边缘间距重叠问题**

最常见的场景就是子元素使用margin，父元素却移动了的现象，这是因为他们可能满足了BFCmargin重叠的三个条件，1.处于同一个BFC，2.都为块级元素，3.margin相邻

给父级添加overflow：hidden使我们最常用的解决方式，从原理上来讲，它是打破了上述的第一条规则，给父级开启了BFC(一个块级元素的overflow不为visible时就会开启BFC)，这样子级会处在父级的BFC中，父级则处于更外层的BFC中（根元素默认BFC），这样第一个条件就会被打破，三条规则其中任意一条不满足就不会出现margin重叠现象 [具体参照→](../小实例/CSS2.1/BFC盒模型边缘间距重叠问题.html)

**清除浮动**

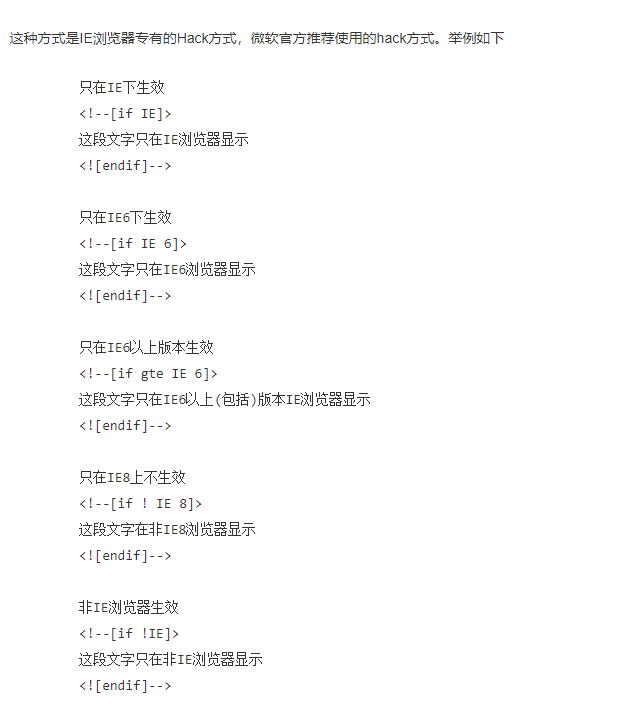


CSS hack

Hack标识符是浏览器之间的区分符，可以达到不同浏览器添加不同样式或文本内容的效果



条件注释法



PS：IE10级以上的版本已经取消hack

左右查询

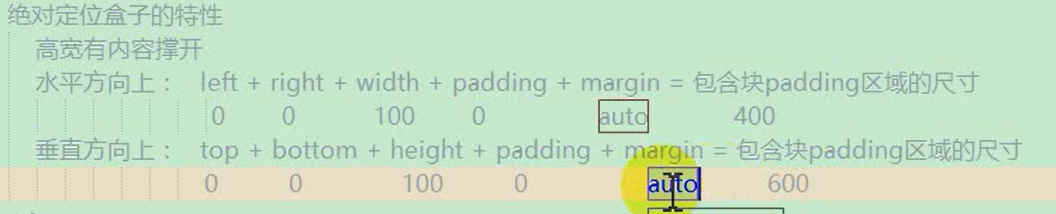
左查询：JS引擎会沿着作用链向上查找，找不到会自动全局创建

右查询：同上，只不过查不到会报错

置换元素

偶然之间发现的东西，img标签是一个行内元素，但是它可以设置宽高，img其实还是一个置换元素，所以它本身就具有宽高，同理的还有input，textare

垂直居中问题



绝对定位四个方向全设为0，margin为auto，上面的父级宽减去子元素本身的宽最后margin左右各分了150，高同理

字体与文本

Chrome默认字体为16px，最小为12px，小于12默认为12,负值默认为16px，为0的情况下无法显示字体

Font-family可选多种字体，还可从字体族中选择字体，PS：字体如果中间有空格要加引号，字体族不可加引号

Line-height：1是设置以字体大小为参照物的倍数行高，他与%不同的是，%会继承下算好的值，而行高因子会将倍数值继承下去