**盒子模型**

内容区（content）

内边距（padding）

边框（border）

外边距（margin）

Width和high只是内容区的大小 不是盒子的大小

布局：盒子将不同的盒子放在对应的位置

可见大框的值的设置=pdding\*2+中间盒子+border\*2

全部占地：pdding\*2+中间盒子+border\*2+margin\*2

**border的属性**

border-style: solid dashed/\*虚线;\*/ /\*外框的样式\*/\*/  
border-color: red;/\*外框的颜色,同上\*/

border-width: 10px;/\*外框的宽度,可顺时针设置上右下左来设置边框的宽度，两个就是上下一组。左右一组\*/

border-bottom-width: 20px;/\*用来设置底部边框宽度\*/

border-width: 10px 20px 30px; /\*上 左右 下\*/  
**三者合一：**

**border:20px blue soild;/\*必须设置三个属性，顺序不重要\*/**

padding 内容区与外边框的距离 子元素会覆盖父元素

**margin外边距**

当前盒子和其他盒子之间的距离，不影响可见框 但是会影响盒子的位置

贴左为准

* 上和左是会改变盒子位置
* 改变右和下会改变其他盒子的位置
* 外边距也可以指定一个负值，则元素会向反方向移动

margin：后面会把前面覆盖，不设置为0的时候 会有默认值8px

* margin:0 auto; /\*上下为0，左右自动 就是水平居中，auto 在的地方则那侧是最大值\*/
* 垂直方向设置成auto，则外边距默认为0

**相邻的垂直方向的外边距会重叠**

* 相邻垂直的子元素会和父元素一起变化（解决：加内容 加margin 加padding 就不会一起动了
* 垂直方向：垂直方向取最大值
* 水平方向：边距相加

设置段落文本中的内容样式 可以加一个span，

span不能设置内容区宽高

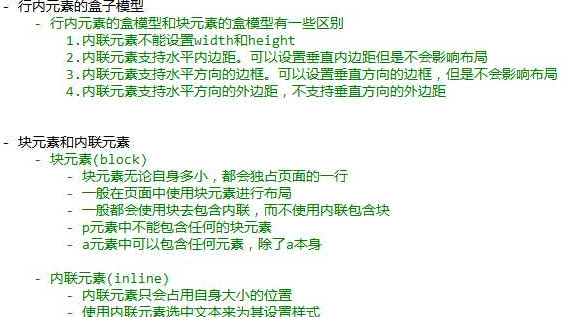
大小直接覆盖别人，不像DIV直接排斥，在行内元素里面span的border、padding magin不设置垂直方向的内边距，可设置水平方向的內边距，垂直会覆盖其他的。

**区分：**

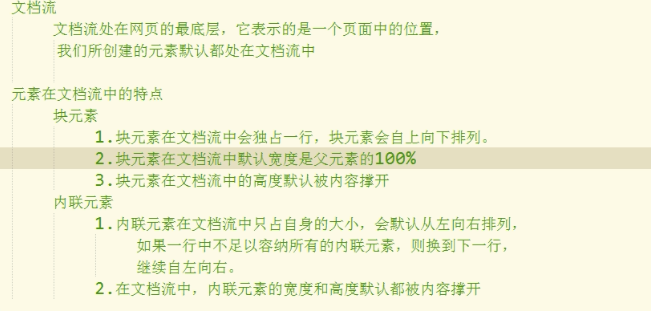
Display：none是隐藏也消失

Visility：hidden是消失了但是位置还在

**行内元素的盒子模型;**



**文档流**



**浮动：**

脱离文档流，

只能在自己的边框中浮动

紧挨着第一个浮动

盒子里面的文字浮动起来不会盖住，会环绕前面的边框显示。

**clear清除浮动**

clear可以用来清除浮动元素对其他元素的影响(浮动元素在那边就设置那边的元素)

none 不清除浮动

left：清除左侧浮动元素对当前元素的影响

right：清除右侧元素对当前元素的影响

both:清除两侧浮动元素对当前元素的影响

清除b1浮动对b1产生的影响（清除浮动后，元素会回到原来的位置）

其他元素设置clear，清除影响最大

**高度塌陷**

**高度塌陷：**在文档流中，父元素的高度是被子元素撑开的

当设置float时，子元素就浮起来父元素就塌陷，由于子元素浮起来了下面的元素就会上移

1..**BFC的核心思想**是形成一个独立空间，形成闭关自守的姿态，尽量与外界断绝联系

2.元素都有隐藏的属性BFC该属性可设置打开或关闭 默认关闭

3.**开启的特性:**

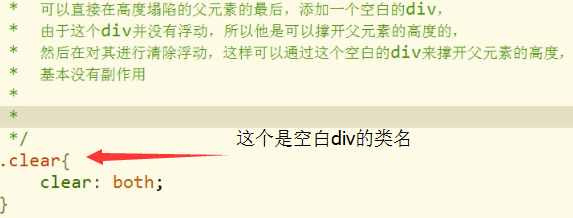
* 父元素不会跟着子元素一起移动
* 开启的BFC元素不会被浮动元素覆盖
* 开启BFC元素可以包含浮动元素

4.**开启BFC的方法**：

* 让元素浮动（前后两个都浮动，虽然可撑开，但是会导致父元素宽度丢失）
* 设置元素绝对定位
* 设置元素display为：inline-block（会导致宽度丢失）
* 将元素overflow设置为一个非visiblie的值hidden、auto（副作用更小） （把高度去掉再写）

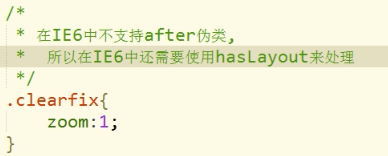
**5.解决高度塌陷的方法:**

* **overflow:hidden**
* **外边距重叠** 子元素和父元素一起走的问题: 加overflow:hidden 就可以让小的在大的里面自己移动
* **浮动覆盖**(左侧固定，右侧自适应)在浮动的元素加 overflow:hidden 元素不会被后面的浮动的元素覆盖
* **父元素自适应子元素大小**（在网页中大的div高度可以不设置，用此方法设置会让大的跟着小的大小变而变）： overfowl:hidden
* **添加一个空白的div**



* **在后面添加：after伪类 内容为空，化成块元素再清除浮动（可以两个一起写，就都兼）**





**盒子并排**

1. display：inline-bloack
2. float:lift

**定位**

**Position可选值**

static默认值，元素没有开启定位

relative:开启元素的相对位置

absolute;开启元素的绝对位置

fixed：开启元素的固定定位（也是绝对定位的一种）

当开启元素定位（position属性值是一个非static的值）时

可以通过left right top bottom四个属性来设置元素的偏移量

**Life：**元素相对于其定义位置的左偏移量

Right: 元素相对于其定义位置的右偏移量

**Top:** 元素相对于其定义位置的上偏移量

Bottom: 元素相对于其定义位置的下偏移量

**相对定位（relative）:**

* 相对于**自身的位置**移动，
* 灵魂出窍，位置还在，
* 相对定位会提升一个级别，不会改变其性质

**绝对定位（absolute）：**

* 相对于离他**最近的开启了定位的祖先元素**进行原点定位的，如果都所有的祖先元素都没有开启定位元素的，就相对于窗口进行定位
* 一般情况开启了子元素的绝对定位，也会给父元素开启定位
* 开启绝对定位会脱离文档流，移动了，原先位置也移动了（和区别relation区别）
* 开启之后如果不设置偏移量、则位置不会发生偏移在原来的位置（和浮动的区别）
* 绝对定位会使元素提高一个层级
* 绝对定位会改变元素性质（内联元素变成块元素）（块元素的宽度和高度默认都被内容撑开）

**固定定位（fixed）：**

* + - 固定定位也是一种绝对定位，大部分特点都相同与绝对定位

**与绝对定位的不同：**

* 固定定位永远都会相对于浏览器窗口进行定位，
* 不管浏览器怎么移动 它的位置都完全不变（像那种广告咨询和客服窗口跟着你转）
* IE6不支持固定定位

**遮罩:**

用background-color:rgba（0，0,0,0.5）红绿蓝透明度，这个不会继承

.COVER{  
 width: 100%;  
 height: 100%;  
 background-color: #fff;  
 position: fixed;  
 top:0;  
 left: 0;  
 opacity: 0.5;设置全部的透明的，会继承  
 }

#login{  
 width: 500px;  
 height: 300px;  
 background-color: black;  
 margin: 50px auto;  
}

**元素的层级：**

如果定位元素层级一样，下边元素会盖住上边的

可以用z-index:整数值 ; 可以提高层级，对于没有开启定位的不能使用z-index:整数值 ;

父元素层级再高也不会盖住子元素

**导航条的练习**

**/\*清除默认样式\*/**

\*{  
 margin: 0px;  
 padding:0;  
 }  
 ul{  
 list-style: none;/\*去除dian\*/  
 }/\*清除默认样式\*/

a{ text-decoration: none; }

**设置导航条背景颜色**

.Div {  
 width: 400px;  
 background-color: lightgreen;  
 overflow: hidden;  
 margin: 50px auto;  
} /\*设置背景颜色，注意背景颜色的塌陷用overflow\*/

**设置li的样式** li{  
 width: 100px;  
 color: lightgoldenrodyellow;  
 text-align: center;  
  
 float: left;  
 padding: 5px 0;  
 }  
 li:hover{  
 background-color: lightcoral;  
 }

**设置a的样式**

a:hover{  
 color: lightgoldenrodyellow;  
 }  
 a{  
 color: lightgoldenrodyellow;  
 display: inline-block;  
 width: 100%;  
 padding: 5px 0;  
 }

在浏览器一定范围最合适 可以设置最大最小宽度

min-width：;

max-width:;