# 布局：display，position，float，媒体查询，inline-block，column

Display:none用于隐藏元素，不占据空间

visibility: hidden; 还会占据空间

既然没有比这更好的方法，一些CSS开发者想要页面上所有的元素都有如此表现。所以开发者们把以下CSS代码放在他们页面上：

\* {

-webkit-box-sizing: border-box;

-moz-box-sizing: border-box;

box-sizing: border-box;

}

Position的用法

## Static

static 是默认值。任意 position: static; 的元素不会被特殊的定位。一个 static 元素表示它不会被“positioned”，一个 position 属性被设置为其他值的元素表示它会被“positioned”。

Relative

只是relative值作用和static一样

但设置 top 、 right 、 bottom 和 left 属性会使其偏离其正常位置。其他的元素的位置则不会受该元素的影响发生位置改变来弥补它偏离后剩下的空隙。

Fixed

相对于视窗来定位，这意味着即便页面滚动，它还是会停留在相同的位置

## absolute

相对于最近的“positioned”祖先元素

注：

当设置float的元素高于父级元素，会超出，而不能撑开父级元素，父级加overflow：auto

一个div里的图片随div大小变化，width：xx%

inline-block

**布局模型**

**http://www.cnblogs.com/chaixiaozhi/p/8481253.html**

1.流动模型（Flow） 默认的  
2、浮动模型 (Float)  
3、层模型（Layer）

**浮动模型 (Float)**

浮动元素的定位还是基于正常的文档流，然后从文档流中抽出并尽可能远的移动至左侧或者右侧，**文字内容会围绕在浮动元素周围**。它只是改变了文档流的显示，而没有脱离文档流

**层模型（Layer）**

**相对定位**

相对于**以前的位置移动，偏移前的位置保留不动。在使用相对定位时，就算元素被偏移了，但是他仍然占据着它没偏移前的空间。**

**绝对定位**

相对于其最接近的一个具有定位属性的父包含块进行绝对定位。

**在文档流中是不占据空间的，如果某元素设置了绝对定位，那么它在文档流中的位置会被删除；**

**我们可以通过z-index来设置它们的堆叠顺序**

**固定定位**

相对移动的坐标是视图（**屏幕内的网页窗口**）本身

**z-index**

z-index值较大的元素将叠加在z-index值较小的元素之上。

### 只对定位元素有效

### 父子关系处理

如果父元素z-index有效，那么子元素无论是否设置z-index都和父元素一致，会在父元素上方

如果父元素z-index失效（未定位或者使用默认值），那么定位子元素的z-index设置生效

如果两个元素都没有设置z-index，使用默认值，一个定位一个没有定位，那么定位元素覆盖未定位元素

**Float**

[**http://www.cnblogs.com/chaixiaozhi/p/8481778.html**](http://www.cnblogs.com/chaixiaozhi/p/8481778.html)

float被设计出来的初衷是用于——文字环绕效果

1. **破坏性：**

**父级div会有坍塌现象，可以清除浮动来解决**

1. **包裹性：**

**元素设置了float，边界会包裹文字**

1. **清空格**

本来块对象之间有空格，设置float之后就紧密项链

1. 清除浮动
2. 为父元素添加overflow:hidden
3. 浮动父元素
4. 所有浮动元素下方添加一个clear:both的元素
5. 

布局方式