

1. 定位

1.1. 为什么需要定位

元素可以自由的在一个盒子内移动位置，并且压住其他盒子

有些元素是固定在页面的某个位置上的

以上两种情形标准流和浮动都无法快速实现，此时需要**定位**来实现

所以：

- 浮动可以让多个块级盒子一行没有缝隙排列显示，经常用于横向排列盒子
- 定位则是可以让盒子自由的在某个盒子内移动位置或者固定屏幕中某个位置，并且可以压住盒子

1.2. 定位组成

定位：将盒子定在某一个位置，所以定位也是在摆放盒子，按照定位的方式移动盒子

定位=定位模式+边偏移

定位模式用于指定一个元素在文档中的定位方式，边偏移则决定了该元素的最终位置

定位模式：

定位模式决定元素的定位方式，它通过css的position属性来设置，其值可以分为四个：

值	语义
static	静态定位
relative	相对定位
absolute	绝对定位
fixed	固定定位

边偏移

边偏移就是定位的盒子移动到最终位置。有top, bottom, left, right四个属性

1.3. 静态定位static（了解）

静态定位是默认定位方式，无定位的意思

语法：

```
选择器 {position:static;}
```

- 静态定位按照标准流的特性摆放位置，它没有边偏移
- 静态定位在布局中很少用到

1.4. 相对定位（relative）（重要）

相对定位是元素在移动位置时，是相对于它原来的位置来说的（自恋型）

语法：

```
选择器 {position:relative;}
```

相对定位的特点：（务必记住）

- 它是相对于自己原来的位置来移动的（移动的时候的参照点是自己原来的位置）
- 原来在标准流的位置继续占有，后面的盒子仍然以标准流的方式对待他。（不脱标，继续保留原来位置）

1.5. 绝对定位absolute(重要)

绝对定位是元素在移动位置的时候，是相对于它祖先元素来说的（拼爹型）

语法：

```
选择器 {position:absolute;}
```

绝对定位的特点：（务必记住）

- 如果没有祖先元素或者祖先元素没有定位，则以浏览器为准定位（document文档）
- 如果祖先元素有定位（相对，绝对，固定定位），则以最近的一级的有定位祖先元素为参考点移动位置
- 绝对定位不再占有原先的位置（脱标）

1.6. 子绝父相的由来

子绝父相的意思是：子级要用绝对定位的话，父级要用相对定位

子级绝对定位，不会占有位置，可以放到父盒子里面的任何一个地方，不会影响其他的兄弟盒子
父盒子需要加定位限制子盒子在父盒子内显示
父盒子布局时，需要占有位置，因此父亲只能是相对定位

1.7. 固定定位fixed（重要）

固定定位是元素固定于浏览器可视区的位置。

主要使用场景：可以在浏览器页面滚动时元素的位置不会改变

语法：

```
选择器:{position:fixed;}
```

固定定位的特点：（务必记住）

- 以浏览器的可视窗口为参照点移动元素
- 跟父元素没有任何关系
- 不随滚动条的滚动滚动
- 固定定位不再占有原先的位置
- 固定定位也是脱标的，其实固定定位也可以看做是一种特殊的绝对定位

固定定位小技巧：固定在版心右侧位置

小算法：

- （1）让固定定位的盒子left: 50%，走到浏览器可视区（也可以看做版心）的一半位置
- （2）让固定定位的盒子margin-left:版心宽度的一半距离。多走版心宽度的一半位置

就可以让固定定位的盒子贴着版心右侧对齐了

1.8. 粘性定位sticky（了解）

粘性定位可以被认为是相对定位和固定定位的混合

语法：

```
选择器 {position:sticky;top:10px;}
```

粘性定位的特点：

- 以浏览器的可视窗口为参照点移动元素（固定定位的特点）
- 粘性定位占有原先的位置（相对定位特点）
- 必须添加top,left,right,bottom其中一个才有效

- 跟页面滚动搭配使用，兼容性较差，IE不支持

1.9. 定位叠放次序z-index

在使用定位布局时，可能会出现盒子重叠的情况。此时，可以使用z-index来控制盒子的前后顺序

语法：

选择器 {z-index:1;}

- 数值可以是正整数，负整数或0，默认是auto，数值越大，盒子越靠上
- 如果属性值相同，则按照书写顺序，后来居上
- 数字后面不能加单位
- 只有定位的盒子才会有z-index属性

1.10. 定位的拓展

绝对定位的盒子居中

加了绝对定位的盒子不能通过margin: 0; 水平居中，但是可以通过一下计算方法实现水平和垂直居中：

定位特殊特性

- 绝对定位和固定定位也和浮动类似
- 行内元素添加绝对或者固定定位，可以直接设置宽度和高度
- 块级元素添加绝对定位或者固定定位，如果不给宽度或者高度，默认是内容的大小

脱标的盒子不会触发外边距塌陷

浮动元素、绝对定位（固定定位）元素都不会触发外边距合并的问题

浮动元素会压住下面标准流的盒子，但是不会压住下面标准流盒子里面的文字

但是绝对定位（固定定位）会完全压住盒子

浮动之所以不会压住文字，因为浮动产生的目的最初是为了做文字环绕效果的，文字会围绕浮动元素

如果一个盒子既有left属性又有right属性，则默认会执行left属性，同理top bottom会执行 top属性

2. 网页布局总结

通过盒子模型，清楚知道大部分HTML标签是一个盒子

通过浮动和定位，可以让每个盒子排列成为网页

一个完整的网页，是标准流、浮动、定位一起完成布局的，每个都有自己的专门用法

2.1. 标准流

可以让盒子上下排列或者左右排列，垂直的块级盒子显示就用标准流布局

2.2. 浮动

可以让多个块级元素一行显示或者左右对齐盒子，多个块级盒子水平显示就用浮动布局

2.3. 定位

定位最大的特点是有层叠的概念，就是可以让多个盒子前后叠压来显示。如果元素自由在某个盒子内移动就用定位布局

3. 元素的显示与隐藏

本质：让元素在一个页面中隐藏或者显示出来

display显示隐藏

visibility显示隐藏

overflow溢出显示隐藏

3.1. display属性

display属性用于设置一个元素应如何显示

display:none;隐藏对象

display:block;除了转换为块级元素之外，同时还有显示元素的意思

display属性隐藏元素后，不再占有原来的位置

后面应用极其广泛，搭配JS可以做很多网页特效

3.2. visibility可见性

visibility属性用来指定一个元素应可见还是隐藏

visibility:visible;元素可视

visibility:hidden;元素隐藏

visibility隐藏元素后，继续占有原来的位置

如果隐藏元素想要原来的位置，就用visibility:hidden;

如果不要原来的位置，就用display:none; (用的更多)

3.3. overflow溢出

overflow属性指定了如果内容溢出一个元素的框（超过其指定高度及宽度）时，会发生什么

属性值	描述
visible	不剪切内容也不添加滚动条
hidden	不显示超过对象尺寸的内容，超出的部分隐藏掉
scroll	不管超出内容与否，总是显示滚动条
auto	超出自动显示滚动条，不超出不显示滚动条

一般情况下，我们都不想让溢出的内容显示出来，因为溢出的部分会影响布局

但是如果有定位的盒子，请慎用overflow: hidden 因为它会隐藏掉多余的部分