

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 = padding*2 + 中间盒子 + border*2

全部占地: padding*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 solid; /*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 margin 不设置垂直方向的内边距, 可设置水平方向的内边距, 垂直会覆盖其他的。

区分:

Display: none 是隐藏也消失

Visibility: hidden 是消失了但是位置还在

行内元素的盒子模型:

- 行内元素的盒子模型
 - 行内元素的盒模型和块元素的盒模型有一些区别
 1. 内联元素不能设置width和height
 2. 内联元素支持水平内边距。可以设置垂直内边距但是不会影响布局
 3. 内联元素支持水平方向的边框。可以设置垂直方向的边框，但是不会影响布局
 4. 内联元素支持水平方向的外边距，不支持垂直方向的外边距
- 块元素和内联元素
 - 块元素(block)
 - 块元素无论自身多小，都会独占页面的一行
 - 一般在页面中使用块元素进行布局
 - 一般都会使用块去包含内联，而不使用内联包含块
 - p元素中不能包含任何的块元素
 - a元素中可以包含任何元素，除了a本身
 - 内联元素(inline)
 - 内联元素只会占用自身大小的位置
 - 使用内联元素选中文本来为其设置样式

文档流

文档流

文档流处在网页的最底层，它表示的是一个页面中的位置，我们所创建的元素默认都处在文档流中

元素在文档流中的特点

块元素

1. 块元素在文档流中会独占一行，块元素会自上向下排列。
2. 块元素在文档流中默认宽度是父元素的100%
3. 块元素在文档流中的高度默认被内容撑开

内联元素

1. 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。
2. 在文档流中，内联元素的宽度和高度默认都被内容撑开

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 visible 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的大小变而变): overflow:hidden
- 添加一个空白的 div

```
/* 可以直接在高度塌陷的父元素的最后, 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,
* 然后在对其进行清除浮动, 这样可以通过这个空白的div来撑开父元素的高度,
* 基本没有副作用
*
*
*/
.clear{
    clear: both;
}
```

这个是空白div的类名

- 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)

```
.clearfix:after{
    /*添加一个内容*/
    content: "";
    /*转换为一个块元素*/
    display: block;
    /*清除两侧的浮动*/
    clear: both;
}
```

```
/*
* 在IE6中不支持after伪类,
* 所以在IE6中还需要使用hasLayout来处理
*/
.clearfix{
    zoom: 1;
}
```

盒子并排

1. display: inline-block
2. float:left

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

Left: 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 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:;
```

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*/
```

