

# 1：盒模型

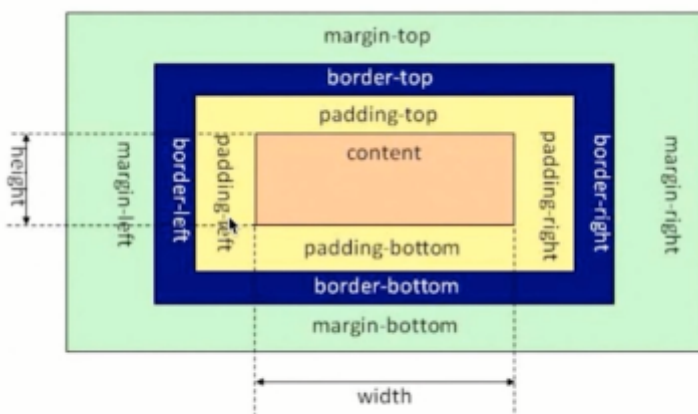
如果你在面试的时候面试官让你谈谈对盒模型的理解，你是不是不知从何谈起。这种看似简单的题其实是最不好答的。

下面本文章将会从以下几个方面谈谈盒模型。

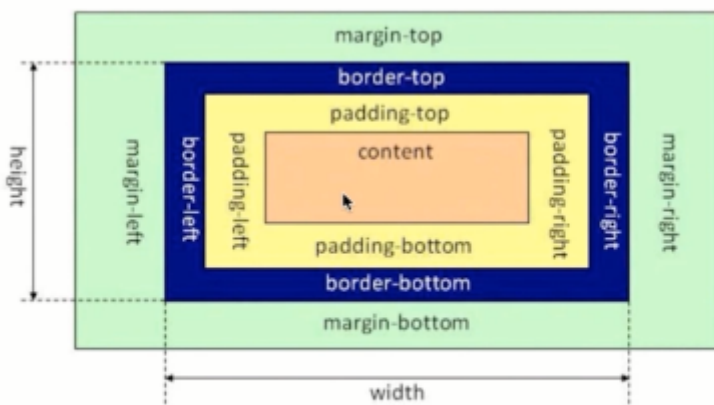
- 基本概念：标准模型 和IE模型
- CSS如何设置这两种模型
- JS如何设置获取盒模型对应的宽和高
- 实例题（根据盒模型解释边距重叠）
- BFC（边距重叠解决方案）

盒模型的组成大家肯定都懂，由里向外content,padding,border,margin.

盒模型是有两种标准的，一个是标准模型，一个是IE模型。



标准模型



IE模型

从上面两图不难看出在标准模型中，盒模型的宽高只是内容（content）的宽高，而在IE模型中盒模型的宽高是内容(content)+填充(padding)+边框(border)的总宽高。

## css如何设置两种模型

这里用到了CSS3 的属性 box-sizing

```
/* 标准模型 */
```

```
box-sizing:content-box;
```

```
/*IE模型*/
```

```
box-sizing:border-box;
```

## JS获取宽高

通过JS获取盒模型对应的宽和高，有以下几种方法：

为了方便书写，以下用dom来表示获取的HTML的节点。

### 1. dom.style.width/height

这种方式只能取到dom元素内联样式所设置的宽高，也就是说如果该节点的样式是在style标签中或外联的CSS文件中设置的话，通过这种方法是获取不到dom的宽高的。

### 2. dom.currentStyle.width/height

这种方式获取的是在页面渲染完成后的结果，就是说不管是哪种方式设置的样式，都能获取到。

但这种方式只有IE浏览器支持。

### 3. window.getComputedStyle(dom).width/height

这种方式的原理和2是一样的，这个可以兼容更多的浏览器，通用性好一些。

### 4. dom.getBoundingClientRect().width/height

这种方式是根据元素在视窗中的绝对位置来获取宽高的

### 5.dom.offsetWidth/offsetHeight

这个就没什么好说的了，最常用的，也是兼容最好的。

## 边距重叠

什么是边距重叠

如下图，父元素没有设置margin-top，而子元素设置了margin-top: 20px;可以看出，父元素也一起有了边距。

此部分是能更容易看出让下面的块的margin-top。

## 子元素

margin-top:20px;

## 父元素

没有设置margin-top

上图的代码

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Document</title>
  <style>
    *{
      margin:0;
      padding:0;
    }
    .demo{
      height:100px;
      background: #eee;
    }
    .parent{
      height:200px;
      background: #88f;
    }
    .child{
      height:100px;
      margin-top:20px;
      background: #0ff;
      width:200px;
    }
  </style>
```

```
</head>
<body>
  <section class="demo">
    <h2>此部分是能更容易看出让下面的块的margin-top。 </h2>
  </section>
  <section class = "parent">
    <article class="child">
      <h2>子元素</h2>
      margin-top:20px;
    </article>
    <h2>父元素</h2>
    没有设置margin-top
  </section>
</body>
</html>
```

## 边距重叠解决方案(BFC)

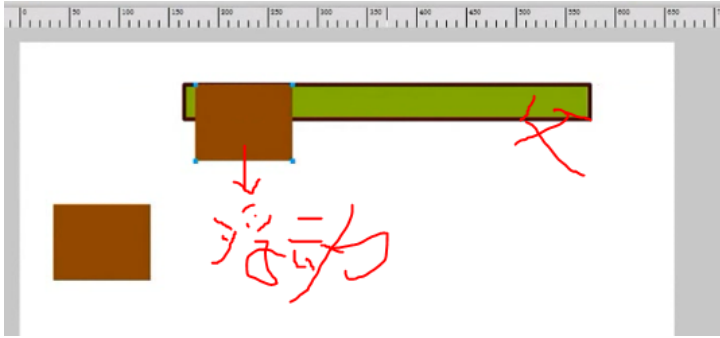
首先要明确BFC是什么意思，其全英文拼写为 Block Formatting Context 直译为“块级格式化上下文”

1:是一个独立的渲染区域，只有block-level box参与，它规定了内部的block-level box如何布局（bfc是一个页面上的独立的容器，外面的元素不会影响bfc里的元素，反过来，里面的也不会影响外面的）

2:display 为block ,list-item,table的元素，会产生bfc

### BFC的原理（bfc特性）

1. 内部的box会在垂直方向，一个接一个的放置
2. 每个元素的margin box的左边，与包含块border box的左边相接触（对于从做往右的格式化，否则相反）
3. box垂直方向的距离由margin决定，属于同一个bfc的两个相邻box的margin会发生重叠
4. bfc的区域不会与浮动区域的box重叠



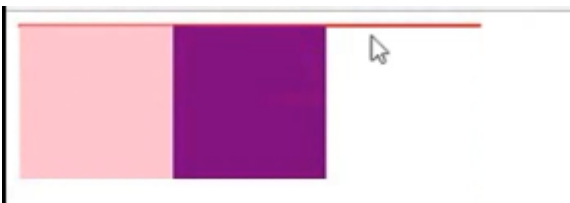
5. bfc是一个页面上的独立的容器，外面的元素不会影响bfc里的元素，反过来，里面的也不会影响外面的

### 6. 计算bfc高度的时候，浮动元素也会参与计算

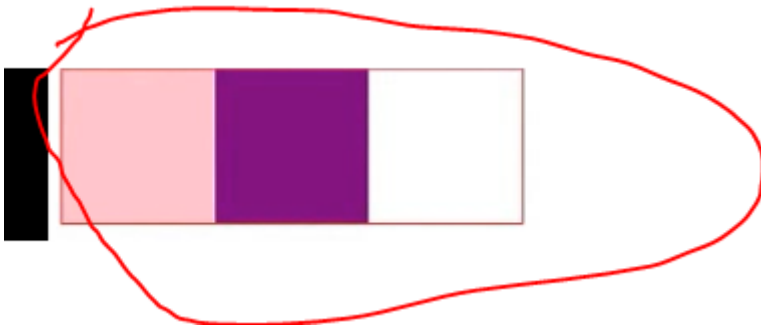
```

html head style
6   <style>
7     .father {
8       width: 300px;
9       border: 1px solid red;
10    }
11    .son1, .son2 {
12      width: 100px;
13      height: 100px;
14      background-color: pink;
15      float: left;
16    }
17    .son2 {
18      background-color: purple;
19    }
20  </style>
21 </head>
22 <body>
23 <div class="father">
24   <div class="son1"></div>
25   <div class="son2"></div>
26 </div>
27 </body>
28 </html>

```



图一：没有触发bfc,加了浮动，父级div没有被撑开



图二：触发了bfc

```
html head style
6   <style>
7     .father {
8       width: 300px;
9       border: 1px solid red;
10      overflow: hidden;
11    }
12    .son1, .son2 {
13      width: 100px;
14      height: 100px;
15      background-color: pink;
16      float: left;
17    }
18    .son2 {
19      background-color: purple;
20    }
21  </style>
22 </head>
23 <body>
24   <div class="father">
25     <div class="son1"></div>
26     <div class="son2"></div>
27   </div>
28 </body>
```

## 怎么取创建bfc

1. float属性不为none（脱离文档流）
2. position为absolute或fixed
3. display为inline-block,table-cell,table-caption,flex,inline-flex
4. overflow不为visible
5. 根元素

## 应用场景

1. 自适应两栏布局
2. 清除内部浮动
3. 防止垂直margin重叠

请参考该文章：

<https://www.cnblogs.com/chengzp/p/cssbox.html>