

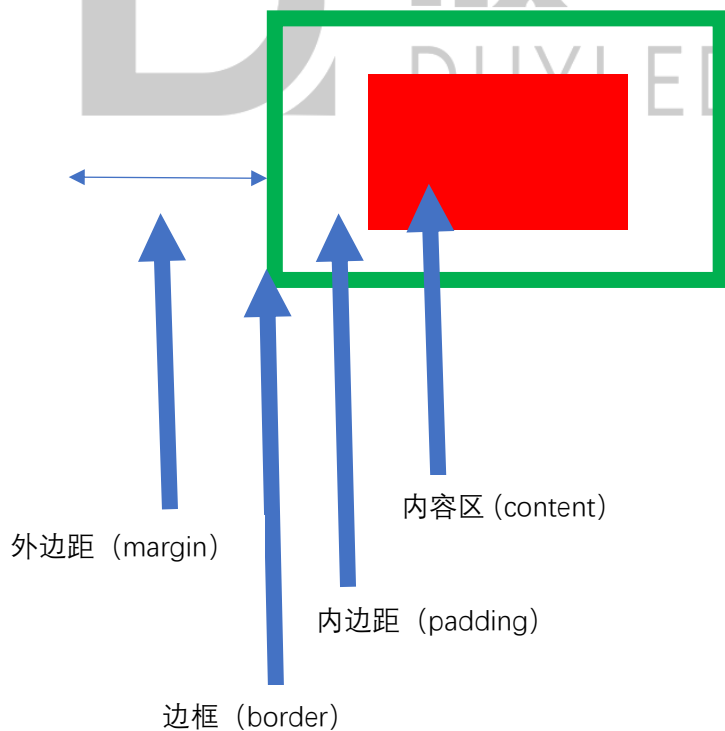
## 对于标准盒模型和 IE 盒模型你了解多少？

今天我们来探究一下盒模型，一说到盒模型，相信同学们都了解。在我们日常的布局当中我们经常的使用。但是你真的了解盒模型么？

## 盒模型讲解

我们今天只会提到 **标准盒模型**，和 **IE 盒模型**，两种盒模型模式讲解。首先来看标准盒模型。

只要是盒模型都会有这四个属性 **margin, border, padding, content** 这四个部分。也就是说一个基本的盒模型由着四部分组成。样子如下：



这里要说明的是，为了体现出盒模型的层次感。这里的 content 和 padding 用了

---

不痛的颜色（红色，白色），正常来说，我发单独设置 padding 的颜色。也就是说 padding 和 content 内容区的背景颜色是一致的。

盒模型，就像盒子一样。也是 box 直译过来的。这四部分合理的配合，能做出页面布局。每个元素都可以看成一个一个盒子，并且每个元素在页面中，给我们的直接感受，就是占据屏幕的大小。但是标准模式盒模型，它是很“贪心”的，这个要从它的占地面积说起。

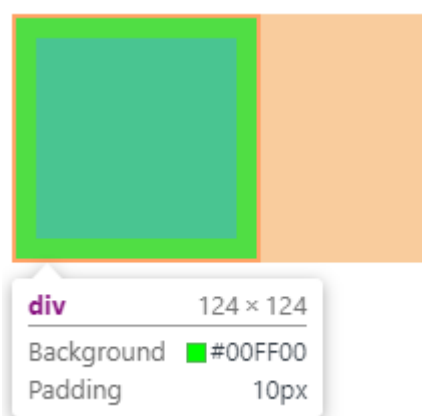
我们都知道标准模式的盒模型的宽高的计算公式

**元素宽度 = borderLeft + paddingLeft + width + paddingRight + borderRight。**

**元素高度=borderTop+paddingTop+height+paddingBottom+borderBottom**

这里说到的元素宽度和高度，指的是实际占据的宽高，而且在标准模式盒模型下，我们设置的 width 和 height 指的是 'content'（内容区）的宽高。并不是他的实际宽高。

Like this:



CSS:

---

```
div {  
    width: 100px;  
    height: 100px;  
    background: lime;  
    padding: 10px;  
    border: 2px solid #f20;  
}
```

根据上面的公式：

Div 的实际占地宽度为： $100 + 20 + 4 = 124\text{px}$ ；通过我们上面的截图，我们也能看出来是这样的。

对于这种贪婪特性（尽可能向外扩张）。对于布局来说，可能有些同学已经习惯了这种布局的方式，毕竟刚开始我们就是这么学过来的。随着代码的积累，发现一些问题：*我们先定制内容的大小，然后在修改 padding 和 border 导致的盒子的具体大小由这三部分决定。对于一个盒模型来说变数太大。*

大家想想开发商买地盖房子。先是圈地，然后在块土地里面进行操作，盖几幢房子。绿化面积比，太阳高度角多大等等等。这样的话，我们位置一旦分配好了。

就不用我们管了。你只能在内部改造。不可动用外部的区域。更容易限定，更容易布局。**土地交易局不管你，房屋多大（content），绿植面积（padding），栅栏宽度（border）怎么分配，你就告诉我，你要多大就完了。**

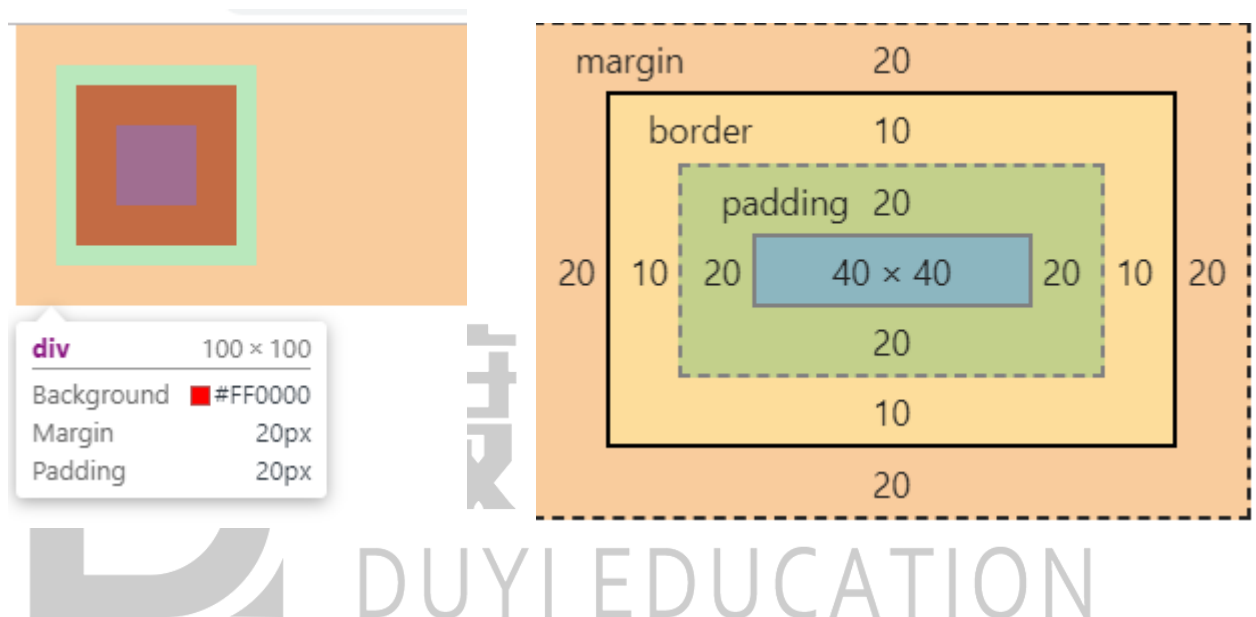
对于这种情况，就涉及到 IE 盒模型，也有一个尊称叫“怪异盒模型”。对于 IE 盒模型他还是很传统的，只在特定区域内去分配自己的空间。

举个栗子：

，

代码：

```
div {  
    box-sizing: border-box;  
    width: 100px;  
    height: 100px;  
    padding: 20px;  
    border: 10px;  
    margin: 20px;  
    background: red;  
}
```



我们看到的这个盒模型就是 IE 盒模型，给定宽度 100 边框 10 内边距 20 自己所剩下的宽度为：

$100 - 10 - 20 - 20 - 10 = 40;$

这里面我们看到除了外边框，以及内边距剩下的区域才是 content 的区域。

在这里面涉及到一个新的属性 `box-sizing`。用于设置盒模型的模式。

`Box-sizing` 存在两个属性值

`content-box` 默认值，也就是标准模式盒模型。

`Border-box`，设置为 IE 模式盒模型。

---

## border-box 的优势:

border-box 的诞生，主要就是解决 content-box 的最大缺点。border-box 意味着子容器的 padding 和 border 的厚度都算在 50%之内，这样，你可以随意的修改 padding 和 border 的厚度值，根本不用担心父容器被撑爆。

## border-box 是事实的标准?

现在最著名的 CSS 框架几乎都采用了 border-box，所以它才是事实的标准。框架想要具备栅格系统，就肯定要去使用 border-box，如果用 content-box 会有一种想死的感觉。

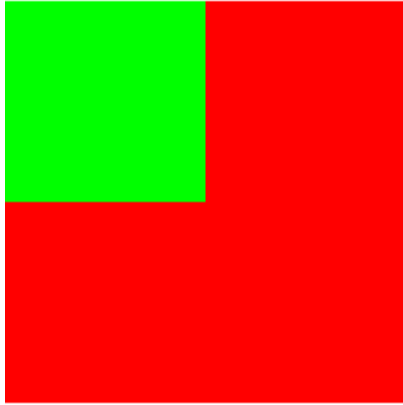
接下来说说盒模型存在两个经典 BUG——margin 塌陷和合并

### margin 塌陷:

父子嵌套元素在垂直方向的 margin,父子元素是结合在一起的,他们两个的 margin 会取其中最大的值.

正常情况下,父级元素应该相对浏览器进行定位,子级相对父级定位.

但由于 margin 的塌陷,父级相对浏览器定位.而子级没有相对父级定位,子级相对父级就像坍塌了一样.



margin 塌陷的解决办法

1.给父级设置边框或内边距(不建议使用)

2.触发 bfc(块级格式上下文),改变父级的渲染规则

方法:

改变父级的渲染规则有以下四种方法,给父级盒子添加

(1)`position:absolute/fixed`

(2)`display:inline-block;`

(3)`float:left/right`

(4)`overflow:hidden`

margin 合并

两个兄弟结构的元素在垂直方向上的 margin 是合并的。



---

在 margin 合并这个问题上,我们一般不用 bfc,而是通过只设置上面的元素的 margin-bottom 来解决距离的问题。

