

深入理解css盒子模型

一些基本概念

HTML的大多数元素都是块级(block)元素或行内(inline)元素

块级元素

默认情况下，块级元素会另起一行，并尽可能的充满整个容器。

块级元素可以包含行内元素和其他块级元素，相比于行内元素可以创建更复杂和大型的结构

块级元素列表：

html5新增的元素：

figcaption：图文信息组标题，article：文章，figure：图文信息组

output：表单输出，aside：侧栏内容，footer：区段尾或页尾，audio：音频播放

video：视频播放，section：页面区段，canvas：画布、绘制图形，header：区段头或页头

hgroup：标题组，

address：联系方式信息，ol：有序列表，p：行，form：表单，pre：预格式化文本，blockquote：块引用

h1-h6：标题，table：表格，dd：列表中条目描述，dl：定义列表，div，

hr：水平分割线

行内元素

行内元素不会另起一行只占据它对应的标签的边框所包含内容的空间，只能包含数据和其他行内元素

行内元素列表

b, big, i, small, tt,

abbr, acronym, cite, code, dfn, em, kbd, strong, samp, var,

a, bdo, br, img, map, object, q, script, span, sub, sup,

button, input, label, select, textarea

块级元素和行内元素的区别

块级元素：

- 1.会另起一行，
- 2.可以设置width, height, margin, padding, border属性
- 3.默认宽度是容器的100%

行内元素：

- 1.和其他元素在同一行内
- 2.高度和宽度就是内容的高度和宽度
- 3.可以设置margin-left和margin-right属性，无法设置margin-top和margin-bottom属性
- 4.border和padding可以设置，但是border-top和padding-top到页面顶部后就不再增加

正常流

正常流指：从左到右，从上到下显示。要让一个元素不在正常流中，唯一的办法是让元素浮动或定位

非替换元素

如果元素的内容包含在文档中，则称之为非替换元素。比如一个段落的文本都在该元素本身之内，这个段落就是一个非替换元素。

替换元素

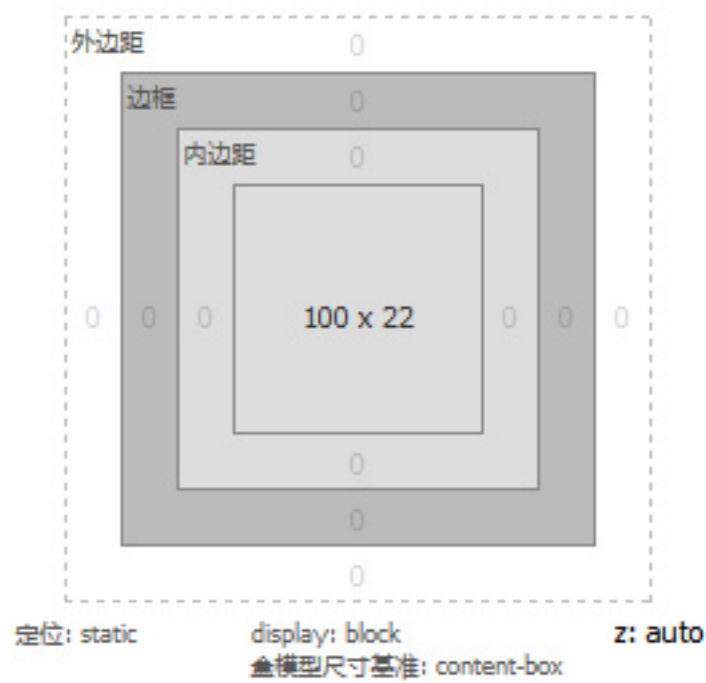
作为其他内容占位符的一个元素称为替换元素，根据标签和属性的值来显示内容的元素。比如img元素，它只是指向一个图像文件，这个文件插入到文档流中。大多数表单元素（input，根据type属性来显示内容）也是替换元素。

根元素

位于文档树的顶端，在html文档中就是html元素

盒模型

html文档中的每个元素都被描绘成矩形盒子，这些矩形盒子通过一个模型来描述其占用空间，这个模型称为盒模型。盒模型通过四个边界来描述：margin（外边距），border（边框），padding（内边距），content（内容区域），如图所示：



CSS盒模型

几点提示

1.padding，border，margin都是可选的，默认值为0，但是浏览器会自行设置元素的margin和padding，通过在css样式表中设置

1	<code>*{</code>
2	<code>margin:0;</code>
3	<code>padding:0</code>
4	<code>}</code>

来覆盖浏览器样式。注意：这里的*表示所有元素，但是这样性能不好，建议依次列出常用的元素来设置

2.如果给元素设置背景，并且边框的颜色为透明，背景将应用于内容，内边距和边框组成的区域。

3.浏览器兼容性

一旦为页面设置了恰当的 DTD，大多数浏览器都会按照上面的图示来呈现内容。然而 IE 5 和 6 的呈现却是不正确的。

根据 W3C 的规范，元素内容占据的空间是由 width 属性设置的，而内容周围的 padding 和 border 值是另外计算的。

不幸的是，IE5.X 和 6 在怪异模式中使用自己的非标准模型。这些浏览器的 width 属性不是内容的宽度，而是内容、内边距和边框的宽度的总和。

水平格式化

非替换元素的水平格式化

水平格式化的7大属性是：margin-left, border-left, padding-left, width, padding-right, border-right, margin-right。这7个属性值加起来往往是父级元素的width值。

其中margin-left, width, margin-right可以设置为auto。

主要有下面几种情况：

一个属性设置成auto

如果三个属性中某个属性设置了auto，其余两个为特定的值，那么设置auto的属性为确定所需的元素，从而使得元素框的宽度等于父级元素的width。

例子

HTML代码

```
<div class="parent">
  <span class="block">块级元素</span>
</div>
```

CSS代码

```
.parent{
  width:600px
}
```

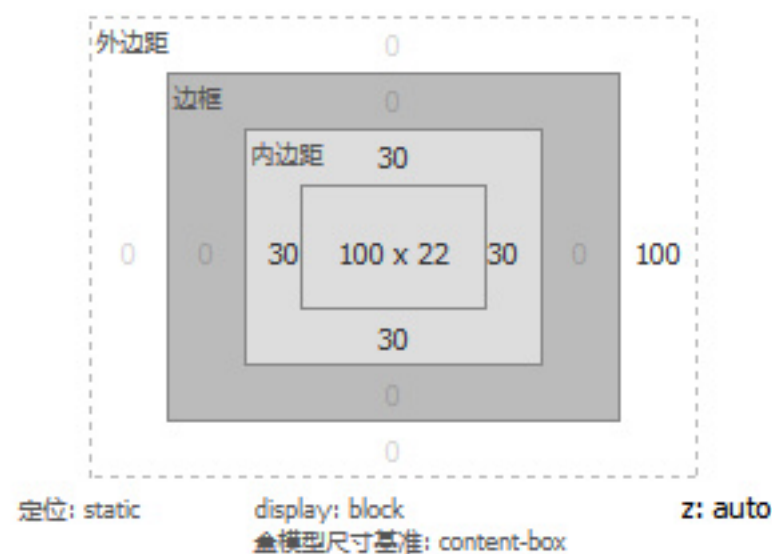
```
div {
```

```

background: #eeb3b3 none repeat scroll 0 0;
}

.block {
background: #ffd800 none repeat scroll 0 0;
display: block;
margin-left: auto;
margin-right: 100px;
padding: 30px;
width: 100px;
}

```



被设置为auto的margin-auto属性值为340px，即margin-left+border-left+padding-left+width+padding-right+border-right+margin-right=340+0+30+100+30+0+100=600

总和不等于父级元素的width

使用auto可以弥补实际值与所需总和的差距，如果三个属性都设置了特定值，但是总和不等于父级元素的width。

修改上面例子中的margin-left为100px，即

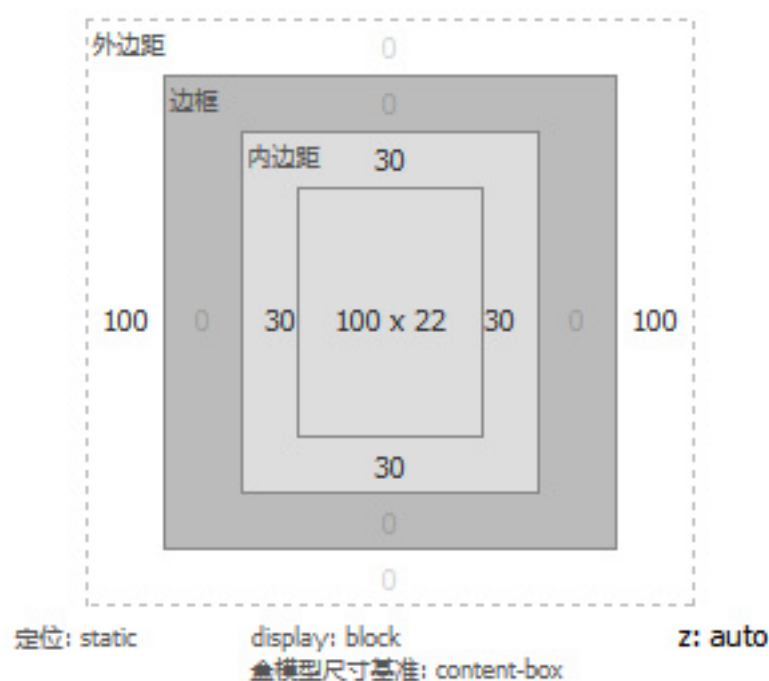
```

.block {
background: #ffd800 none repeat scroll 0 0;
display: block;
margin-left: 100px;
}

```

```
margin-right: 100px;
padding: 30px;
width: 100px;

}
```



在上面的CSS中，margin-left，width，margin-right都设置了特定值，但是七大属性总和并不等于父级元素的width。这种情况下：
 在FF中，margin-right的值为开发者设定的值
 在Chrome中，margin-right被强制为auto

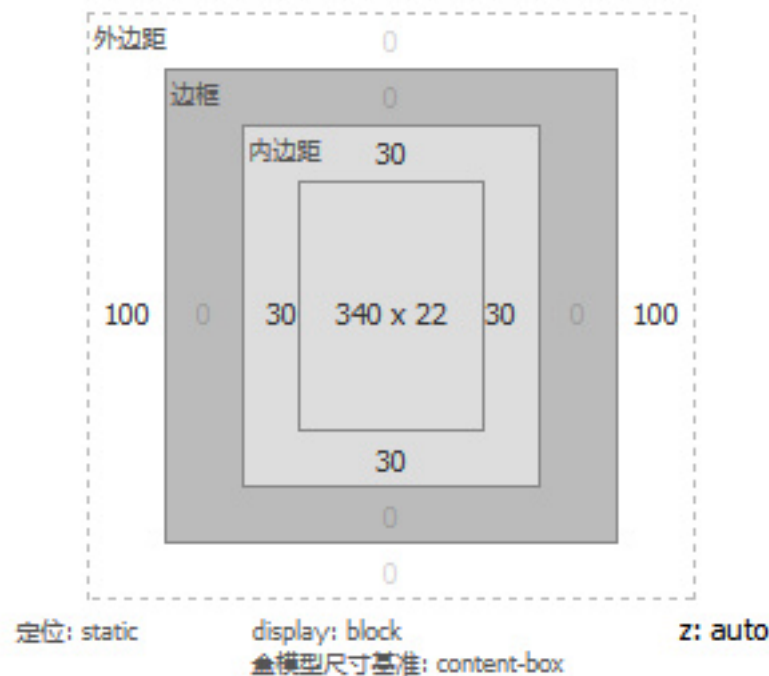
width设置为auto

如果margin-left和margin-right都设置特定值，width设置为auto，则width将会等于某个特定值以达到父级元素的width。
 如果将width修改为auto，即：

```
.block {
  background: #ffd800 none repeat scroll 0 0;
  display: block;
  margin-left: 100px;
  margin-right: 100px;
```

```
padding: 30px;
width: auto;

}
```

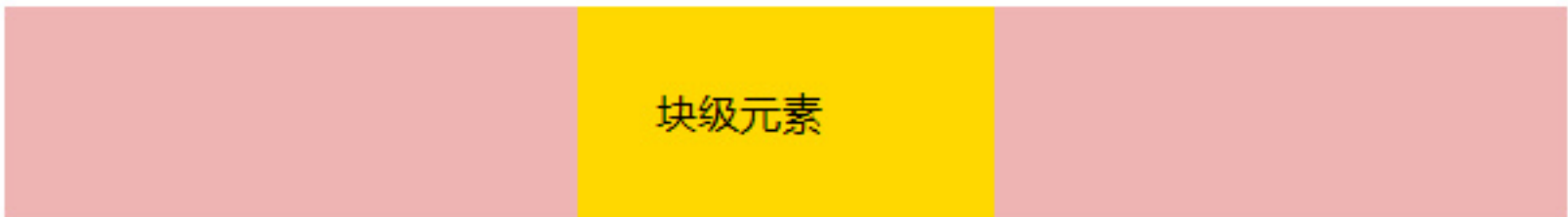


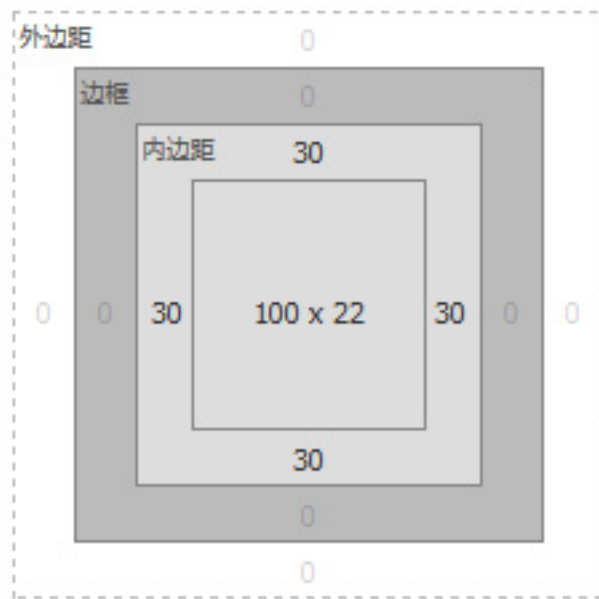
元素的width将被被设定为340px来使总和达到父级元素的width

margin-left和margin-right设置成auto

如果margin-left和margin-right都设置为auto，则它们会设置相等的值，因此元素将在父级元素中居中。这是将块级元素居中的一种方法。注意：text-align设置为center只适用于块级元素中的内联内容居中，并不能使块级元素居中。

设置margin属性为margin:0 auto





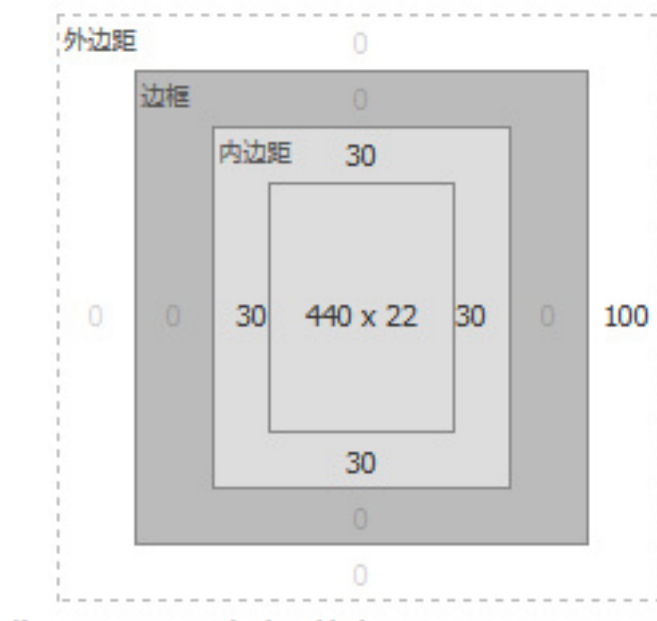
margin-left和margin-right的值会被设置为相等，使得元素居中

某个外边距和width设置成auto

如果设置某个外边距和width为auto，则设置为auto的外边距会为0，width会设置为所需的值来填充父级元素。

```
.block {  
  background: #ffd800 none repeat scroll 0 0;  
  display: block;  
  margin-left: auto;  
  margin-right: 100px;  
  padding: 30px;  
  width: auto;  
}
```

块级元素



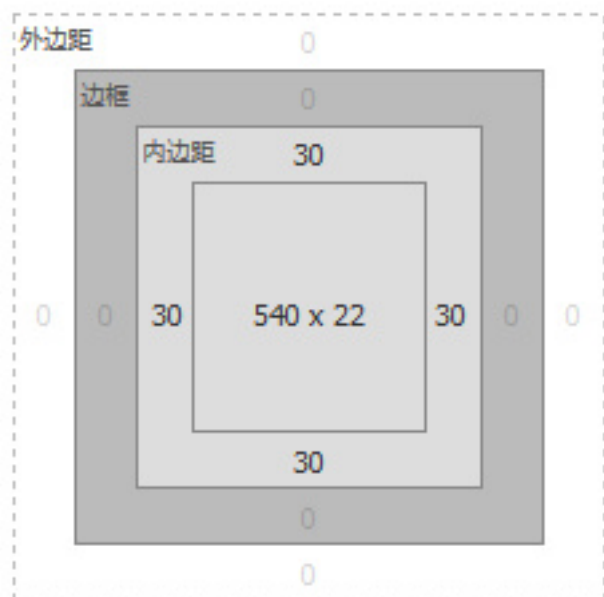
设置margin-left和width为auto，则margin-left将被设置为0，width会被设置为440px来满足父级元素的width

全设置成auto

如果margin和width都设置为auto，则两个外边距会设置为0，width会尽可能宽。

```
.block {  
  background: #ffd800 none repeat scroll 0 0;  
  display: block;  
  margin-left: auto;  
  margin-right: auto;  
  padding: 30px;  
  width: auto;  
}
```

块级元素



三个值都设置为auto，则两个外边距会设置为0，width会被设置为540px

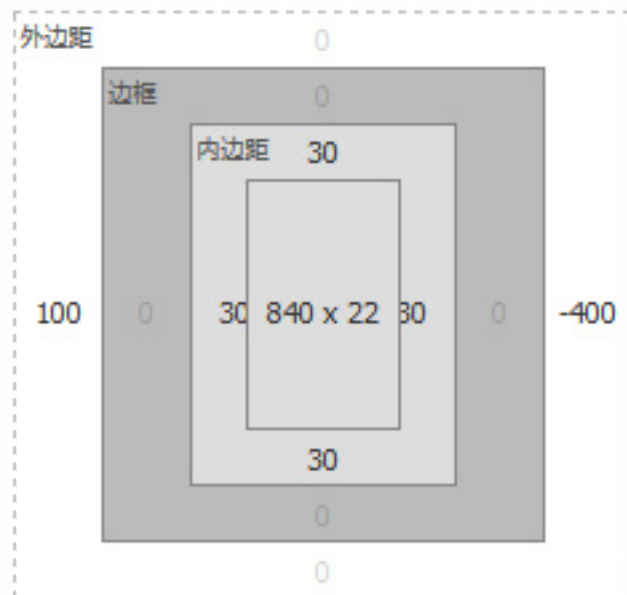
负外边距

7个属性只要都是大于等于0的值，总和总是等于父级元素的width，不会超过父级元素的区域

但是可以通过制定负外边距来得到比父级元素width更大的区域

```
.block {  
    background: #ffd800 none repeat scroll 0 0;  
    display: block;  
    margin-left: 100px;  
    margin-right: -400px;  
    padding: 30px;  
    width: auto;  
}
```

块级元素



设置margin-right为-400px，则元素会大于父级元素width，因为 $100+0+30+840+30+0-400=600$ ，元素的width为840px

替换元素的水平格式化

替换元素的水平格式化规则和非替换元素的规则类似，只有一个width有区别，如果width设置为auto，则元素的宽度是内容的固有宽度。注意：对于img标签，如果width不等于其固有宽度，则height也会等比例增加，除非设置特定值。反过来如果height设置高度，width也会等比例增加

垂直格式化

垂直格式化和水平格式化类似，也有7个相关属性：margin-top，border-top，padding-top，height，padding-bottom，border-bottom，margin-bottom，这7个属性的总和必须等于父级元素的height属性。

其中margin-top，margin-bottom和height可以设置成auto

一个正常流中的块元素的margin-top和margin-bottom设置为auto后，会被设置为0，即不能将元素垂直居中，实际上元素没有外边距。定位元素如果设置成auto有不同的处理结果。

如果正常流元素的height设置为auto，则其高度将会被设置为其内容元素的高度总和。

垂直外边距合并

垂直外边距合并：当两个垂直外边距相遇时，它们将形成一个外边距。合并后的外边距的高度等于两个发生合并的外边距的高度中的较大者。

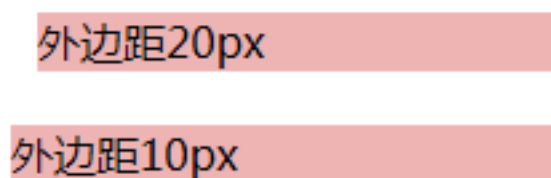
例子：

html代码

```
<div class="m20"> 外边距20px </div>  
<div class="m10"> 外边距10px </div>
```

css代码

```
.m20 { padding: 20px; }  
.m10 { padding: 10px; }
```



如图所示，两个div标签的外边距分别是20px，10px，但是最终两个div之间的距离是20px，而不是20+10=30px

行内元素的盒模型

行内元素也是有盒模型的，但是有几点要注意：

1.对于非替换元素，比如a，span标签等

（1）可以设置margin-left和margin-right属性，无法设置margin-top和margin-bottom属性

（2）行内元素border和padding可以设置，但是border-top和padding-top到页面顶部后就不再增加

2.对于替换元素，比如input，img标签

margin，padding，border都有效果