

【前端面试】offsetHeight、scrollHeight、...

前言

offsetHeight、scrollHeight、clientHeight这三个属性我们经常在开发中遇到，如果小伙伴们没有经常使用的话，很容易把这些属性搞混，比如说什么窗口高度、元素高度、内容高度等等。当然，现在的前端框架很多时候帮我们封装了这些属性，但是我们也不能太过依赖框架，底层的原理我们还是需要了解的，今天就来理一理这三个属性分别代表什么？

1.盒子模型

介绍这三个属性之前先来了解一下CSS盒子模型，因为面试中如果遇到这道题，面试官通常也是想要考察你盒子模型的相关知识。

盒子模型顾名思义就是盒子的意思，盒子里面可以装很多东西。比如我们的div就可以比作一个盒子，那么组成一个完整的盒子主要包括下面几个部分：

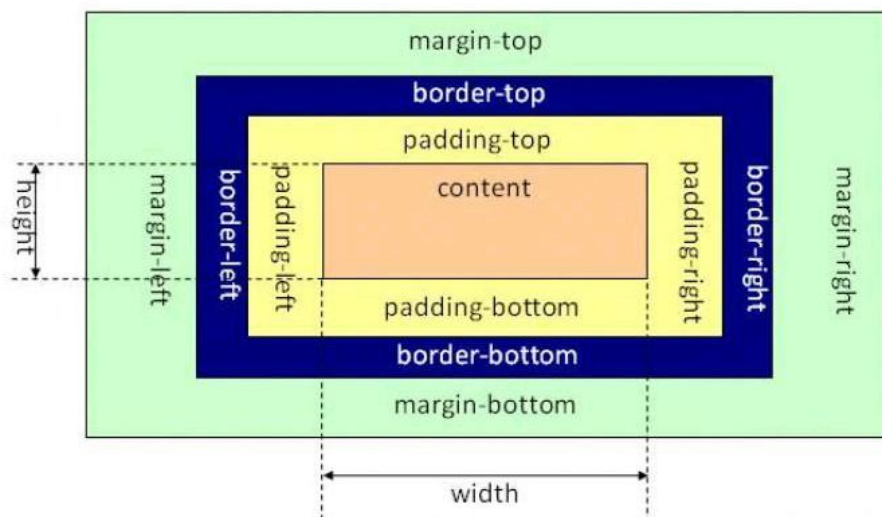
- width
- height
- padding
- border
- margin
- box-sizing（用来区别正常盒模型和怪异盒子模型）

一个盒子大概由上面一些属性组成，盒子模型又分为了正常和模型和怪异和模型。

1.1 正常盒模型

我们借用一张图就可以很清楚的知道什么是正常盒模型：

■ 标准盒子模型



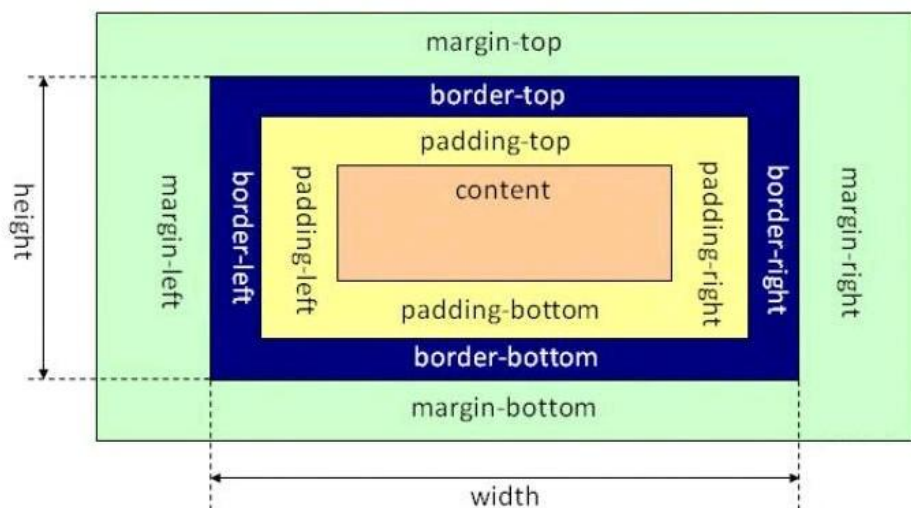
从上图可以看到标准 W3C 盒子模型的范围包括 margin、border、padding、content，并且 content 部分不包含其他部分

通过上图可以看出：宽度就是内容的宽度。当我们修改padding或者border属性时，盒子的总宽度会改变。

1.2 怪异盒子模型

同样，我们借助一张图来理解怪异盒子模型：

■ IE 盒子模型



从上图可以看到 IE 盒子模型的范围也包括 margin、border、padding、content，和标准 W3C 盒子模型不同的是：IE 盒子模型的 content 部分包含了 border 和 padding

通过上图可以看出：盒子的宽度 = 内容宽度 + padding + border。

简单了解了CSS中的盒子模型后，我们再了解这三个高度属性。

2.offsetHeight

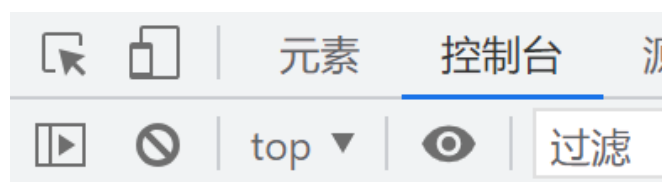
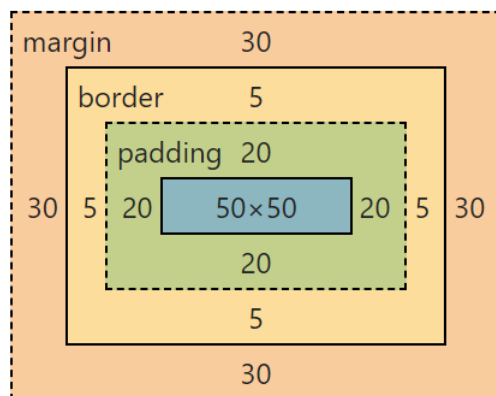
简单总结为一句话：获取元素的高度，包含padding和border。

需要注意的是，如果我们的盒子是正常盒子，那么高度只有内容高度，所以通常我们需要将盒子模型改为怪异盒子模型，使用box-sizing属性。

代码如下：

```
1 <head>
2   <style>
3     .box1 {
4       width: 100px;
5       height: 100px;
6       padding: 20px;
7       margin: 30px;
8       border: 5px solid yellow;
9       background-color: #ccc;
10    }
11  </style>
12 </head>
13 <body>
14   <div class="box1">盒子1</div>
15 </body>
16 <script>
17   const box1 = document.getElementsByClassName("box1")[0];
18   console.info("盒子1的offsetHeight",box1.offsetHeight)
19 </script>
```

打印结果：



结果解释：

最终offsetHeight = 150px，我们说过offsetHeight是获取元素的高度，上面代码中我们元素的高度设置为100px，但是offsetHeight却是150px，这就是因为正常盒子模型的宽度等于内容宽度的原因，实际上box1占据的宽度要更宽。

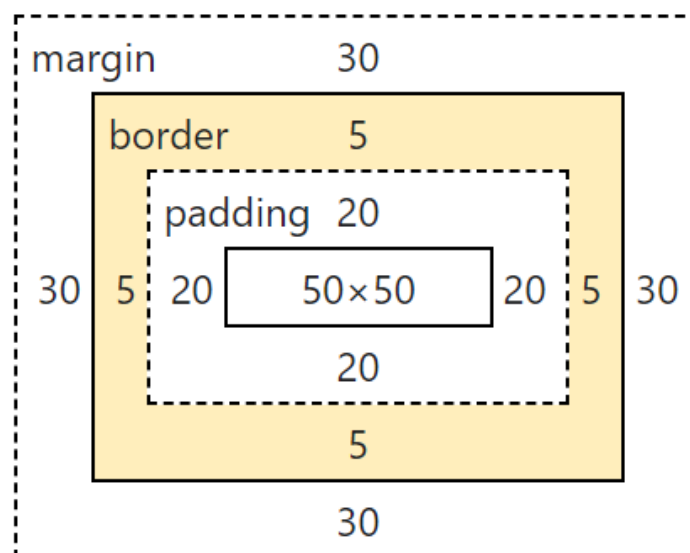
实际上 $\text{offsetHeight} = 100 + 20(\text{padding}) + 20(\text{padding}) + 5(\text{border}) + 5(\text{border})$ 。

在实际项目中，为了获得更准确，或者最真实的元素宽度，我们需要把正常盒子模型转为怪异盒子模型，添加属性`box-sizing: border-box`。

代码如下：

```
1 <style>
2   .box1 {
3     width: 100px;
4     height: 100px;
5     padding: 20px;
6     margin: 30px;
7     border: 5px solid yellow;
8     box-sizing: border-box;
9     background-color: #ccc;
10  }
11 </style>
```

输出结果：



总结： $\text{offsetHeight} = \text{content} + \text{padding} + \text{border}$

3.clientHeight

简单总结为一句话：获取元素的高度，包含padding。

这个属性和offsetHeight类似，唯一的区别就是它不包含border，具体来看代码演示。

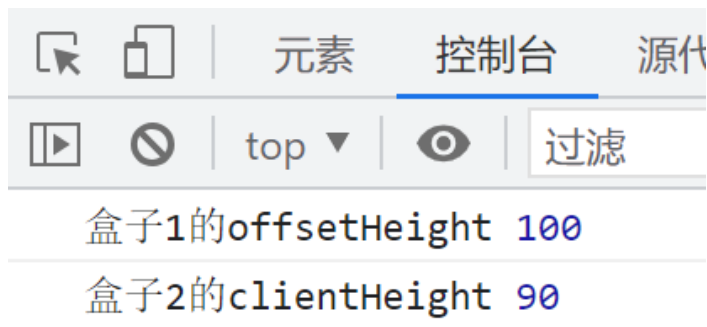
代码如下：

```

1 <style>
2 .box2 {
3   width: 100px;
4   height: 100px;
5   padding: 20px;
6   margin: 30px;
7   border: 5px solid yellow;
8   box-sizing: border-box;
9   background-color: #ccc;
10 }
11 </style>
12 <body>
13   <div class="box1">盒子1</div>
14   <div class="box2">盒子2</div>
15 </body>
16 <script>
17   const box1 = document.getElementsByClassName("box1")[0];
18   const box2 = document.getElementsByClassName("box2")[0];
19   console.info("盒子1的offsetHeight", box1.offsetHeight);
20   console.info("盒子2的clientHeight", box2.clientHeight);
21 </script>

```

输出结果:



总结: $clientHeight = content + padding$

4.scrollHeight

简单总结为一句话: 获取元素的高度, 包含padding。

这个属性和clientHeight类似, 都不包含border, 那么具体有什么不一样呢, 我们看代码演示。

代码如下:

```

1 <style>
2 .box3 {
3   width: 100px;
4   height: 100px;
5   padding: 20px;
6   margin: 30px;
7   border: 5px solid yellow;
8   box-sizing: border-box;

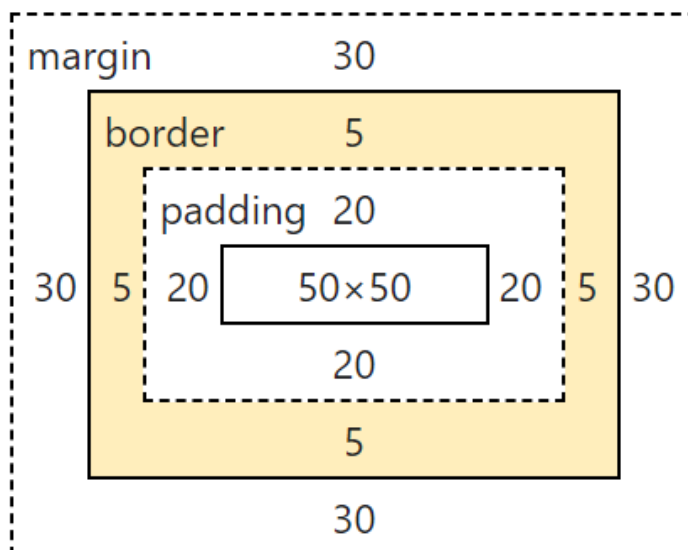
```

```

9     background-color: #ccc;
10  }
11  </style>
12  <body>
13    <div class="box1">盒子1</div>
14    <div class="box2">盒子2</div>
15    <div class="box3">盒子3</div>
16  </body>
17  <script>
18    const box1 = document.getElementsByClassName("box1")[0];
19    const box2 = document.getElementsByClassName("box2")[0];
20    const box3 = document.getElementsByClassName("box3")[0];
21    console.info("盒子1的offsetHeight", box1.offsetHeight);
22    console.info("盒子2的clientHeight", box2.clientHeight);
23    console.info("盒子3的scrollHeight", box3.scrollHeight);
24  </script>

```

输出结果:



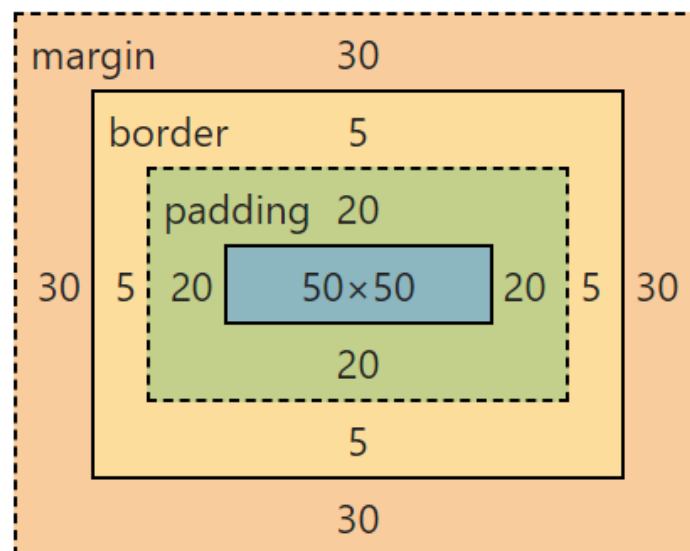
可以看到scrollHeight和clientHeight输出结果一样，那么它们之间有什么区别呢？

其实它们的区别就一个：scrollHeight的高度需要更具内容的实际尺寸决定，比如我们修改一下我们的代码。

代码如下：

```
1 .box3 {  
2   width: 100px;  
3   height: 100px;  
4   padding: 20px;  
5   margin: 30px;  
6   border: 5px solid yellow;  
7   box-sizing: border-box;  
8   background-color: #ccc;  
9   overflow: auto;  
10 }  
11 <div class="box3">  
12   <div style="height: 300px;">盒子3</div>  
13 </div>
```

输出结果：



上段代码中我们在box3中添加了一个div，并且设置高度300px，此时的scrollHeight输出340px，也就代表我们的scrollHeight是需要根据实际的内容尺寸计算的。

总结：scrollHeight = 实际内容尺寸+ padding

总结

这三个属性虽然都很类似，但是每一个又有一些不一样的地方，总结起来就下面三点：

- $\text{offsetHeight} = \text{内容高度} + \text{padding} + \text{border}$
- $\text{clientHeight} = \text{内容高度} + \text{padding}$
- $\text{scrollHeight} = \text{内容实际尺寸} + \text{padding}$