

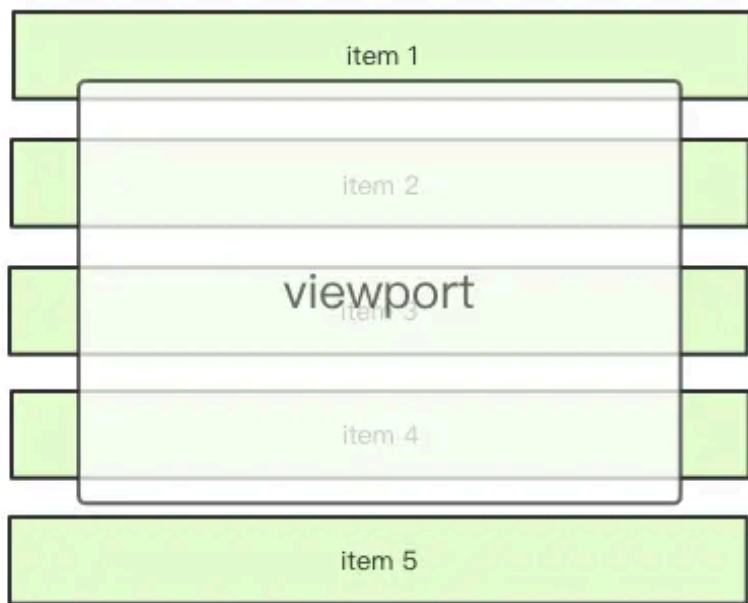
虚拟列表的实现

分享人：白涵

什么是虚拟列表

什么是虚拟列表

虚拟列表本质就是使用少量的 DOM 节点来模拟一个长列表



为什么使用虚拟列表

解决当列表数量较多时，页面内引入大量的 DOM 元素，从而导致页面卡顿的情况。

常用使用场景：table和select等

```
document.getElementById('button').addEventListener('click', function(){
  // 记录任务开始时间
  let now = Date.now();
  // 插入一万条数据
  const total = 10000;
  // 获取容器
  let ul = document.getElementById('container');
  // 将数据插入容器中
  for (let i = 0; i < total; i++) {
    let li = document.createElement('li');
    li.innerText = ~~(Math.random() * total)
    ul.appendChild(li);
  }
  console.log('JS运行时间: ', Date.now() - now);
  setTimeout(()=>{
    console.log('总运行时间: ', Date.now() - now);
  }, 0)

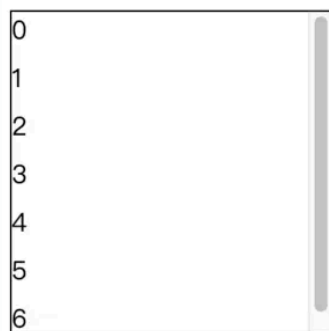
  // print JS运行时间: 38
  // print 总运行时间: 957
})
```


定高虚拟列表的实现

创建适合容器高度的 DOM 元素

假设有一个拥有1000元素的列表

- 真实 DOM 数量(actualRows) = $\text{Math.ceil}(\text{容器高度} / \text{条目高度})$
- 获取出现在视窗内的数据切片 `data.slice(0, actualRows)`

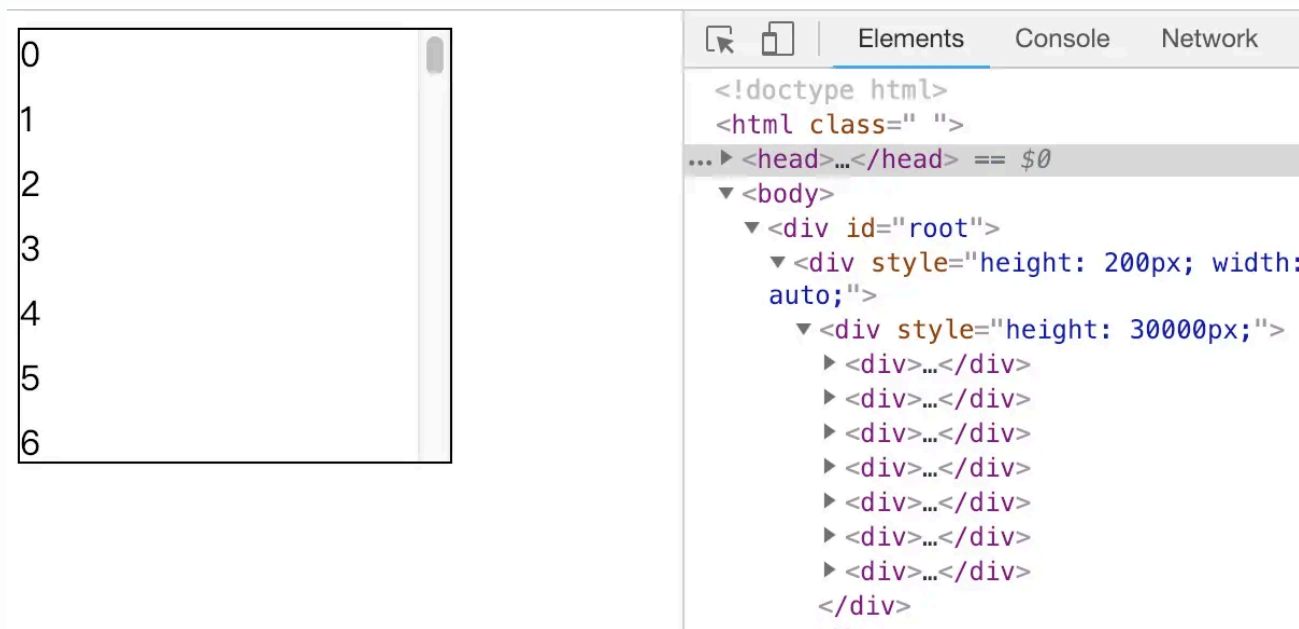


```
Elements Console Network Sources Application Performance >> X
<!doctype html>
<html class=" ">
  <head>...</head>
  ... <body> == $0
    <div id="root">
      <div style="height: 200px; width: 200px; border: 1px solid black; overflow: auto;">
        <div>...</div>
        <div>...</div>
        <div>...</div>
        <div>...</div>
        <div>...</div>
        <div>...</div>
        <div>...</div>
      </div>
    </div>
  </body>
</html>
```

现在存在一个问题，容器的滚动条明显不符合 1000 条数据该有的高度，因为我们只有 7 条真实 DOM，没有办法将容器撑开。

撑开容器

- 真实容器高度 = 每条 item 的高度 * 数据总数量



但是滚动之后发现，下面全是空白的，由于列表只存在7个元素，空白是正常的。而我们期望随着滚动，元素能正确的出现在视野中。

滚动列表

列表滚动到某一位置的时候，去计算出当前的视窗中列表的索引，有了索引就能得到当前时刻的数据切片，从而将数据渲染到视图中。

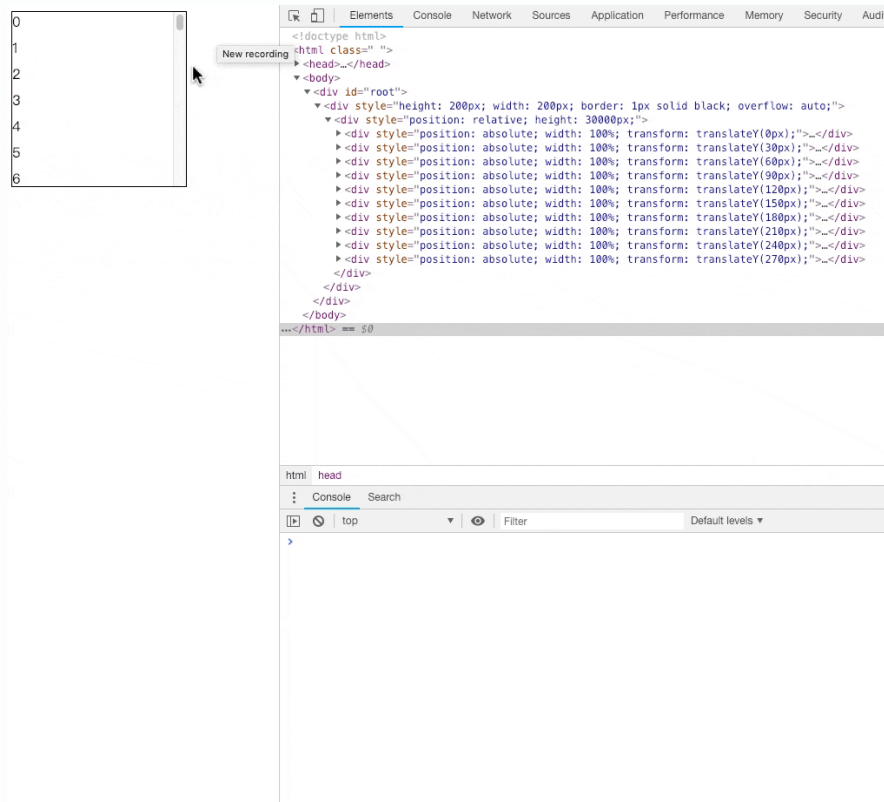
- $\text{firstIndex} = \text{Math.floor}(\text{scrollTop} / \text{height})$
- $\text{lastIndex} = \text{firstIndex} + \text{actualRows} - 1$

虽然我们随着滚动的发生计算出了正确的数据切片，但是正确的数据却没有出现在正确的位置，因为他们的位置是固定不变的

- $\text{transform: translateY}(\text{currentIndex} * \text{height} + \text{px})$

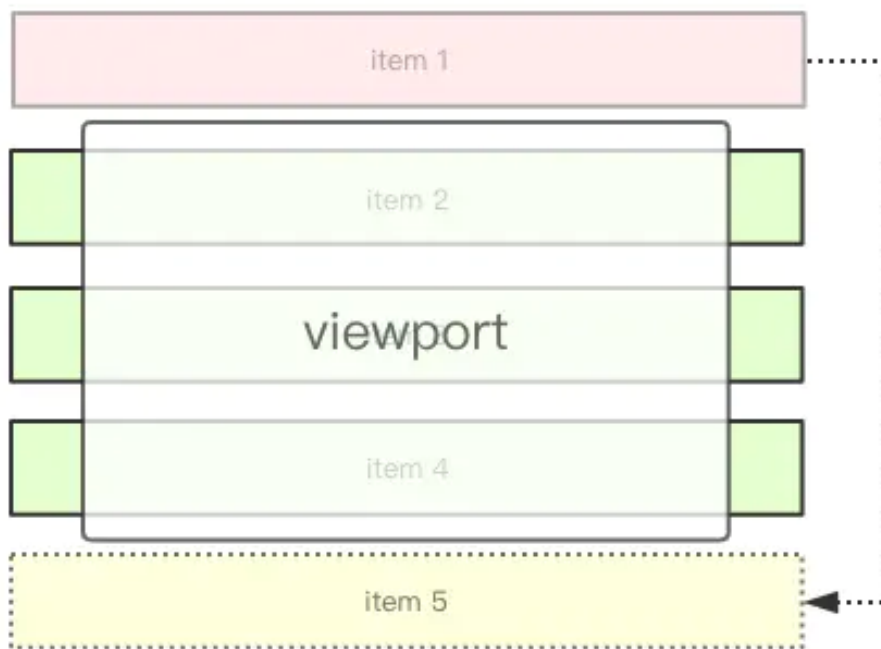
问题：

1. 计算浪费
2. DOM 节点的创建和移除



复用 DOM + 按需更新列表

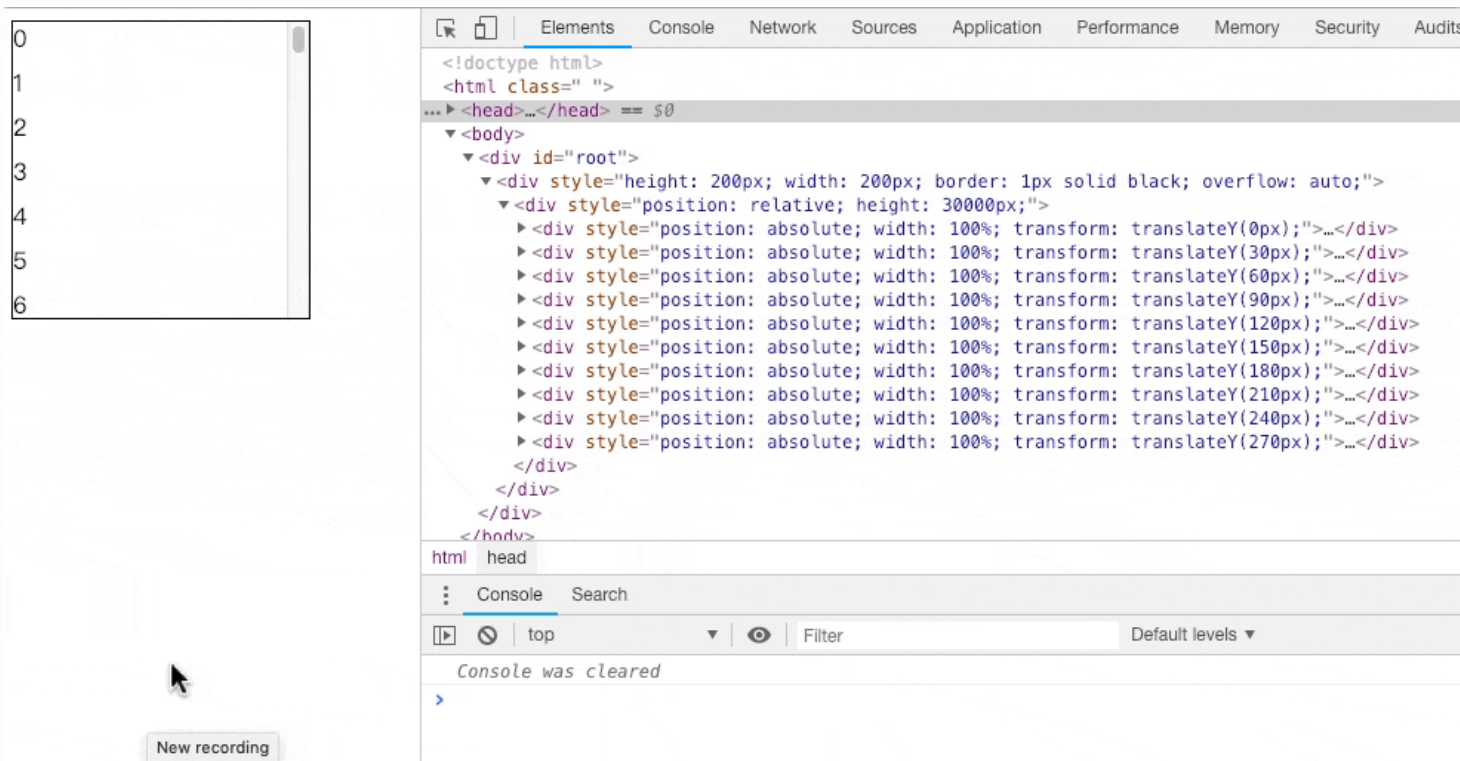
为了能让列表只按照需要进行更新，而不是全部重刷，我们就需要知道有哪些 DOM 节点被移出了视野范围，然后操作这些视野范围外的节点来补充列表，从而完成列表的按需更新



索引差异

比如滚动前的索引为 [0,1,2,3]，滚动后的索引为 [1,2,3,4]，那么他们的差集就是 [0]，说明老数组中的第一个元素被移出了视野，那么就需要用这第一个元素来补充到列表最后，成为最后一个元素

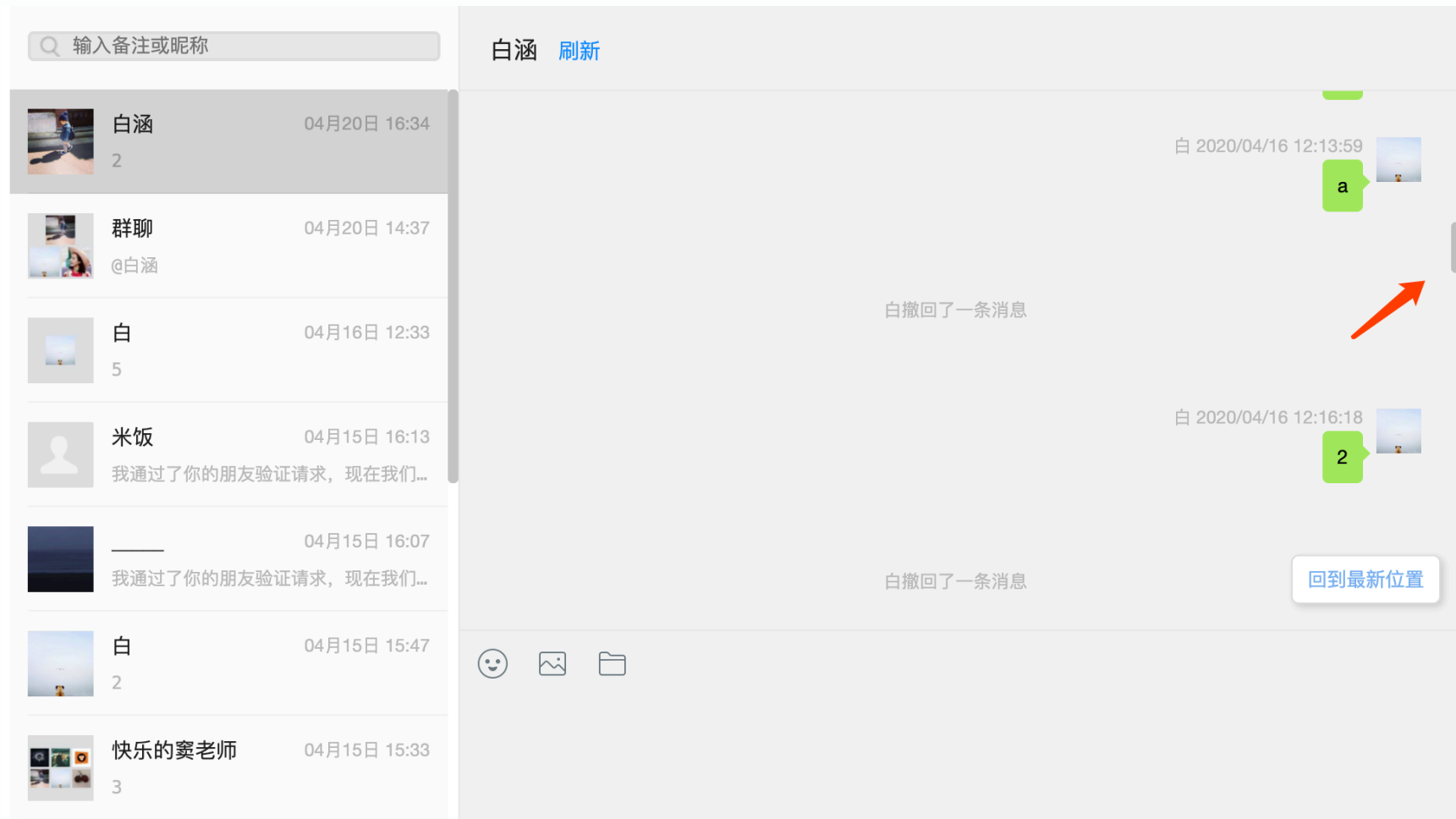
oldSlice [0,1,2,3] -> newSlice [1,2,3,4]



不定高虚拟列表

哆啦的实现

通过IntersectionObserver来实现懒加载





动态高度的虚拟列表

解决办法

- 将列表项渲染到屏幕外，对高度测量并缓存后再渲染到可视区
- 预估高度

图片异步加载撑开列表

- 使用ResizeObserver来监听列表项内容区域的高度改变，从而实时获取每一列表项的高度

Thanks