1. DOM树加CSS结构体最终会生成Render树，Render树相较于DOM树是不包括display:none的节点，但包括visibility:hidden节点。
2. 引起回流的操作
3. 添加或者删除DOM节点。
4. 元素的几何属性发生变化，width，height等等发生变化
5. 页面第一次加载。
6. 浏览器窗口发生变化—resize事件发生时。
7. 查询DOM元素的OffsetLeft,OffsetTop,OffsetHeight,OffsetWidth  
   scrollLeft,scrollTop,scrollHeight,scrollWidth  
   clientLeft,clientTop,clientHeight,clientWidth  
   getComputedStyle（）
8. 回流的代价很高，浏览器不可能每次你有回流操作就马上执行，它很聪明，浏览器会维持一个队列，等队列中的操作到了一定的数量或者到了一定间隔，浏览器会做一个批处理。但是有些操作会提前flush队列，比如上述操作
9. 我们可以做到：
10. 如果是操作CSS，可以使用className，cssText等等。
11. 如果业务不需要实时的元素高宽位置信息，那么尽量少访问flush的操作，可以使用变量缓存下来
12. 对于可能会多次回流重绘的元素，让其脱离文档流，不要让它再影响其他的元素了。例如，有动画的元素最好使用绝对定位。
13. 尽量不要使用表格布局，因为如果表格没有定下宽度，那么表格的每一列的宽度将由最宽的一列决定，那么很可能最后一行的宽度超过前面的列宽，将会引起整体的回流，造成table可能需要多次计算才能确定好其在渲染树中节点的属性。
14. 操作DOM树中的节点时，让要操作的元素进行离线操作，处理完后再一起处理。  
    使用document.createDocumentFragment()进行缓存操作，只引起一次回流与重绘。使用cloneNode，replaceChild等技术，引起一次回流与重绘。