1. **什么是防抖和节流？有什么区别？如何实现？**

防抖和节流严格算起来应该属于性能优化知识，但实际上遇到的频率相当高，处理不当或者放任不管就容易引起浏览器卡死。

防抖：对于短时间内连续触发的事件，防抖的含义就是让某个时间期限（setTimeOut）内，事件处理函数只执行一次。

节流：可以设计一种类似控制阀门一样定期开放的函数，也就是让函数执行一次，在某个时间段内暂时失效，过了这段时间后在重新激活（类似于技能冷却时间）。

效果：如果短时间内大量触发同一事件，那么函数在执行一次之后，该函数在指定的时间期限内不再工作，直到过了这段时间才重新生效。

例子：

1. 搜索框input事件，例如要支持输入实时搜索可以使用节流方案（间隔一段时间就必须查询相关内容），或者实现输入间隔大于某个值（如500ms），就当做用户输入完成，然后开始搜索，具体使用哪种方案要看业务需求。
2. 页面resize事件，常见于需要做页面适配的时候。需要根据最终呈现的页面情况进行dom渲染（这种情形一般是使用防抖，因为只需要判断最后一次的变化情况）
3. **Promise 构造函数是同步执行还是异步执行，那么 then 方法呢？  
   React 中 setState 什么时候是同步的，什么时候是异步的？并举例**

promise本身是同步的。then方法和catch方法是异步的。

setState只在合成事件和钩子函数中是“异步的”，在原生事件和setTimeout中都是同步的。

合成事件：就是react在组件中的onClick等都是属于它自定义的合成事件。

原生事件：比如通过addeventListener添加的，dom中的原生事件。

1. **介绍下重绘和回流（Repaint & Reflow），以及如何进行优化**

什么是重绘：

当render tree中的一些元素需要更新属性，而这些属性只是影响元素的外观，风格，而不会影响布局的，比如background-color。则就叫称为重绘。

什么是回流：

当render tree中的一部分（或全部）因为元素的规模尺寸，布局，隐藏等改变而需要重新构建。这就称为回流（reflow）。每个页面至少需要一次回流，就是在页面第一次加载的时候，这时候是一定会发生回流的，因为要构建render tree。在回流的时候，浏览器会使渲染树中受到印象的部分失效，并重新构造这部分渲染树，完成回流后，浏览器会重新绘制受影响的部分到屏幕中，该过程称为重绘。

区别：

回流必将引起重绘，而重绘不一定引起回流。比如：只有颜色改变的时候只会发生重绘而不会引起回流

当页面布局和几何属性改变时就需要回流

比如：添加或者删除可见的DOM元素，元素位置改变，元素尺寸改变-边距、填充、边框、宽度和高度，内容改变。

**4、随机数抽取不重复的数组元素，例如1000人中抽100人中奖，如何防止极端的随机数重复问题？**