

requestIdleCallback 和 requestAnimationFrame 区别

都是浏览器提供的API，用于优化页面的渲染和性能，但它们在使用场景和执行机制上存在显著的区别。

1. 执行时机和频率：

- requestAnimationFrame (rAF) 是一个高级任务，它的回调函数会在每一帧都会被浏览器调用，无论这一帧是否有空闲时间。这意味着，如果你的动画需要更新，rAF会在下一次重绘之前调用你提供的回调函数，从而确保动画的流畅性。回调函数执行的次数通常是每秒60次，但这取决于浏览器的屏幕刷新次数。
- requestIdleCallback (rIC) 则是一个低级任务，它的回调函数只会在浏览器的一帧渲染完成后，且存在空闲时间时才会被调用。因此，rIC的执行频率是不确定的，它取决于浏览器在每一帧中的工作负载。

2. 使用场景：

- requestAnimationFrame主要用于实现更流畅和性能更好的动画。如果你需要在每一帧中更新动画的状态，那么rAF是一个理想的选择。
- requestIdleCallback则主要用于在浏览器的渲染空闲时间执行优先级不高的操作，以避免阻塞渲染。这些操作可能包括一些后台任务、非关键的数据获取或处理等。

3. DOM操作：

- 由于rIC发生在一帧的最后，此时页面布局已经完成，因此不建议在rIC中操作DOM，因为这可能会导致页面再次重绘。相反，DOM操作建议在rAF中进行，因为此时浏览器正准备进行下一帧的渲染，对DOM的修改可以立即反映出来。

4. 优先级和可预测性：

- requestIdleCallback的回调函数执行的优先级较低，且其执行时间取决于浏览器的空闲时间，因此不具备可预测性。
- 相比之下，requestAnimationFrame的回调函数在每一帧都会被调用，其执行时间相对更可预测。

总的来说，requestAnimationFrame和requestIdleCallback各有其适用的场景。选择使用哪一个，取决于你的具体需求以及希望优化的方面。

他们是宏任务还是微任务

他们是宏任务，因为每次都要等到 DOM 渲染完成才继续执行。

```
window.onload = () => {
  console.info("start");
  setTimeout(() => {
    console.info("setTimeout");
  });
  requestAnimationFrame(() => {
    console.info("requestAnimationFrame");
  });
  requestIdleCallback(() => {
    console.info("requestIdleCallback");
  });
}
```

```
});  
console.info("end");  
}
```

requestIdleCallback 和 requestAnimationFrame 返回