## 每日一题 - Vue.nextTick()有什么作用？实现原理是什么？

### 信息卡片

* 时间：2019-11-12
* tag：VUE nextTick

### 题目描述

Vue.nextTick()有什么作用？实现原理是什么？

### 参考答案

#### 什么是 nextTick

在 vue 里面，数据更新后，我们是无法立马拿到响应数据更新后的 dom 的，nextTick 就是 vue 的 api，如果想要操作 dom，可以通过这个 api 获取

以下是[vue 官方文档](https://cn.vuejs.org/v2/api/#Vue-nextTick)

在下次 DOM 更新循环结束之后执行延迟回调。在修改数据之后立即使用这个方法，获取更新后的 DOM。

#### 为什么需要用到 nextTick

浏览器里面频繁操作 dom 会非常影响性能，某个 dom 的宽高变化可能都会引起整个页面的重排重绘

vue.nextTick 就是上述背景的一个产物，我们在更改响应式数据的时候，vue 不会检测到一次数据更新，就去更新一下 dom，它会先有一个 diff 的计算，如果你在真实dom修改之前就想用到真实dom的数据，那么可能就需要用到这个API了。

#### nextTick 实现原理

上述下一个Tick其实就是下一次事件循环，所以只要是可以放到下一次事件循环的函数都可以用来实现 nextTick，vue 里面采取了以下策略去实现 nextTick

1. 如果浏览器支持 promise，就用 promise.resolve(fn) 来实现
2. 否则，如果浏览器支持 MutationObserver，用 MutationObserver
3. 如果浏览器支持 setImmediate，用 setImmediate
4. 如果上述都不支持，用 setTimeout 兜底

Vue 2.0 的源码见这 [nextTick 源码](https://github.com/vuejs/vue/blob/dev/src/core/util/next-tick.js)

自己实现一个不用 mutionObserver 的 nextTick

const nextTick = cb => {  
 const queueMicrotask =  
 typeof queueMicrotask === "function"  
 ? queueMicrotask  
 : Promise && (cb => Promise.resolve().then(cb))  
 ? setImmediate && (cb => setImmediate(cb))  
 : cb => setTimout(cb);  
  
 queueMicrotask(cb);  
};

*ps: 看了下 vue3 的* [*nextTick*](https://github.com/vuejs/vue-next/blob/555e3be69d39f4b35a312916253d9f38cbcab495/packages/runtime-core/src/scheduler.ts#L10) *好像只用了 promise，可能之后会补充*