

单线程：js执行环境中负责执行代码的线程只有一个，安全简单。

笔记本： 1.有关印象笔记

创建时间： 2020/5/28 23:19

更新时间： 2020/7/12 1:14

作者： 1639079350@qq.com

标签： 1.执行队列

单线程：js执行环境中负责执行代码的线程只有一个，安全简单。

有两种模式：同步模式（排队执行，会产生阻塞）/异步模式（代码执行顺序混乱）

任务类型：

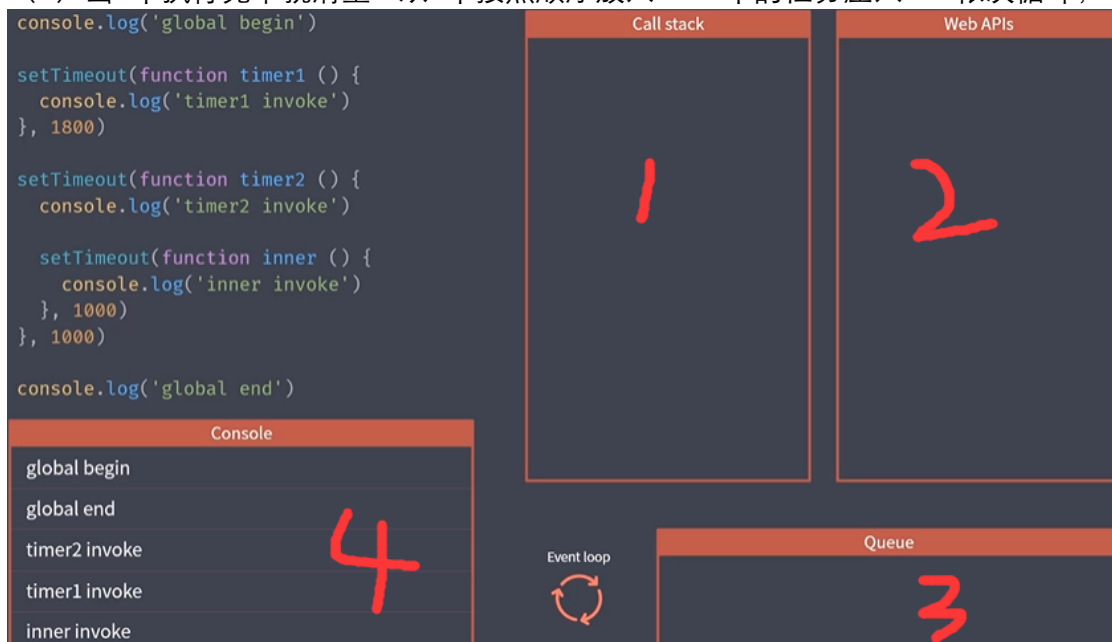
宏任务：回调队列中的任务。绝大多数异步任务都是宏任务；

微任务：在当前任务结束后立即执行；promise/MutationObserver、
process.nextTick

执行队列：

（1）任务放入1，有输出放入4，开启的异步放入2；（2中压入3的顺序由异步事件时间来计算）

（2）当1中执行完毕就清空-- 从2中按照顺序放入3 -- 3中的任务压入1 -- 依次循环；



回调函数：作为参数的函数，理解不方便：

promise执行时序：promise是微任务【微任务比宏任务快？】

```
const { log } = console;
```

```
// 没有异步的promise:
```

```
log('start');
```

```
Promise.resolve()
  .then(() => {
    log('promise 1');
  })
  .then(() => {
    log('promise 2');
  })
  .then(() => {
    log('promise 3');
  })
log('end');
// start - end - 1 - 2 - 3

// 定时器
log('start');
setTimeout(() => {
  log('setTimeout');
}, 0)
Promise.resolve()
  .then(() => {
    log('promise 1');
  })
  .then(() => {
    log('promise 2');
  })
  .then(() => {
    log('promise 3');
  })
log('end');
// start - end - 1 - 2 - 3 - setTimeout
```