基础进阶(事件循环机制,Promise,详解Module)

2020年1月14日

10:19

* 1. **事件循环机制**

**1.1 基本结论**

--JS一大特色是单线程,该线程拥有唯一的一个任务循环

--JS代码执行的过程,除了依靠函数调用栈来确定函数的调用顺序,还依赖任务队列(task queue)来确定一些代码的执行

--一个线程中,事件循环只有一个,任务队列可以有多个

--任务队列又分宏任务(macro-task)和微任务(micro-task),他们又被称为task和jobs

--macro-task大概包括：script(整体代码), setTimeout, setInterval, setImmediate, I/O, UI rendering。(setTimeout, setInterval同队列)

--micro-task大概包括: process.nextTick, Promise, Object.observe(已废弃), MutationObserver(html5新特性)

--setTimeout/Promise等我们称之为任务源。而进入任务队列的是他们指定的具体执行任务。

--来着不同任务源的任务会进入不同的任务队列.其中setTimeout与setInterval是同源的.

--事件循环的顺序解决代码的执行顺序.它从script开始第一次循环.之后全局上下文进入函数调用栈.直到调用栈清空(只剩全局),然后执行所有的micro-task.当所有的micro-task执行完毕之后,循环再次从macro-task开始,找到其中一个任务队列执行完毕,然后再执行所有的micro-task,如此循环

--每一个任务的执行,无论是micro-task还是macro-task,都是借助函数调用栈来完成

Node循环机制

