

node 高级 学员预习

node 的 buffer 、 stream、 Event、 OS 等等

Buffer

Buffer 的声明方式

和字符串的转换

Buffer 的拼接

Buffer 的截取

类型判断

Buffer 的应用场景

文件读取

Stream

EventEmitter

OS 一些小技巧

Global 全局变量解析

事件循环

node 的 buffer 、 stream、 Event、 OS 等等

Buffer

Buffer 的声明方式

和字符串的转换

Buffer 的拼接

在我们客户端给服务端之间进行大数据传输时，一般会把大量的数据进行拼接。

Buffer 的截取

类型判断

Buffer 的应用场景

文件读取

Stream

Stream 是 Node.js 最基本的概念之一，Node.js 内部的大部分与 IO 相关的模块，比如 http、net、fs，都是建立在各种 Stream 之上的。

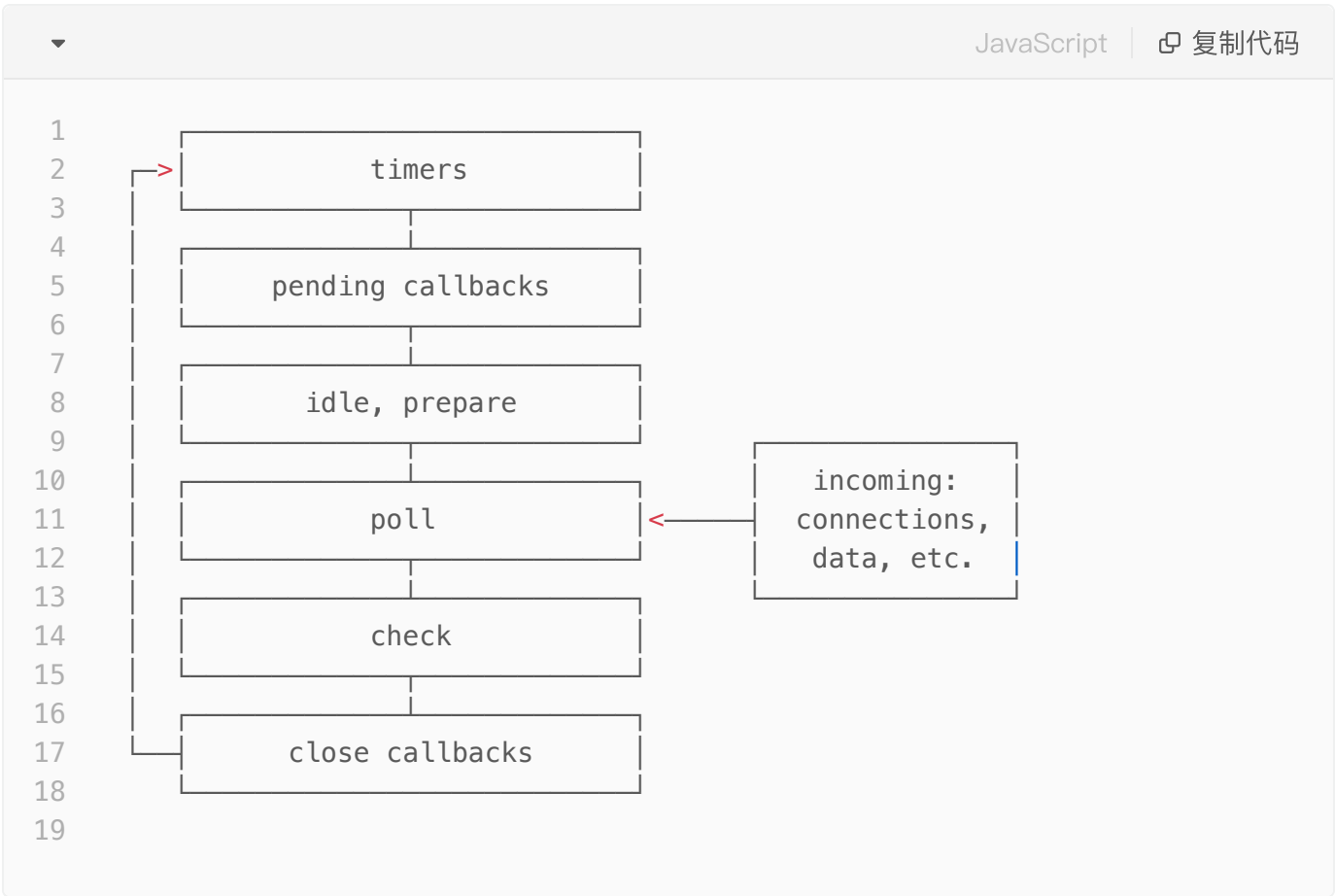
EventEmitter

一起写一个

OS 一些小技巧

Global 全局变量解析

事件循环



- **定时器**：本阶段执行已经被 `setTimeout()` 和 `setInterval()` 的调度回调函数。
- **待定回调**：执行延迟到下一个循环迭代的 I/O 回调。
- **idle, prepare**：仅系统内部使用。
- **轮询**：检索新的 I/O 事件; 执行与 I/O 相关的回调（几乎所有情况下，除了关闭的回调函数，那些由计时器和 `setImmediate()` 调度的之外），其余情况 node 将在适当的时候在此阻塞。
- **检测**：`setImmediate()` 回调函数在这里执行。
- **关闭的回调函数**：一些关闭的回调函数，如：`socket.on('close', ...)`。