

node 高级 学员预习

课程大纲

课程内容

- 说在前面
- node 的应用场景
- node 的历史
- node 的一些弊端
- node 的 buffer 和 stream
 - 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
- Global
- EventEmitter
- 事件循环

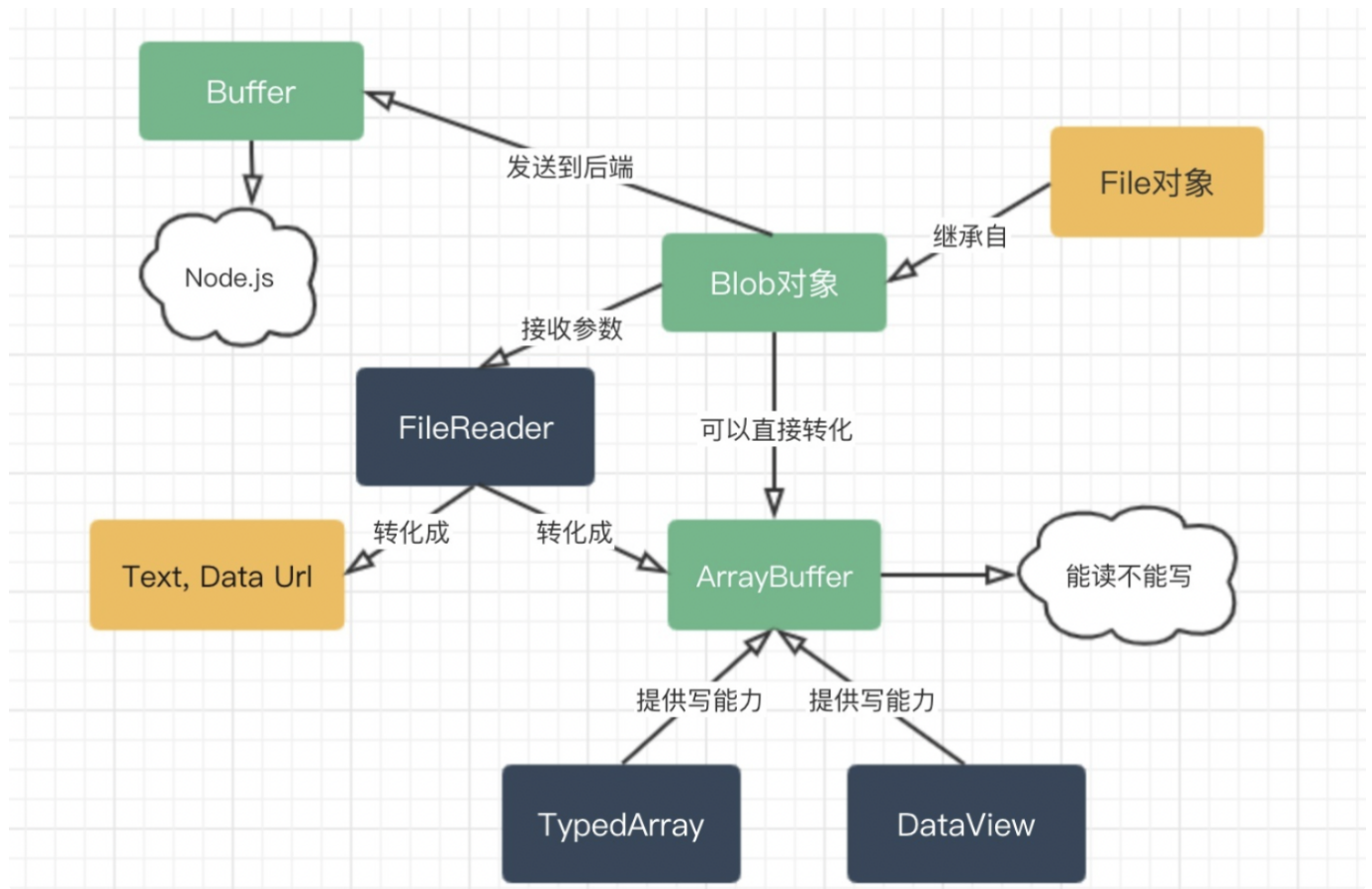
课程大纲

node.js 原理详解	Buffer / stream / events 等内容详解
	全局对象解析
	node.js 事件循环模型

课程内容

说在前面

JS 中和二进制相关的一些API和数据类型：



node 的应用场景

- 跨平台开发：几乎能覆盖你能想象到的 面向用户的所有平台：传统的PC Web 端、移动端、HTML5、RN、Weex以及硬件等（车载的一些框架）；
- 后端开发：面向网站、API、RPC服务；
- 前端开发：等前端工具链；
- 工具开发：脚手架、命令行工具；

node 的历史

LAMP 和 MEAN ；

- MEAN : mongodb \ express \ angular \ node.js ； 2014年；

在当下，node 的必要性太多了

分类	举例
压缩	UglifyJS, JSMin, CSSO
依赖管理	npm, Bower
模块系统	CommonJS, AMD, ESM
模块加载器	Require.js, jspm, Sea.js, System.js
模块打包器	Browserify, webpack
CSS预处理器	postcss, less, sass, scss, stylus
构建工具	grunt, gulp
模板引擎	jade, handlebars, nunjucks
JavaScript 友好语言	coffeescript, babel, typescript
跨平台打包工具	electron, nw.js, cordova
生成器	yeoman, slush, vue-cli, cra
其他	imagemin, dataurl...

node 的一些弊端

- 单线程很脆弱，但是可以通过 cluster / pm2 模块实现多核并发处理网络请求；
- node 对 MongoDB, MySQL, Redis 等方面的支持还不错，但是其他数据库就不是特别好了；
- 安全问题，伪命题。

浏览器和node.js 之间可能有多种协议，比如html, restful, bigpipe, comet, socket 等。

node 的 buffer 和 stream

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

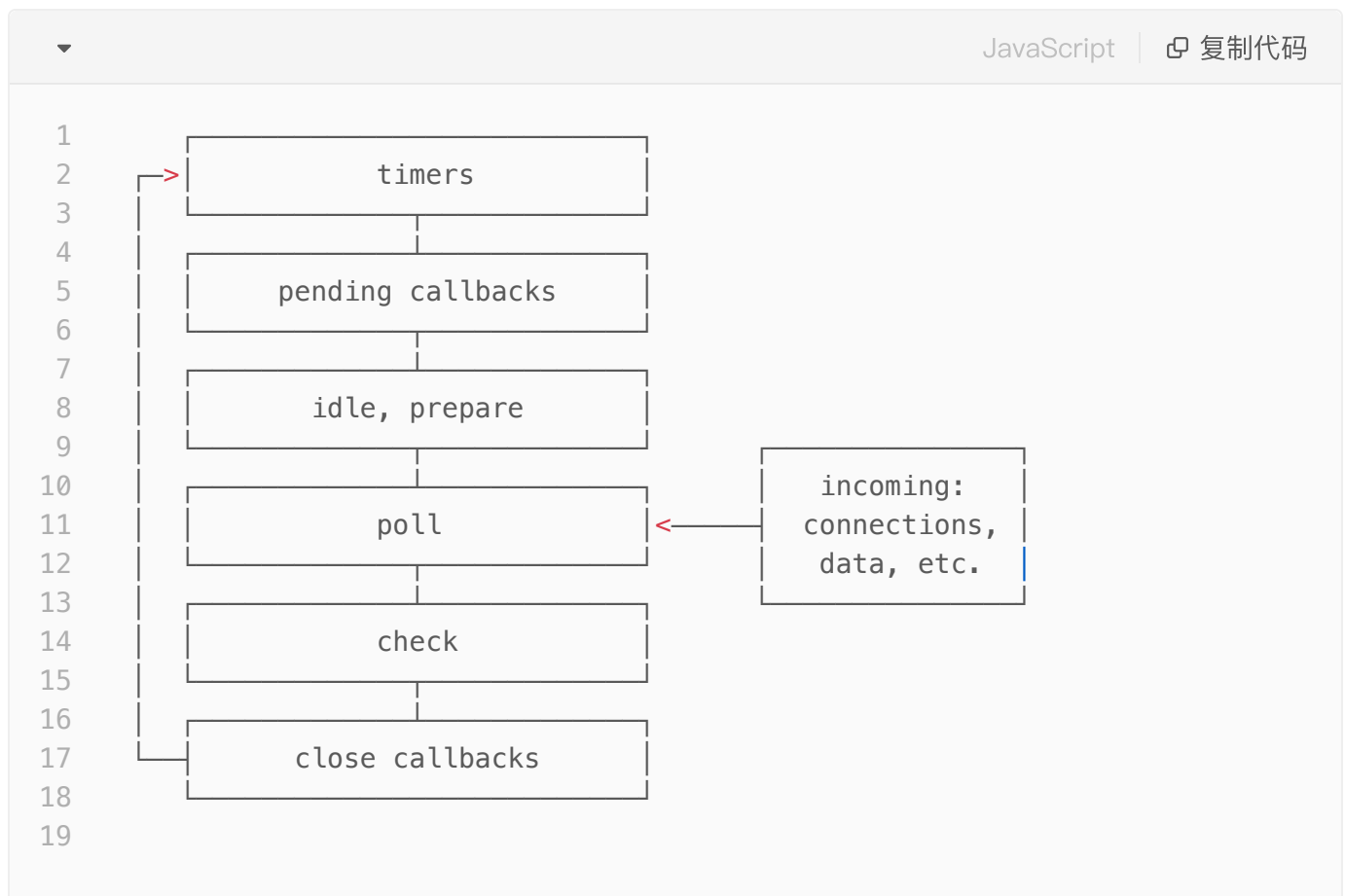
Stream

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。

Global

EventEmitter

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', ...)`。