指令

cls 清楚终端

参考文档

<https://koa.bootcss.com/>

如何编程 - 编写程序

每一项技术都有其源头，追根朔源理解其含义。

归属于node

npm包管理 / package / 事件循环机制

vue项目

react项目

前后端的响应和请求

# 回调函数

把函数作为参数传入就是简单的回调。把函数赋值给变量，作为参数传入高阶函数。

# 事件循环机制

<https://www.runoob.com/nodejs/nodejs-event-loop.html> 概念参考文档

在解析JS时，根据其特点内置有两个任务队列，一个是宏任务(macro-task)，一个是微任务(micro-task)。

宏任务是正常的JavaScript代码片段，代码都是同步执行。并没有体现优先级。

在微任务中则是有优先级。

宏观任务中: javascript代码片段 process promise I/O setMmediate setTimeout setInterval

微观任务 process.nextTick promise.then async/await

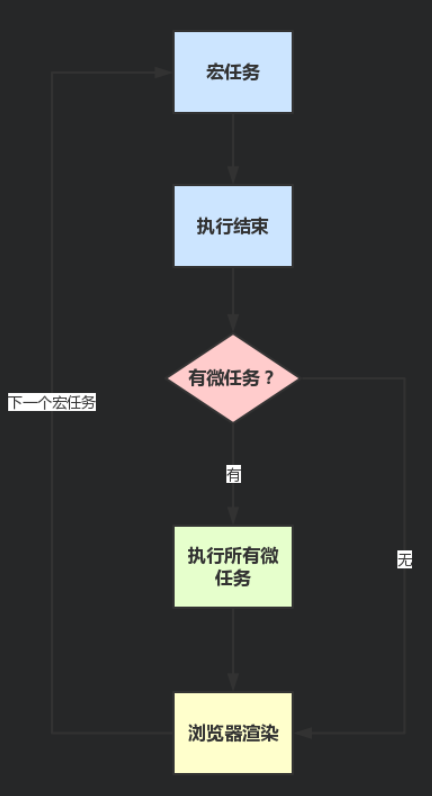
在宏观任务中**process promise**是同步的，**process.nextTick/promise.then**是异步的。

在微观任务中process.textTick优先级高于Promise.then, async/await的原理是Promise，

setTimeout/setInterval 顺序执行 ，并且比setImmediate的优先级要高。

回调函数Promise.then, async/await正常的执行循序都是**先近先出。实际谁的数据准备就绪就先执行。**node中Koa中间件函数是**先进后出。**

**事件循环机制，执行栈反复的检查任务队列是否有可执行的事件。**



这种不用等所有代码执行完而执行回调，只是在每一次宏任务结束后，回调已准备好即可执行。

async/await 是同时出现的。

# Package.json一个文件

管理包的文件

dependencies 生产环境依赖的包

devDependencise 开发环境依赖的包

# Webpack一个工具

本质上，webpack是一个用于现代 JavaScript 应用程序的静态模块打包工具

# 项目

技术

node.js

koa koa-static koa-views path

koa 搭建服务器

处理静态文件

路径解析

扩展文件名解析

路由-动态路由，返回值

步骤

前端和后端

先做接口

# **Koa库**

搭建服务器，以及一系列相依赖的模块。

# **pug模板引擎**

简单的语法， 后台返回的数据，脚步和样式的引入。同步异步。

# mongodb

Node.js链接mongodb 参考

<https://www.runoob.com/nodejs/nodejs-mongodb.html>

<https://www.cnblogs.com/fightjianxian/p/12263784.html> 【增删查改比较详细】

在node中是以模块的形式的使用。因此需要安装对应的模块。

mongodb看做一个程序，操作数据的程序，有客户端和服务器端。

使用时必须启动服务器端。

指令 mongod 指的是启动服务端

mongo 启动客户端

在客户端对数据库的数据进行各种花式操作，主要时增、删、查、改。

<https://www.cnblogs.com/web-fengmin/p/6435681.html>

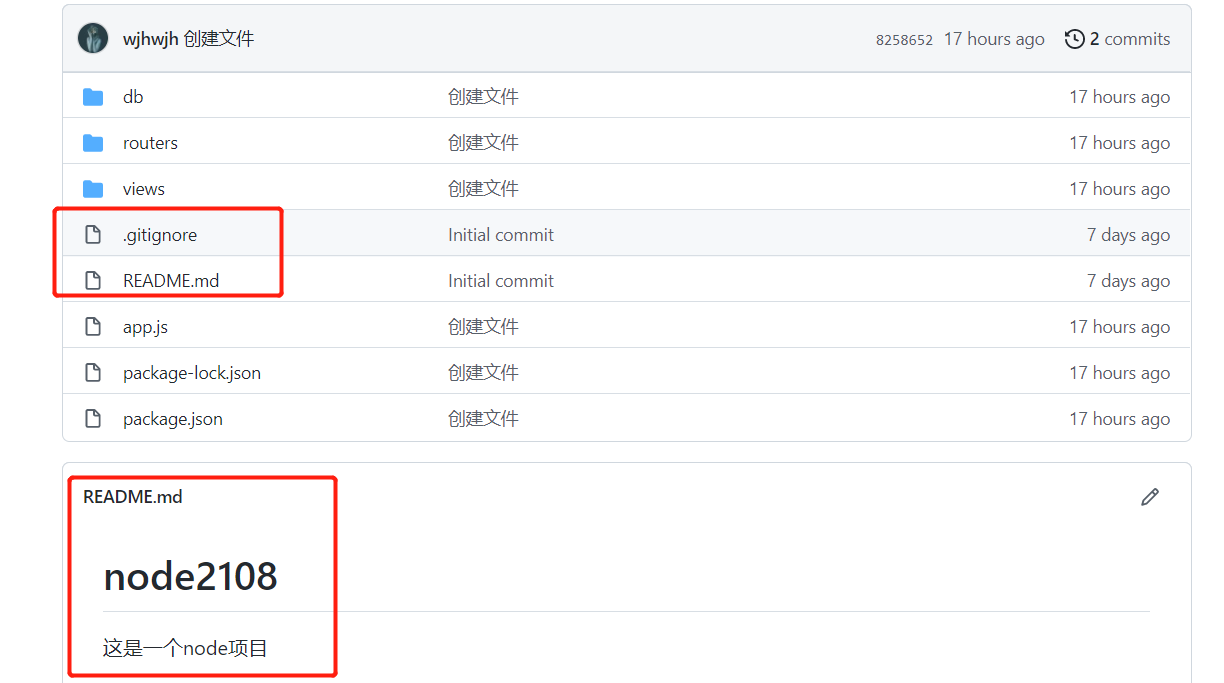
# **git配置**

# git好用是真的，配置不好搞也是真的。这个笔记本是前一个阶段从单位申请的，算是一台裸机，想用什么完全都是从新配。所以git也不例外。

第一步先到git官网下载安装

第二步

我在github上创建一个仓库，存放我的代码。



**开始只有2个文件**

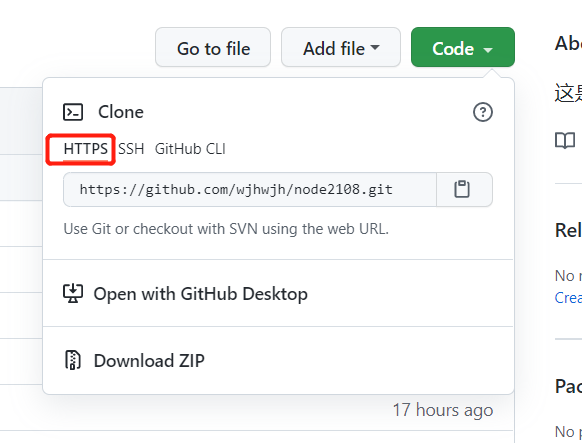
第三步

克隆到本地

开始我使用的是htts格式clone

git clone 地址

这样做没毛病，也成功的下载到本地。可是拉去和提交却始终不成功。



第四步

因为提交始终不成功，再push时还迷惑性的提示打开网址输入邮箱和密码，看似正确却缕缕失败。于是各种找，1> 使用SSH链接远程仓库不需要总是输入账号和密码 2> 使用SSH就要配置ssh key

https://www.cnblogs.com/hafiz/p/8146324.html [参考设置ssh key]

https://www.cnblogs.com/guanqiweb/p/9839443.html

配置好ssh key 提交和拉取都没有问题了。

https和ssh都能clone远程仓库的项目，但是如果要提交项目则必须要配置ssh key。再远程仓库中与每台电脑都要有相应的配置。

开始是用htts克隆的代码，为了不用输密码换成ssh方式,执行步骤

git remote rm origin 删除远程仓库

git remote add origin [git@github.com](mailto:git@github.com):wjhwjh/node2108.git 重新链接

git remote -v 查看远程仓库

git config --global --list 查看用户信息配置，邮箱名和用户名

<https://www.cnblogs.com/moyiwang/p/12739573.html> git指令