

node1

node1

配置环境变量
前端的模块化
node特点
基于commonjs规范，对模块的操作
如何使用模块
npm

定义模块
如何导出模块
require(第三方模块)导入机制
global
定时器
计算时间差
process
区分开发和上线环境
process.nextTick

nodejs继承
nodejs的发布订阅
学习的nodejs核心模块

配置环境变量

- node可以在命令行下执行的原因
- 将可执行文件配置到了环境变量下

前端的模块化

闭包
单例模式
requirejs(AMD)依赖前置，seajs(CMD)就近依赖
commonjs规范，是node模块实现的module.exports

node特点

异步io(定时器，回调函数)

单线程（其他语言多线程）多线程实现就是切换执行上下文，（速度很快感觉不出来，好想在干很多件事）

进程是包含线程的，node一个进程只能开一个线程

基于commonjs规范，对模块的操作

如何使用模块

require() 将模块导入，这个是同步的，有回调函数的都是异步，多次require，并不会多次导入，他会缓存，require的文件会执行一次，模块分为三种

- 自己写的模块，用的时候要采用相对路径./../
- 第三方模块，直接写模块的名字即可，不需要./../ 全局安装第三方模块 要加一个-g 例如 `npm install http-server -g` 只能在命令行使用 默认下载是通过npm下载，可以使用nrm进行切换。本地第三方包只能在当前文件夹下使用，又分为开发依赖 `--save-dev/-D` 和发布依赖 `--save/-S`
- 核心模块
安装包之前，一定要初始化 `npm init -y` ,如果没有初始化，并且外层文件夹有node_modules,直接安装包就会向上安装。在初始化的时候，不能用汉字或者特殊字符，不能用已经存在的包名命名，.json文件里面不能写注释，安装的包会默认安装到node_modules里面

nrm

安装 `npm install nrm -g`
`nrm ls` 列出所有可用源
`nrm use` 用哪个源
`nrm test` 测试所有可用源的网速
之后依然这样使用 `npm install some`
两个第三方包 `http-server nodeppt`

定义模块

在node中，一个js文件就是一个模块

如何导出模块

- `exports`
如果使用`exports`导出，必须通过属性一个个添加的方式，否则无法挂载到`module.exports`
- `module.exports`
这个值可以随便写，因为返回的就是它，不管是基本数据类型还是引用数据类型

require(第三方模块)导入机制

首先会到当前目录下找`node_modules`文件夹，找到名字叫`mime`的文件夹，然后找到`package.json`问价，找到入口文件，将其执行，如果没有找到，则就去上一级目录查找，一直到盘的根目录，最后报错。 `module.paths` 查找路径

global

闭包中的`this`才是`global`，普通的是导出对象`module.exports`

定时器

`setImmediate` `clearImmediate` 立即执行

`setTimeout`

`setInterval`

定时器里面的回调函数的 `this` 代表的定时器自己

改变里面 `this` 的方法

1. `call` `apply` 方法不可取，因为会导致函数执行
2. `bind` 方法可以，（他只绑定一次，只绑定不执行）
3. 使用 `that = this` 进行缓存
4. 使用 箭头函数处理，箭头函数没有自己的`this`，它的`this`指向的是外层的`this`

定时器回调函数传递参数的问题

可以从第二个参数位置开始，一般是在时间的后面

这三个定时器都是异步的， `setImmediate<=setTimeout`

计算时间差

```
console.time('name');  
console.timeEnd('name')
```

process

区分开发和上线环境

在电脑上设置变量process.env (代表当前进程的环境变量) `set BIANLIANGMING=development`

```
if(process.env.bianliangming==='development'){  
  // 开发环境  
}  
else{  
  // 生产环境  
}
```

process.nextTick

下一队列，当前队列的底部，重要的事情放在nextTick中，稍微不重要的用setImmediate

```
process.nextTick(function(){})
```

nodejs继承

常用的方法 `util.inherits(Child,Parent)` 只继承公有属性

nodejs的发布订阅

订阅代表一对多的关系，发布就是当事件触发，执行所有订阅的方法

一个核心模块events

两个方法 on监听 emit触发 removeListener(类型，回调函数名字)移除事件

学习的nodejs核心模块

http url

fs

util

'events