nodeJs的理解

nodeJs模块化的作用

基于common.js的模块化

命名冲突解决

文件依赖解决

nodeJs模块化的形式

module对象下的属性和方法

module的作用域

全局

局部

块

module模块作用域对象

模块分类：

文件模块

核心模块

目录（文件夹）模块

nodeJs模块化的作用

基于common.js的模块化

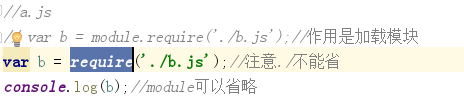
命名冲突解决：每个文件都是独立的，每个模块都有自己的作用域

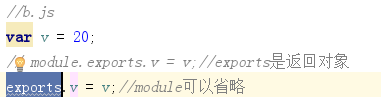
文件依赖解决

nodeJs模块化的形式

不用闭包，

每个文件里面都有一个module对象，对象下面有属性和方法。





module对象下的属性和方法

每个页面都有自己的对象：module，不需要创建——类似每个浏览器窗口有window

id：当前模块的唯一名称，当前模块的编号，该编号默认是当前模块的路径

**exports**：一个对象，该对象里面存放了该模块对外导出的数据

parent：当前面模块的父模块

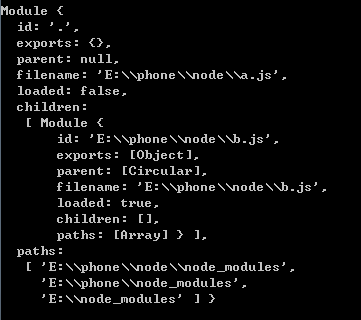
children：当前模块的子模块

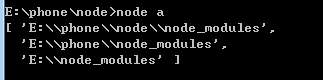
paths：模块查找路径

loaded：是否是执行模块

**require()**：加载其他模块，reuqire()方法的返回值就是被加载模块的exports对象

在nodejs还提供了一个全局方法 require() 等同与 module.require() 这个方法





module模块作用域对象

在nodejs中，每一个文件都有自己独立的作用域-模块作用域，该作用域下一些内置的作用域对象与方法

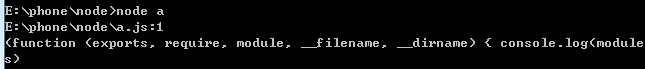
- \_\_dirname：返回当前模块做在的目录

- \_\_filename: 返回当前模块的路径包括文件名称

- exports

- module

- require()



模块分类：

文件模块

**require('./b.js')**

核心模块

**require（‘assert’）**——注意不要加./，表示是核心模块

目录（文件夹）模块

**require（‘miaov’）**

在当前项目下面新建一个文件夹node\_modules

在node\_modules文件夹下面再建文件夹，比如miaov

在miaov中建文件

比如：E:\phone\node\node\_modules\miaov

存在问题：核心模块和目录（文件夹）模块执行语句是一样的，系统是怎么判断是什么语句的。

在module下面有一个属性paths，该属性是一个数组，里面存放了一组路径字符串，那么这个路径是我们通过require方法加载非文件模块的时候的查找路径。

require如果加载的是非路径的模块（文件模块），那么就会从当前目录下的node\_modules目录下去找对应的文件夹，但是一个文件夹模块下会有很多的文件模块，那么到底加载哪个文件模块，这个需要通过一个叫做 package.json 的文件来进行说明，这个文件称为模块配置文件，同时我们把这种文件夹模块称为 - 包。

当我们通过require去加载这个包模块的时候，首先会读取这个包下的package.json文件，返回的就是这个文件中main指定的js文件。

如果在当前目录下找不到，则会按照module.paths中指定路径，也就是上一级目录进行查找，如果一直找到根目录还找不到，则会查找核心模块，如果还没有，报错。

**require（‘miaov’）**

