1、模块

console.log(\_\_filename);//当前模块绝对路径和模块名称

console.log(\_\_dirname);//当前模块绝对路径不包含文件名称

2、模块分类

|  |  |  |
| --- | --- | --- |
|  | 以路径开头 | 不以路径开头 |
| 文件模块 | require('./ab.js')  用于引入自定义模块 | require('url')  用于引入官方的核心模块 |
| 目录模块 | require('./02ran') //02ran目录  首先到目录下寻找package.json文件中main属性对应的文件，如果找不到则会自动寻找index.js | require('ran') //ran目录，  会到当前目录下的node\_modules中寻找ran目录，如果找不到会一直往上一级目录下的node\_modules中寻找，  用于引入第三方模块 |

json是一种数据类型，通常包含对象和数组两种，要求属性名必须使用双引号包裹，属性值字符串必须使用双引号包裹。

3、包和npm

包package，表示第三方模块，也就是目录模块

npm原来管理第三方模块的工具，包含下载安装、卸载、升级、上传

4、查询字符串模块querystring

浏览器向服务器发请求传递数据的一种方式，位于浏览器的地址栏中，

查询字符串模块用于解析和格式化的查询字符串的工具

parse()将查询字符串解析为对象。

5、URL模块

统一资源定位符，互联网上的任何的资源都有对应的网址。

URL模块是用于处理与解析URL

6、timer定时器。

全局API，不需要引入。

// 全局函数

// 一次性定时器

//开启 let time = setTimeout(回调函数,间隔时间),当间隔时间到了调用回调函数

let timer = setTimeout(()=>{

console.log("炸弹爆炸了")

},5000)

//清除定时器,需要把开启的定时器存到变量，变量名放到clearTimeout

clearTimeout(timer)

//周期性定时器

let t = setInterval(回调函数,间隔时间) 每隔一段时间执行一次回调函数

//清除周期性定时器

// clearInterval(t)

var total = 1;

let t = setInterval(() => {

console.log("炸弹爆炸了");

total++;

if (total == 4) {

clearInterval(t)

}

}, 1000)

//执行多少次后清除定时器。

3、立即执行定时器。执行耗费时间的函数，放到事件队列。主程序执行完在执行事件队列

console.log(2)

let time = setImmediate(()=>{

console.log("1")

})

//可以清除，clearImmediate(time)

process.nextTick(()=>{ //插队

console.log("444")

})

console.log(3) ====>2 3 4 1

process.nextTick执行顺序早于setImmediate

文件系统模块：

服务器端文件包括文件形式和目录形式。

const fs = require('fs');

// console.log(fs)

// 查看文件状态

let f = fs.statSync('../04');

// console.log(f)

// 查看是否为文件

console.log(f.isFile())

// 查看是否为目录

console.log(f.isDirectory())