

全局对象和全局变量

了解更多 Node 的最基本成员以及核心概念

[Node.js 中文文档](#)

[Node.js 官网](#)

1、全局对象

- 所有模块都可以调用
- ① global:
 - 表示Node所在的全局环境，类似于浏览器中JavaScript的window对象。
 - node.js 中的全局对象是 `global`，全局对象中的成员直接使用不需要包含它
 - [全局变量](#)

```
global.console.log("aaaaa");

global.setTimeout(()=>{console.log("bbbbbb")},3000);
```

- ② process:
 - process是nodejs中的全局对象，不需要require进来，就可以直接使用
 - 指向Node内置的process模块，允许开发者与当前进程互动。
 - process可以用来做：[获取进程信息](#) 和 [操作进程](#)。
 - [在线文档](#)

```
console.log(process.execPath); //可执行文件的绝对路径,如 /usr/local/bin/node
console.log(process.version);  //版本号
console.log(process.versions); //依赖库的版本号
console.log(process.platform); //运行平台。如 darwin、freebsd、linux、sunos、win32
console.log(process.env);      //操作系统环境信息
//process.chdir('/tmp');      //切换当前工作目录位置
console.log(process.cwd());     //此方法返回当前目录，不使用任何参数
```

- 获取命令行中的参数信息:

```
//打印命令行中的参数: node 文件名.js 参数1 参数2
process.argv.forEach((val, index) => {
  console.log(`${index}: ${val}`);
});
```

- 从标准输入设备中获取信息

```
//接收键盘输入值，并发送服务器端
process.stdin.setEncoding('utf8');//设置字符编码
process.stdin.on('readable', () => {
  let msg;
  // 使用循环确保我们读取所有的可用数据。
  while ((msg = process.stdin.read()) !== null) {
    //判断输入q退出
    if(msg == "q\r\n"){
      console.log("exit! ");
      process.exit();
    }
    console.log(msg.trim());
  }
});
```

- 当进入NODE的命令行方式（REPL环境）。退出要退出的话，可以输入 process.exit();
- ③ console:
 - 指向Node内置的console模块，提供操作控制台的输出功能，常见使用方式与客户端类似。
 - 通常是写console.log(), 无须多言

2、全局函数：

- ① 定时器函数（共计6个）
 - setTimeout()
 - clearTimeout()
 - setInterval()
 - clearInterval()
 - setImmediate() //这是node中增加的
 - clearImmediate()
- ② require：用于加载模块。
 - 常用核心模块

模块名称	作用
util	实用工具
url	url 地址操作
path	路径操作
os	操作系统信息
fs	文件操作
http	网络操作
events	事件触发器
net	网络操作
querystring	解析查询字符串
...	...

3、全局变量：

- ① `__filename`：指向当前运行的脚本文件名(双下划线开头)
- ② `__dirname`：指向当前运行的脚本所在的目录。

```
const path = require("path");

console.log(__filename); //获取当前模块的文件名（含文件）

console.log(__dirname); //获取当前模块的目录名（含路径）
//或
console.log(path.dirname(__filename)); //通过path获取同上效果
```

4、准全局变量

- 模块内部的局部变量，指向的对象根据模块不同而不同，但是所有模块都适用，可以看作是伪全局变量。
- 主要有：`module`, `module.exports`, `exports`等。
- `module`变量指代当前模块。
 - `module.id` 模块的识别符，通常是模块的文件名。
 - `module.filename` 模块的文件名。
 - `module.loaded` 返回一个布尔值，表示模块是否已经完成加载。
 - `module.parent` 返回使用该模块的模块。
 - `module.children` 返回一个数组，表示该模块要用到的其他模块。
- `module.exports`变量表示当前模块对外输出的接口，其他文件加载该模块，实际上就是读取 `module.exports` 变量。

- `exports` 是对于 `module.exports` 的更简短的引用形式（别名）。