**学习目标**

（1）全局对象

（2）模块系统

**1.全局对象**

js的全局对象——window

NodeJS的全局对象——global

说明：在脚本模式下，NodeJS中声明的变量和函数都属于局部变量和局部函数

，不能使用global访问

**（1）console对象**

console.time('FOR-LOOP');

//要检测的程序

console.timeEnd('FOR-LOOP');

作用：检测程序的执行时间

**（2）process(查看进程相关的信息)**

process.arch查看CPU架构类型

process.env查看操作系统的环境

process.platform查看操作系统的类型

process.cwd()查看当前的工作目录

process.version/versions查看当前nodejs的版本信息

process.uptime()查看nodejs的运行时间

process.memoryUsage()查看内存使用情况

process.pid 查看当前nodejs的进程编号

process.kill() 杀死某一个nodejs进程

**（3）Buffer对象**

缓冲区，用于存储数据的内存区域

new Buffer(32);//指定缓冲区的长度，默认值未0

new Buffer([10,12,45])// 在缓冲区存储数组

new Buffer('abcd');//在缓冲区存储字符串

**（4）全局函数**

定时器

①一次性定时器

setTimeout(func,delay); -->setTimeout(()=>{ },delay);

func:要执行的任务，是一个回调函数，可以用箭头函数形式，

delay 间隔时间，单位是毫秒

clearTimeout（timer）; 清楚一次性定时器

②周期性定时器

setInterval(func,delay);--->setInterval(()=>{ },delay);

func:要执行的任务

delay 间隔的时间

clearInterval(timer) 清除周期性定时器

练习：每个2秒钟，打印出12345678910

③process.nextTick(( )=>{ }); 开启定时器

④setImmediate(( )=>{ }); 开启定时器

clearImmediate(timer);

同步和异步

同步：按照程序顺序执行，先执行前边，如果前边程序没有执行完，后边的程序就无法执行，也称为**阻塞模式**

异步：在程序执行的过程中，遇到异步函数等异步操作，会把这放到整个进程的最后边，整个进程会跳过这段继续执行，也称**为非阻塞模式**

在nodejs中，异步操作主要是结合回调函数来实现。

**2.模块系统**

在nodejs下模块分为三类，核心模块，第三方模块，自定义模块。

（1）核心模块，直接引入

require('querystring')

（2）自定义模块

在nodejs的文件中，任何一个js文件都是自定义模块，在js文件中称为模块作用域。

在任意一个js文件中，自动包含一个function，任何自己写的代码自动被function所包含。

(function(exports,require,module,\_\_filename,\_\_dirname){

//程序员自己写的代码

//\_\_filename 文件的完整路径和名称

//\_\_dirname 文件的完整路径

//require 用于引入其他的模块

//module.exports 导出某些模块，是一个对象

})

课后练习：创建一个主模块，在创建一个功能模块（求长方形的面积），把功能模块导出，在主模块中导入，并调用方法