# day01

## 1.ES6

(1)函数中的参数

如果没有为形参设置默认值，入股没有传递实参，则自动调用形参的默认值

function add(a,b,c=0){ };add(2,5,8);//15

(2)模板字符串

`在模板字符串中可以写任何形式的字符串${js表达式}`

练习：创建一个图书对象，包含有图书的编号，名称，价格，上架时间，是否在售(1->是/0->否)，使用模板字符串打印书的内容

## 2.nodejs概述

基于谷歌的V8引擎，运行在服务器端的语言。

对比JS和NodeJS

1. JS运行在浏览器端，存在多个浏览器，容易产生兼容性的问题；而NodeJS运行在服务器端，只有V8引擎一种解释器，不存在兼容性问题。
2. 两者都有内置对象、自定义对象，宿主对象(根据执行环境的不同)
3. JS用于浏览器中的交互效果，而NodeJS用于服务器端，例如web服务器创建，数据库操作，文件操作。

官网 nodejs.org

中文版 nodejs.cn

NodeJS执行方式：

脚本模式：node c:/xampp/...1.js 回车

交互模式：node 回车进入交互模式

退出交互模式，两次Ctrl+c 或者 .exit

### 1.全局对象

NodeJS:global

在交互模式，声明的变量和创建的函数默认都属于全局对象下的，可以使用global来访问。var a=1;global.a

练习：在脚本模式下，使用global来访问变量和函数

在脚本模式，声明的变量和创建的函数默认都是局部作用域下，不能被global访问，好处在于不能污染全局。

JS:window

在浏览器下，脚本中声明的变量和创建函数默认都是全局作用下的，可以使用window来访问，var a=1;window.a

(1)console对象

console是global下的成员，可以使用global.console访问

global.console.log() 打印日志

console.info() 打印消息

console.warn() 打印警告

console.error() 打印错误

console.time('自定义字符串') 开始计时

console.timeEnd('自定义字符串') 结束计时

开始计时和结束计时中的字符串要保持一致

练习：查看for/while/do-while循环相同次数的耗时情况。

(2)process对象

进程： 当操作系统启动一个程序时，会将必须的可执行文件和数据文件从系统调入到内存，分配到必要的内存空间。

对应操作系统中的任务管理器

process.arch 查看当前CPU架构

process.platform 查看当前的操作系统

process.env 查看当前操作系统的环境变量

process.version 查看当前NodeJS版本

process.pid 查看当前进程的编号

process.kill() 结束某个编号的进程

(3)Buffer对象

缓冲区：在内存中存储数据的区域，常用于存储网络传输时的资源

创建Buffer:

Buffer.alloc(5,'abcde');

将Buffer数据转为字符串:

buf.toString()

(4)全局函数

parseInt/parseFloat...

①一次性定时器

开启:

var timer=setTimeout(回调函数，间隔的时间);

当间隔的时间到了，执行一次回调函数，单位是毫秒

清除:

clearTimeout(timer);

②周期性定时器

开启:

var timer=setInterval(回调函数，间隔的时间);

当间隔的时间到了，执行一次回调函数，单位是毫秒

清除:

clearInterval(timer);

③立即执行

开启:

var timer=setImmdiate(回调函数);

回调函数在事件队列中执行，在事件的最前面执行

清除:

clearImmdiate(timer);

④立即执行

process.nextTick(回调函数)

在事件的最后执行

### 2.模块

模块是一个独立的功能体

在NodeJS下模块分为3类：自定义模块、核心模块、第三方模块

自定义模块:

在NodeJS下，每一个文件都是模块

NodeJS自动为每个文件添加了构造函数，所有的代码被构造函数所包含

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

程序员写的代码

})

require('08\_2.js'); //用于引入一个模块

module:当前的模块对象

module.exports:当前模块的导出对象，可以提供其他模块使用的属性和方法

课后任务：

(1)复习，整理思维导图

(2)练习：创建两个模块main.js(主模块),circle.js(功能模块);在功能模块下

创建两个函数，传递1个参数(半径),分别获取圆的面积和周长，导出这两个函数; 在主模块中引入功能模块，并调用两个方法

(3)预习模块querystring(查询字符串),url,fs