复习

1.es6

ECMA

ECMAScript

ES6

(1)块级作用域

使用let关键字声明变量，阻止变量提升，添加到块级作用域中，成为局部变量。

块级作用域：{}、if、else、while、for...

(2)参数增强

给参数设置默认值

|  |
| --- |
| function add(a,b,c=0){  } |

(3)箭头函数

回调函数的另一种写法，但是不等价于匿名函数

|  |
| --- |
| sort( (a,b)=>{  return a-b;  } ); |

如果函数体中只有一行代码，并且是return形式的，可以省略为 sort( (a,b)=>a-b );

(4)模板字符串

|  |
| --- |
| `在模板字符串之间放任何形式的字符 ${JS表达式} ` |

练习：创建一个人对象，包含人的姓名，性别(0/1)，年龄；在模板字符串之间打印 个人的介绍。

1.NodeJS概述

基于谷歌V8引擎，运行在服务器端的环境

**对比JS和NodeJS**

(1)JS运行在浏览器端，存在多种浏览器解释器，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(不相同)

(3)JS用于浏览器端的交互效果，NodeJS用于服务器端操作，例如创建web服务器、操作数据库、文件操作

**NodeJS执行方式**

脚本模式 node c:/xampp/.../01.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

NODEJS API

http://nodejs.cn/api/globals.html 中文版

https://nodejs.org/dist/latest-v10.x/docs/api/ 英文版

**适用场景**

基于社交网络的大规模web应用(属于I/O密集型)

1.全局对象

NodeJS: global

在交互模式下，声明的变量和创建函数都属于全局对象下的，可以使用global来访问 global.a global.fn

在脚本模式，声明的变量和创建函数都不属于全局对象下的，

不能使用global来访问

JS：window

在JS脚本中声明的变量和创建的函数都属于全局对象下的，可以使用window访问 window.a window.fn()

2.process对象

进程: 执行程序的时候，操作系统分配内存空间

process.arch 查看当前CPU架构

process.platform 查看当前的操作系统

process.env 查看当前的环境变量有哪些

process.version 查看当前NodeJS版本号

process.pid 查看当前进程的编号

process.kill() 结束某个编号的进程

3.console对象

标准输出

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

注意：开始计时和结束计时的字符串保持一致。

练习：循环100万次，分别使用for，while，do-while查看耗时情况。

4.Buffer对象

缓冲区: 在内存中存储数据的区域，常用于网络传输时的资源

创建Buffer

var buf=Buffer.alloc(5, 'abcde')

将buffer数据转为字符串

String(buf)

buf.toString()

5.模块

模块：就是一个独立的功能体

在NodeJS下，模块分为：自定义模块、核心模块、第三方模块

自定义模块

**在NodeJS，每一个文件都是一个模块**

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){  程序员自己写的代码  }) |
| require(): 用于引入一个模块，同一级目录写路径 ./  module: 指代当前的模块对象  module.exports: 当前模块的导出对象，公开的属性和方法 |

课后任务

(1)复习今天内容，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，传递半径，计算圆的周长和面积，并导出两个函数，在主模块下引入功能模块，并调用两个函数。

(3)预习核心模块 querystring url fs 文件操作