复习

学习一门编程语言的基本步骤

(1)了解背景知识：历史、现状、特点、应用场景。

(2)搭建开发环境，编写hello world

(3)变量和常量

(4)数据类型

(5)运算符

(6)逻辑结构

(7)通用小程序

(8)函数和对象

(9)第三方库、框架

(10)实用的项目

1.ES6新特征

(1)函数中的参数

ES6允许为形参设置默认值，如果没有传递实参，自动调用形参的默认值

(2)模板字符串

|  |
| --- |
| ` 在此之间可以写任意形式的代码 ${JS语法} ` |

练习：创建一个员工对象，包含姓名、性别、生日、工资；使用模板字符串打印员工的信息。

'姓名：\*\*，性别：\*\*，生日：\*\*，工资：\*\*'

2.nodejs概述

nodejs基于谷歌V8引擎(JS解释器)，运行在服务器端的语言，基于JS。

http://nodejs.cn 中文

http://nodejs.org 英文

**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在服务器端只有一个运行环境，不存在兼容性。

(2)两者都有内置(ES)对象、自定义对象、宿主对象(根据执行环境的不同)

(3)JS用于网页中的交互效果，而NODEJS用于服务器的数据库操作、文件操作...

**NODEJS的执行方式**

脚本模式 node c:/xampp/..../1.js

交互模式

node

两次ctrl+c 或者输入 .exit

3.全局对象

**NODEJS: global**

在交互模式下，声明的变量或者创建的函数都属于全局对下的，可以使用global访问，例如 var a=1; global.a

在文件中声明的变量或者创建的函数都属于是局部作用域下的，不能使用global来访问。

**JS: window**

在浏览器下，文件中声名的变量或者创建的函数都属于是全局作用域下的，可以使用全局对象访问；

例如: var a=1; window.a

(1)console对象

global.console.log() 打印消息

global.console.info() 打印消息

global.console.warn() 打印警告消息

global.console.error() 打印错误消息

global.console.time('自定义字符串') 开始计时

global.console.timeEnd('自定义字符串') 结束计时

自定义字符串前后要保持一致。

练习：使用计时查看for，while，dowhile循环10000的耗时。

(2)process对象

查看当前计算机的进程

process.arch 查看当前CPU架构 X64

process.platform 查看当前的操作系统 win32

process.env 查看当前计算机的环境变量

process.version 查看当前nodejs的版本号

process.pid 查看当前的进程编号

process.kill( ) 杀死某一个编号的进程

(3)Buffer对象

缓冲区：在内存中存储数据区域，存储网络传输时的资源

创建buffer

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

将buffer存储的数据转为普通字符

buf.toString()

(4)全局函数

parseInt/parseFloat/encodeURI/decodeURI/isNaN

isFinite/eval

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

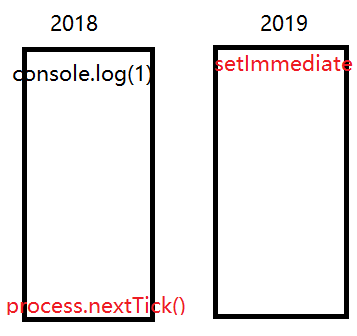
|  |
| --- |
| 开启  var timer=setInterval(回调函数，间隔的时候);  当间隔时间到了，执行回调函数；  清除  clearInterval(timer); |

练习：使用周期性定时器每隔3秒打印hello，打印三次后，清除定时器。

(3)立即执行(了解)

|  |
| --- |
| process.nextTick(回调函数)  在当前事件循环的结尾(2018年底)执行 |

|  |
| --- |
| var timer=setImmediate(回调函数)  clearImmediate(timer)  在下一个事件循环的开头(2019年初)执行 |



4.模块

模块可以理解为是一个功能体(积木块)

在NODEJS下模块分为自定义模块、核心模块(官方提供)、第三方模块

在NODEJS下，任意一个文件都是一个模块，文件中的代码默认是被一个构造函数所包含。

**以下代码中红色代码都是NODEJS自动为每一个文件添加的**

|  |
| --- |
| (function(exports,require,module,\_\_dirname,\_\_filename){  //程序员编码写的代码  }) |
| \_\_dirname 当前模块(文件)的完整路径  \_\_filename 当前模块(文件)的完整路径和文件名称  require() 引入一个模块  module 指代当前的模块对象  module.exports 当前模块的导出对象(公开)，可以供其它的模块使用的属性和方法  exports 等价于 module.exports |

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)；在功能模块中创建两个函数，传递1个参数，分别获取圆的周长(getLength)和面积(getArea)，导出这两个函数；在主模块中引入功能模块，调用两个方法。

(3)预习 querystring、url、fs模块