一.错误处理

1.常见的错误

语法错误(SyntaxError)：不符合语法规范，例如出现了中文的符号，缺少括号..

引用错误(ReferenceError)：引用错误，找不到变量，使用了未声明的变量

类型错误(TypeError)：类型错误，当前调用的不是一个函数类型

自定义错误：开发人员自己指定的错误，用于调试

throw 错误内容

2.错误处理

即使出现错误，也不影响后续代码的执行

|  |
| --- |
| try{  尝试执行，可能会出现错误  }catch(err){  一旦try中出现错误，会被捕获，将错误放入到err  执行错误处理  } |

二.ES6

ECMAScript6 是JS的第6套标准规范

ES2015 ES2016 .. ES2021..

1.块级作用域

let声明的变量会进入暂时性死区(提升，不允许访问)，不允许重复声明

|  |
| --- |
| let在全局作用域下声明的变量不是全局变量 |

大括号之间的语句块就是块级作用域，例如：if、else、for、while...块级作用域下let和const声明的都是局部的，不允许外部访问的。

练习：计算1~100之间所有整数的和，使用let声明变量

2.参数增强

可以给参数设置默认值

|  |
| --- |
| function add(a,b,c=0){  b=b||0; //ES6之前的默认值设置方法  }  add(8000) |

3.箭头函数

简化了匿名函数的写法，但不等价于匿名函数

()=>{ }

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

如果箭头函数的函数体中只有一行代码，并且是return形式的，则可以进一步简化， sort( (a,b)=>a-b )

练习：使用之前的匿名函数创建函数getAvg，传递任意3个数字，返回平均值。改用箭头函数

4.模板字符串

简化了字符串的拼接

|  |
| --- |
| `模板字符串 ${JS表达式} ` |

三.Node.js概述

Node.js是JS在服务器端的运行环境，可以让JS实现后端开发

www.nodejs.org 官网

www.nodejs.cn 中文版

1.对比JS和Node.js

JS运行在客户端浏览器，有多款浏览器，存在代码兼容性问题；Node.js运行在服务器端，只有一种解释器，不存在代码兼容性问题。

两者都有相同的自定义对象和内置对象，不同的宿主对象

JS用于开发浏览器端交互效果，Node.js用于服务器端开发，例如：访问数据库、调用其它服务器..

2.运行JS

(1)脚本模式

node 拖拽文件 回车

(2)交互模式

node 回车 进入交互模式

两次 ctrl+c 或者 ctrl+d 退出交互模式

四.全局对象

1.global对象

用来检测是否为全局，JS下global名称为window

|  |
| --- |
| 一个js文件在Node.js下是在一个独立的作用域；  一个js文件在JS下是在全局作用域下 |

练习：编写脚本文件04\_global.js，声明变量和创建函数，使用global检测是否为全局的

2.console对象

console.log(1); //打印日志

console.info(2); //打印消息

console.warn(3); //打印警告

console.error(4); //打印错误

console.time() //开始计时

console.timeEnd() //结束计时

注意事项：开始计时和结束计时的值要保持一致

3.Buffer对象

缓冲区：是内存中用于临时存储数据的区域

|  |
| --- |
| var buf=Buffer.alloc(5,'abcde'); //创建Buffer，分配内存空间为5个字节，并存储字符，每个中文字符占3个字节  buf.toString() //将Buffer数据转为字符串 |

五.模块系统

模块：代表的是一个功能体，每个文件就是一个模块

module.exports：暴露的对象，默认是一个空对象

require()：是一个函数，用于引入其它的模块，回到暴露的对象

课后任务

(1)复习今天内容，整理思维导图

(2)练习：创建主模块(main.js)和功能模块(circle.js)，在主模块下创建两个函数，传递半径，返回周长和面积；最后在主模块下引入功能模块，并调用两个函数

(3)预习nodejs第2天

http://www.codece.com/archives/132