1.ES6

(1)块级作用域

let 声明的变量不存在变量的提升，在同一个作用域下不允许重复声明变量

|  |
| --- |
| 块级作用域：在大括号之间的区域就是块级作用域，例如if，else，while，do-while，for..  在块级作用域下使用let声明的变量是局部变量，不允许外部访问 |

练习：使用for循环计算1~100之间所有整数的和，变量使用let声明

(2)箭头函数

简化了匿名函数，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，则可以简化为

sort( (a,b)=>a-b )

练习：使用匿名函数创建函数 ，把匿名函数改为箭头函数，计算任意两个数字相加的和，并返回结果。

(3)模板字符串

解决了字符串的拼接

|  |
| --- |
| ` 之间的内容就是模板字符串 ${JS表达式} ` |

练习：假设从后端获取到了一条商品的数据，格式为对象，分别有商品的编号，标题，价格，最后打印出以下格式

编号：xxx

标题：xxx

价格：xxx.00

-------------------------------------------------------------------------

1.Nodejs概述

(1)对比JS和Nodejs

JS运行在客户端浏览器，存在多种解释器，存在代码兼容性问题；Nodejs运行在服务器端，只有一种解释器，不存在兼容性问题

两者都有自定义对象、内置(ES)对象，不同的宿主对象

JS用于网页中交互功能开发，Nodejs用于服务器端操作，例如数据库访问，文件的操作，其它服务器的调用。

(2)下载网址

www.nodejs.org 官网

www.nodejs.cn 中文镜像

查看版本 node -v

node 回车 进入交互模式

两次ctrl+c 退出交互模式

2.全局对象

(1)global对象

全局变量和全局函数都是全局对象global下的，可以通过global访问

在交互模式下所处是全局作用域，可以使用global访问变量和函数；

在脚本模式下不是全局作用域，不能使用global访问变量和函数，目的是为了防止全局污染。

练习：在脚本文件中声明变量a，创建函数fn，查看是否为全局的。

JS全局对象是window，在JS下脚本文件属于全局作用域，存在全局污染。

(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

开始计时和结束计时两者的参数要保持一致

练习：使用while循环计算1~10000之间所有整数的和，统计耗时。

(3)process对象

进程对象，任何应用程序运行都会代表相应的进程

process.arch 查看当前CPU架构

process.platform 查看当前操作系统

process.version 查看当前nodejs版本号

process.pid 查看当前的进程编号

process.kill(编号) 结束指定编号的进程

(4)Buffer对象

Buffer：缓冲区，缓冲器

buffer数据，是内存中一块存储的临时数据，常用于存储网络传输时的资源。

|  |
| --- |
| let buf=Buffer.alloc(5, 'abcde'); //创建buffer为5个字节，并存储数据  buf.toString() / String(buf) 转为字符串 |

3.模块系统

每个模块都是一个独立的功能体

nodejs模块分为自定义模块、第三方模块、核心模块

每个文件就是一个模块，创建一个文件就是创建一个模块

|  |
| --- |
| 每个文件都是一个模块作用域，里边的变量和函数外部都是无法访问  require(): 用于引入其它的模块，得到的是对方导出的对象  module: 当前模块的对象  module.exports: 当前模块导出的内容，默认是一个对象，导出的内容都往这个对象下添加 |

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)，在功能模块下创建两个函数，传递参数为半径，计算圆的周长和面积；在主模块下引入功能模块，调用函数（传递半径）

(3)预习nodejs第2天内容