1、参数增强。

2、块级作用域（在大括号之间的取余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声明的变量都是局部变量，不允许外部访问）。

①声明的变量不存在变量提升。

②同一作用域下不允许重复声明同一变量。

3、箭头函数。

定义匿名函数：

var fn = function(){ }

(function(){ })

匿名函数自调用：(function(){})()

简化了匿名函数，不等价于匿名函数。

arr.sort(function (a, b) { return a - b; })

arr.sort( (a,b) => {return a-b})

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

如果箭头函数中只有一行代码而且是return，可以简化为 (a,b)=>a-b

4、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}

NodeJS

1、对比Nodejs和JavaScript

①JS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，只存在一种解释器，不存在兼容性问题

②两者都有自定义对象、内置对象（ES对象）、不同的宿主对象

③JS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，其他服务器的调用。

2、node 回车，进入交互模式，退出：ctrl+c 两次

全局对象

①global对象：全局对象下，函数是全局函数，变量是全局变量

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于，不能使用global访问变量和函数，目的是为了防止全局污染。

JS的全局对象是window，在JS下脚本文件属于全局作用域，存在全局污染

②console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("aa") //开始计时 两者参数保持一致

console.timeEnd("aa") //结束计时 两者参数保持一致

③process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 查看当前进程编号

process.kill(进程编号) 结束指定编号的进程

④Buffer对象：

Buffer缓冲器/缓冲区。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。

// 创建Buffer

let buf = Buffer.alloc(9,'abc小小')//(5个字节，存储的数据) 代表分配5个字节的空间

console.log(buf) //<Buffer 61 62 63 64 65> 16进制

// console.log(String(buf)) //buffer数据转字符串

console.log(buf.toString()) //buffer数据转字符串

模块系统：

每个模块都是一个独立的功能体。

在NodeJS下，模块分为三大模块，每个文件就是一个模块，创建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的，

require() 用于引入其他的模块，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，默认是一个空对象，导出的内容都往这个对象下添加。

主模块main.js 功能模块circle.js 在功能模块下创建两个函数，传输参数为半径，计算圆的周长面积，在主模块下引入功能模块，调用函数。传递半径

自定义模块：

第三方模块：

核心模块：