复习

循环条件，循环体

while(循环条件){ 循环体 }

do{ 循环体 }while(循环条件);

for(初始值;循环条件;增量){ 循环体 }

continue和break

跳过，继续下一次

结束循环

|  |
| --- |
| var i=0,sum=0;  do{  i++; //5  if(i%2===0) continue;  if(i%5===0) break;  sum+=i;//1+3  }while(i<10);  console.log(sum); |

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)第三方的库和框架

(10)实用的项目

1.函数

Number/parseInt/parseFloat/isNaN...

函数分为系统函数和自定义函数

function: 功能体，函数(方法)，接收若干个数据，返回处理的结果。—— 用于封装多次执行的代码

(1)创建普通函数

|  |
| --- |
| function 函数名称(){  函数体; //封装的代码  } |

调用

函数名称()

练习: 创建函数，封装计算两个数字(2,3)相加的和，调用3次

练习: 创建函数，封装计算1~100之间所有整数的和，调用3次

(2)创建带有参数的函数

|  |
| --- |
| function 函数名称(参数列表){ //用于接收传递的数据  函数体;  } |

调用

函数名称(参数列表) //实际要传递的数据

参数: 创建函数时的参数称为形参，调用函数时的参数称为实参，实参会赋值给形参；多个参数之间用逗号隔开，实参的值可以传递0个或者多个，如果未赋值，形参的值为undefined。

练习: 创建函数，计算1~任意1个数字之间所有整数的和，调用多次

练习: 创建函数，计算任意两个年份之间的闰年个数，调用多次

(3)创建带有返回值的函数

|  |
| --- |
| function 函数名称(参数列表){  函数体;  return 值; //返回值，函数调用后产出的结果  } |

调用

函数名称(参数列表)

注意事项: 如果函数中没有return，或者return后没有任何的值，则返回undefined；return执行后，后续的代码不会再执行。

练习: 创建函数，传递两个参数，返回任意两个数字的最大值。

练习: 创建函数，传递三个参数，返回任意三个数字的最大值。

练习: 创建函数，传递1个参数，根据订单的状态码返回对应的中文状态

1-等待付款 2-等待发货 3-运输中 4-已签收 5-已取消 其它-无法追踪

|  |
| --- |
| 对比break和return  break用于结束switch语句和循环  return 用于结束函数，函数后续代码不再执行 |

练习: 创建函数，传递1个参数，查看任意一个年份是否为闰年，如果是闰年返回true，否则返回false

2.变量作用域

(1)作用域

变量和函数的可访问范围，分为两种

全局作用域: 在函数外使用var关键字声明的变量，可以在任意位置访问到

函数作用域: 在函数内使用var关键字声明的变量，只能在函数内部访问到

(2)变量声明提升

JS程序在执行前，会将var声明的变量提升到所在作用域的最前边，但赋值还是在原来的位置

3.函数的作用域

函数的可访问范围，和变量的作用域一样，也分为全局函数和局部函数；全局函数可以在任意位置调用，局部函数只能在所在的函数作用域调用。

函数提升

和变量提升一样，JS程序在执行前，会将使用function关键字创建的函数提升到所在作用域的最前边。

4.递归

在函数内部调用自身，本身是一个死循环

用法：需要有结束条件，结合着return

斐波那契数列

1 1 2 3 5 8 13 21 34 55

第1，第2项值是1，从第3项开始，每项的值是前两项相加的和。

课后任务

(1)复习今天内容，整理思维导图

(2)练习

使用函数计算斐波那契数列的第n项

(3)预习**匿名函数**、对象