复习

循环 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; }

isNaN

break/continue

|  |
| --- |
| var i=0,sum=0;  do{  i++;  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)常量和变量

(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)第三方框架、库

(10)实用的项目

1.函数

isNaN(数据)/parseInt/parseFloat/Number/prompt...

函数分为系统函数和自定义函数

function: 功能体，函数(方法)，可以接受若干个数据，返回处理的结果。用于封装反复执行的代码——饺子机

(1)创建普通函数

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

调用

函数名称()

练习：创建函数，封装两个数字相加的和，调用多次。

练习：创建函数，封装计算1~100之间所有整数的和并打印，调用多次。

(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) //实际传递的数据

参数: 创建函数时的参数称为形参，调用时参数称为实参，实参会赋值给形参，多个参数之间用逗号隔开。参数可以有0个或者多个，如果形参未被赋值，值为undefined

练习：创建函数，封装计算1~任意数字之间所有整数的和，调用多次。

练习: 创建函数，封装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，函数执行完返回的结果  } |

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中没有加任何值，返回undefined；执行完return后的代码不会再执行。

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，传递订单的状态码，返回对应的汉字状态

1-待付款 2-待发货 3-运输中 4-已签收 5-已取消 其它-无法追踪

练习：创建函数，传递任意一个年份，返回是否为闰年 是->true 不是->false

练习：创建函数，传递任意一个年份，如果是闰年返回366，否则返回365

练习：计算1~任意数字之间所有整数阶乘的和

步骤1：计算任意数字的阶乘。 n~1之间所有整数的乘积

1+ 2+ 3+ 4 + 5

4!=4\*3\*2\*1

5!=5\*4\*3\*2\*1

**对比break和return用法**

break用于结束switch语句，还可以用于提前结束循环。

return 用于函数中返回值，可以结束函数后续代码执行。

2.作用域

变量或者函数的可访问范围，分为两种

全局作用域：在全局使用var声明的变量，可以在任意范围访问到

函数作用域：在函数使用var声明的变量，只能在函数内部访问到

注意事项：在函数内不使用var声明的变量，是全局变量，可以在函数外部访问——不推荐这样写。

(1)变量提升

JS程序执行前，会将使用var关键字声明的变量提升到**所在作用域**的最前边，但是赋值还是在原来的位置。

(2)函数的作用域

函数的可访问范围，在全局作用域下创建的函数可以在任意位置访问；在函数作用域下创建的函数只能在函数内部访问。

(3)函数提升

和变量提升一样，JS程序执行前，会把function关键字创建的函数提升到所在作用域的最前边，在调用的时候才会执行函数体中的代码

课后任务

(1)复习今天内容，整理思维导图

(2)课后练习

1 1 2 3 5 8 13 21 34 ...

使用循环，结合函数，计算斐波那契数列 的第n项

fib(20)

(3)预习对象