学习一门编程语言步骤

(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，框架

(10) 实用的项目

复习

循环

两要素：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 }

break

continue

|  |
| --- |
| var i=0,sum=0;  do{  i++;  if(i%2==0){  continue;  }  If(i%5==0){  break;  }  sum+=i; } while(i<10);  console.log(sum); //4 |

计算1~100之间的和，当和大于4000 ，结束循环，打印当前的和以及i的值。

1. 函数

Number()/parselnt()/parseFloat()/alert()/prompt()

JS中函数分为系统函数和自定义函数

自定义函数

function: 功能体，函数，用于封装重复执行的代码 可以接收若干个数据，返回处理的结果。。——饺子机

1. 创建普通函数

|  |
| --- |
| function 函数名称(){  函数体; //封装的重复执行的代码  } |

调用

练习；使用函数封装1+2的计算结果 ，调用 3次

函数名称()

(2创建带有参数的函数

|  |
| --- |
| function 函数名称(参数列表){ //形参 -> 形式上的参数  函数体;  } |

调用

函数名称(参数列表) //实参-> 传递的数据

参数列表：可以是0个或者多个，之间用逗号隔开，创建时的参数成为形参，调用时的参数 成为实参，实参会赋值给形参，

练习：创建函数，传递1个参数。计算1~任意数字之间的和

练习：创建函数getRun,传递2个参数，计算任意两个年份之间的闰年个数

创建函数时的参数叫形参，也称为形式上参数，用于接收数据。

调用函数时的参数叫实参，这才是实际传递的数据

形参可以理解为是一个变量，只是未赋值。默认是undefined

注意：形参就相当于是声明未赋值的变量，实参的个数如果少于形参

则未赋值的形参就是undefined

练习：创建函数getSum,传递1个参数，计算1~任意数字之间所有数字的和

练习：创建函数getRun,传递2个参数，计算任意两个年份之间的闰年个数

（3）带有返回值的函数

|  |
| --- |
| function函数名称(参数列表){  函数体  **return 返回值;**  } |

调用

函数名称( 参数列表 );

return 表示函数执行后的返回结果

注意：使用return 目的 为了保存函数的调用结果。

return后的所有代码都无法执行，return 有终止函数的执行。

如果函数中没有return,或者return后没有加任何的值，则返回undefined.

练习：创建函数，getMax，返回任意两个数字的最大值。

练习：创建函数getMax，返回任意3个数字的最大值。

练习：创建函数，传递1个参数（状态码）。根据订单状态码，返回对应的中文。

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

function getStatus(n){

switch(n){

case 1:

return '待付款';

break;

case 2:

return '待发货';

break;

case 3:

return '运输中';

break;

case 4:

return '已签收';

break;

case 5:

return '已取消';

break;

default:

return '无法追踪';

}

}

var s=getStatus(5);

console.log(s);

练习：创建函数isRun,传递1个参数（年份），返回对应的天数；是->true 否->false

对比return,break和continue

return用于函数中 ，可以终止函数执行，并返回结果

break用于终止循环，终止switch语句

continue用于跳过某一次循环，还会继续下一次循环

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

（局部）函数作用域：在函数中使用var使用关键字声明的变量，只能在函数的内部 访问

全局作用域：在函数外部使用var声明的变量，可以在任意的位置访问

注意：在函数内使用var声明的变量是局部变量，不使用var声明的变量是全局变量

（2）变量声明提升

JS程序执行前，会将使用var声明的变量提升到所在作用域的最前边，赋值还是在原来 的位置。

在局部里不加var，，调用 会在全局中找var

3.函数的作用域

函数的可访问范围，也分为全局作用域和函数作用域

（1）函数声明提升

和变量声明提升一样，JS程序执行前，会将function

创建的函数提升到所在作用域的最前边，调用 可以在任意合法的位置

4.递归调用

在函数的内部调用自身

递归要有结束的条件，结合return来使用

练习：使用递归计算1~任意数字之间的和

斐波那契数列

1 1 2 3 5 8 13 21 34

练习：

创建函数，计算斐波那契数列的第n项：使用递归和普通函数两种方法完成

（3）预习匿名函数和对象