课后拓展：创建函数getSum，传递任意一个数字，返回1~任意数字之间所有整数阶乘的和

1!+2!+3!+4!+5!

一.函数的作用域

全局函数：在全局作用域下创建的函数，可以在任意的作用域下调用

局部函数：在函数作用域下创建的函数，只能在所在的函数作用域下调用

作用域链：多个作用域嵌套形成的结构，在查找变量的时候，首先会到当前作用域下查找，如果当前没有会不断的往上一级作用域查找

函数提升：程序执行前，会将创建的函数提升到所在作用域的最前边。

二.递归

在函数的内部调用自身这个函数

递归默认是一个死循环，使用不当造成内存泄漏

内存泄漏：操作系统分配给JS空间被占用，无法存储新的数据

使用：在递归之前有边界条件，结合着return

斐波那契数列：第1、2项固定是1，从第3项开始每项的值是前两项相加的和；使用递归的算法计算任意一项的值

fib(10)

三.匿名函数

function (){ }

1.创建函数

|  |
| --- |
| 函数表达式  var fun=function(){  } |
| 函数声明  function fn(){  } |

对比**函数名称()**和**函数名称**的区别

函数名称() 调用函数，执行函数体中的代码

函数名称 本质上就是一个变量，保存了函数类型数据

对比**函数声明**和**函数表达式**创建函数的区别

函数声明创建的函数存在函数的**整体提升**

函数表达式创建的函数只是存在**变量声明提升**

2.匿名函数自调用

全局污染：全局变量的出现产生影响，任何作用域都可以访问全局变量

|  |
| --- |
| (function(){  //在一个函数作用域下，声明的变量是局部变量，可以防止全局污染  })(); |

3.回调函数

将函数作为参数传递

|  |
| --- |
| function tao(madai){  madai(); //调用传递的回调函数  }  function xin(){}  tao( xin );  tao( function(){ } ); |

四.系统函数

就是JS提供的函数，可以直接调用.

isNaN() 检测一个字符串中是否含有非数字，会隐式转换为数值查看是否为NaN， 是->true 不是->false

练习：声明变量保存用户输入的值，如果输入的值含有非数字，打印‘非法的数量’

eval() 执行字符串中的表达式

五.对象

对象是一组属性和方法的集合

一部手机，属性有颜色、品牌、CPU、内存... 方法有打电话、发短信、听音乐、看视频、玩游戏、办公...

涛哥的煎饼机，属性有大小、材质、颜色.. 方法有摊煎饼...

万物皆对象

1.JS中的对象

内置对象，JS提供的对象

自定义对象，自己创建的对象

宿主对象，根据不同的执行环境划分

2.创建自定义对象

(1)对象字面量

(2)内置构造函数

(3)自定义构造函数

3.对象字面量

{属性名:属性值, 属性名:属性值...}

属性名的引号可以省略，如果含有特殊字符必须添加

练习：创建一个员工对象，包含有员工的编号，姓名，性别，工资

4.属性的访问

对象.属性名 对象['属性名']

如果访问的属性不存在，则返回undefined

练习：创建一个图书对象，包含的属性有编号，标题，作者，价格；修改价格属性，添加出版社属性；最后打印输出对象

5.内置构造函数

new Object()

创建一个空对象{}，需要单独添加属性

练习：创建班级的对象，包含的属性有编号，中心名称，学员人数。

6.遍历属性

依次访问对象中每个属性

|  |
| --- |
| for(var k in 对象){  k 每个属性名  对象[k] 属性值  } |

课后任务

(1)复习今天内容，整理思维导图

(2)练习：字面量创建对象，包含一组成绩，遍历对象得到每个成绩，计算出总成绩和平均成绩

(3)预习JS第7天 数组

http://www.codece.com/archives/98