**05**

**复习：**

do-while循环结构:先执行一次，在进行判断

do{

循环体

}while(布尔表达式); //布尔表达式范围更广泛

for循环

for(初始值；循环条件；自加减){循环体}

强制终止循环break，跳过本次循环continue

循环的嵌套：while do-while for三者都可以进行嵌套

**今日学习目标**

函数相关（重点）

变量的作用域（重点）

函数的作用域（重点）

递规调用（掌握）

**1.自定义函数**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| alert(); | prompt(); | parseInt(); | parseFloat(); | toString(); | typeof(); |

以上都属于js提供的函数：

（1）函数（function）也可以成为方法（method），具有特定的功能。

函数是一段预定义好的代码段，并且可以反复执行。

预定义：提前定义好的，不会马上执行

代码段：可以包含多条件可执行的语句。

反复执行：允许被调用多次。

（2）定义函数和调用函数：

自定义函数就是用来封装代码段。

**①普通函数的定义（声明）和调用。**

定义：

|  |
| --- |
| function函数的名称(){//命名规则和变量一致  //函数体，就是可执行的代码段  } |

调用：

在任意的JavaScript中合法的位置，包括在另一个函数体中，通过【函数名称()】形式进行调用。

练习:声明一个函数calc，在函数体中实现两个数字相加，并打印出结果。

**②带有参数的函数声明和调用**

声明：

|  |
| --- |
| function函数名称(参数列表){  //函数体  } |

参数列表：可以是0个或者多个，多个之间用逗号隔开，声明的参数称为形参（形式上的参数），只是负责占位，函数调用的时候，形参会被传入的值替换

调用

函数名称（参数列表）

注意：尽量按照声明的函数格式进行调用（有几个形参，调用的时候，就提供几个参数），调用时所传递的参数称为实参。

练习1：声明函数sayHi，函数体中打印一段个人介绍，不同的人要求介绍不同的内容：

练习2：定义一个函数叫做getsum，要求传递一个参数，或者从1~100参数之间的和

**③带有返回值的函数声明和调用**。

声明：

|  |
| --- |
| function函数名称（形参列表）{  //函数体  //return值：return关键字出现后，立马终止执行函数体后面的代码，然后把结果带出来。  } |

调用：

函数名称（实参列表）

没有return，或者return后面没有值的情况下，结果返回undefined

练习：声明函数getMax，传递两个参数......

对比break，continue和return用法

break终止循环的进行

continue终止某一次的循环，还会继续下一次循环

return终止函数的执行

**2.变量的作用域**

（1）什么是作用域

作用域表示变量或者函数的可访问范围。

js中的作用域，分为两种：

①局部（函数）作用域：只能在某个函数的内部生效。

②全局作用域：在任何的位置都可以生效。

（2）变量的声明提升

js程序在执行前，声明的变量，都会预加载到作用域程序的最前端，但是赋值还是在原来的位置。函数的内部同样也存在变量的提升。

（3）按值传递

按值传递：指将实参的值拷贝一份然后传递给形参。

如果传递的值是原始数据类型（5个基本类型），则在方法内修改参数变量，不会影响到外部的参数变量。

**3.函数的作用域**

和变量的作用域类似，函数的作用域也分为局部作用域和全局作用域。

局部作用域：在函数内部声明的函数，只能在函数内部调用

全局作用域：可以在任意位置调用

函数在程序执行前，也存在把函数名提升到前边，所以在任何位置都可以定义函数

**对比函数名称和函数的调用**

function myfun(){

return 123;

}

函数的调用myfun()得到的结果是函数的返回值——>123

函数名称myfun() 得到的是函数结构

**4.递归**

就是在函数的内部调用自身

递归的使用：需要有一个边界条件--->递归的结束

课后练习：使用递归计算菲波那切数列的第n项。

1 1 2 3 5 8 13 21 34 55 89 144

**06**

**今日学习目标**

匿名函数(重点)

全局函数

对象(重点)

**1.arguments对象**

arguments.callee, 只能在函数内部使用，用于访问当前函数的函数名称。

**2.匿名函数**

（1）使用匿名函数创建函数

声明

var 函数名称 = function(){ }

声明的变量名称就是函数名称

调用

函数名称();

对比匿名函数和普通函数的区别

普通函数存在函数提升，匿名函数不存在提升，所有在使用匿名函数的时候，需要先声明再调用

（2）回调函数

把匿名函数以实参的形式传递给形参。

（3）匿名函数的调用

(function(形参列表){

//函数体

})(实参列表);

使用匿名函数自调用可以创建一个局部作用域。

**3.全局函数**

ECMAScript提供一组全局函数

encodeURI()对一个url进行编码

decodeURI()对编码后的url进行解码

eval()执行一个字符串表达式（创建网页计算器）

isNaN()判断一个值是否为NaN，是->true 不是->false

isFinite() 判断是否为有限值，是有限值-->true 不是有限值-->false

parseInt() 将一个值转成整型

parseFloat() 将一个值转换成浮点型

练习：使用eval()函数，结合弹出提示框，来显示计算器功能

**4.对象**

js中的数据类型分为

值类型：存储在栈内存中

引用类型：也称为对象型，存储在堆内存中。

对象是由一组属性(property)和方法(method)的集合

(1)js中的对象

内置对象(ES原声对象)，宿主对象，自定义

（2）自定义对象的创建方式

①对象字面量(直接量)

②通过new Object()

③function对象模板

**5.对象自变量创建对象**

使用大括号创建对象

属性名和属性值之间用冒号隔开

多组属性之间用逗号隔开

注意：如果对象中包含（ - ， 空格）需要用引号包来

**6.通过new Object创建对象**

new Object创建的是一个空的对象，括号中没有值，括号可以去掉。

需要单独往对象里面添加属性

laptop.属性名 = '属性值'

laptop['属性名'] = '属性值'

获取对象中的属性laptop.color或者laptop['color']

注意：如果访问的属性不存在，返回结果是undefined

练习：创建分页对象page，对象中含有总数量（total），每页大小（size），当前页（pno）

上一页（prev:'<'），下一页(next:'>')

**7.遍历（枚举）属性**

for(var proName in emp){

//proName表示遍历的所有属性名

//emp要遍历的对象

//要通过属性名proName访问属性值-->emp[proName]

}

注意：使用for-in但只能遍历对象自定义属性，不能遍历预定义(对象本身自带的)的属性值.

练习：创建汽车对象

**8.检测对象中是否含有属性**

①'属性' in 对象 ， 属性名要加引号

②对象hasOwnProperty(属性)

③对象，属性 === undefined //不存在为true，存在为false

**（以上结果全为布尔型的值）**

**9.对象中的方法**

var person = { sayHi:function(){ } }

在成员方法中访问成员属性 this.name

课后练习：

创建一个圆对象（circle），添加属性半径和圆周率，添加方法计算圆的周长和面积。

getLength getArea

复习:

使用匿名函数声明函数

var fn = function(){} fn()

回调函数

把函数名（匿名函数）以实参的形式传递给形参

getSum(getNum1,getNum2)

getSum(function(){ }, function(){ })

匿名函数的自调用创建一个局部函数

(function(){ })(); // 防止污染全局变量

对象:

万物皆是对象：区别对象和类的关系，是一组属性和方法的集合。

字面量（直接量） var obj ={name: 'tom',age: 20}