1.函数

Number()/parseInt()/parseFloat()/alert()/prompt()...

JS中函数分为系统函数和自定义函数

函数，是一个功能体，需要提供若干个数据，返回处理的结果，用于将重复执行的代码封装起来

(1)创建函数

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

调用

函数名称()

会执行函数体中的代码

练习：创建函数getSum，封装计算1~100之间所有整数的和，并调用多次

(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)

参数：创建函数时的参数称为形参，用于接收数据，调用函数时的参数称为实参，实际要传递的数据，实参会赋值给形参，多个参数之间用逗号隔开，实参的数量可以和形参的数量不等，如果形参未赋值为undefined。

练习：创建函数getSum，传递任意一个数字，计算1~该数字之间所有整数的和，调用多次

练习：创建函数getRun，传递任意两个年份，计算两个年份之间的闰年个数，调用多次

(3)带有返回值的函数

创建函数

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

调用

函数名称()

return用于在函数调用后返回的结果，如果函数中没有return或者return后没有加任何值，则返回的结果为undefined；一旦return执行后，就会结束函数的调用，就不再执行函数体中任何代码。

练习：创建函数getMax1，传递两个数字，返回最大值；调用多次

练习：创建函数getMax2，传递两三数字，返回最大值

练习：创建函数getStatus，传递状态码，返回对应的汉字状态，使用switch-case

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-case语句或者循环  return用于函数中，结束函数的调用 |

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

2.变量的作用域

变量的可访问范围

**全局作用域：**在函数外使用var声明的变量，可以在任何的作用域下访问——**全局变量**

**函数作用域：**在函数内使用var声明的变量，只能在当前的函数下访问到——**局部变量**

函数中不加var声明的变量是全局变量，不推荐这种方式，后期严格模式下会报错。

变量提升：JS程序执行前，会将var声明的变量提升到所在作用域的最前边，只是提升声明，赋值不提升。

3.函数的作用域

全局函数：在全局作用域下创建的函数，可以在任意作用域下调用

局部函数：在函数中创建的函数，只能在当前函数作用域下调用

函数提升：JS程序执行前，会将function关键字创建的函数提升到所在作用域的最前边，可以先调用再创建。

4.递归

在函数内调用自身这个函数，本身是一个死循环

使用递归：要有结束的条件，结合return

斐波那契数列：第1项和第2项是1，从第3项开始，每项的值是前两项相加的和

1 1 2 3 5 8 13 ...

课后任务

(1)复习今天内容，整理思维导图

(2)练习：创建函数，传递一个参数(为任意一项)，在函数中返回该项对应的值

循环 递归

(3)预习js第6天内容 http://www.codece.com/archives/95