day4-javascript函数上

笔记本： 凡尘：二阶段
创建时间： 2017/7/23 16:58 **更新时间：** 2017/9/27 17:31
作者： 437389128@qq.com

回顾

1、怎么让数字保留n位小数？
数字.toFixed(n);
var num = 123.567; num.toFixed(2);

2、三元表达式怎么写？
表达式（条件） ? 值1 : 值2;

3、将字符串转化为数字的方法有哪些？ 数学方法有哪些？（Math）
var str = '123.45';
parseInt(str);
parseFloat(str);
Number(str);

Math数学方法
随机数：Math.random() 0 - 1 3 - 5 0-2 + 3
向上取整：Math.ceil();
向下取整：Math.floor();
四舍五入：Math.round();
次方函数：Math.pow(x,y); x的y次方

4、name 和 top 不要轻易使用 他们都是系统自带的变量

1、什么是函数

生活场景

那些年我们洗过的衣服

全自动洗衣机




生活越来越方便了

工具的好处在哪里？
1 可重复使用
2 隐藏处理细节
3 控制代码的执行时机

怎么定义函数（功能）？（重）

0、定义函数使用关键字 function

1、关键字定义方式：

```
function 函数名(参数1,参数2...){
    功能//函数体
}

function showInfo(name,age,hobby){
    alert('大家好，我叫' + name + ',今年' + age + '岁，我喜欢'+hobby);
}
```

2、字面量定义方式：

```
var 函数名 = function(参数1,参数2...){
    //函数体
}

var showInfo = function(name,age,hobby){
```

```
        alert('大家好，我叫' + name + ',今年' + age + '岁，我喜欢'+hobby);
    }
```

说明：函数名就是一个变量

3、Function构造函数(很少用)

var 函数名 = new Function(参数,函数体);

var showAge = new Function('age','alert(age)');

暂时不讲，可以自己研究

怎么调用函数？（重）

```
定义一个打印功能
var print = function(){
    //需要打印的信息
}
```

如果()跟在一个函数名后面 代表立即执行这个功能

为什么说JS是基于事件驱动的语言？

事件即是函数的执行触发条件。

所有的程序，都要依靠事件的产生来驱动。

常用的事件有什么？

鼠标事件
单击事件 onclick
双击事件 dblclick
鼠标划入 onmouseover
鼠标划出 onmouseout
鼠标滚轮（滚动，单击） onmousewheel
鼠标的移动 onmousemove

键盘
按下某个键不松开 onkeydown
抬起某个键 onkeyup
按下并抬起来 onkeypress

输入框
获取焦点 onfocus
失去焦点 onblur
改变输入框的内容 onchange

BODY标签
加载完成 onload

函数有什么用？（理解）

将重复的代码放到一个函数中 ，可以做到：

1、减少代码的编写（代码重用）

2、便于修改和控制（维护）

3、需要的时候再使用

函数的特点

1、函数自己无法运行，调用时才会执行函数体内的代码块

2、可以多次使用

参数（重）

1、什么是参数？
在定义函数的时候，在()里面可以定义一些变量，用于接收调用的时候传过来的数据

函数的()中定义的变量及调用时传入的数据都是参数

参数：
1、函数的()中定义的变量
2、调用函数时传入的数据

2、为什么要定义参数？
1、让函数变得更加灵活（实现函数的多功能）
2、实现函数的作用

3、怎么定义参数（形参）？(形式参数)

1、在函数定义的时候括号里面写的变量
2、形参可以定义多个
3、每个形参之间用逗号隔开 ，
4、形参其实就是个变量（遵循变量规则）

function print(arg1,arg2,arg3...){}

4、怎么传入参数（实参）？（实际参数）
1、调用函数时，写在括号里的值（变量）就称之为 实参
2、多个实参之间使用逗号隔开

参数解读（重）

1、形参和实参什么关系？
a、形参和实参之间是一一对应的关系
b、如果对应的形参没有传值，那么值是undefined

2、javascript函数的参数与大多数其他语言的函数的参数有所不同。函数不介意传递进来多少个参数，也不在乎传进来的参数是什么数据类型，甚至可以不传参数。

参数初体验

1、写一个功能，传入一个数字，计算数字的平方。

2、定义一个加法功能，传入一个数字，计算从1到该数字整数的和。

3、定义一个功能，传入两个数字m,n(m<n)，随机生成m-n之间的随机数

3、定义一个功能，传入两个数字m,n，如果m<n,随机生成m-n之间的随机数,如果n<m，随机生成n-m之间的随机数

编写一个函数，计算两个数字的和差积商

编写一个函数，计算三个数字的大小，按从小到大的顺序输出。

编写函数，判断一个字符串的内容是不是纯数字

编写一个函数，在页面上输出一个N行M列的表格，表格内容填充1~100的随机数字

```
function printTable(n,m){
    //.....
}

printTable(5,6);
```

函数如何将处理结果返回？

```
1、什么是return？
      函数的返回值

      1、 调用者期望拿到的数据
      2、 一个功能的返回值
      3、 如果函数没有写return，默认返回undefined
      4、 函数中遇到return就直接返回，后面的代码将不再执行（类比break）
      5、 只有函数中才有return
      6、 只能return一个结果

continue只能出现在循环中
break只能出现在switch或者循环中
return只能出现函数中

2、为什么要return？
      ATM机取500钱，银行卡扣钱之后，你期望ATM吐出来500元
      一个加法功能，调用者期望函数返回计算的结果（如果调用者期望返回，函数一定要返回）

      当调用者希望得到数据时，函数就要返回结果
      调用者拿到返回值做什么事情是不确定的

3、什么时候使用return？
      1、需要返回结果
      2、不需要程序继续往下执行
```

```
/*
    逐个排除
*/
function judge(year){
    //判断是不是数字
    if(Number(year) != year){
        console.log('不是数字');
        return;
    }
    //判断是不是 > 0
    if(year <= 0 || parseInt(year) != year){
        console.log('sorry,wrong!');
        return;
    }
    console.log('恭喜，输入的是合法的年份');
}
```

4、有什么用？

1、 返回运算的结果
2、 中断代码执行

return 关键字 ，用来让函数返回结果

如果函数没有return，那么执行完后，将返回什么？？
undefined

return 要返回的内容;

练习：
编写一个函数，生成4位数字的验证码

求m-n之间数字的和

求圆的面积

编写函数digit(num, k)，函数功能是：求整数num从右边开始的第k位数字的值，如果num位数不足k位则返回0。

编写函数计算一个数字的长度

综合练习：
编写一个函数，计算任意两个数字之间所能组成的奇数个数，数字必须是个位数。
比如：计算0~3之间能组成的奇数是： 01、03、13、21、23、31

某个公司采用公用电话传递数据，数据是四位的整数，在传递过程中是加密的，
加密规则如下：每位数字都加上5,然后用除以10的余数代替该数字，再将第一位和第四位交换，
第二位和第三位交换，请编写一个函数，传入原文，输出密文

扩展：

变量的声明提升

浏览器在执行代码之前，会对所有的声明语句，进行提升（把声明代码放到整个代码的最前面）然后再按顺序执行代码

1、js代码的编译和执行

2、变量的提升：变量声明和函数声明从他们代码中出现的位置被移动到执行环境的顶部，这个过程就叫做提升 只有声明操作会被提升，赋值和逻辑操作会被留在原地等待执行

Js编译器会把变量声明看成两个部分分别是声明操作(var a)和赋值操作(a=2)

```
<script type="text/javascript">
    /*
        预编译  执行

        预编译过程做的事情：

            把var 和 function定义的变量提升到script的最上方

            赋值语句是不会被提升的(哪怕=后面是一个function)

        变量提升

        注意：
            使用变量形式定义的函数只能在赋值语句之后调用

            test();
            // TypeError: test is not a function  test不是一个函数

            var test = function(){

            }

            test();
    */
```

函数声明、函数表达式、匿名函数的不同

函数声明:在执行的时候函数声明会进行'函数声明的提升'

函数表达式:
1、函数表达式是自上而下的执行函数表达式
2、函数表达式可以在后面加括号立即掉用该函数，而函数声明则不会

例子:

```
fnName();
function fnName(){
    ...代码块
}
//因为函数进行了函数声明提升，所以可以在函数定义之前进行调用

fnName();
var fnName = function(){
```

```
        ...代码块
    }
    //报错，因为不会进行函数的声明提升，所以逐行进行解析的时候没有执行到下面的函数


    函数表达式加括号可以立即执行
    var fnName =function(){
        ....代码块
    }()
```

面试题

```
var n = 10;
function test(){
    console.log(n);
    var n = 20;
    console.log(n);
}
test();
变量的声明提升原则： 提升至该作用域的最前面
```

```
1、
    href="javascript:;"
    href="javascript:void(0);"

    1、代表是一个js操作,不会发生页面跳转
    2、能有有效防止文字被选中

2、  元素.innerHTML = '123'；将元素的内容改为 ‘123’

   var content = 元素.innerHTML 获取元素的内容
```

预习

```
0、变量提升

1、arguments
```

2、作用域

3、全局作用域和局部作用域

4、全局变量局部变量

5、递归函数