正课:

1. String:

2. RegExp:

3. Math:

4. Date:

1. String:

切割: 将一个字符串，按指定分隔符，切割为多段子字符串

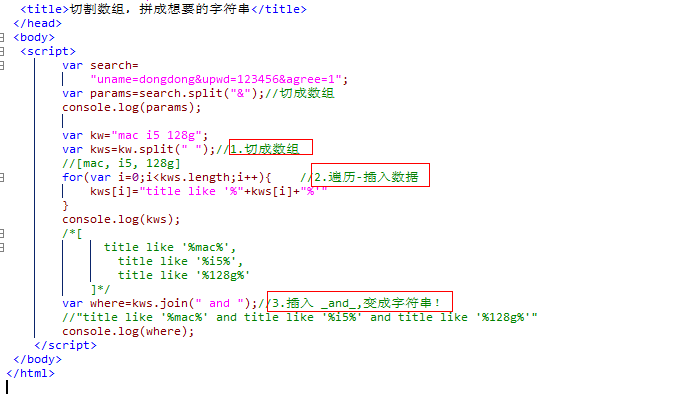
简单切割: 切割符是固定的

var arr=str.split("切割符")

强调: 切割后的结果中，不包含切割符

复杂切割: 切割符是变化的

var arr=str.split(/正则/)





2. RegExp:正则表达式封装函数

什么是: 封装一条正则表达式，并提供使用正则进行验证和查找的API

何时: 只要在程序中存储一条正则，并使用正则进行验证和查找

如何:

创建: 2种:

1. 如果正则表达式是固定不变的: 用//

var reg=/正则/ig

2. 如果正则表达式是动态变化的:

var reg=new RegExp("正则表达式", "ig")

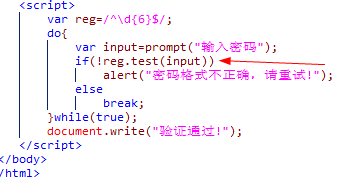
API:

验证字符串的格式是否符合要求:

var bool=reg.test(str)

用reg检查 str 是否符合格式要求

返回bool:



问题: 默认，只要str中部分内容满足reg要求，就返回true。

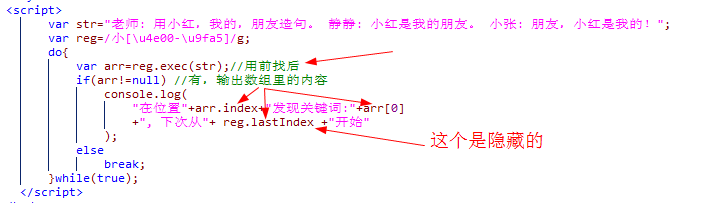
解决: 凡是验证，必须前加^，后加$

即查找所有关键词的内容，又查找关键词的位置

var arr=reg.exec(str)

在str中查找下一个reg匹配的关键词的位置和内容

返回: arr的内容和match不加g时返回的arr结构一样



[0: 关键词, index: 位置]

强调: exec要想找所有，也必须加g

3. Math:

什么是: 保存所有数学计算的函数和常量的对象

何时: 只要执行算术计算时，都用Math

如何:

强调: Math不能new！所有的API，都用大写Math直接调用

API:

1. 取整:

1.上取整: Math.ceil(num)

2.下取整: Math.floor(num) 不识别单位，无法转换

parseInt(str) 可去掉数字后非数字的单位

3.四舍五入: Math.round(num)

缺: 只能取整

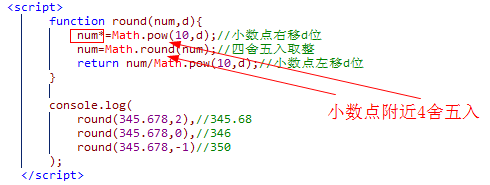
优: 返回number类型，可直接参与计算

忍不了：自定义一个Math.round():如下！

num.toFixed(d)

优: 可灵活指定小数位数

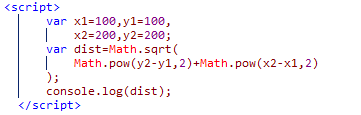
缺: 返回字符串，无法参数+运算



2. 乘方和开平方

乘方: Math.pow(底数, 幂)

开平方: Math.sqrt(num)



3. 最大值和最小值:

Math.max(值1, 值2, ....)

Math.min(值1, 值2, ....)

问题: 不支持查找数组中的最大值和最小值

解决: spead: ...数组

4. 随机数:

Math.random() 生成0~1之间的随机小数

公式: min~max之间取随机

parseInt(Math.randon()\*(max-min+1)+min)

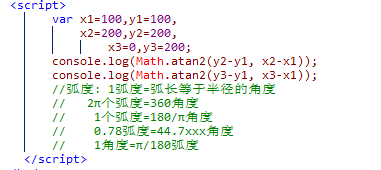
eg：抽奖

5. 计算角度:

反三角函数: Math.atan(对/临)

问题: 只能计算正负90度内的角度变化，无法区分象限

解决: Math.atan2(对, 临);



4. Date:

什么是: 封装一个时间，并提供操作时间的API

何时: 只要在程序中保存时间或操作时间

如何:

创建: 4种:

1. 创建日期对象，自动获得当前系统时间

var now=new Date()

强调: 只能拿到客户端系统时间

2. 创建日期对象，保存一个自定义时间

var birth=new Date("1993/12/26")

3. 将ms转化为日期对象:

为什么: 数据库中存储时间，都是保存时间的毫秒数

人看不懂

何时: 只要希望在内存中保存数据库中的时间时，都将ms转为date对象

如何: var date=new Date(ms);

4. 复制一个日期对象:

为什么: 日期的计算都是直接修改原日期对象，计算后，无法保留原日期

何时: 只要希望同时保留计算前后的开始和结束时间时，都要先将开始时间复制一个副本，再用副本计算结束时间

如何: var date2=new Date(date1)