正课:

String:

1. 替换

2. 切割

RegExp:

1. 验证:

2. 查找:

Math:

1. 取整

2. 乘方开方

3. 最大值/最小值

4. 随机数

String

1. 替换: 2种:

1. 简单替换: 将所有关键词替换为一个统一的值

如何: str=str.replace(/正则/i,"替换值")

2. 高级替换: 根据本次找到的关键词不同，动态选择不同的关键词替换

如何: str=str.replace(/正则/i,function(kword){

return 根据kword，动态返回不同的替换值

})

衍生: 删除: 将关键词替换为""

2. 切割:

什么是: 将字符串，按指定分隔符，切割为多段子字符串

如何: 2种:

1. 简单切割: 切割符是固定的字符串

var subs=str.split("切割符");

切割后的结果中，不包含切割符

固定套路: 将字符串打散为字符数组

var chars=str.split("");

2. 复杂切割: 切割符不是固定的

var subs=str.split(/正则/i);

RegExp:

什么是: 专门封装一条正则表达式，并使用正则表达式，执行验证和查找功能的对象

何时: 1. 查找, 2. 验证

如何:

1. 创建: 2种:

1. 直接量: var reg=/正则/ig

何时: 只要正则表达式是固定不变的！

2. 用new: var reg=new RegExp("正则","ig");

何时: 如果正则表达式需要动态生成！

2. 验证: 验证一个字符串的格式，是否符合正则表达式的要求

var bool=reg.test(str);

问题: 默认，正则表达式只要部分匹配，就返回true

解决: 只要验证，必须前加^，后加$，要求，从头到尾完整匹配

3. 查找: 即找每个关键词的内容，又找每个关键词的位置

var arr=reg.exec(str);

在str中，找下一个符合reg要求的关键词

返回值: 数组:[

0:"关键词内容", index:本次找到关键词的位置

]

如果找不到，返回null

原理: 每次查找3件事:

1. 将本次找到的关键词，放入arr中0位置

2. 将本次关键词的位置，放入arr中index位置

3. 自动修改reg的lastIndex属性为当前位置+关键词长度

Math

什么是: 专门封装数学计算的常量，并提供数学计算的API的对象

何时: 只要执行数学计算时

如何:

创建: 不能new！必须用Math直接调用API

1. 取整: 3种:

上取整: 只要超过，就取下一个整数！

Math.ceil(num)

下取整: 舍弃小数部分

Math.floor(num) 只能处理纯数字内容

vs parseInt(str) 可去掉数字结尾的非数字字符

四舍五入取整:

Math.round(num) 缺: 只能取整！优: 返回number类型

vs num.toFixed(d)

优: 按任意小数位数四舍五入

缺: 返回字符串类型

自定义:

2. 乘方和开平方:

Math.pow(底数,幂)

Math.sqrt(num)

3. 最大值、最小值

Math.max(值1,值2,...)

Math.min(值1,值2,...)

问题: 不支持获得数组中的最大值，最小值

解决: Math.max/min.apply(null,arr)

其中: apply，负责将arr打散为单个元素，再传给max/min

4. 随机数:

Math.random() 在0~1之间生成一个随机小数

在任意min~max之间，取一个随机整数的公式

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

简写: 从0~max之间取随机:

parseInt(Math.random()\*(max +1))