个人项目:

1. 静态界面

2. 收集业务数据

正课:

1. String

2. RegExp

3. Math

4. Date

1. String:

替换:

衍生: 删除: 将关键词替换为""

切割: 将一个字符串，按指定的字符切割为多段子字符串

如何: 2种:

1. 简单切割: 按一个固定的分隔符，分割字符串

var substrs=str.split("分隔符")

固定套路: 打散字符串为字符数组:

var chars=str.split("")

2. 复杂切割: 如果分隔符是变化的，可用正则切割

var substrs=str.split(/正则/)

2. RegExp:

什么是: 专门保存一条正则表达式，并提供用正则表达式执行验证和查找操作的API 的对象

何时: 只要在js中使用正则表达式，都要先创建RegExp对象

主要两个场景:

1. 验证字符串格式

2. 即查找所有关键词的内容，又查找每个关键词的位置

如何:

创建: 2种:

1. 直接量:

var reg=/正则/ig;

何时: 如果正则表达式不需要动态生成!

2. 用new:

var reg=new RegExp("正则","ig")

何时: 专门用于动态生成正则表达式

API:

1. 验证字符串格式:

var bool=reg.test(str)

用reg检查str是否符合正则表达式格式要求

问题: 正则，默认只要找到符合条件的内容，即使只是一部分，也返回true。

解决: 今后，凡是验证，必须前加^，后加$

表示从头到尾，完整匹配!

2. 即找所有关键词内容，又找所有关键词位置

var arr=reg.exec(str)

在str中找下一个关键词的内容和位置

返回值: 数组:[

"0": 本次找到的关键词内容,

"index": 本次找到的关键的位置

]

如果找不到，返回null

3. Math:

什么是: 专门保存算术计算的常量，并提供算术计算的API

何时: 今后，只要进行数学计算时

如何:

创建: 不能new!

所有API通过Math直接.调用

API:

1. 取整: 3种:

上取整: 只要小数部分超过，就取下一个整数

Math.ceil(num)

下取整: 舍弃小数部分

Math.floor(num)

vs parseInt(str) 可去掉数字后非数字字符(单位)

四舍五入取整: 整数后一位小数，够5进1，不够舍弃

Math.round(num);

vs .toFixed(2)

1. Math.round只能取整!

.toFixed(d) 按任意小数位数四舍五入

2. 返回值: Math.round() 返回数字类型

.toFixed() 返回字符串类型

2. 乘方和开平方:

Math.pow(底数,幂)

Math.sqrt(n)

3. 最大值和最小值

Math.max(值1,值2,.....)

Math.min(值1,值2,.....)

问题: max和min默认不支持在数组中找最大和最小

解决: Math.max(...arr)

4. 三角函数:

Math.sin Math.cos Math.tan

反三角函数:

Math.asin Math.acos Math.atan

5. 随机数:

0<=Math.random()<1

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

简写: 当最小值为0时，

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

4. Date:

什么是: 保存一个时间，提供操作时间的API

何时: 只要在程序中保存或操作时间，都要使用Date对象

如何:

创建: 4种:

1. 创建日期对象，并获得当前系统时间

var now=new Date();

强调:无法获得服务器时间，仅能获得客户端时间

2. 创建日期对象，封装自定义时间

var date=new Date("yyyy-MM-dd hh:mm:ss")

3. 可用一个毫秒数转化为时间

原理: 起始日期对象中保存的也是一个毫秒数(1970年1月1日0点至今的毫秒数)

为什么: 时间点受所在时区的影响，有所不同

时间段不受时区影响。

每个时区，将毫秒数时间段转为时间点时，各自的时间起点不一样，导致计算之后的终点时间各有不同。

何时: 只要将毫秒数转为日期对象执行操作时，都用new Date()

如何: var date=new Date(ms)

4. 复制一个日期对象:

var date2=new Date(date1);