正课:

1. RegExp

2. Math

3. Date

1. 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查找和验证的API

何时: 1. 使用正则表达式验证用户输入的格式

2. 即查找关键词内容，又查找关键词位置

如何:

创建: 2种:

1. 直接量: var reg=/正则/ig;

何时: 如果正则表达式是固定不变的

强调: //之中如果再出现/，就必须改为\/

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

何时: 如果正则表达式需要动态拼接生成

强调: ""中再出现\或"，都要改为\\和\"

API: 2个:

1. 即查找内容，又查找位置:

var arr=reg.exec(str) 在str中查找下一个符合reg规则的关键词。

返回值: 返回一个数组arr:

arr[0] 本次找到的关键词的内容

arr["index"] 本次找到的关键词的位置

如果没找到，就返回null

原理: 每次查找3件事:

1. 将本次找到的关键词保存在arr的0位置

2. 将本次找到的关键词位置保存在arr的index位置

3. 每个reg对象都有一个lastIndex属性, 标识下次开始的查找位置。开始时为0。

每执行完一次exec，exec会自动修改lastIndex为当前位置+关键词长度

2. 验证格式:

var bool=reg.test(str) 验证str是否符合reg规则要求

验证通过返回true，不通过返回false

问题: 默认正则只要找到部分匹配就返回true

解决: 凡是验证，必须前加^，后加$

2. Math:

什么是: 封装算数计算的常量和API

何时: 只要执行数学计算

如何:

创建: Math不用创建，不能new，直接使用！

API:

取整: 3种:

1. 上取整: 只要超过，就取下一个整数

Math.ceil(num)

2. 下取整: 只要超过，就省略小数部分

Math.floor(num)

vs parseInt(str)

3. 四舍五入取整:

Math.round(num)

缺. 只能取整,不能指定小数位数

优. 返回number, 可直接计算

vs n.toFixed(d)

优. 可按任意小数位数四舍五入

缺. 返回字符串，要先转化，再计算

可自定义round函数:

乘方和开平方:

Math.pow(底数,幂)

Math.sqrt(n)

最大值和最小值:

Math.max(值1,值2,值3,...)

Math.min(值1,值2,值3,...)

问题: max/min不支持数组参数

解决: Math.max.apply(null,arr)

随机数:

Math.random() 0<=r<1

公式: 在任意min~max之间取一个随机整数:

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

简写: 如果0~max之间

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

3. Date

什么是: 封装一个时间值，并提供操作时间的API

何时: 今后只要保存，操作时间，都要使用Date

如何:

创建: 4种:

1. 自动获得当前系统时间: var now=new Date();

强调: 只能获得客户端本地时间

2. 创建自定义时间点:

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

如果只关心日期，不关心时间，后半部分时分秒可省略。省略后，默认是00:00:00

var date=new Date(yyyy, MM-1,dd,hh,mm,ss);

3. 使用毫秒数创建日期对象:

原理: 日期对象中保存的是1970年1月1日0点至今的毫秒数——整数

为什么: 时间段，不受时区影响

相同的毫秒数: 在不同时区可显示不同的时间点

总结: 今后在数据库中存储时间或在程序中存储时间都用毫秒数，不要用文字。

问题: 人看不懂毫秒数

解决: 将毫秒数转化为程序中的日期对象，再输出

var date=new Date(ms);

4. 复制一个日期对象:

为什么: 日期的计算只能直接修改原日期

后果: 计算后，原日期无法保存

解决: 今后如果希望同时保存开始时间和结束时间，

都要先将开始时间复制一份，再用副本计算结束时间。

如何: var date2=new Date(date1);

API: 3句话:

1. 单位: FullYear Month Date Day

Hours Minutes Seconds Milliseconds

2. 每个单位都有一对儿getXXX()/setXXX()方法

getXXX() 获取指定单位的值

setXXX() 设置指定单位的值

特例: Day没有setXXX()方法

3. 取值范围:

FullYear: 和现实中年份一致

Month: 0~11 现实中: 1~12

计算机中的月份值，比现实中少1

需要修正

Date: 1~31 和现实一样

Day: 0~6 现实: 日~六 和现实中一致

Hours: 0~23 和现实一样

Minutes/Seconds: 0~59

日期计算:

1. 两日期对象可直接相减计算时间差(ms差)

2. 对任意单位做加减:3步:

1. 取值: var date=now.getDate();

2. 做加减: date+=30;

16+30=46

3. 放回去: now.setDate(date);

setDate可自动调整进制

简写: now.setDate(now.getDate()+30)

问题: setDate直接修改原日期对象，无法保留开始时间。

解决: 在计算前，都要先将开始时间复制一个副本，再用副本计算结束时间。

转字符串:

String(date) 将date转为当地时间的完整格式

date.toLocaleString() 将date转为当地时间简化版格式

date.toLocaleDateString()将date转为当地时间简化版格式，仅保留了日期部分.

date.toLocaleTimeString()将date转为当地时间简化版格式，仅保留了时分秒部分.

date.toGMTString() 将date转为0时区国际标准时间