**Math对象**

Math对象不需要使用关键字new创建

**Math.PI**  取圆周率

**Math.abs(n)**  取绝对值

**Math.floor()** 向下取整 parseInt()

**Math.ceil()**  向上取整

**Math.round()** 四舍五入取整

**Math.max()**/**Math.min**  取最大/小值

**Math.random()** 返回0~1之间的一个随机数

Math.pow(x,y) 返回x的y次幂 Math.pow(2,4) ->16

Math.sqrt() 返回数的平方根 Math.sqrt(4)->2

**Date对象**

1.创建Date对象

var now=**new Date()**  //获取当前的系统时间

var d=new Date(2018,11,20,0,0,0) //2018-12-19T16:00:00:00.000Z 月份：0~11 时区：0

var d=new Date(‘2018-11-20 0:0:0’) 获取距离某个时间的Date对象

var d=new Date(1) //1970-01-01T00:00:00.001Z //获取距离计算机元年的Date对象

2.Date方法

set/getFullYear() 设置/获取年份

set/getMonth() 设置/获取月份 0~11

set/getDate() 设置/获取日期

getDay() 获取星期 0~6

set/getHours() 设置/获取小时

set/getMinutes() 设置/获取分钟

set/getSeconds() 设置/获取秒

set/getMilliseconds() 设置/获取毫秒

**set/getTime()** 设置/获取距离计算机元年的毫秒数

例：计算现在距离某一天的时间

var n1=new Date(),

n2=new Date('2018-12-31 0:0:0'),

x=n2-n1, //两个Date对象相减，结果是距离计算机元年的差

s=Math.floor(x/1000),

d=Math.floor(s/(24\*60\*60)),

h=Math.floor(s%(24\*60\*60)/(60\*60)),

m=Math.floor(s%(60\*60)/60),

s0=Math.floor(s%60);

console.log('距离2018年12月31日还有'+d+'天'+h+'小时'+m+'分'+s0+'秒');

本地化Date对象

toLocaleString() //2018/8/16 下午2:19:42

toLocaleDateString() //2018/8/16

toLocaleTimeString() //下午2:23:00

例1：获取时间的方法

var d=new Date();

console.log(d.toDateString()); //Thu Aug 16 2018

console.log(d.toTimeString()); //14:21:19 GMT+0800 (中国标准时间)

console.log(d.toString()); //Thu Aug 16 2018 14:19:42 GMT+0800 (中国标准时间)

console.log(d.toGMTString()); //Thu, 16 Aug 2018 06:23:44 GMT

console.log(d.toUTCString()); //Thu, 16 Aug 2018 06:24:27 GMT

console.log(d.toLocaleString()); //2018/8/16 下午2:19:42

console.log(d.toLocaleDateString()); //2018/8/16

console.log(d.toLocaleTimeString()); //下午2:23:00

例2：初始化员工入职时间，求合同到期时间、续签时间（提前一月，遇上周六周日提前至周五）、提醒时间（续签前一周）

var w=new Date('2018-5-10'),

m=new Date(w),

y=m.getFullYear()+3;

m.setFullYear(y); //

console.log('合同到期时间：'+m.toLocaleDateString());

var r=new Date(m);

r.setMonth(r.getMonth()-1);

var d=r.getDay();

if(d==6){

r.setDate(r.getDate()-1);

}

if(d==0){

r.setDate(r.getDate()-2);

}

var a=new Date(r);

a.setDate(a.getDate()-7);

console.log('续签时间：'+r.toLocaleDateString());

console.log('提醒时间：'+a.toLocaleDateString());

**Number对象**

1.创建Number对象

var num1=new Number(16); // Number {16} 可以执行数学运算

var num2=new Number(‘abc’); // Number {NaN}

var num3=new Number(‘150abc’); // Number {NaN}

var num4=new Number('123'); // Number {123} 使用Number()进行隐式转换

\*获取计算机存储的最大值和最小值(数值型)

console.log(Number.MAX\_VALUE); //1.7976931348623157e+308

console.log(Number.MIN\_VALUE); //5e-324

\*解决小数计算问题

例：console.log(0.1+0.2); //十进制->二进制->十进制 0.30000000000000004

console.log((0.1+0.2).toFixed(4)); //toFixed(4)表示保留小数点后4位 0.3000

**Boolean对象**

创建Boolean对象

var b1=new Boolean(1); // false : 0 NaN undefined null ‘’ Boolean {true}

var b2=!!100; //true

**Error对象**

1.错误类型：

(1)语法错误 SyntaxError

出现语法错误，整个程序不执行。

(2)引用错误 ReferenceError

使用了不存在的变量，会影响后续代码的执行

(3)类型错误 TypeError

错误地使用了括号，会影响后续代码的执行

(4)范围错误 RangeError

参数超出了范围，会影响后续代码的执行

2.异常(错误)处理

try{

//尝试执行的代码，代码可能有错误

var arr=new Array(-1);

}catch(err){

//把产生的错误捕获到err中

console.log(err);

var arr=new Array(1);

}

console.log(arr);

console.log(100);

**ES6新特性**

1.块级作用域

{

let a=1; //只能在块级作用域内访问变量a，使用let声明局部变量（在{}之间）

}

2.箭头函数

回调函数中传递的匿名函数可以使用箭头函数

var arr=[100,79,23,99];

/\*arr.sort(function(a,b){

return a-b;

});\*/

arr.sort((a,b)=>{

return a-b;

});

arr.sort((a,b)=>a-b) //  大括号之间只有一行，且含有return [23, 79, 99, 100]

3.模板字符串

使用反引号(`)包含的字符串就是模板字符串，在模板字符串之间可以放任意的字符

在模板字符串中使用变量或其他值——${变量名称}

例：var bname='三国演义',

author='罗贯中',

publish='人民邮电出版社';

var str=`书名：${bname}

作者：${author}

出版社：${publish}`;

console.log(str);

4.函数的拓展

例：function fn(a,b,c=0){ //c默认值为0，如果c不赋默认值，fn(10,25)调用后结果为NaN

console.log(a+b+c);

}

fn(10,25) //35,c赋值后会代替默认值