复习

push()/unshif()/pop()/shift()/slice(0,3)

名称

参数

作用

返回值

[ [],[],[] ]

New String()//构造函数，String()//普通函数

var str=’hello’;

转移字符 \n \’

length/charAt(1)/charCodeAt()/indexOf(‘@’)/lastlndexOf()

toUpperCase()/toLowerCase()/slice(0,5)/substr(0,3)/split(‘ ’)

replace(/tom/gi,’...’ )/match()/search()

Math 对象

PI/abs()/ceil()/floor()/round()/max()/min()/pow(x,y)/random()

1. Date 对象

用于对日期和时间进行存储和计算

1. 创建Date对象

new Date(‘2019/3/16 10:30:50’)

new Date(2019.0,16,10,30,50);//第2个参数月份范围0~11

new Date() //存储当前系统时间的Date对象

new Date(1000\*60\*60\*24);//存储距离计算机元年的毫秒数对应的日期时间

1. 获取Date对象中的信息

getFullYear 年/Month 月/Date 日/Hours 时/Minutes 分/Seconds秒/

Milliseconds 毫秒/Day 星期/Time 距离计算元年毫秒

月份结果0~11 1月~12月

星期结果0~6 星期日~星期六

练习：根据Date对象(2019/2/5 0:0:0);

打印2019年2月5日0时0分0秒 星期二

var d=new Date('2019/2/5 0:0:0');

var year=d.getFullYear();

var month=d.getMonth()+1;

var date=d.getDate();

var hour=d.getHours();

var minute=d.getMinutes();

var second=d.getSeconds();

var day=d.getDay();//0~6

var arr=['星期日','星期一','星期二','星期三','星期四','星期五','星期六'];

//判断是否小于10

if(month<10){

//在数字前拼接字符0

month='0'+month;

}

console.log(

year+'年'+

month+'月'+

date+'日'+

hour+':'+

minute+':'+

second+arr[day]

);

//计算当前距离春节还有？ 天？ 小时？ 分？ 秒？

//计算两个Date对象的时间差

//通过计算两个时间距离计算机元年的毫秒差值

//当前系统时间的对象

（3）设置(修改)Date中的信息

setFullYear/ Month/Date/Hours/Minutes/Seconds/Milliseconds/Time

月份0~11

设置距离计算机元年毫秒数，有可能会覆盖所有的日期时间。

不能设置星期，星期是由日期来决定。

（4）转为本地字符串格式

toLocaleString() //年-月-日 时：分：秒

toLocaleDateString() //年-月-日

toLocaleTimeString() //时：分：秒

练习：创建2019-2-5 10:20:30日期时间的Date对象，计算15天以后，8小时后的本地字符串

var d=new Date('2019/2/5 10:20:30');

//15天以后，先获取当前的日期，加15，把结果设置为日期

d.setDate( d.getDate()+15);

d.setHours(d.getHours()+8);

console.log(d.toLocaleString());

（5）拷贝一个Date对象

var d1=new Date(Date对象)

拷贝后会生成一个新的对象，和之前的对象没有关系

练习：创建对象保存员工的入职时间‘2018-11-16’3年后到期，计算到期时间；合同到期前1个月续签合同，假设是周末，提前到周五，计算续签时间，

打印以上3个时间的本地字符串格式

var d1=new Date('2018/11/16');

//拷贝一个入职时间

var d2=new Date(d1);

//设置日期为3年后

d2.setFullYear(d2.getFullYear()+3);

//拷贝一个到期时间，作为续签日间

var d3=new Date(d2);

//提前一个月，设置月份减1

d3.setMonth(d3.getMonth()-1);

//查看是否为周末，获取星期6 0

var day=d3.getDay();

//如果是周六，提前1天

if(day==6){

d3.setDate(d3.getDate()-1);

}else if(day==0){

d3.setDate(d3.getDate()-2);

}

console.log('入职时间:'+d1.toLocaleDateString());

console.log('到期时间:'+d2.toLocaleDateString());

console.log('续签时间:'+d3.toLocaleDateString());

1. Number对象

new Number(值) 将任意的数据转为数值型，返回对象

Number(值) 返回数值型数据

获取能存储的最大值和最小值

Numger.MAX\_VALUE

Number.MIN\_VALUE

保留小数点后n位

toFixed()

例如： 0.30000004.toFixed(1) ->0.3

1. Boolean 对象

new Boolean(值) 将任意的数据转为布尔型，返回对象

Boolean(值) 返回布尔型数据

false的5种情况

‘’ 0 NaN undefined null

1. 常用的错误

SyntaxError 语法错误，错误的使用中文，缺少括号...程序不执行

ReferenceError 引用错误，使用了未声明的变量，会影响后续代码的执行

TypeError 类型错误，错误的使用了括号，会影响后续代码的执行

RangeError 范围错误，参数的使用超出了范围，会影响后续代码的执行

5.错误处理

|  |
| --- |
| try{  尝试执行，可能出现错误的代码  }catch(err){  err:捕获到的错误信息  处理错误的代码  } |

6.ES6新特征

ECMAScript 6

1. 函数的参数

|  |
| --- |
| 设置参数的默认值  function add(a,b=0){  console.log(a+b)  }  add(5)  如果没有给参数传递值，则自动使用默认值 |