**字符串对象**

JS把字符串、数值、布尔型包装成对象——包装对象

包装类型：专门封装原始类型的数据，并提供对数据常用操作的内置引用类型。——让原始数据类型像引用类型一样，拥有方法和属性。

1.创建字符串

var str1=’hello’; //直接量

var str2=new String(‘world’); //构造函数——类型是object

var str3=String(‘tom’);

2.转义字符

\b 退格符

\n 换行符

\r 回车符

\t 水平制表符

\’ 单引号

\’’ 双引号

\\ 反斜线

3.字符串API

(1)将英文字符串转为大/小写

str.**toUpperCase()**  str.**toLowerCase()**

(2)获取指定位置的字符

**charAt()**  获取指定下标处的字符

**charCodeAt()** 获取指定下标处字符的Unicode码

(3)查找某个字符对应的下标

**IndexOf()** 查找某个字符对应的下标

**lastIndexOf()** 查找某个字符最后一次出现位置的下标

例1：var str='javascript';

console.log(str.indexOf('a',5)); //从下标5的字符开始查找 没有返回-1

console.log(str.lastIndexOf('a',5)); //3

例：验证邮箱格式是否正确（含有‘@’）

var str='1589456@qq.com';

if(str.indexOf('@')>-1){

console.log(true);

}else{

console.log('false');

}

(4)字符串截取

**slice()**

例：var str=’javascriptabc’;

console.log(str.slice(4,10)); //’script’

console.log(str.slice(-13，-9)); //’java’

**substring()**

当参数为负数，自动将参数转换为0；将较小的数作为开始的位置，较大的位置作为结束的位置。

**substr()**

传入2个参数：开始位置的索引，如果为负数n，表示从倒数第n位开始截取；

截取的长度，如果省略，表示截取到末尾

例：通过身份证号获取出生日期

var id='450892199908184926';

console.log(id.slice(6,14)); //19990818

console.log(id.substring(10,6)+'-'+id.substring(10,14)); //1999-0818

console.log(id.substr(-6,4)); //从倒数第6位开始，截取4个字符 1849

(5)分隔字符串

**split()** 根据分隔符把字符串切割成数组 ——join()的反向操作

例：将一句英文转换为每个单词首字母大写

var str='hoW oLd arE yOu?',

arr=str.split(' ');

for(var k in arr){

var u=arr[k].slice(0,1).toUpperCase(); //分割首字母,转大写 H O A Y

var l=arr[k].slice(1).toLowerCase(); //分割剩余字母，转小写

arr[k]=u+l;

str=arr.join(' ');

}

console.log(str); //How Old Are You?

(6)匹配模式

**replace()** 查找并替换特定的字符串，或一个与正则表达式匹配的字符串

例：str.replace(‘china’,’中国’)（只替换第一个’china’） str.replace(/china/ig,’中国’)

修饰符：**i** 执行对大小写不敏感的匹配 /china/i

**g** 执行全局匹配 /china/ig

**m** 允许执行多行匹配 /^is$/m(m会对^和$匹配会产生影响)

**match()**  在字符串内查找特定的字符串，或找到一个或多个与正则表达式匹配的子串，返回一个数组。

str.match(/china/ig) //[‘china’,’China’,’CHINA’,’china’]

**search()**  查找某个字符串，找到后返回第一个子串下标，找不到则返回-1，不支持全局匹配，只能使用i

str.serch(/china/i)