

replace的实例

一 全部替换

1.小写数字（A）替换成大写数字（B）

二 部分替换

1.字符首字母小写变大写

三 替换且储存obj

1.获取字符串出现最多的字符及次数

2.获取URL中参数地址并保存在obj中

四 模板引擎——固定部分+非固定（捕获）部分

1.初步实现模板引擎

2.改变时间格式

备注：本实例中都在 正则全局g

一 全部替换

1.小写数字（A）替换成大写数字（B）

分析：

- ①是一类替换一类，A在字符串中等待被替换，B是替换内容，既然为一类，可以用数组来包起来
- ②是一一对应的关系，可以利用数字作为索引 查找数组内容
- ③全体替换，正则g

```
var ary=["零","壹","贰","叁","肆","伍","陆","柒","捌","玖","拾"]
var str="12354元"
var reg=/\d/g
var m= str.replace(reg,function(){
    return ary[arguments[0]]
})
```

//牢记：每次捕获的内容都是 arguments 值为arguments[0] 前提是要有 function

//同理可实现 a-A 这种一一对应的关系； 数组也是我们自己定义的，数组内容是有限的

二 部分替换

1.字符首字母小写变大写

```
var str="zhu fen pei xun"
var reg=/([a-z])([a-z]+)/g
str.replace(reg,function(){
    return arguments[1].toUpperCase()+arguments[2]
})
```

三 替换且储存obj

1.获取字符串出现最多的字符及次数

分析：

- ①全局捕获 exec（一般不用，while循环） match replace
- ②存储字符及对应次数（是两个值，一一对应，应该存在object里面）
- ③求最值（遍历对象 求最大值）

```
var str="adkfkdkjaadhkjdlksajjfa63423fdadfladljfk"
```

```

var reg=/\w/ig
//两种写法: ①match+for循环 ②利用replace里内置循环
var ary= str.match(reg)
var obj={}
for(var i=0;i<ary.length;i++){//利用属性名覆盖, 属性值叠加
    if( obj[ary[i]]>=1){obj[ary[i]]+=1}
        else{obj[ary[i]]=1}
    }
console.log(obj);
var n=0;
var val=null;
for(var k in obj){
    if(obj[k]>=n){n=obj[k];val=k} //在这获取属性名, 只能获取到一个
}
//再一次循环
var ary2=[]
for(var k in obj){
    if(obj[k]==n){ary2.push(k)}
}
console.log(ary2);
console.log(ary2.toString()+"出现了"+n+"次")

//②利用replace里内置循环
var str="adkfkdkjaadhkjdlkskajjfa63423fdadfjladljfk"
var reg=/\w/ig
var obj={}
    str.replace(reg,function(){
        if(obj[arguments[0]]>=1){obj[arguments[0]]+=1}
        else {obj[arguments[0]]=1;}
        //其余步骤同上, 还是replace 简单 内置循环, 不需要for循环
    })

```

2.获取URL中参数地址并保存在 obj中

```

//var str="http://kbs.sports.qq.com/game?mid=100000&cid=1467086&app=1.0"
// 转化成:    obj={
//            mid:10000,
//            cid:1467086,
//            app:1.0}

var str="http://kbs.sports.qq.com/game?mid=100000&cid=1467086&app=1.0"
var reg=/([a-z]+)=([d+])&([a-z]+)=([d+])&([a-z]+)=([d.\d])/g
var obj={}
    str.replace(reg,function(){
        obj[arguments[1]]=arguments[2]
        obj[arguments[3]]=arguments[4]
        obj[arguments[5]]=arguments[6]
    })
console.log(obj);

//换一个简单的正则 :格式都是 xxx=xxx
var reg=/([?&]+)=([?&]+)/g//想捕获的用 ( ) 分组, 在arguments里获得 exec

```

```

    var obj2={}
    str.replace(reg,function(){
        obj2[arguments[1]]=arguments[2]
    })
    console.log(obj2);
    //正则分组简单点，function里面的对arguments的操作就能简单点

```

四 模板引擎——固定部分+非固定（捕获）部分

1.初步实现模板引擎

```

var str="my name is {0},my age is {1}, my hobby is {2}, my corporation is {3}"
//将 {} 替换成 指定字符
//模板引擎原理：固定不变的部分，改变的部分；我们将改变的部分替换即可；

```

分析：①有数字可考虑数组索引

```

var ary=["guoxin",19,"Relaxing","Google"]
var reg2=/{\d}/g //注意 这个不是直接获取数字，所以要利用到小分组
str=str.replace(reg2,function(){
    return ary[arguments[1]]
})
console.log(str);
//用到了 分组捕获
//整体捕获， 分组替换

```

2.改变时间格式

2015-2-3 12:9:9 变成 2015年02月03日 12时09分09秒

分析：

- ①-变为 年 月 日 ： 变成 时 分 秒 没有对应关系，考虑利用位置
- ② 添0占位

//用到了 模板引擎的思想

//定义字符串的固定部分 和 改变部分

```

var str="2015-2-3 12:9:0"
var str2="{0}年{1}月{2}日{3}时{4}分{5}秒"
var reg=/\d+/g
var ary=str.match(reg)
var reg2=/{\d}/g
//不要单独写for循环，replace里有内置循环
var val=str2.replace(reg2,function(){
    var m=arguments[1]//注意：m是字符串 不能和数字比较
    ary[m]<10? ary[m]="0"+ ary[m]:null
    return ary[m]
})
console.log(val);

```

//格式时间 不是从时间入手，而是分析固定和非固定的形式，再捕获数字；

总结：

repalce: 捕获并替换,

①具备exec、match的功能;

②捕获的内容传递给arguments

arguments[0]:捕获内容

arguments[1-i]: 分组捕获内容

③内置循环功能, 不需要在外面循环

格式: `str.replace(reg,function(){return 替换值})`