

JS中关于字符串的一些细节知识

在JS中所有用单引号或者双引号包起来的都是字符串，每一个字符串是由零到多个字符组成

```
1. var str = 'zhufengpeixun';
2. str.length -> 字符串长度
3. str[0] -> 'z'
4. str[str.length-1] -> 'n'
5. str[100] -> undefined
6.
7. //=> 字符串中的每一个字符都有一个自己对应位置的索引，也有类似于数组一样的length代表自己的长度
8.
9. //=> 循环遍历字符串，输出每一项字符
10. for(var i=0; i<str.length; i++){
11.     console.log(str[i]);
12. }
```

关于字符串中常用的方法

字符串是基本数据类型，字符串的每一次操作都是值直接的进行操作，不像数组一样是基于空间地址来操

作的，所以不存在原有字符串是否改变这一说，肯定都是不变的

charAt/charCodeAt

作用：charAt根据索引获取指定位置的字符，charCodeAt不仅仅获取字符，它获取的是字符对应的Unicode编码值(ASCII码值)

参数：索引

返回：字符或者对应的编码

```
> var str = 'zhufengpeixun';
```

```
< undefined
```

```
> str.charAt(0)
```

```
< "z"
```

和直接操作索引方式获取的区别：

```
> str[0]
```

```
< "z"
```

1. 当索引不存在的时候, str[x]获取的结果是undefined，运行的机制和对象是一样的，而charAt(x)获取的结果是空字符串

```
> str[100]
```

```
< undefined
```

```
> str.charAt(100)
```

```
< ""
```

```
> var str = 'zhufengpeixun';
```

```
< undefined
```

charCodeAt返回的是字符对应的编码

```
> str.charCodeAt(0)
```

```
< 122
```

fromCharCode返回的是编码对应的字符

```
> String.fromCharCode(122)
```

```
< "z"
```

indexOf/lastIndexOf

基于这两个方法，可以获取字符在字符串中第一次或者最后一次出现位置的索引，有这个字符，返回大于等于零的索引，不包含这个字符，返回的结果是-1，所以可以基于这两个方法，验证当前字符串中是否包含某个字符

```
1. var str='zhufengpeixun';
2. if(str.indexOf('@')>-1){
3.     //=>条件成立说明包含@符号
4. }
```

slice

作用：str.slice(n,m) 从索引n开始找到索引为m处(不包含m)，把找到的字符当做新字符串返回

```
> var str = 'zhufengpeixun';
```

```
< undefined
```

```
> str.slice(2,7)
```

```
< "ufeng"
```

和数组中的slice操作是一样的

```
> str.slice(2)
```

1. 不写m是查找到字符串的末尾

```
< "ufengpeixun"
```

2. n/m都不写是字符串的克隆

```
> str.slice()
```

3. 支持负数索引：用字符串的总长度+负数索引做运算

```
< "zhufengpeixun"
```

```
> str.slice(-3,-1)
```

```
< "xu"
```

substring

和slice语法一模一样，唯一的区别在于：slice支持负数索引，而substring不支持负数索引

```
> var str = 'zhufengpeixun';
< undefined
> str.slice(2,7)
< "ufeng"
> str.substring(2,7)
< "ufeng"
> str.substring(-3,-1)
< ""
> str.slice(-3,-1)
< "xu"
```

substr

也是字符串截取的方法，用法是：str.substr(n,m)，从索引n开始截取m个字符

```
> var str = 'zhufengpeixun';
< undefined
> str.substr(2,7)
< "ufengpe"
> str.substr(2)
< "ufengpeixun"
> str.substr(-3,1)
< "x"
> str.substr(-6,3)
< "pei"
```

和substring一样，第二个参数不传，截取到末尾，但是它支持第一个索引为负数，负数也是总长度+负数索引

toUpperCase/toLowerCase

实现字母的大小写转换，toUpperCase小写转大写，toLowerCase大写转小写

```
> var str = 'ZhuFengPeiXun';  
< undefined  
-----  
> str.toLowerCase()  
< "zhufengpeixun"  
-----  
> str.toUpperCase()  
< "ZHUFENGPEIXUN"-----
```

split

和数组中的join相对应，数组中的join是把数组们一项按照指定的连接符变为字符串，而split是把字符串按照指定的分隔符，拆分成数组中每一项

```
> var ary=[12,23,34];  
< undefined  
-----  
> ary.join('+')  
< "12+23+34"  
-----  
> var str="12+23+34";  
< undefined  
-----  
> str.split('+')  
< ► (3) ["12", "23", "34"]  
-----
```

replace

作用：替换字符串中的原有字符

参数：原有字符，要替换的新字符

返回：替换后的字符串

```
1. //=>把“zhufeng”替换为“珠峰”
2. var str = 'zhufeng2017zhufeng2018';
3. str = str.replace('zhufeng','珠峰'); //=>在不使用正则的情况下，没执行一次replace只能替换一个“珠峰2017zhufeng2018”
4. str = str.replace('zhufeng','珠峰'); //=>“珠峰2017珠峰2018”
5.
6. //=====
7. str = str.replace(/zhufeng/g,'珠峰');
```

字符串中还有很多常用方法，回去后大家可以自己扩展一下：（String.prototype）

- includes
- localeCompare
- search
- trim

• ...

真实项目中的需求

1. 时间字符串格式化

有一个时间字符串“2018-4-4 16:26:8”，我们想基于这个字符串获取到“04月04日 16时26分”

```
1.  /*
2.   * 1. 基于SPLIT按照空格把字符串拆成两部分(数组中的两项)
3.   * 2. 左边这一部分继续以SPLIT按照中杠来拆
4.   * 3. 右边这一部分继续以SPLIT按照冒号来拆
5.   * 4. 把需要的信息拼接在一起即可（拼接的时候不足十位的补零）
6.   */
7.  function addZero(val) {
8.      return val < 10 ? '0' + val :
9.      val;
10. }
11. var str = '2018-4-4 16:32:8';
12. var ary = str.split(' '),//=>["20
```

```

    18-4-4", "16:32:8"]
13.     aryLeft = ary[0].split('-
    '),//=>["2018", "4", "4"]
14.     aryRight = ary[1].spli
    t(':');//=>["16", "32", "8"]
15. var month = addZero(aryLeft[1]),
16.     day = addZero(aryLeft[2]),
17.     hour = addZero(aryRight[0]),
18.     minute = addZero(aryRigh
    t[1]);
19. var result = month + '月' + day +
    '日' + hour + '时' + minute +
    '分';
20. console.log(result);

```

暂时提高眼界的：

```

1. ~function (pro) {
2.     pro.formatTime = function (te
    mplate) {
3.         template = template ||
        '{0}年{1}月{2}日 {3}时{4}分{5}秒';
4.         var ary = this.matc
            h(/\d+/g);
5.         template = template.repla
            ce(/\{(\d+)\}/g, function () {

```



```
6.         var n = arguments[1],
7.         val = ary[n] ||
            '0';
8.         val < 10 ? val = '0'
+ val : null;
9.         return val;
10.    });
11.    return template;
12. }
13. }(String.prototype);
```

URL地址问号传参解析

有一个URL地址“<http://www.zhufengpeixun.cn/stu/?lx=1&name=AA&sex=man>”地址问号后面的内容是我们需要解析出来的参数信息

```
{
  lx:1,
  name:'AA',
  sex:'man'
}
```

```
1.  /*
```

```
2.    * 1.先找到问号，把问号后面的信息截取下来即可
3.    * A.首先我们需要验证是否存在#哈希值，存在我们从问号开始截取到#，不存在我们直接截取到字符串的末尾
4.    * 2.以&进行拆分(数组)
5.    * 3.遍历数组中的每一项，把每一项在按照=进行拆分，把拆分后的第一项作为对象的属性名，第二项作为属性值进行存储即可
6.    */
7.    var str = 'http://www.zhufengpeixun.cn/stu/?lx=1&name=AA&sex=man#teacher';//=>#后面的称为哈希(HAS H)值,这个值可能有可能没有,我们需要处理,有的话我们截取的时候需要过滤掉
8.
9.    //=>获取问号和井号在字符串中索引位置
10.   var indexASK = str.indexOf('?'),
11.       indexWell = str.indexOf('#');
12.   //=>#可能有可能没有
13.   if (indexWell > -1) {
14.       //=>存在井号,我们截取到井号的位置即可
15.       str = str.substring(indexASK + 1, indexWell);
16.   } else {
17.       //=>没有井号,我们截取到末尾即可
18.       str = str.substr(indexASK +
```

```

    1);
19. }
20.
21. //=>str='lx=1&name=AA&sex=man'
22. var ary = str.split('&'),//=>["lx=1", "name=AA", "sex=man"]
23.     obj = {};
24. for (var i = 0; i < ary.length; i++) {
25.     var item = ary[i],
26.         itemAry = item.split('=');
27.     //console.log(itemAry);//=>
    ["lx", "1"] ["name", "AA"] ...
28.     var key = itemAry[0],
29.         value = itemAry[1];
30.     obj[key] = value;
31. }
32. console.log(obj);//=>{lx: "1", name: "AA", sex: "man"}

```

提高眼界：

```

1. ~function (pro) {
2.     pro.queryURLParameter = function () {

```

```
3.         var obj = {},
4.         reg = /([^?=&#]+)(?:=
    ([^?=&#]+)?)/g;
5.         this.replace(reg, function
    n () {
6.             var key = argument
    s[1],
7.             value = argument
    s[2] || null;
8.             obj[key] = value;
9.         });
10.        return obj;
11.    }
12. }(String.prototype);
13.
14. var str = 'http://www.zhufengpeix
    un.cn/stu/?lx=1&name=&sex=#teache
    r';
15. console.log(str.queryURLParamete
    r());
```