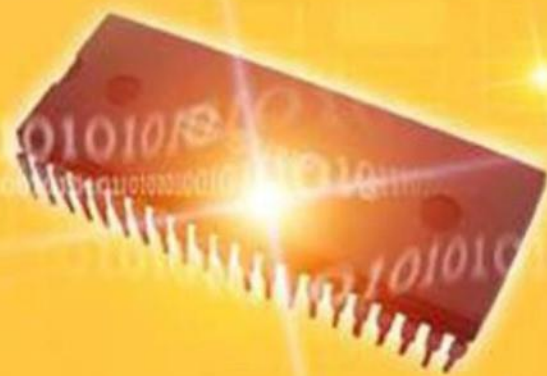


字符串



- 掌握字符串的常用方法

- 字符串的定义
- 字符串的常用方法

◆ 什么是字符串

String类型用于表示零或多个16位的Unicode字符串组成的字符序列，即字符串，字符串可以由双引号（"）或单引号（'）表示。

Unicode（统一码、万国码、单一码）是一种在计算机上使用的字符编码。解决传统的字符编码方案的局限而产生的，它为每种语言中的每个字符设定了统一并且唯一的二进制编码，以满足跨语言、跨平台进行文本转换、处理的要求。

◆ 创建字符串的两种方式

```
var str= "hello world" ;
```

```
var str= new String("hello world");
```

String函数是一个内置的构造函数

- String类型都有一个length属性，表示字符串包含多少个字符
- 当然需要注意的是js中的中文每个汉字及空格也代表一个字符

◆ 字符串既不是对象，为什么会有属性？ 比如：str.b=1

只要引用了字符串str的属性，js就会将字符串通过调用new String()的方式转换成对象，这个对象继承了字符串的方法，并被用来处理性的引用。一旦属性引用结束，这个新创建的对象就会销毁。不能自定义属性，属于不可变类型。

```
var str= "test" ;
```

```
str.b=9;
```

```
console.log(str.b);//undefined
```

修改只是发生在临时对象身上，而这个临时对象并未继续保留下来。

◆ 把其它类型转化为字符串

- toString() 默认按十进制返回,基模式是会把十进制换成相应的进制数

```
var a = 10;  
alert(a.toString());  
alert(a.toString(2)); //基模式
```

◆ 把其它类型转化为字符串

- toString() 默认按十进制返回,基模式是会把十进制换成相应的进制数

练一练:

```
var x={x:1};  
var y=[1,2,3];  
var z=true;  
var w=null;  
var h=undefined;  
var j=20;  
var g=function(){  
    console.log(123);  
};
```

```
console.log(x.toString());  
console.log(y.toString());  
console.log(z.toString());  
console.log(w.toString());  
console.log(h.toString());  
console.log(j.toString());  
console.log(j.toString(16));  
console.log(g.toString());
```


◆ 把其它类型转化为字符串

- String() 强制转换成字符串的方法

```
var a = 10;  
alert(String(a));  
alert(typeof String(a));
```

◆ 把其它类型转化为字符串

- String() 强制转换成字符串的方法

练一练：

```
var x={x:1};  
var y=[1,2,3];  
var z=true;  
var w=null;  
var h=undefined;  
var j=20;  
var g=function(){  
    console.log(123);  
};
```

```
console.log(String(x));  
console.log(String(y));  
console.log(String(z));  
console.log(String(w));  
console.log(String(h));  
console.log(String(j));  
console.log(String(g));
```

◆ charAt()

- 返回字符串的某一个；若不存在这个范围将返回一个空字符串，有效值是0与字符串长度减一之间的索引值，默认值为0
- 括号里的参数为所需查找字符的索引

```
var str = "abcdef";  
console.log(str.charAt()); //a,默认的位置为0  
console.log(str.charAt(2)); //c,找到了str中索引为2的字符串c  
console.log(str.charAt(6)); //不存在这个范围返回空字符串
```

练一练：

```
var str= “明天的我会感谢今天努力的我” ;  
console.log(str.charAt(4));  
console.log(str.charAt(2));  
console.log(str.charAt(6));  
console.log(str.charAt());  
console.log(str.charAt(15));
```

◆ search()

- 查找并返回的是找到的第一个字符串的索引位置，如果找不到返回-1；

```
var str = "abcdef";  
console.log(str.search("b"));  
console.log(str.search("e"));  
console.log(str.search("f"));  
console.log(str.search("s"));
```

练一练：

```
var str="明天的我会感谢今天努力的我";  
console.log(str.search("我"));  
console.log(str.search("天"));  
console.log(str.search("会"));  
console.log(str.search("你"));
```

◆ charCodeAt ()

- 返回一个整数，代表指定位置字符的Unicode编码。若不存在这个范围将返回NaN，有效值是0与字符串长度减一之间的索引值，默认值为0
- 阿斯科码 0~9 对应48~57，a~z对应97~122，A~Z对应65~90
- 中文unicode编码参考：

<http://www.qqxiuzi.cn/bianma/zifuji.php>

`var str='84935784';`
`console.log(str.charCodeAt());`//56,没有参数,默认的索引值为0

`console.log(str.charCodeAt(2));`//57,找到了str中索引为2的字符串的unicode编码

`console.log(str.charCodeAt(10));`//NaN,不存在这个范围返回NaN

练一练：

```
var str='今天天气晴朗';
console.log(str.charCodeAt(1));//22825
console.log(str.charCodeAt(3));//27668
console.log(str.charCodeAt(5));//26391
console.log(str.charCodeAt(7));//NaN
```


◆ fromCharCode ()

- 从一个或多个Unicode编码中返回一个字符串
- 可以有多个参数，如果参数为空，则返回一个空字符串
- 它是String对象里的静态方法，需要通过String来调用

```
console.log(String.fromCharCode(97,98,99));//abc
```

```
console.log(String.fromCharCode(20170,22825,22825));//今天天
```

```
console.log(String.fromCharCode(20116,24180,21518,30340,33258,24049));//五年后的自己
```

```
console.log(String.fromCharCode());//空字符串
```

练一练：

```
console.log(String.fromCharCode(48,50,77));//02M
```

```
console.log(String.fromCharCode(26126,22825,30340,20250,24863,35874));//明天的会感谢
```

```
console.log(String.fromCharCode());//返回空字符串
```


◆ substring(start,end)

- 返回一个新的字符串，参数为非负整数
- 参数start：字符串的起始位置从0开始
- 参数end：字符串的结束位置（延伸至end，但不包括end）

```
var str= "未来的你会感谢今天努力的你" ;  
console.log(str.substring(3,7));//你会感谢
```

- substring方法使用start和end两者中的较小值作为子字符串的起点

```
console.log(str.substring(7,3));//你会感谢
```

- 如果start或end为NaN或者为负数，那么将其替换为0

```
console.log(str.substring(-5,3));//未来的  
console.log(str.substring(7,"abc"));//未来的你会感谢
```

- 如果只有一个数的时，这个数就会默认为开始的位置

```
console.log(str.substring(5));//感谢今天努力的你
```

- 如果end超出范围时，返回从起始位置到结束的值

```
console.log(str.substring(5,30));//感谢今天努力的你
```

◆ substring(start,end)

- 返回一个新的字符串，参数为非负整数
- 参数start：字符串的起始位置从0开始
- 参数end：字符串的结束位置（延伸至end，但不包括end）
- substring方法使用start和end两者中的较小值作为子字符串的起点
- 如果start或end为NaN或者为负数，那么将其替换为0
- 如果只有一个数的时，这个数就会默认为开始的位置
- 如果end超出范围时，返回从起始位置到结束的值

练一练：

```
var str='今天天气好晴朗处处好风光';  
console.log(str.substring(1,4));  
console.log(str.substring(3,20));  
console.log(str.substring(3,-5));  
console.log(str.substring(7));  
console.log(str.substring("e",5));  
console.log(str.substring(7,5));
```

◆ substr(start,length)

- 返回一个从指定位置开始的指定长度的字符串
- 参数start: start所需的字符串的起始位置。字符串的位置从0开始
- 参数length: 返回的字符串中应包括的字符个数

```
var str= "未来的你会感谢今天努力的你" ;
```

```
console.log(str.substr(3,7));//你会感谢今天努
```

- 如果start为NaN, 那么将其替换为0

```
console.log(str.substr( "abc" ,7));//未来的你会感谢
```

- 如果start为负数, 负数从后面倒着往前数。

```
console.log(str.substr(-5,2));//天努
```

- 如果只有一个数的时, 这个数就会默认为开始的位置

```
console.log(str.substr(7));//今天努力的你
```

- 如果length超出范围时, 返回从起始位置到结束的值

```
console.log(str.substr(5,30));//感谢今天努力的你
```

◆ substr(start,length)

- 返回一个从指定位置开始的指定长度的字符串
- 参数start: start所需的字符串的起始位置。字符串的位置从0开始
- 参数length: 返回的字符串中应包括的字符个数
- 如果start为NaN, 那么将其替换为0
- 如果start为负数, 负数从后面倒着往前数。
- 如果只有一个数的时, 这个数就会默认为开始的位置

练一练:

```
var str= '今天天气好晴朗处处好风光' ;  
console.log(str.substr(1,4));  
console.log(str.substr(3,20));  
console.log(str.substr(-5,3));  
console.log(str.substr(7));  
console.log(str.substr( "e" ,5));  
console.log(str.substr(7,5));
```


◆ slice(start,end)

- 返回一个新的字符串，参数为非负整数
- 参数start: start所需的字符串的起始位置。字符串的位置从0开始
- 参数end: 字符串的结束位置（延伸至end，但不包括end）

```
var str= "未来的你会感谢今天努力的你" ;  
console.log(str.slice(3,7));//你会感谢
```

- 如果start为NaN，那么将其替换为0

```
console.log(str.slice( "abc" ,3));//未来的
```

- 如果只有一个数的时，这个数就会默认为开始的位置

```
console.log(str.slice(3));//你会感谢今天努力的你
```

- 如果end超出范围时，返回从起始位置到结束的值

```
console.log(str.slice(3,20));//你会感谢今天努力的你
```

- 如果start和end为负数，负数从后面倒着往前数。

```
console.log(str.slice(-5,-2));//天努力
```

- 其它情况

```
console.log(str.slice(-5,2));//返回空字符串
```

```
console.log(str.slice(2,0));//返回空字符串
```

◆ slice(start,end)

- 返回一个新的字符串，参数为非负整数
- 参数start: start所需的字符串的起始位置。字符串的位置从0开始
- 参数end: 字符串的结束位置（延伸至end，但不包括end）
- 如果start为NaN，那么将其替换为0
- 如果只有一个数的时，这个数就会默认为开始的位置
- 如果end超出范围时，返回从起始位置到结束的值
- 如果start和end为负数，负数从后面倒着往前数。
- 其它情况

练一练：

```
var str= '今天天气好晴朗处处好风光' ;  
console.log(str.slice(3,7));  
console.log(str.slice("abc",3));  
console.log(str.slice(3));  
console.log(str.slice(3,20));  
console.log(str.slice(-5,-2));  
console.log(str.slice(-5,2));  
console.log(str.slice(2,0));
```


◆ replace(old,new)

- 返回一个新的字符串
- 只会替换匹配到的第一个字符串

```
var str="你忘了吗?不知道,不知道,可能没忘";  
console.log(str.replace("不知道","忘记了")); //只会替换匹配到的第一个字符串  
console.log(str.replace("忘","记起"));
```

练一练:

```
var str='aaaaabbbb';  
console.log(str.replace("a",0));  
console.log(str.replace("b","d"));
```

◆ replace(old,new)

- 返回一个新的字符串
- 只会替换匹配到的第一个字符串

```
var str="你忘了吗?不知道,不知道,可能没忘";  
console.log(str.replace("不知道","忘记了")); //只会替换匹配到的第一个字符串  
console.log(str.replace("忘","记起"));
```

练一练:

```
var str='aaaaabbbb';  
console.log(str.replace("a",0)); //0aaaaabbbb  
console.log(str.replace("b","d")); //aaaaadb主bb
```

◆ split()

- 用于把一个字符串分割成字符串数组
- 第一个参数为字符串

```
var str = "abacdef";  
console.log(str.split("")); //[a,b,a,c,d,e,f]
```

- 第二个参数可选，规定返回的数组长度

```
console.log(str.split("",3)); //[a,b,a]
```

练一练：

```
var str1 = '2015-11-2-23-07';  
console.log(str1.split("-"));  
console.log(str1.split(""));  
console.log(str1.split("0"));
```

```
var str2="http://www.baidu.com";  
console.log(str2.split("."));
```

//如果想得到str2里的http这个词,该怎么做呢?

◆ concat()

- 将一个字符串或多个字符串拼接起来，获得一个新的字符串

```
var str = "abcdef"  
console.log(str.concat('123')); //abcdef123
```

- 可以有多个参数

```
var str2 = '123' ;  
console.log(str.concat(str2, "你好吗")); //abcdef123你好吗
```

练一练：

```
var str3 = '未来的你';  
var str4 = '会感谢';  
var str5 = '现在的你';  
console.log(str3.concat(str4, str5));
```

◆ indexOf("str" , "index")

- 获取第一次出现字符串位置。大小写敏感，如果没有则返回-1
- 参数str： 是在String对象中查找的子字符串
- 参数index： 该整数值指出在String对象内开始查找的索引。省略，则从字符串的开始处查找。

```
var str = "abcdedf";  
console.log(str.indexOf('a',3)); //-1,从3的位置开始查找  
console.log(str.indexOf('d',2)); //3  
console.log(str.indexOf('a')); //0
```

练一练：

```
var str2 = '未来的你会感谢现在的你';  
console.log(str2.indexOf("你",6));  
console.log(str2.indexOf("会",6));  
console.log(str2.indexOf("会"));
```

◆ lastIndexOf("str" , "index")

- 获取字符最后出现的位置。如果没有则返回-1
- 参数str: 是在String对象中查找的子字符串
- 参数index: 该整数值指出在String对象内结束查找的索引。省略, 则从字符串的开始处查找。

```
var str = "abcdedfa";  
console.log(str.lastIndexOf('a',3));//0  
console.log(str.lastIndexOf('d',6));//5  
console.log(str.lastIndexOf('a'));//7
```

练一练:

```
var str2 = '未来的你会感谢现在的你';  
console.log(str2.lastIndexOf("你",6));  
console.log(str2.lastIndexOf("会",6));  
console.log(str2.lastIndexOf("会"));  
console.log(str2.lastIndexOf("你"));
```


◆ toLowerCase()

- 将整个字符串转成小写字母

```
var str = "ABCDedfa";  
console.log(str.toLowerCase()); // abcdedfa
```

练一练:

```
var str2 = 'ABCDklSDFKDLfidsfj';  
console.log(str2.toLowerCase());
```

◆ toUpperCase()

- 将整个字符串转成大写字母

```
var str = "ABCDedfa";  
console.log(str.toUpperCase());//ABCDEDEFA
```

练一练：

```
var str2 = 'ABCDklSDFKDLFidsfj';  
console.log(str2.toUpperCase());
```

◆ navigator.userAgent

- 获取用户浏览器识别码

```
var str=window.navigator.userAgent;  
console.log(str);
```

- 转换成小写

```
var str=window.navigator.userAgent;  
str.toLowerCase( );
```

- 检索字符串

```
if(str.indexOf('chrome')!=-1){  
    alert('chrome');  
}
```

记住，在js中字符串是固定不变的，所有字符串方法都返回新字符串，原字符串本身并没有发生改变。



值得信赖的教育品牌

Tel: 400-705-9680 , Email: edu@sunplusapp.com , BBS: bbs.sunplusedu.com

