与数字类似，基本类型String也有一个对应的String对象，并且提供了很多方法。



1. 创建字符串对象（new String()）

与Number类似的，可以通过new String()创建一个String对象。

<script>

var x = "5";

var y = new String(x);

document.write("变量x的值是：" + x);

document.write("<br />");

document.write("变量x的类型是：" + (typeof x));

document.write("<br />");

document.write("变量y的值是：" + y);

document.write("<br />");

document.write("变量y的类型是：" + (typeof y));

document.write("<br />");

</script>

1. 字符串长度（length）

属性length返回字符串的长度。

<script>

var y = new String("Hello JavaScript");

document.write(y.length);

</script>

1. 返回指定位置的字符（charAt()、charCodeAt()）

charAt()返回指定位置的字符。

charCodeAt()返回指定位置的字符对应的Unicode码。

<script>

var y = new String("Hello JavaScript");

document.write("字符串y的值：" + y);

document.write("<br />");

document.write("通过charAt(0)获取位置0的字符串：" + y.charAt(0));

document.write("<br />");

document.write("通过charCodeAt(0)获取位置0的字符的Unicode码：" + y.charCodeAt(0));

</script>

1. 字符串拼接（concat()）

concat()用于进行字符串拼接。

<script>

var x = new String("Hello ");

var y = new String("JavaScript");

document.write("字符串x的值：" + x);

document.write("<br />");

document.write("字符串y的值：" + y);

document.write("<br />");

document.write("通过函数concat()把x和y连接起来：" + x.concat(y));

</script>

1. 子字符串出现的位置（indexOf()、lastIndexOf()）

indexOf()返回字符串第一次出现的位置。

lastIndexOf()返回字符串最后一次出现的位置。

<script>

var y = new String("Hello JavaScript");

document.write("字符串y的值：" + y);

document.write("<br />");

document.write("通过indexOf('a')获取子字符串“a”第一次出现的位置" + y.indexOf("a"));

document.write("<br />");

document.write("通过lastIndexOf('a')获取子字符串“a”最后出现的位置" + y.lastIndexOf("a"));

</script>

1. 比较两段字符串是否相同（localeCompare()）

localeCompare比较两段字符串是否相同，0表示相同，非0表示不同。

<script>

var x = new String("Hello");

var y = new String("Hello");

var z = new String("aloha");

document.write("通过localeCompare()判断x和y是否相等" + x.localeCompare(y));

document.write("<br />");

document.write("通过localeCompare()判断x和z是否相等" + x.localeCompare(z));

document.write("<br />");

document.write("0表示相等<br>");

document.write("1表示字母顺序靠后<br>");

document.write("-1表示字母顺序靠前<br>");

</script>

1. 截取一段子字符串（substring()）

substring()截取一段子字符串。

注：第二个参数，取不到。

<script>

var x = new String("Hello JavaScript");

document.write("字符串x的值：" + x);

document.write("<br />");

document.write("x.substring(0,3)获取位0到3的字符串：" + x.substring(0, 3));

document.write("<br />");

document.write("左闭右开，取得到0，取不到3");

</script>

1. 根据根据分隔符，把字符串转换为数组（split()）

split()根据分隔符，把字符串转换为数组。

注：第二个参数可选，表示返回数组的长度。

<script>

var x = new String("Hello This Is JavaScript");

document.write("字符串x的值：" + x);

document.write("<br />");

var y = x.split(" ");

document.write("通过空格分隔split(“ ”)，得到数组" + y);

document.write("<br />");

var z = x.split(" ", 2);

document.write("通过空格分隔split(“ ”,2)，得到数组，并且只保留前两个" + z);

</script>

1. 替换子字符串（replace()）

replace(search,replacement)

找到满足条件的子字符串search，替换为replacement。

注：默认情况下只替换找到的第一个子字符串，如果要所有都替换，需要写成：

x.replace(/a/g, "o");

或者

var regS = new RegExp("a","g");//

x.replace(regS, "o");

<script>

function p(s) {

document.write(s);

document.write("<br />");

}

var x = new String("Hello JavaScript");

p("这个是原字符串：" + x);

var y = x.replace("a", "o");

p("只替换第一个a：" + y);

var regS = new RegExp("a", "g");

var z = x.replace(regS, "o");

p("替换掉所有的a：" + z);

</script>

1. 返回基本类型（typeof）

需要注意的是，所有返回字符串类型的方法，返回的都是基本类型的String。

<script>

function p(s) {

document.write(s);

document.write("<br />");

}

var x = new String("Hello JavaScript");

var temp = x.charAt(0);

p("charAt返回的值" + temp);

p("其类型是" + typeof temp);

var temp = x.substring(0, 5);

p("substring返回的值" + temp);

p("其类型是" + typeof temp);

</script>