//字符串对象基础

//1.声明

//var 名 = ""/''

var s1 = "Aale45";

var s2 = 'fjjjjl4';

var s3 = '45';

document.write( (s1 + s2).length ); //13

//2.处理

//1.属性：length

//2.方法：

//1）字符

//1> strObj.charAt(index):返回指定索引位置处的字符

document.write( '<br/>' + s1.charAt( 0 ) + '<br/>' + s1.charAt( 1 ) ); //A,a

//2> strObj.charCodeAt(index):返回一个整数，代表指定位置上字符的 Unicode 编码

document.write( '<br/>' + s1.charCodeAt( 0 ) + '<br/>' + s1.charCodeAt( 1 ) );//65,97

//2)操作

//1>拼接

//strObj.concat([item1, item2, . . . ):返回字符串值,该值包含了两个或更多个提供的字符串的连接

//等价于 操作符 +

document.write( '<br/>' + s1.concat(s2, s3) ); //Aale45fjjjjl445

document.write( '<br/>' + s1 + s2 + s3 ); //Aale45fjjjjl445

//2>截取（负值的处理: +length / 转换为0 ；NAN转换为0 截取位置一般不包含end）

//a、stringObj.slice(start, [end]):返回字符串的片段

document.write( '<br/>' + s1.slice( 0 ) );//Aale45

document.write( '<br/>' + s1.slice( 0, -1 ) );//Aale4

//b、strVariable.substring(start, end):返回位于 String 对象中指定位置的子字符串

document.write( '<br/>' + s1.substring( 3, -1 ) ); //Aal

//如果 start 或 end 为 NaN 或者负数，那么将其替换为0==》substring( 3, 0)

//substring 方法使用 start 和 end 两者中的较小值作为子字符串的起始点==》substring( 0, 3)

//substring 方法将返回一个包含从 start 到最后（不包含 end ）的子字符串的字符串==>Aal

//c、stringvar.substr(start [, length ]):返回一个从指定位置开始的指定长度的子字符串

document.write( '<br/>' + s1.substr( 0 ) );//Aale45

document.write( '<br/>' + s1.substr( 1, 3 ) );// 3表示长度//ale

//3>空格处理

//stringObj.trim():清除前置和后缀空格,需要自己写函数实现

//3）位置

//1>strObj.indexOf(subString[, startIndex]):在一个字符串中的指定位置(从0开始)从前向后搜索，返回 String 对象中子字符串第一次出现的位置的下标（从左往右）

document.write( '<br/>' + s1.indexOf( 'e45', 0 ) );

//Aale45,确认'e45'是子字符串，等价于从A开始从前向后搜索'e'，返回的是下标3

document.write( '<br/>' + s1.indexOf( 'e45', 2 ) );

//Aale45,确认'e45'是子字符串，等价于从l开始从前向后搜索'e'，返回的是下标3

//2>strObj.lastIndexOf(substring[, fromindex]):在一个字符串中的指定位置(从0开始)从后向前搜索，返回 String 对象中子字符串第一次出现的位置的下标（从左往右）

document.write( '<br/>' + s1.lastIndexOf( "e45",0) );

//Aale45,确认'e45'是子字符串，等价于从A开始从后向前搜索'e'，找不到返回-1

document.write( '<br/>' + s1.lastIndexOf( 'e45', 3 ) );

//Aale45,确认'e45'是子字符串，等价于从e开始从前向后搜索'e'，返回的是下标3

//相同点:指定位置都是从左向右从0开始，返回的都是第一次出现的位置的下标

//不同点：搜索的方向不同，indexOf()从前往后，lastIndexOf()从后往前

//4)大小写转换

//1>strVariable.toLowerCase():返回一个字符串，该字符串中的字母被转换为小写字母

document.write( '<br/>' + s1.toLowerCase() );

//2>strVariable.toUpperCase():返回一个字符串，该字符串中的所有字母都被转化为大写字母

document.write( '<br/>' + s1.toUpperCase() );

//5)匹配方法

//1>stringObj.match(rgExp):使用正则表达式模式对字符串执行查找，并将包含查找的结果作为数组返回

var s4 = "The rain in Spain falls mainly in the plain";

var re1 = /ain/i; // 没有设置全局标志g,返回第一次出现 "ain" 的地方

var re2 = /ain/ig; // 设置全局标志g,返回的数组包含了所有 "ain"

document.write('<br/>' + s4.match(re1) +'<br/>'+ s4.match(re2));// 尝试匹配搜索字符串并打印

//2>stringObj.search(rgExp):返回与正则表达式查找内容匹配的第一个子字符串的位置

var re3 = /falls/i;

document.write( '<br/>' + s4.search(re3) );

//3>stringObj.replace(rgExp, replaceText):返回根据正则表达式进行文字替换后的字符串的复制

var s5 = "The man hit the ball with the bat while the fielder caught the ball with the glove.";

var re4 = /The/g;

document.write( '<br/>' + s5 + '<br/>' + s5.replace(re4, "A") ); // 用 "A" 替换 "The"

//4>stringObj.split([separator, limit):将一个字符串分割为子字符串，然后将结果作为字符串数组返回

//separator:可选项,字符串或 正则表达式 对象，它标识了分隔字符串时使用的是一个还是多个字符。如果忽略该选项，返回包含整个字符串的单一元素数组

//limit: 可选项,该值用来限制返回数组中的元素个数

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

var s5 = s4.split(" ", 3);

for (i in s5){

if( i <= s5.length-1 ){

document.write( s5[i] + '--' );

}else{

document.write( s5[i] + '<br/>' );

}

}

//6)静态方法

//String.fromCharCode(code1, code2, ...)：从一些 Unicode 字符值中返回一个字符串

document.write( String.fromCharCode(112, 108, 97, 105, 110) );

|  |  |
| --- | --- |
| 属性 | 描述 |
| innerHTML | 获取元素内容的最简单方法是使用 innerHTML 属性 |
| nodeName | * nodeName 是只读的 * 元素节点的 nodeName 与标签名相同 * 属性节点的 nodeName 与属性名相同 * 文本节点的 nodeName 始终是 #text * 文档节点的 nodeName 始终是 #document |
| nodeValue | * 元素节点的 nodeValue 是 undefined 或 null * 文本节点的 nodeValue 是文本本身 * 属性节点的 nodeValue 是属性值 |
| nodeType | * nodeType 是只读的  |  |  | | --- | --- | | 元素 | 1 | | 属性 | 2 | | 文本 | 3 | | 注释 | 8 | | 文档 | 9 | |