# 第14天【字符串和数组】

## 主要内容

1. 字符串概述
2. 字符串常用方法
3. 数组介绍
4. 数组的创建
5. 数组的遍历
6. 数组的调用
7. 数组的常用方法

## 学习目标

|  |  |  |
| --- | --- | --- |
| 节数 | 知识点 | 要求 |
| 第一节 | 字符串概述 | 了解 |
| 第二节 | 字符串常用方法 | 掌握 |
| 第三节 | 数组的概述 | 了解 |
| 数组的创建 | 掌握 |
| 数组的遍历 | 掌握 |
| 数组的调用 | 掌握 |
| 第四节 | 数组的常用方法 | 掌握 |

1. **字符串**

**1.1字符串的概述**

字符串就是用单引号或者双引号包裹起来的，零个或多个排列在一起的字符。JavaScript 字符串用于存储和处理文本。

’javascript‘,“”,“345”,’9-11a$‘,“xiao\_yuanLian”

字符串可以嵌套，在单引号包裹的字符串内部，应该使用双引号进行嵌套。在双引号包裹的字符串内部，应该使用单引号进行嵌套。

“I am ‘coolMan’”, ’are u “kidding” me’

总之字符串嵌套不能够出现容易引起混淆的使用方式。

**1.2转义字符**

用特殊的符号来替代在字符串当中，容易引起歧义的内容。反斜杠是一个转义字符。 转义字符将特殊字符转换为字符串字符：

转义字符 (\) 可以用于转义撇号，换行，引号，等其他特殊字符。

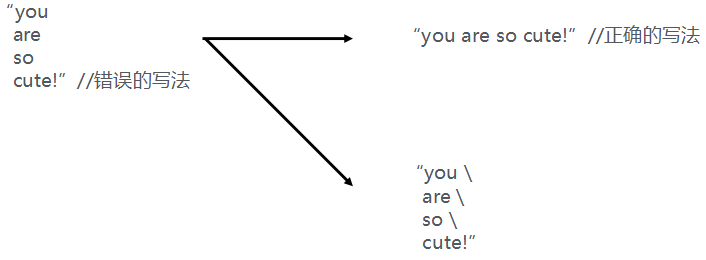


在嵌套时提到，字符串的嵌套不能够写引起混淆的方式。如果非要在字符串中使用相同的标识，就需要使用转义字符。

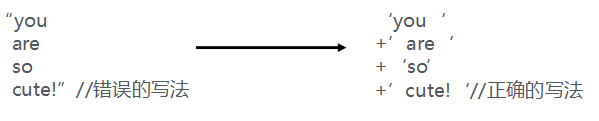
'Did she say \’Hello\'?'

"Did she say \"Hello\"?“

强调1：字符串需要在一行内定义，分成多行会报错。但是如果必须分成多行，则在每一行的末尾使用反斜杠\作为连接符。



强调2：也可以使用【(+)连接运算符】来连接多个字符串，模拟多行字符串。



**1.3字符串的属性**

**1.3.1length 属性**

1、定义

length 属性返回字符串的长度(字符数)。

2、语法

|  |
| --- |
| string.length |

3、例子

|  |
| --- |
| var txt = "Hello World!"; document.write(txt.length);//12 |

**1.4字符串的方法**

**1.4.1charAt()方法**

1、定义

charAt() 方法可返回指定位置的字符。

第一个字符位置为 0, 第二个字符位置为 1,以此类推.

2、语法

|  |
| --- |
| string.charAt(index) |

3、例子

|  |
| --- |
| var str = "HELLO WORLD"; var n = str.charAt(2)  var str = "HELLO WORLD"; var n = str.charAt(str.length-1); |

**1.4.2charCodeAt()方法**

1、定义

charCodeAt() 方法可返回指定位置的字符的 Unicode 编码。

字符串中第一个字符的位置为 0， 第二个字符位置为 1，以此类推。

2、语法

|  |
| --- |
| string.charCodeAt(index) |

3、例子

|  |
| --- |
| var str = "HELLO WORLD"; var n = str.charCodeAt(0);  var str = "HELLO WORLD"; var n = str.charCodeAt(str.length-1); |

**1.4.3concat()方法**

1、定义

concat() 方法用于连接两个或多个字符串。

该方法没有改变原有字符串，但是会返回连接两个或多个字符串新字符串。

2、语法

|  |
| --- |
| string.concat(string1, string2, ..., stringX) |

3、例子

|  |
| --- |
| var str1 = "Hello "; var str2 = "world!"; var n = str1.concat(str2);  var str1="Hello "; var str2="world!"; var str3=" Have a nice day!"; var n = str1.concat(str2,str3); |

**1.4.4substring()方法**

1、定义

substring() 方法用于提取字符串中介于两个指定下标之间的字符。

substring() 方法返回的子串包括 开始 处的字符，但不包括 结束 处的字符。

2、语法

|  |
| --- |
| string.substring(from, to) |

3、例子

|  |
| --- |
| var str="Hello world!"; document.write(str.substring(3)+"<br>"); document.write(str.substring(3,7)); |

**1.4.5substr()方法**

1、定义

substr() 方法可在字符串中抽取从 开始 下标开始的指定数目的字符。

提示： substr() 的参数指定的是子串的开始位置和长度，因此它可以替代 substring() 和 slice() 来使用。

注意： substr() 方法不会改变源字符串。

2、语法

|  |
| --- |
| string.substr(start,length) |

3、例子

|  |
| --- |
| var str="Hello world!"; var n=str.substr(2,3)  var str="Hello world!"; var n=str.substr(2) |

**1.4.6slice()方法**

1、定义

slice(start, end) 方法可提取字符串的某个部分，并以新的字符串返回被提取的部分。

使用 start（包含） 和 end（不包含） 参数来指定字符串提取的部分。

字符串中第一个字符位置为 0, 第二个字符位置为 1, 以此类推。

提示： 如果是负数，则该参数规定的是从字符串的尾部开始算起的位置。也就是说，-1 指字符串的最后一个字符，-2 指倒数第二个字符，以此类推。

2、语法

|  |
| --- |
| string.slice(start,end) |

3、例子

|  |
| --- |
| var str="Hello world!"; var n=str.slice(1,5);  var str="Hello world!"; var n=str.slice(3);  var str="Hello world!"; var n=str.slice(0,1);  var str="Hello world!"; var n=str.slice(-1); |

**1.4.7indexOf()方法**

1、定义

indexOf() 方法可返回某个指定的字符串值在字符串中首次出现的位置。

如果没有找到匹配的字符串则返回 -1。

注意： indexOf() 方法区分大小写。

2、语法

|  |
| --- |
| string.indexOf(searchvalue,start) |

3、例子

|  |
| --- |
| var str="Hello world, welcome to the universe."; var n=str.indexOf("welcome");  var str="Hello world, welcome to the universe."; var n=str.indexOf("e");  var str="Hello world, welcome to the universe."; var n=str.indexOf("e",5); |

**1.4.8lastIndexOf()方法**

1、定义

lastIndexOf() 方法可返回一个指定的字符串值最后出现的位置，如果指定第二个参数 start，则在一个字符串中的指定位置从后向前搜索。

注意： 该方法将从后向前检索字符串，但返回是从起始位置 (0) 开始计算子字符串最后出现的位置。 看它是否含有字符串。

开始检索的位置在字符串的 start 处或字符串的结尾（没有指定 start 时）。

如果没有找到匹配字符串则返回 -1 。

注意：lastIndexOf() 方法是区分大小写的！

2、语法

|  |
| --- |
| string.lastIndexOf(searchvalue,start) |

3、例子

|  |
| --- |
| var str="I am from runoob，welcome to runoob site.";  var n=str.lastIndexOf("runoob");  var str="I am from runoob，welcome to runoob site.";  var n=str.lastIndexOf("runoob", 20);  var str="I am from runoob，welcome to runoob site.";  var n=str.lastIndexOf("runoob", 9); |

**1.4.9trim()方法**

1、定义

trim() 方法用于删除字符串的头尾空格。

trim() 方法不会改变原始字符串。

2、语法

|  |
| --- |
| string.trim() |

3、例子

|  |
| --- |
| var str = " Runoob ";  alert(str.trim()); |

**1.4.10toUpperCase()方法**

1、定义

toUpperCase() 方法用于把字符串转换为大写。

2、语法

|  |
| --- |
| string.toUpperCase() |

3、例子

|  |
| --- |
| var str="Runoob";  document.write(str.toUpperCase()); |

**1.4.11toLowerCase()方法**

1、定义

toLowerCase() 方法用于把字符串转换为小写。

2、语法

|  |
| --- |
| string.toLowerCase() |

3、例子

|  |
| --- |
| var str="Runoob";  document.write(str.toLowerCase()); |

**1.4.12replace()方法**

1、定义

replace() 方法用于在字符串中用一些字符替换另一些字符，或替换一个与正则表达式匹配的子串。

2、语法

|  |
| --- |
| string.replace(searchvalue,newvalue) |

3、例子

|  |
| --- |
| var str="Visit Microsoft! Visit Microsoft!";  var n=str.replace("Microsoft","Runoob"); |

**1.4.13search()方法**

1、定义

search() 方法用于检索字符串中指定的子字符串，或检索与正则表达式相匹配的子字符串。

如果没有找到任何匹配的子串，则返回 -1。

2、语法

|  |
| --- |
| string.search(searchvalue) |

3、例子

|  |
| --- |
| var str="Visit Runoob!";  var n=str.search("Runoob");  var str="Mr. Blue has a blue house"; document.write(str.search("blue")); |

**1.4.14match()方法**

1、定义

match() 方法可在字符串内检索指定的值，或找到一个或多个正则表达式的匹配。如果没有找到任何匹配的文本， match() 将返回 null。

2、语法

|  |
| --- |
| string.match(regexp) |

3、例子

|  |
| --- |
| var str="The rain in SPAIN stays mainly in the plain";  var n=str.match(ain); |

**1.4.15split()方法**

1、定义

split() 方法用于把一个字符串分割成字符串数组。

提示： 如果把空字符串 ("") 用作 separator，那么 stringObject 中的每个字符之间都会被分割。

注意： split() 方法不改变原始字符串。

2、语法

|  |
| --- |
| string.split(separator) |

3、例子

|  |
| --- |
| var str="How are you doing today?"; var n=str.split(" ");  var str="How are you doing today?"; var n=str.split();  var str="How are you doing today?"; var n=str.split("");  var str="How are you doing today?"; var n=str.split("o"); |



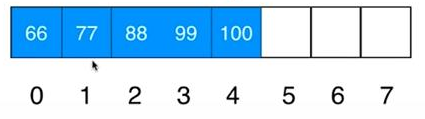
1. **数组**

**2.1数组的概念**

数组(array)是按照一定顺序排列的一组值，每个值都拥有自己的编号，编号从0开始。

|  |
| --- |
| var arr = [item1,item2,item3]; |

上述代码中的item1、item2和item3元素就构成一个数组。两端的方括号是数组的标志。因为数组元素编号从0开始，所以item1是数组的第0个元素。



item1、item2和item3是泛指，javascript中的数组元素可以是任何类型的数据。

var arr = [‘123’,123,true,null,undefined,{'name':'frank'}];这是数组么？

**2.2数组的本质**

typeof运算符会返回数组的类型是object。本质上，数组是对象类型的一种特殊表现形式。因此创建的时候我们可以使用new方式来创建。

|  |
| --- |
| var nameArr = new Array('LiLei', 'HanMeiMei', 'LiMing');  console.log(typeof nameArr);//object |

**2.3数组的创建**

1、字面量法

|  |
| --- |
| var 数组名 = [数组元素1,数组元素2,...]; |

2、new方法

|  |
| --- |
| var 数组名 = new Array(数组元素1,数组元素2,...); |

**2.4数组的长度**

计算数组长度的时候只需要计算数组元素的个数即可

数组的长度 = 最后一个下标 + 1

|  |
| --- |
| arr.length |

|  |
| --- |
| var ageArr = [18,24,30];  console.log(ageArr.length);//3 |

length是一个可写属性：

如果设置length长度小于数组本身长度，那么多余元素舍弃。

如果设置length长度大于数组本身长度，那么缺少元素用空位补齐。

如果设置length长度不是合法数值，那么会报错Invalid array length

|  |
| --- |
| var arr = [55,66,77,88];  arr.length= 2;  console.log(arr);//[55, 66]  arr.length = 4;  console.log(arr);//[55, 66, empty × 2]  arr.length = -1;  console.log(arr);// RangeError |

**2.5数组的遍历**

如果想要连续访问数组中的每个元素，可以使用for in快速遍历或者for循环。

|  |
| --- |
| var heroArr = ['GreenJuPeople','BlackGuaFu','MetalXia'];  for(var i = 0; i < heroArr.length; i++){  console.log(heroArr[i]);  }  var heroArr = ['GreenJuPeople','BlackGuaFu','MetalXia'];  for(var i in heroArr){  console.log(heroArr[i]);  } |

需要引起注意的是，循环遍历数组的时候，for循环中的i表示的数组的下标。并不是表示数组元素！

**2.6数组的空位**

当数组的某个位置是空元素，即两个逗号之间没有任何值，我们称该数组存在空位(hole)。有趣的是：如果最后一个元素后面有逗号，这种情况并不会产生空位。也就是说最后一个元素后面的逗号有和没有结果都是一样的。(当然这种写法并不推荐。)

|  |
| --- |
| var countryNameArr = ['China',,'Japan'];  countryNameArr.length//3  var countryName = ['China','American','Japan',];  countryName.length//3 |

**2.7数组的调用**

数组的调用实际上指的是数组元素的调用。数组元素通过【数组名+下标】的方式来进行访问。数组元素一次只能访问一个，不能一次连续访问多个数组元素。

|  |
| --- |
| var arr = ["张一山","李现","宋仲基"];  console.log("我的男盆友是"+arr[0]);//我的男盆友是张一山 |

**2.8数组的方法**

**2.8.1isArray方法**

1、定义

isArray() 方法用于判断一个对象是否为数组。

如果对象是数组返回 true，否则返回 false。

2、语法

|  |
| --- |
| Array.isArray(obj) |

3、例子

|  |
| --- |
| var fruits = ["Banana", "Orange", "Apple", "Mango"];  console.log(Array.isArray(fruits)); |

**2.8.2push()方法**

1、定义

push() 方法可向数组的末尾添加一个或多个元素，并返回新的长度。

注意： 新元素将添加在数组的末尾。

注意： 此方法改变数组的长度。

2、语法

|  |
| --- |
| array.push(item1, item2, ..., itemX) |

3、例子

|  |
| --- |
| var fruits = ["Banana", "Orange", "Apple", "Mango"]; fruits.push("Kiwi")  var fruits = ["Banana", "Orange", "Apple", "Mango"]; fruits.push("Kiwi","Lemon","Pineapple") |

**2.8.3pop()方法**

1、定义

pop() 方法用于删除数组的最后一个元素并返回删除的元素。

注意：此方法改变数组的长度！

2、语法

|  |
| --- |
| array.pop() |

3、例子

|  |
| --- |
| var fruits = ["Banana", "Orange", "Apple", "Mango"]; fruits.pop(); |

**2.8.4unshift()方法**

1、定义

unshift() 方法可向数组的开头添加一个或更多元素，并返回新的长度。

注意： 该方法将改变数组的数目。

2、语法

|  |
| --- |
| array.unshift(item1,item2, ..., itemX) |

3、例子

|  |
| --- |
| var fruits = ["Banana", "Orange", "Apple", "Mango"]; fruits.unshift("Lemon","Pineapple"); |

**2.8.5shift()方法**

1、定义

shift() 方法用于把数组的第一个元素从其中删除，并返回第一个元素的值。

注意： 此方法改变数组的长度！

2、语法

|  |
| --- |
| array.shift() |

3、例子

|  |
| --- |
| var fruits = ["Banana", "Orange", "Apple", "Mango"]; fruits.shift() |

**2.8.6slice()方法**

1、定义

slice() 方法可从已有的数组中返回选定的元素。

slice()方法可提取字符串的某个部分，并以新的字符串返回被提取的部分。

注意： slice() 方法不会改变原始数组。

2、语法

|  |
| --- |
| array.slice(start, end) |

3、例子

|  |
| --- |
| var fruits = ["Banana", "Orange", "Lemon", "Apple", "Mango"]; var citrus = fruits.slice(1,3);  var fruits = ["Banana", "Orange", "Lemon", "Apple", "Mango"]; var myBest = fruits.slice(-3,-1);  var str="www.runoob.com!"; document.write(str.slice(4)+"<br>"); // 从第 5 个字符开始截取到末尾 document.write(str.slice(4,10)); // 从第 5 个字符开始截取到第10个字符 |

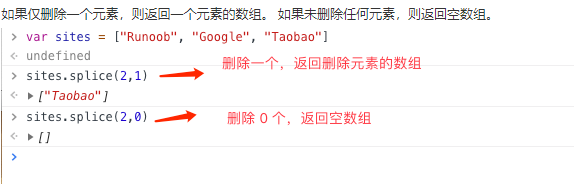
**2.8.7splice()方法**

1、定义

splice() 方法用于添加或删除数组中的元素。

注意：这种方法会改变原始数组。

如果仅删除一个元素，则返回一个元素的数组。 如果未删除任何元素，则返回空数组。



2、语法

|  |
| --- |
| array.splice(index,howmany,item1,.....,itemX) |

3、例子

|  |
| --- |
| //数组中添加新元素：  var fruits = ["Banana", "Orange", "Apple", "Mango"]; fruits.splice(2,0,"Lemon","Kiwi");  //移除数组的第三个元素，并在数组第三个位置添加新元素:  var fruits = ["Banana", "Orange", "Apple", "Mango"]; fruits.splice(2,1,"Lemon","Kiwi");  //从第三个位置开始删除数组后的两个元素：  var fruits = ["Banana", "Orange", "Apple", "Mango"]; fruits.splice(2,2); |

**2.8.8join()方法**

1、定义

join() 方法用于把数组中的所有元素转换一个字符串。

元素是通过指定的分隔符进行分隔的。

2、语法

|  |
| --- |
| array.join(separator) |

3、例子

|  |
| --- |
| //把数组中的所有元素转换为一个字符串：  var fruits = ["Banana", "Orange", "Apple", "Mango"]; var energy = fruits.join();  //使用不同的分隔符：  var fruits = ["Banana", "Orange", "Apple", "Mango"]; var energy = fruits.join(" and "); |

**2.8.9indexOf()方法**

1、定义

indexOf() 方法可返回数组中某个指定的元素位置。

该方法将从头到尾地检索数组，看它是否含有对应的元素。开始检索的位置在数组 start 处或数组的开头（没有指定 start 参数时）。如果找到一个 item，则返回 item 的第一次出现的位置。开始位置的索引为 0。

如果在数组中没找到指定元素则返回 -1。

2、语法

|  |
| --- |
| array.indexOf(item,start) |

3、例子

|  |
| --- |
| //查找数组中的 "Apple" 元素：  var fruits = ["Banana", "Orange", "Apple", "Mango"];  var a = fruits.indexOf("Apple");  //查找数组中 "Apple" 的元素, 在数组的第四个位置开始检索：  var fruits=["Banana","Orange","Apple","Mango","Banana","Orange","Apple"]; var a = fruits.indexOf("Apple",4); |

**2.8.10lastIndexOf()方法**

1、定义

lastIndexOf() 方法可返回一个指定的元素在数组中最后出现的位置，从该字符串的后面向前查找。

如果要检索的元素没有出现，则该方法返回 -1。

该方法将从尾到头地检索数组中指定元素 item。开始检索的位置在数组的 start 处或数组的结尾（没有指定 start 参数时）。如果找到一个 item，则返回 item 从尾向前检索第一个次出现在数组的位置。数组的索引开始位置是从 0 开始的。

如果在数组中没找到指定元素则返回 -1。

2、语法

|  |
| --- |
| array.lastIndexOf(item,start) |

3、例子

|  |
| --- |
| //查找数组元素 "Apple"出现的位置：  var fruits = ["Banana", "Orange", "Apple", "Mango"]; var a = fruits.lastIndexOf("Apple");  //查找数组中"Apple"的位置：  var fruits=["Banana","Orange","Apple","Mango","Banana","Orange","Apple"]; var a = fruits.lastIndexOf("Apple");  //数组中的第四个位置查找字符串 "Apple"出现的位置：  var fruits=["Banana","Orange","Apple","Mango","Banana","Orange","Apple"]; var a = fruits.lastIndexOf("Apple",4); |