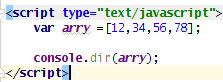
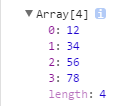
JS数组基础知识

声明数组，并且查看数组的属性值



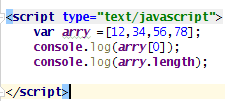
数组也是对象，我们看到的是属性值，它的属性名是它的下标值（也成为索引），从零开始。

控制台输出结果：

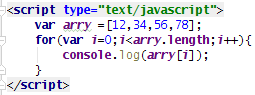


Length:代表的是数组的长度。

通过索引可以获取数组的属性值以及通过length获取数组的长度：



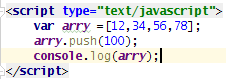
通过循环遍历数组：



注意：若是这样打印出的是数组属性值得坐标：console.log(i);

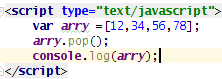
JS数组的常用方法

尾部添加属性值：



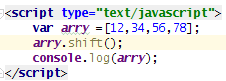
注意：返回值是数组的长度。通过arry[arry.length]=100；也可以增加。length是数组的长度，length-1是数值下标值得长度。

删除尾部属性值：



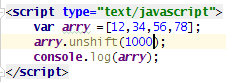
注意：此方法有返回值：是被删除的那一项的值 即var reg=arry.pop(); console.log(reg); 可获取被删除的属性值。同arry.length--；具有同样的功能。

删除首部部属性值：

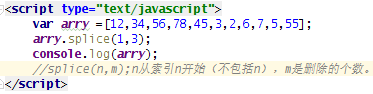


注意：有返回值，是被删除的那一项的属性值。

首部添加一项属性值：



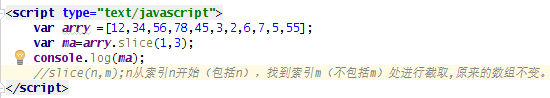
查找删除/增加/复制数组属性值：



注意：有返回值：是被删除的属性值（一个新的数值）；splice(n);从n索引开始，一直删除到数值尾部。n=0清空数组。Splice()；返回一个空数组。Splice(n,m,x);删除后有x替换；

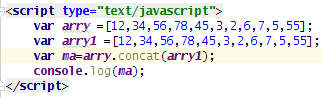
Spalice(n,0,m);一项都不删除，在索引n前后面增加属性值。

截取数组的某一部分：



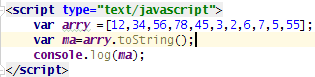
注意：slice(0)/slice();克隆一份。Slice(n);从n开始截取到末尾。

拼接数组：

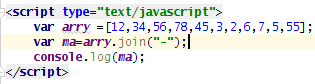


注意：原来的数组没有变（数组1和数组2）；

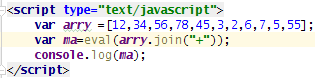
把数组转化为字符串：



注意：原来的数组没变。 下面是按照拼接符来把数组转化为字符串（原数组不变）：

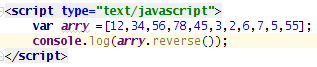


数组求和：



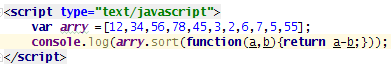
注意：eval是JS中把有运算符的字符串转变为表达式进行计算的方法。

数组的排序倒过来：



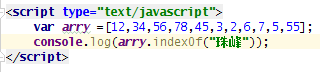
注意：原来的数组改变了

数组由小到大排序：



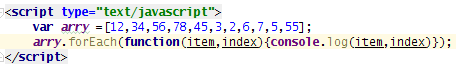
注意：由大到小return b-a；就可以了。原来数组会发生改变

判断是否为该数组的成员属性：



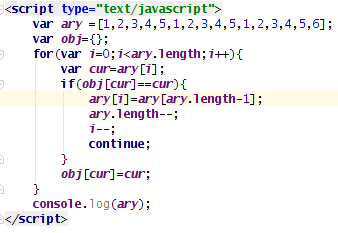
注意：该方法IE6-8下不兼容，若是不是成员属性则返回-1；

遍历数组中的每一项：



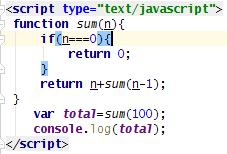
注意：该方法在IE6-8中不兼容，原来的数组不变

数组的去重：



注意：以上方法时利用对象的键值对来进行数组的去重。

递归：



注意：以上是利用递归进行1-100的累加

冒泡排序：

快速排序：

插入排序：