

## aardio 范例: 多维数组

```
//多维数组
import console;

/*
aardio 中的数组可以包含一切, 数组成员可以是任意类型,
当然数组的成员也可以是数组, 数组可以任意的嵌套包含其他数组。
aardio数组的成员不必限定类型、限定维数, 数组也不必先声明大小就可以使用, 是非常灵活方便的动态数组。
参考文档: https://www.aardio.com/zh-cn/doc/language-reference/datatype/table/\_.html
*/
var array = {};

array[1] = {};
array[1][1] = {};
array[1][1][1] = {};

//这是一维数组
var array = { 11;12;13;14 };

//这是一个二维数组, 其实就是数组的数组
var array = {
    {11;12;13;14};
    {21;22;23;24};
    {31;32;33;34};
}
console.dump( array[2][3] ); //显示第2行第三列的数据

//其实所谓多维数组, 只是方便人们理解的一种说法, 多维数组也可以用一维数组来表示, 例如:
var array = {
    11;12;13;14;
    21;22;23;24;
    31;32;33;34;
    get = lambda(row,col) owner[ (row-1) * 4 + col ];
}

console.dump( array.get(2,3) ); //获取第2行第3列的数据,

/*
其实在编程中, 大多时候, 我们使用多维数组, 并不必要先声明大小和类型,
在aardio中也提供 table.array 函数可以预分配指定大小、维数、并拥有指定默认值的数组。

可以在参数中使用任意个参数指定下一维的数组长度,
前面的参数指定父数组长度, 后面的参数指定子数组长度, 最后一个参数指定默认值
*/
var array = table.array( 5/*行*/,5/*列*/,7/*默认值*/ );
console.dump(array);

//默认值也可以指定一个table对象(数组最后一维的成员都会复制一个新的table对象作为默认值, 而不是指向同一个table对象)
var array = table.array( 5,5,{ } );
console.log( array[1][1] != array[2][2] );

//默认值还可以是一个用来创建默认值的函数(或 lambda 表达式 ), 并且可以在函数后面指定调用该函数的参数
var array = table.array( 5, table.array,5,{ } );
console.dump( array );

//我们甚至可以不用指定数组长度, 仅指定一个函数, table.array会循环调用该函数并将所有非null返回值返回为数组
var array = table.array( string.gmatch("abcdef",".") ); //调用string.gmatch返回的迭代器恰好符合要求
console.dump( array );

var arr = {
    {50,80};
    {20,24};
    {100,103};
    {4,8};
}

//多维数组排序, 注意两值相等不要返回 true
table.sort(arr,function(v) {
    for(i=1;#owner) if(owner[i]!=v[i]) return owner[i] < v[i];
})

console.dump(arr);
console.pause();
```