

aaudio 范例: 含 null 数组

```
//含 null 数组
import console.int;
```

```
/*
有很多编程语言甚至不允许数组出现不同的数据类型，
或者根本就不允许改变数组长度。
```

aaudio 的数组虽然非常灵活，
可包含任意类型，且长度可以任意改变，但也不是完全没有限制。
aaudio 数组包含 null 值时，#操作符取到的数组长度可能包含 null，也可能不包含。
这是一种性能优化措施。

所以如果你在寻找十全十美设计的编程语言，aaudio 肯定不是！
即使在支持含 null 数组的编程语言中，数组中包含 null 值的做法也非常罕见且不必要。
*/

```
//数值索引不连续的数组，正规的叫法是『稀疏数组』。
var tab = {1,[9000000000000000]=90000000000000};
```

```
//如果数组可以包含 null 值，试想一下屏幕会不会炸掉？！
console.dump( tab );
```

```
/*
如果表包含 null 值，也就是数值索引不连续，那么就不应当再作为『有序数组』使用。
aaudio 中未加特别说明时『数组』一律特指数值索引连续不间断的『有序数组』。
并且数组索引自 1 开始。
*/
```

```
/*
用 table.array 可创建包含 null 值的定长数组。
参数 @2 指定数组最小长度，如果为 -1，则自动获取最大索引作为最小长度。
*/
var arr = table.array({1,null,null,null,null,5,6},-1);
```

```
/*
aaudio 对这种定长数组提供有限支持，
# 操作符与大部分依赖 # 操作符的数组函数都不支持这种数组。
```

但是我们可以用 table.len 取到这种数组的长度。
可以看看这些函数的源码，实际上 aaudio 对来自各种编程语言接口的稀奇古怪的数组都尽可能做了兼容，
兼容与完美都是要付出运算成本的，不是没有代价的。
完美也总是会伴随着新的缺陷。

```
*/
console.log(table.len(arr))
```

```
//可以用 console.dumpTable, util.table.stringify 显示 null 值。
console.dumpTable(arr)
```

```
//支持 web.json, console.dumpJson 输出含 null 值的数组。
console.dumpJson(arr)
```

```
//也可以创建空数组
var arr = table.array()
```

```
//显示 [] 而不是 {}
console.dumpJson(arr)
```

```
//支持 table.type(), table.isArray() 等判断数组类型的函数
console.log(table.type(arr))
```

[Markdown 格式](#)