

aaudio 范例: 动态指针

```
//动态指针
import console;

/*
动态指针可以自主控制内存何时分配何时释放，在拼接字符串时速度快。
但是动态指针也比较危险，每次重新分配都有可能改变指针的地址，原来的指针会失效，新的指针在必要时需要释放，
这个过程中非常容易出错，必须非常小心的使用。
如果没有必要使用动态指针，应当优先使用 raw.buffer 分配内存创建 buffer 对象。
*/

/*
重新分配内存大小，可选在参数2中传入原来的内存指针，
调整内存大小时动态指针的地址是可能会动态改变的，所以我们称其为动态指针，
切记总是要接收 raw.realloc, raw.concat 返回的新指针并更新原来的动态指针变量。

如果不指定初始值，aaudio不会初始化内存，
如果原来已经有数据，重新分配内存大小时不会清除原来的数据。
但如果指定了初始值会清空原来的数据。
*/
var p = raw.realloc(100);

for(i=1;2000;1){
    //追加字符串,如果参数是字符串,aaudio会在最后一个字符串尾部放上'\u0000'
    p = raw.concat(p,tostiring(i)); //必须随时更新动态指针变量p的实际地址
    p = raw.concat(p," ") //必须随时更新动态指针变量p的实际地址
}

console.log( "存储内容长度,内存大小",raw.sizeof(p) );
console.log( raw.tostiring(p) ); //转为字符串,对于文本,因为有'\u0000'所以可自动得到长度

//内存长度为0则释放动态指针,记住动态指针变量p也要用返回值设为空值
p = raw.realloc(0,p);
console.pause(true);
```

[Markdown 格式](#)