

# aaudio 范例: 结构体

```
//结构体
import console;

/*
在 aaudio 中结构体就是表对象，
但在每个成员名前指定了该成员的原生类型。
如下 x,y 成员的类型就是 int 类型（32位有符号整数，大写表示32位无符号整数）。

相关文档： https://www.aaudio.com/zh-cn/doc/library-guide/builtin/raw/struct.html
*/
var pt = {
    int x;
    int y;
}
/*
注意：原生类型必须写在结构体或者 API 声明内部。
*/

//结构体传入 API 参数会自动转换为原生结构体指针（调用结束自动释放原生指针）
::User32.GetCursorPos(pt);
console.log(pt.x,pt.y);

/*
不声明直接调用 API 时，所有结构体参数都是输出参数。
所以下面这样写也是可以的：
*/
var ret,pt = ::User32.GetCursorPos({int x:int y;});

//结构体会自动初始化，所以 0, null, false 等值不必如下显式指定。
var ret,pt = ::User32.GetCursorPos({int x = 0:int y = 0;});

//我们可以用 class 声明结构体类型
class PT{
    int x;
    int y;
}

//用结构体类型创建结构体
var pt = PT();
::User32.GetCursorPos(pt);

//aaudio 已经默认声明了 ::POINT, ::RECT 这些常用结构体
var pt = ::POINT();
::User32.GetCursorPos(pt);
console.log(pt.x,pt.y);

//结构体在调用 API 时转换为原生指针，调用结束释放原生指针。
//如果不想在调用结束自动释放结构体指针，就需要先将结构体复制到内存，如下。
var ptBuffer = raw.buffer({int x:int y;});
::User32.GetCursorPos(ptBuffer); //使用结构体指针作为参数调用 API
var pt = raw.convert(ptBuffer, {int x:int y;}); //自内存中还原结构体
console.log(pt.x,pt.y);

console.pause();
```