

aardio 范例: 编码转换

```
//编码转换
import console;
/*
aardio 使用 Unicode 内核, 并以 UTF-8 为默认编码, 可以完全避免乱码的危害。
aardio 字符串存在神奇的 UTF 自动标记功能, 用 aardio 调用 ANSI 编码的控制台程序,
又或是编写 UTF-8 编码的网页, 又或是调用 UTF-16 编码的操作系统接口, 都可以做到自动识别, 自动转换。
我们几乎不需要用到下面这些编码转换函数, 因为 aardio 都会自动处理好。
```

注意转换编码的目的仅仅是改变存储编码 —— 并保持显示的字符不改变。
当然 ANSI 编码可显示的字符比 Unicode 少得多, 转换后可能出现无法正常显示的字符。

参考文档: <https://www.aardio.com/zh-cn/doc/language-reference/datatype/string.html>
*/

```
//aardio 字符串默认为 UTF-8 编码
var str = "aardio代码中的字符串字面量为UTF-8编码";
console.log("是否UTF8",str,string.isUtf8(str));

//转换为 Unicode(UTF-16) 编码
str = string.toUtf16(str,65001);
console.log("是否UTF16",str,string.isUtf16(str));

//Unicode 转换为 ANSI 编码
str = string.fromUtf16(str,0);
console.log("ANSI",str,string.getUtf(str));

//从一种编码转换另一种编码可以使用 string.fromto 函数
console.log("UTF8->ANSI",string.fromto("转换编码",65001,0) );

console.pause();
```