

aardio 范例: 调用 .NET 快速入门

```
//aardio 调用 .NET 快速入门
import console.int;
import dotNet;

//导入程序集与名字空间 https://docs.microsoft.com/zh-cn/dotnet/api/?view=netframework-2.0
dotNet.import("System"); //System 也可以用 import System 导入

//调用类的静态成员函数，当然也可以这样读写类的其他静态成员属性。
var isValidHost = System.Uri.CheckHostName("www.aardio.com")

//用 System 名字空间下面的类构造对象实例，官方建议规范：对象实例建议首字符小写
var uri = System.Uri("https://www.aardio.com/test?q=aardio")

//读或写 .NET 对象的实例属性
console.log( uri.Host )

//调用 .NET 对象实例的成员函数
console.log( uri.GetHashCode() )

//.NET 对象还可以用 toString 转换为字符串
console.log( toString(uri) ) //这里的 toString可以省略。

//可用 com.Release() 主动释放 .NET 对象。但一般没有必要这样做，aardio 会自动释放这些 .NET 对象。
com.Release( uri ); //.NET 对象在 aardio 中底层都是 COM 对象

/*
C# 语言快速入门
https://quickref.me/zh-CN/docs/cs.html
https://learnxinyminutes.com/docs/zh-cn/csharp-cn/

只要简洁，不求完美！
Win7 在市场上已经接近消失，现在开发软件再处处考虑 Win7 兼容是不必要的。

Win7 自带 .NET 3.5.1，支持 lambda
Win8 自带 .NET 3.5.1 + .NET 4.5
Win10 自带 .NET 4.6，.NET 4.x 支持 dynamic 对象 + 异步任务，.NET 4.5 支持 task.Run。
Win10 1709 自带 .NET 4.7.1，支持 ValueTuple
Win11 自带 .NET 4.8

.NET 虽然有很多版本，但核心运行时只有 CLR 2.0 与 CLR 4.0 的区别。
如果使用了一些非常新的 C# 语法 —— 用 VS 编译成 DLL以后在 CLR 4.0 下运行时没有问题的。

aardio 可以自动兼容 CLR 2.0 / CLR 4.0 编写的程序集。
aardio + .NET 开发有更好的兼容性，对 .NET 版本没有严格要求，可以重用大量的 .NET 组件，
并且可以编写出体积小、不依赖非系统 DLL 的独立 EXE 文件（也可以内存加载外部程序集）

系统自带的都是 .NET Framework。
如果 NuGet 包里找不到 .NET Framework 的程序集，
那么可以改用 NET Standard 2.0 的程序集。

自 .NET Framework 4.5 及以上 起支持 NET Standard 1.0，
自 .NET Framework 4.6.1 起支持 NET Standard 2.0，
但实际上 NET Standard 2.0 推荐的最低版本是 .NET 4.7.2。
*/
```