**【DSAL 快速入门】**

1. **什么是DSAL？**
2. DSAL（Develop Script Active Language）扩展脚本流行语言

**二、我能用DSAL做什么？**

1、DSAL以支持Microsoft ActiveX的脚本语言为基础

2、DSAL可以跨脚本语言运行

3、DSAL具有易读性，高效性，精确性

4、DSAL可以面向各种经验的开发者供他们创建脚本

5、DSAL Studio是轻量级的程序系统

6、DSAL Console作为DSAL的核心控制台，运行代码具有不可逆性

7、在运行过程中，DSAL脚本基本保证不在计算机留下数据

**三、我该怎么运行DSAL脚本？**

DSAL Console是运行DSAL脚本的基础，DSAL Studio则为扩展内容和启动器；

初次体验建议先使用DSAL Console学习一段时间，再尝试使用DSAL Studio为你的工作增添色彩；

**四、这种程序是安全的吗？**

我们保证DSAL Console以及DSAL Studio内无有害内容或程序，但对于你编写的内容的安全性我们不做任何保证，比使用DSAL编写有害脚本将与DSAL 无关

**五、基础语法**

我们知道，DSAL语言基于支持Mircosoft ActiveX的脚本语言，所以我们主要需要掌握的是JS（JavaScript）和VBS（VisualBasicScript）

DSAL控制台的基本语法如下；

【设置基本语言】/Language

【DSAL运行时】/DSAL Out

我们同样清楚的知道：JS的脚本文件是.js为后缀的，VBS的脚本语言是由.vbs为后缀的

DSAL没有像这样增添一种新的后缀！DSAL脚本通常以文本的形式呈现到开发者面前。例如；

function DSAL(){return 'Hello world!'}

/Language

VBScript

如上的代码我们使用普遍的语法来解释，即；

【JS代码】

【/设置语言】

【VBS脚本语言】

当然我们也可以继续写下去

function DSAL(){return 'Hello world!'}

/Language

VBScript

function DSAL()msgbox(“Hello world!”)end function

理解成；

【JS代码】

【/设置语言】

【VBS脚本语言】

【VBS代码】

#你已经看到了DSAL的基础部分，现在我们来稍微学习一些语法！#

function DSAL(){return 'Hello world!'}

/Language

VBScript

function DSAL()msgbox(“Hello world!”)end function

我们可以看到上述例子代码中可以发现循环使用function DSAL()

在DSAL运行环境下，为了避免事件与事件之间的冲突和代码冲突

我们需要声明当前运行的是DSAL的事件

相信VBS和JS足够好的读者就能知道；

在VBS中function的语法

Function 名称()

‘code

End Function

而在JS中则为

Function 名称(){‘code}

但为了保证DSAL脚本的逐行运行规范，我们将每一件function归入同一行中！

【注意】function只能代表此行的代码，下一行需要重新使用function DSAL语法，无论是否改变基本运行语言！

为了防止控制台漏读，有时候代码需要优化或者空行

**#基本教程到此为止#**