AppleScript 的终极入门手册

AppleScript 是 Mac OS X内置的一种功能强大的脚本语言，使用 AppleScript 的目的是把一些重复繁琐并且耗费时间的任务自动化。比如我是一个自由职业者，我十分的讨厌每个星期为不同的客户去创建帐单，为了解决这个问题我写了一个 AppleScript 来读取我在 iCal 里输入的时间，然后自动在 Microsoft Excel 里根据这些时间创建出帐单并且自动通过邮件发送给我的客户，要完成所有的这一切动作只需要轻轻的按一个按钮。

AppleScript 的一个优点是你不必是一个天才程序员才能使用它，事实上你甚至不需要有任何编程的经验！这篇文章会告诉你如何通过隐藏在每个应用程序框架里的简单指令去为几乎任何的应用程序写一个 AppleScript。感兴趣了？那就继续读下去吧～

The Main Window

从这里开始：**Tell** 命令块

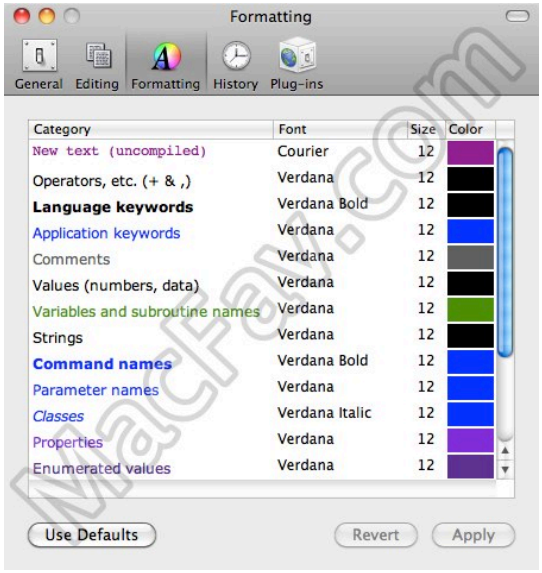
要创建一个 AppleScript时，只要打开 Applications 文件夹里的 Utilities 里面的 AppleScript Editor 就可以开始了。打开后你会看到上图类似的窗口，这个是 AppleScript 编辑器的主窗口，尝试在里面输入以下的代码：

```
tell application "Finder"

     display dialog "Hello World"

end tell
```

AppleScript 这种脚本语言本身尽可能使用平实的英语作为语法来让代码特别容易理解和阅读。AppleScript 的大多数命令会位于 Tell 命令块内，它叫 “Tell 命令块” 是由于你在“告诉”应用程序你想它做些什么。比如上面所说的三行代码告诉 Finder 这个应用程序显示一个包含 “Hello World” 字符串的对话框。当你在 Tell 命令块内写完了你想要应用程序做的一个或多个命令后，你必须要以 “end tell” 命令来结束这个 Tell 命令块。

小技巧：初学者最容易犯的错误就是忘了最后加上 ***“end tell”*** 来结束 ***Tell*** 命令块，如果把它忘了，脚本将会编译失败。

当你在编辑器内完成输入上面的代码后，按下上面的 “Compile” 铁锤图标。如果你的语法是正确的话，你所输入的代码会被自动格式化并改变颜色。如果你输入的时候出现了错误，当你编译的时候编辑器会告诉你哪一块出错了， 并尝试告诉你它认为错误是什么。下面是编辑器默认的一些颜色分别代表什么，你可以在菜单栏的 AppleScript Editor->Preferences里看到并修改。



Syntax Color

你的代码成功编译后，再按 “Run” 图标，你应该会看到以下的对话框：

Hello World

现在按下 “OK” 按纽并且看一下编辑器下方。当你运行一脚本时，编辑器会告诉你运行的结果是什么，或者什么被“返回”了，在我们的例子里，它告诉我们 “OD” 按纽被按下了。

声明变量

变量在每一种编程语言里面的意义都是一样的，程序代码利用变量方便的对许多信息进行读取或运算。但在每一种语言里创建或“声明”变量是不同的，在 AppleScript 里你会像下面这样声明变量：

```
set theString to "Hello World"

tell application "Finder"

     display dialog theString

end tell
```

上面的例子中有几处地方需要注意。首先，我们注意到变量的声明是通过 “set” 和 “to” 命令来进行。在例子中我们 “set” 了我们的变量（“theString”）为一个值（“Hello World”）。在很多编程语言里程序员被要求必须声明变量的类型（如整型、浮点型、字符串等等），但 AppleScript 足够的聪明而不需要程序员声明类型。

另外，请注意我是如何对变量命名的。你不能在变量名中间加空格，所以最好使用驼峰型（theString）或下划线型（the_string）变量 命名方法。使用哪一种命名方法并不重要，但在你所写的程序中最好保持统一的命名方法。当你读其他人写的代码时，你一定会很讨厌看到类似 “myVariable” 这样的变量名，因为你从变量名上根本看不出这个变量是用来存什么的。

最后，我们注意到把 “Hello World” 放进了变量中，我可以在代码里一遍又一遍的引用这个变更。如果后面我决定要把 “Hello World” 改成 “Good Morning Dave”，我只需要在声明变量的那一行进行修改就可以了。

使用变量

你可以尽情的使用变量来做各种疯狂的事，但鉴于这是一篇入门文章，我只会告诉你几种方法，输入下面的代码：

```
--Integer Variables
set theFirstNumber to 3
set the theSecondNumber to 2

--Variable Operations
set theAnswer to (theFirstNumber + theSecondNumber)
set theAnswer to (theAnswer + 1)

--String Variables
set theString to "3+2+1="

--Display Dialog
tell application "Finder"

     display dialog theString & theAnswer

end tell
```

当你的脚本变得越来越复杂，你就需要好好的组织一下你的脚本了。在一行代码前加上两个“-”号，代表这一行是注释，你可以使用注释来隔开和解释你的 代码方便以后阅读。在上面的例子里，我创建了一个字符变量和几个整数变量。你可以对变量进行数学运算，在这里我把 “theFirstNumber” 设为3，“theSecondNumber”设为2，然后把他们相加并赋值给 “theAnswer” 变量。

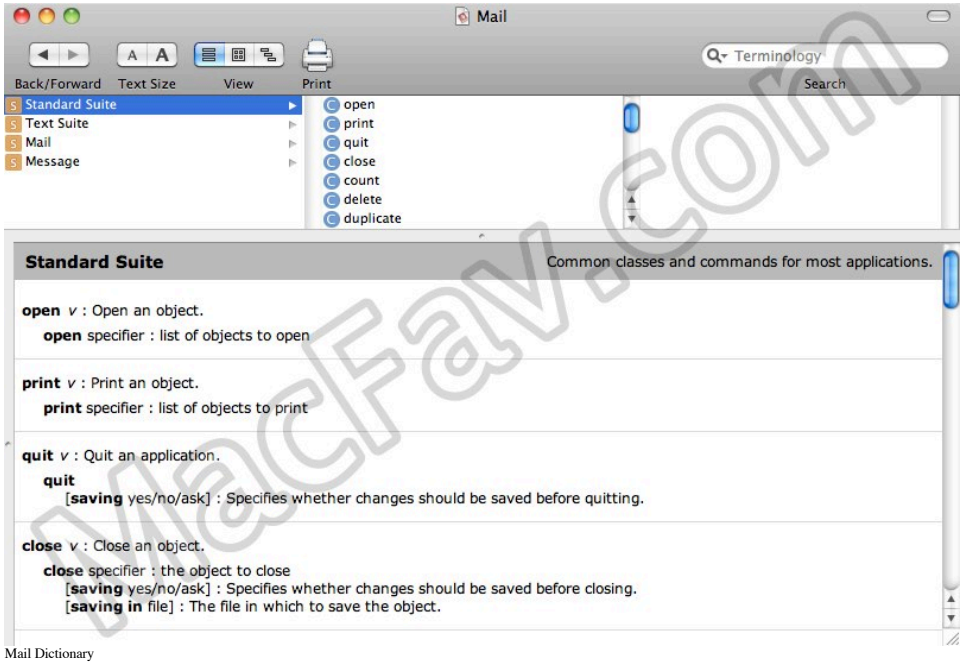
另外，你可以在变量创建后改变变量的值。在 “theFirstNumber” 和 “theSecondNumber” 相加并把结果赋值给 “theAnswer”（结果为5）后，我马上把 “theAnswer” 自己加1（结果为6）。如果你运行这个脚本，你应该会看到以下的结果：

Basic Math

这只是一个很简单的范例，示范你可以对变量做的一些操作。你现在应该明白变量的值并不是不变的，一门编程语言的强大在于你能通过对变量的操作完成各种各样的任务。

重中之重：**AppleScript** 字典

虽然 AppleScript 本身自带有大量可以作用于 Mac 系统中任何程序的指令，但应用程序的作者还是有责任针对他的应用添加 AppleScript 的全面支持。换句话说，开发者必须编写一个他的程序如何与系统里其他程序进行通讯的手册，这些手册被称之为“字典”。要打开字典，在 AppleScript Editor 选择菜单 File->Open Dictionary (或快捷键Shift-Command-O)。例如在打开的字典里向下浏览，找到程序 Mail 然后按 “Choose”，你应该会看到以下的窗口：



Mail Dictionary

在左边的列中包含了指令和物件的 “套件”。当你选择了一个套件，你会在下面的显示窗中看到此套件所包含的所有东西。你可以通过点击第二列及第三列里的命令来缩小预览的范围。套件包

含了“指令”（圆形的C图标）和“类”（方形的C图标），而“类”包含了“属性”（P图标）和“元素”（E图标）。要理解这些东西怎么工作在一起，最好的方法就是用字典来写一个脚本范例。

为脚本创建一个算法

首先我们需要一个算法，其实算法只是“我们需要写下脚本如何工作”的一个高端叫法。我们希望创建一个脚本去编写和发送邮件。我们会使用到变量来令到将来改变邮件内容和送件人变得容易一点。在我们写下算法的时候，我们要时 刻记住 AppleScript 是如何工作的，这些步骤是我得出来的：

1. 为收件人名字、收件人邮件地址、邮件标题以及邮件文本等创建变量
2. 创建新邮件的变量，同时指定邮件信息的属性。
3. 创建一个新的邮件
4. 发送这个新邮件

创建简单的变量

我们从上文已经了解到如何创建储存文本的变量，所以我们在这一步还用不到字典。下面是代码：

```
--Variables
set recipientName to "John Doe"
set recipientAddress to "nobody@nowhere.com"
set theSubject to "AppleScript Automated Email"
set theContent to "This email was created and sent using AppleScript!"
```

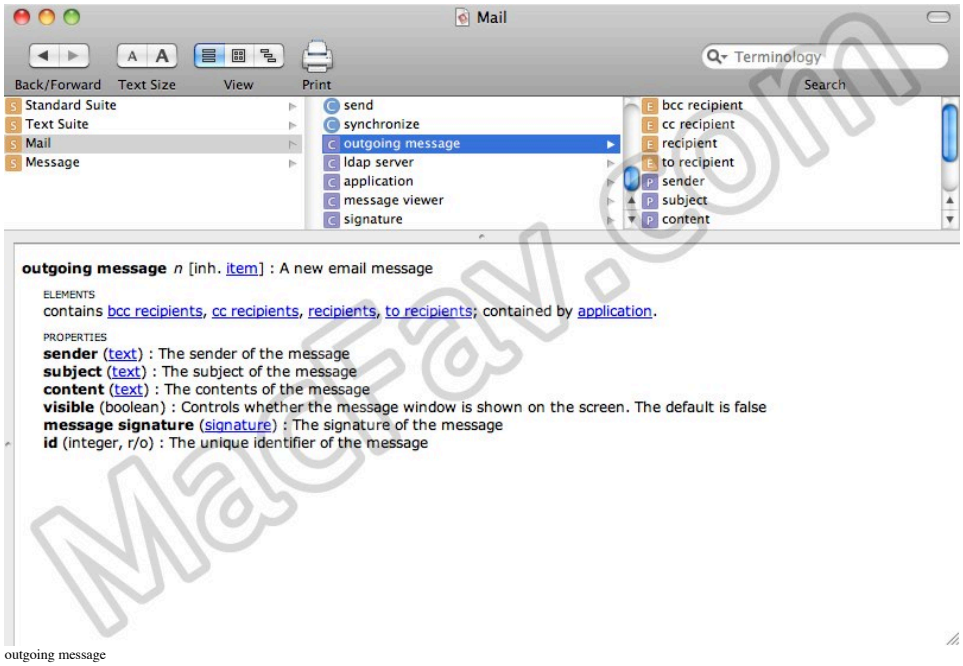
如你所见，我们只是随便放了一些假的数据到这四个变量中，它们分别是收件人名字、收件人邮件地址、邮件标题和邮件正文文本。这些变量里面的内容你喜 欢改成什么都可以，但我建议你把你自己的邮件地址放到收件人邮件地址变量中，这样我们测试脚本的时候就可以通过查自己的邮箱看脚本是否发送邮件成功。

使用 **Mail** 字典来创建邮件变量

由于我们完全不知道怎么告诉 Mail 去创建一个新的邮件信息，所以我们必须借助于 AppleScript 字典。如果你点击一下“Standard Suite”你会看到一些 AppleScript 自带的公共标准命令。由于我们是要“创建”一个新的邮件信息，我们通过向下浏览来找到相关的命令，你会发现里面没有“create”这个命令，但在向下滑到差不多中间的部分你会找到“make”这个命令。听起来貌似不错，所以创建一个新的邮件系统，其实是要告诉 AppleScript 我们要“make”一些东西。

接下来点击最左列的“Mail”套件。我们刚才已经找到动作是“make”，所以在这里我们跳过所有“指令”（动词）不看，直接翻到下面的“类”（名词）中。我们第一个看到的类是“outgoing message”，太好了这就是我们要的，因为我们就是要发出一个新的邮件信息。接着点击“outgoing message”类然后看下面显示的属性说明。

我们需要把第一步创建的几个简单变量插入到新邮件变量中。但在下面的“outgoing message”属性列表中，我们只找到邮件标题和内容的属性，而没看到收件人相关的属性。现在我们知道引用这些属性的正确语法了。注意，字典告诉你的是属性的定义格式。 比如 subject 这个属性，我们定义的方法是“subject:你想打的标题字样”。



outgoing message

同样你会在套件中找到一个叫“send”的指令，我们会使用这个指令来发送邮件。现在我们还需要知道正确表达收件人名字和收件人邮箱地址的语法。由于不在这个套件中，我们点击“Message”套件。向下浏览到差不多一半的时候我们找到“receipient”这个类，点击这个类会看到他的属性列表，同样我们可以通过简单的英文来引用这个类的属性，它的属性包括“name”和“address”。

小技巧：你可以使用字典窗右上角的搜索栏快速的搜索你需要的类或属性等。

现在我们已经准备好用上面学到的语法来创建我们的邮件信息了，代码如下：

```
--Variables
set recipientName to "John Doe"
set recipientAddress to "nobody@nowhere.com"
set theSubject to "AppleScript Automated Email"
set theContent to "This email was created and sent using AppleScript!"

--Mail Tell Block
tell application "Mail"

--Create the message
set theMessage to make new outgoing message with properties {subject:theSubject, content:theContent, visible:true}

end tell
```

注意我创建了一个 tell 命令块来包住所有发送给 Mail 应用程序的命令。然后我创建了一个变量（theMessage）并让他去“make”一个新的“outgoing message”而且指定上面说到的它的属性。需要注意的是属性组需要用{}来包住，每个属性间用逗号分隔。(Jay注：会编程的朋友可以把“make new outgoing message”理解成创建一个“outgoing message”类的实例，然后把实例赋值给theMessage。如有不对请指正）

第一步：设置收件人和发送邮件

我们上面创建了邮件的变量，我们现在会使用另一个 tell 命令块来让这个变量设置好收件人信息并且进行发送，代码如下：

```
--Variables
set recipientName to "John Doe"
set recipientAddress to "nobody@nowhere.com"
set theSubject to "AppleScript Automated Email"
set theContent to "This email was created and sent using AppleScript!"

--Mail Tell Block
```

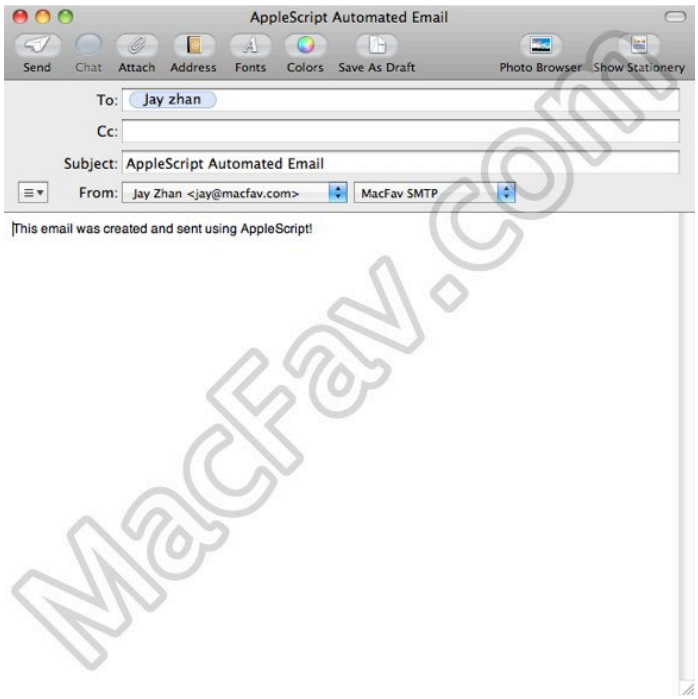
```
tell application "Mail"

     --Create the message
     set theMessage to make new outgoing message with properties {subject:theSubject, content:theContent, visible:true}

     --Set a recipient
     tell theMessage
          make new to recipient with properties {name:recipientName, address:recipientAddress}

          --Send the Message
          send

     end tell
end tell
```

（下面这段我不翻译原文，因为原文说的有点复杂，Jay用自己理解的描述一下，有错请指正）

我们知道 theMessage 指向的是一个 outgoing message 类的实例，我们在前面查 outgoing message 类的时候，看到除了属性列表外，前面还有一个叫 Elements 的说明，这里说明的是这个类具有这些元素，其中就包括一个叫 to recipient 的元素。那么这里就比较好理解了，“tell theMessage make new to recipient” 这句就可以理解为告诉 theMessage 这个实例创建（或者叫设置）它自己的 to recipient 元素，当然用的就是后面跟的 properties 了。

（回到后续的原文译文）

最后我们执行 “send” 指令来发送我们的邮件。注意最下面我们用了两次 end tell，因为有两个 tell 命令块需要关闭。当你修改完一些输入的错误并编译完成后点击 “Run”，Mail 程序应该会自动创建一封邮件并且发出。现在查查你自己的邮箱有没有收到吧～



Automated Email

恭喜你，你已经完成了你第一个 AppleScript 脚本了！你可以把脚本保存成 Script，后面可以随时回来修改和运行，又或者保存成 Application，当成是应用程序一样打开他运行。

总结：不断学习

我希望这个入门手册会让你幻想到那些可以自动化的任务。我展示的 AppleScript 字典和语法仅仅是个开始，但如果你真的对 AppleScript 很感兴趣，你需要看更多的教程才可能掌握。Apple 在官网上提供了丰富的 AppleScript 相关资料，是一个不错的开始！

另外一个值得一提的网站是 T&B。它提供了一些很有深度的解释和教程，虽然有点老，但解释得很全面而且是免费的！无论你是一个 AppleScript 初哥还是高手，都欢迎你留言交流一下，又或者留下你得意的作品，让其他人受益于自动化的 Mac～