**轻松学习E-Prime**

**By福师大2015卡西尔将军**

**2019毕业季发布**

**版权规则：**

**2020年10月13日修改，旧规则无效。**

**该版本（卡西尔将军2019毕业季版）中所有卡西尔将军自己编写制作的示例文件代码、图片都遵循WTFPL协议。而该文档遵循下面的协议：**

**·署名权天然保留。**

**·任何人都可随意修改、打印、复制、保存，允许印刷，商用，但不允许不经修改或加工就直接倒卖该文档的电子版本。**

**·欢迎任何人进行改编再造，添加新内容，但必须保留之前版本所有作者的信息，并必须在后面添加上自己的信息、尽量注明修改的地方和版本号、日期。除非违反该版权规则，任何情况，原作者（卡西尔将军）只保留署名，不收取任何费用。**

**·对于当前版本，所有商用及非商用情况，必须复制该版权规则。其他经由其他作者修改的版本，遵循其他作者留下的版权规则。**

**·若您是修改者，您可以添加自己的版权规则，如不再允许他人复制、修改等。但注意不可掩盖该教程版本存在的信息以及源教程遵循该版权规则的事实，也要遵循其他修改者留下的版权规则。对于该版本的信息，您不需要全部复制，只需留下【一个署名】+【一个链接即可】，详见下方。若基于其他版本做修改再发布，请遵循其他作者添加的版权规则。**

**·若您只是复制打印的人，您可以标注自己的印刷信息和使用规则，但无权对内容添加自己的版权信息。**

**·如果您基于该文档创造出的其他版本引发任何争议或其他纠纷，先前所有版本的凡不涉及该问题的作者不负任何责任。**

**修改者不需要留下该规则和下面的寄语，只需留下下面这两行话，即可用作其他任何用途：**

**基于卡西尔将军的《轻松学习E-Prime》修改，源教程：**

<https://github.com/zkx654321/Learn-E-Prime-Easily>

给学弟学妹们的寄语

By 福师大2015卡西尔将军

2019 毕业季

Github 账号 和 百度ID:zkx654321.

**源教程献给挚爱的福建师范大学心理学院和未来及现在的学弟学妹们，希望对学弟学妹们有所帮助。**

本教程虽然叫轻松学习，不过并不意味着你可以不努力哟！要有看大段文字的耐心，要有探索精神，很多地方，我将会要求你自主动手或发现学习。

由于网上已经有不少E-prime教程，因此本教程只讲最重要的部分以及如何使用E-Prime及其自定义代码功能，偏向实用性，少量涉及总体架构和理论性。本教程对应0基础和其他不同基础的同学提出了不同的学习方案（见第一章最后一节，以及自测），适合各水平阶段同学（大神除外）的使用。

一些相当基础的电脑操作（如保存，打开，新建文件，怎么拿鼠标，怎么移动鼠标，怎么点击，怎么切换大小写，怎么弯曲你的膝盖让你能坐在电脑前，什么是正确的操作电脑的坐姿等）会跳过。

本教程的案例使用E-Prime2.0编写，所以您需要用2.0（Professional）以上版本才能打开。文中打\*的内容表示可以跳过。

我自认为我的教程有两大特色：简单、有趣。

简单在里面的操作都是手把手的（至少在初期），而且对读者要求不高，不要求读者理解，只要会用就可以。

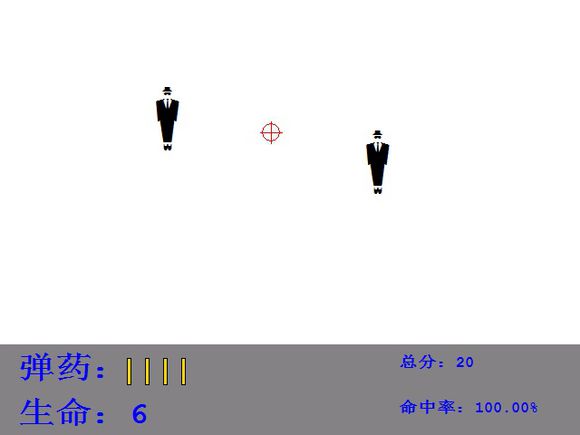
我想达到的效果就是，让这个教程看起来亲切一点，不像那些正规的书，过于严（呆）肃（板）。当然，后果就是牺牲了理论性。**我想让大家知道，E-Prime也可以很好玩。**

下面是我的教程中作为示例程序的用E-Prime两个小游戏。

第一个是记忆字母，用有趣的萌萌哒的魔法师进行正误反馈：



第二个是枪战游戏，包含紧张刺激的音乐，音效，不过BUG很多：



**如果你只有这个word文档而没有示例文件的话，可以从以下的Github链接或网盘链接中下载：**

<https://github.com/zkx654321/Learn-E-Prime-Easily>

**链接：https://pan.baidu.com/s/12BcpBMUkGOc4K2Zn8owZYA   
  
提取码：pw0t**

[第一章-What’s E-Prime 8](#_Toc11249825)

[C:\Program Files\Microsoft Office\MEDIA\OFFICE14\Bullets\BD10297_.gifE-Prime的功能 8](#_Toc11249826)

[除了E-Prime，我还能用什么 8](#_Toc11249827)



[为什么要用E-Prime？ 9](#_Toc11249828)



[看了本文以后我能学到什么程度？ 10](#_Toc11249829)



[我该怎么学？ 10](#_Toc11249830)



[说明: C:\Program Files\Microsoft Office\MEDIA\OFFICE14\Bullets\BD10297_.gif最后你要知道的东西 11](#_Toc11249831)

[第二章- E-Prime低阶教程—基础的应用 12](#_Toc11249832)

[E-Prime的exe程序怎么那么多？我运行那个？ 12](#_Toc11249833)

[开始E-Studio的界面 12](#_Toc11249834)

[从最简单的E-Prime程序开始 14](#_Toc11249835)

[运行E-Prime程序 15](#_Toc11249836)

[开始制作有意义的E-Prime程序 15](#_Toc11249837)

[继续修改属性 17](#_Toc11249838)

[修改呈现的时间（这个属性） 18](#_Toc11249839)

[C:\Program Files\Microsoft Office\MEDIA\CAGCAT10\j0298653.wmf阶段练习 21](#_Toc11249840)

[呈现图片 22](#_Toc11249841)

[C:\Program Files\Microsoft Office\MEDIA\CAGCAT10\j0298653.wmf阶段练习 23](#_Toc11249842)

[相对路径和绝对路径 24](#_Toc11249843)

[为什么使用相对路径？ 25](#_Toc11249844)

[C:\Program Files\Microsoft Office\MEDIA\CAGCAT10\j0298653.wmf阶段练习 25](#_Toc11249845)

[按键反应 26](#_Toc11249846)

[C:\Program Files\Microsoft Office\MEDIA\CAGCAT10\j0298653.wmf阶段练习 27](#_Toc11249847)

[刺激呈现的时间和允许按键的时间 28](#_Toc11249848)

[多次呈现——Procedure和List 28](#_Toc11249849)

[List的一行，可运行一个Procedure 30](#_Toc11249850)

[读取List中的Attribute 32](#_Toc11249851)

[C:\Program Files\Microsoft Office\MEDIA\CAGCAT10\j0298653.wmf阶段练习 33](#_Toc11249852)

[呈现正误反馈 34](#_Toc11249853)

[C:\Program Files\Microsoft Office\MEDIA\CAGCAT10\j0298653.wmf阶段练习 35](#_Toc11249854)

[音频和视频 36](#_Toc11249855)

[数据的记录 36](#_Toc11249856)

[数据的查看 37](#_Toc11249857)

[数据的录入（Excel或SPSS） 38](#_Toc11249858)

[数据的合并 41](#_Toc11249859)

[本章结语 41](#_Toc11249860)

[第二章综合练习 43](#_Toc11249861)

[第三章- E-Prime进阶教程—理解并进一步操作 45](#_Toc11249862)

[C:\Program Files\Microsoft Office\MEDIA\OFFICE14\Bullets\BD14752_.gif导言 45](#_Toc11249863)

[C:\Program Files\Microsoft Office\MEDIA\OFFICE14\Bullets\BD14752_.gifSlide的应用 45](#_Toc11249864)

[C:\Program Files\Microsoft Office\MEDIA\OFFICE14\Bullets\BD14752_.gifWait的应用 47](#_Toc11249865)

[C:\Program Files\Microsoft Office\MEDIA\OFFICE14\Bullets\BD14752_.gifList的进阶应用 47](#_Toc11249866)

[C:\Program Files\Microsoft Office\MEDIA\CAGCAT10\j0298653.wmf阶段练习 51](#_Toc11249867)

[C:\Program Files\Microsoft Office\MEDIA\OFFICE14\Bullets\BD14752_.gifTime Limit!!End Action！！Jump！！ 52](#_Toc11249868)

[C:\Program Files\Microsoft Office\MEDIA\CAGCAT10\j0298653.wmf阶段练习 53](#_Toc11249869)

[C:\Program Files\Microsoft Office\MEDIA\OFFICE14\Bullets\BD14752_.gif实验程序的另一些设定 54](#_Toc11249870)

[C:\Program Files\Microsoft Office\MEDIA\OFFICE14\Bullets\BD14752_.gif你不想让别人知道你的实验程序？ 57](#_Toc11249871)

[C:\Program Files\Microsoft Office\MEDIA\OFFICE14\Bullets\BD14752_.gif修改设备（分辨率、颜色位数、增加设备） 58](#_Toc11249872)

[C:\Program Files\Microsoft Office\MEDIA\OFFICE14\Bullets\BD14752_.gif松开按键时记录反应时 59](#_Toc11249873)

[C:\Program Files\Microsoft Office\MEDIA\CAGCAT10\j0298653.wmf阶段练习 59](#_Toc11249874)

[C:\Program Files\Microsoft Office\MEDIA\OFFICE14\Bullets\BD14752_.gif接收多个按键与中止接收 60](#_Toc11249875)

[C:\Program Files\Microsoft Office\MEDIA\OFFICE14\Bullets\BD14752_.gif输入框制作 60](#_Toc11249876)

[C:\Program Files\Microsoft Office\MEDIA\OFFICE14\Bullets\BD14752_.gifClear After，Echo这一切暗示了什么？ 61](#_Toc11249877)

[C:\Program Files\Microsoft Office\MEDIA\OFFICE14\Bullets\BD14752_.gif本章结语 62](#_Toc11249878)

[第二章综合练习 63](#_Toc11249879)

[第四章- E-Prime高阶教程—使用自定义脚本 66](#_Toc11249880)

[C:\Program Files\Microsoft Office\MEDIA\OFFICE14\Bullets\BD21295_.gifE-Basic 66](#_Toc11249881)

[C:\Program Files\Microsoft Office\MEDIA\OFFICE14\Bullets\BD21295_.gifHello World！ 67](#_Toc11249882)

[C:\Program Files\Microsoft Office\MEDIA\OFFICE14\Bullets\BD21295_.gif声明并赋值一个变量 68](#_Toc11249883)

[C:\Program Files\Microsoft Office\MEDIA\OFFICE14\Bullets\BD21295_.gif变量的运算 69](#_Toc11249884)

[C:\Program Files\Microsoft Office\MEDIA\CAGCAT10\j0298653.wmf阶段练习 70](#_Toc11249885)

[C:\Program Files\Microsoft Office\MEDIA\OFFICE14\Bullets\BD21295_.gif记录呈现的次数——Public声明变量（全局变量） 71](#_Toc11249886)

[C:\Program Files\Microsoft Office\MEDIA\CAGCAT10\j0298653.wmf阶段练习 72](#_Toc11249887)

[C:\Program Files\Microsoft Office\MEDIA\OFFICE14\Bullets\BD21295_.gif查看当前的代码 73](#_Toc11249888)

[C:\Program Files\Microsoft Office\MEDIA\OFFICE14\Bullets\BD21295_.gif在Inline中读取List中的变量 74](#_Toc11249889)

[C:\Program Files\Microsoft Office\MEDIA\CAGCAT10\j0298653.wmf阶段练习 75](#_Toc11249890)

[C:\Program Files\Microsoft Office\MEDIA\OFFICE14\Bullets\BD21295_.gif条件语句（If） 75](#_Toc11249891)

[C:\Program Files\Microsoft Office\MEDIA\CAGCAT10\j0298653.wmf阶段练习 78](#_Toc11249892)

[C:\Program Files\Microsoft Office\MEDIA\OFFICE14\Bullets\BD21295_.gif循环语句（简易） 78](#_Toc11249893)

[C:\Program Files\Microsoft Office\MEDIA\OFFICE14\Bullets\BD21295_.gif查阅帮助以及面向对象编程 79](#_Toc11249894)

[C:\Program Files\Microsoft Office\MEDIA\OFFICE14\Bullets\BD21295_.gif读取按键记录（ResponseData） 83](#_Toc11249895)

[C:\Program Files\Microsoft Office\MEDIA\CAGCAT10\j0298653.wmf阶段练习 84](#_Toc11249896)

[C:\Program Files\Microsoft Office\MEDIA\OFFICE14\Bullets\BD21295_.gif控制Slide及其各页面控件（SlideDisplay Sub-Objects），VB数组 85](#_Toc11249897)

[C:\Program Files\Microsoft Office\MEDIA\CAGCAT10\j0298653.wmf阶段练习 91](#_Toc11249898)

[C:\Program Files\Microsoft Office\MEDIA\OFFICE14\Bullets\BD21295_.gifCanvas Object \* 94](#_Toc11249899)

[C:\Program Files\Microsoft Office\MEDIA\OFFICE14\Bullets\BD21295_.gif制作按钮？读取MouseResponseData中被试点击的位置 98](#_Toc11249900)

[C:\Program Files\Microsoft Office\MEDIA\CAGCAT10\j0298653.wmf阶段练习 103](#_Toc11249901)

[C:\Program Files\Microsoft Office\MEDIA\OFFICE14\Bullets\BD21295_.gif自定义记录数据 104](#_Toc11249902)

[C:\Program Files\Microsoft Office\MEDIA\OFFICE14\Bullets\BD21295_.gif娱乐：枪战游戏\* 107](#_Toc11249903)

[C:\Program Files\Microsoft Office\MEDIA\OFFICE14\Bullets\BD21295_.gif本章结语 107](#_Toc11249904)

[第四章综合练习 109](#_Toc11249905)

[第五章- 后记 111](#_Toc11249906)

[C:\Program Files\Microsoft Office\MEDIA\OFFICE14\Bullets\BD21300_.gif本书的不足 111](#_Toc11249907)

[C:\Program Files\Microsoft Office\MEDIA\OFFICE14\Bullets\BD21300_.gif给学弟学妹以及同学们的一段话 113](#_Toc11249908)

[附录 115](#_Toc11249909)

[本教程的组成部分 115](#_Toc11249910)

[各个按键在按键反应中输入的方法 115](#_Toc11249911)

# 第一章-What’s E-Prime

## C:\Program Files\Microsoft Office\MEDIA\OFFICE14\Bullets\BD10297_.gifE-Prime的功能

让我们从吐槽开始吧！首先E-Prime确实是一个有点坑爹的软件……对高版本Windows兼容性不高（新版的E-Prime好像能够兼容），他的各种各样的呈现图片或音频的控件（“控件”你可以姑且理解为工具）总是连很多常见的文件格式都不支持，软件还动不动崩溃，我看叫“E-Cry”比较好。

不管怎样，吐槽先到此为止。回归正题：E-Prime有什么功能？E-Prime就是用来做心理学实验的，可以呈现图片、文字、声音等，并记录反应时。实际上E-Prime的一些版本甚至可以用来做脑电实验。

由于是专门的软件，而且可视化，面向对象，所以E-Prime还是很好用的，能够很快的做出你先要的程序。除了能够很快的编程，来呈现图片、文字、声音之外，实际上E-Prime还可以用VB来编写自定义代码，这也是我特地写这个教程的主要原因，不然关于基础的部分，网上教程已经不少了。

所以吐槽归吐槽，好用还是好用，好用就要用。最后再吐槽一下，讲真仅仅是呈现图片，并记录反应时这个功能的话根本不难好伐！为什么卖到近200人民币嘞！希望国内有识之士或团体编写一个属于我们的，经过测试时间精度可靠的心理学软件！当然，E-Prime是强大的，你可以运行本教程目录里的“E-Prime能有多强大”里面的“枪战游戏”体会一下。

## 除了E-Prime，我还能用什么



你还能用Matlab，Presentation，Python或其他编程语言甚至PPT等来制作心理学实验。

·Matlab

Matlab是一款数学软件，有人做了一个名为PSYCHTOOLBOX的工具箱，专门用来做心理学实验，功能强大。名为PSYCHTOOLBOX的工具箱是免费的，但Matlab是收费的（当然要找到所谓破解版对各位来说应该不在话下）。要使用Matlab来做心理学实验，首先你要会Matlab编程，其语言特点类似C语言，难度较大。如果你想挑战自己，可以试试。

·Presentation

Presentation也是专门用来做心理学实验的软件，也使用类似C语言的语言进行编程，还能做脑电实验，功能也很强大。我写的一个未完成的程序叫PresentationWriter，是打算让编程过程可视化的，不过没有完成。会编程的朋友可以把它完成，或者另写一个辅助软件，这样就能让她既简单，又好用。

Presentation的难度相比Matlab小得多，学完计算机二级（C语言）后完全可以尝试。

·Python或其他编程语言

Python是一个脚本语言，功能强大。当然，C/C++，VB，Java等其他凡是能够呈现多媒体信息并记录反应时（保证时间精度的前提下）的高级编程语言都可以使用。只要能够实现你的程序，只要有能力，要用什么都可以。

·PPT

不要以为PPT是最简单的，实际上难度不小。

## 为什么要用E-Prime？



因为其他的都挺难的，E-Prime虽然有不少缺点，但简单易学。没有特殊理由，舍弃简单的方法而用复杂方法解决问题，是一件很荒谬的事情。

## 看了本文以后我能学到什么程度？



本文重于应用，因此学完“低阶”以后你已经会熟练使用，在**福师大**的本科阶段是非常足够的了。如果学到目录中的“高阶”，那你已经可以用来装逼了。

## 我该怎么学？



**0基础的同学**：你不仅要看我写了什么，而且一定要实际操作。要盖上本文亲自尝试一遍。你要知道操作类的知识实际操作是极其重要的。遇到问题，先独立解决，然后才问老师，问百度或必应。（请不要问作者，我拒绝回答。）

**已经会用E-Prime设计实验的同学**：快速看完“低阶”部分，认真看“进阶”部分。有需要的场合才看“高阶”部分，“高阶”部分注意独立练习操作。

**0基础但是掌握至少一门编程语言的同学：**迅速扫过“低阶”，主要了解一下界面和使用技巧，其他同下。注意掌握并不是学了个if，while啥的就算掌握了。具体标准请自行判断。

**已经会用E-Prime设计实验并且掌握至少一门编程语言的同学**：直接扫一扫“进阶”部分，“高阶”部分对你来说可能也是小菜一碟。

**大神**：本人能力有限，难免有误。快速浏览本文，进行修改订正，在作者信息里加上你的修改信息和日期，然后重新发布。

**0基础而且电脑基础又差的同学**：除了和0基础的同学一样以外，你平常还要多玩Office，多查看Windows里的文件，熟悉各种Windows操作。嫌无聊的话，玩玩PC游戏也是可以的（尽量小型单机独立游戏，或RTS游戏，而不是3A大作，学会独立安装，查找需要的游戏组件，修改游戏数据等）。

**0基础而且电脑基础又差又不想努力，不独立，有问题不试错，只想问别人颜值又不行的同学**：呵呵。

## 说明: C:\Program Files\Microsoft Office\MEDIA\OFFICE14\Bullets\BD10297_.gif最后你要知道的东西

最后，我要告诉你，亲爱的读者。本书重于应用，尽管描述了一些理论，但可能描述的很有限。当然这不要紧，我这么做就是为了防止你们纠结于理论，因此你们只要继续看下去就够了，更重要的是**动手实践**！**动手实践**！**动手实践**！

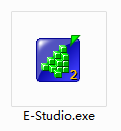
关于作者对本书目前想到的疏漏之处和不足之处，都写在后记里了，您可以去看看。

# 第二章- E-Prime低阶教程—基础的应用

## E-Prime的exe程序怎么那么多？我运行那个？

无论你从开始菜单，打开E-Prime，还是直接去根目录查看并打开E-Prime的程序，都会遇到这个问题：这里有好几个分程序，我该运行哪一个？

这里每个分程序都有其独特的作用，不过实际上，您只要掌握以下这一个就够了：



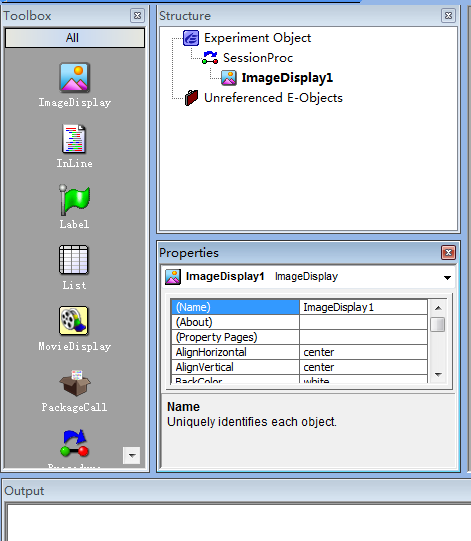
记住这个图标！记住这个方块堆起来的几何体！这个程序叫E-Studio，它的作用是编写一个可以运行的es文件（最终生成ebs文件在E-Run里运行）。

**其他几个分程序在本章最后和进阶篇中会简单介绍，这里不做介绍。**

## 开始E-Studio的界面

打开E-Studio.exe，新建一个空白的es或es2文件（选择Blank(Professional)）。首先为了防止描述困难问题，我不得不简单说一下界面上的东西。

这上面每一个窗口都有名字的（如下图所示）：

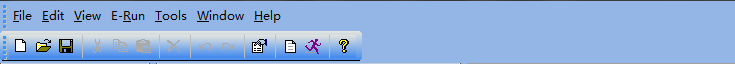


**E-prime的各个窗口名称**

**我已经用红框标记了各个窗口的名字，橙色框标记了要注意的地方。以后为了描述方便，各窗口我也直接用英文描述，请懂得如何找到他们。**

**练习：在Toolbox窗口中找到List。在Structure窗口中找到ImageDisplay1.**

整个E-Studio窗口的最上方也和其他Windows程序一样有菜单。也就是下面这个东东：



**E-prime的菜单，在整个窗口的上方**

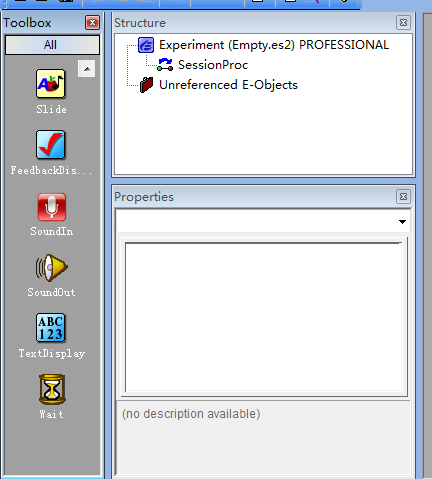
这个我们就叫他菜单。

OK！那么一切都准备好了！让我们从最简单的E-Prime程序开始吧。

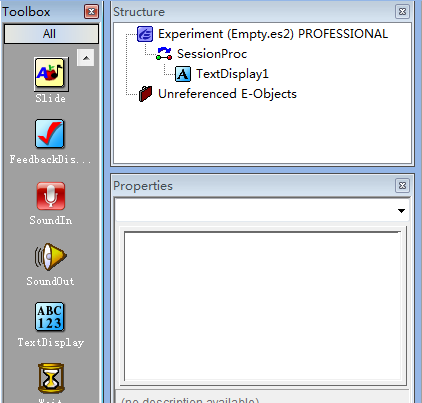
## 从最简单的E-Prime程序开始

那么我们怎么做E-Prime实验呢？接下来我们制作一个非常简单的实验程序，对于凡是智力正常，四肢健全的人，应该都是易如反掌的。

像下图中那样，从Toolbox窗口中找到TextDisplay，用鼠标并把它按住并拖动到Structure窗口里的SessionProc中。



结果应该如下图所示：

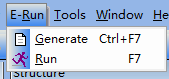


**不要怀疑，一个最简单的E-Prime程序制作完毕了。**

## 运行E-Prime程序

那么做好了程序以后，我们如何运行呢？

找到菜单里的E-Run里面的Run按钮，或者一个跑动的小人按钮，就可以运行你现在编好的程序了。

**可以运行你现在编好的程序的按钮**

首次运行的时候应该会让你保存你现在的程序。任意保存到你想要的位置就好了。

运行后你暂时不要管别的，先一路按OK和Yes就行了。按完一系列对话框后，最后你会看到一个白色屏幕停留了一秒，然后消失，这就说明一切正常。

恭喜你成功完成了最简单的E-Prime程序！

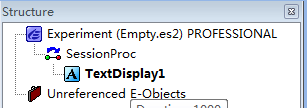
## 开始制作有意义的E-Prime程序

你的可能会说：“学长！可是上面那个程序根本没有意义啊！”没关系，我们总要让一切从简单开始。

上面我们加入了TextDisplay但我们实际上没有呈现任何Text（文本）。那么我们该怎么做才能让它呈现一些内容呢？**我们要修改它的属性。**

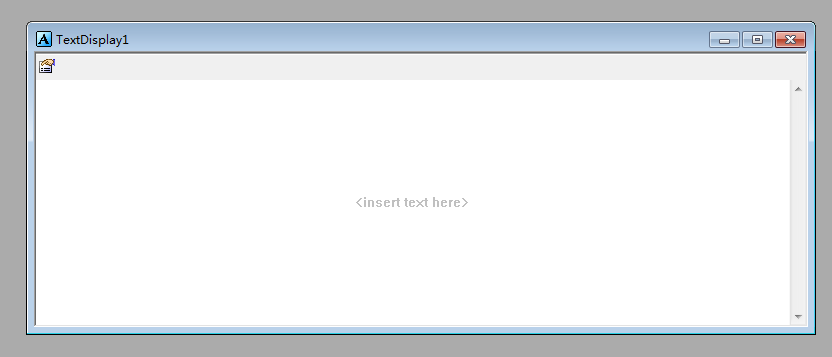
属性是什么？比如一个人有名字、身高、体重、性别，这些都是属性。玩过游戏的同学肯定知道，攻击力、防御力、HP、MP，这些都是属性。所以我们要像程序员设置游戏角色的属性一样，设置TextDisplay的属性，让它呈现我们想要的效果。

首先我们双击一下TextDisplay1：



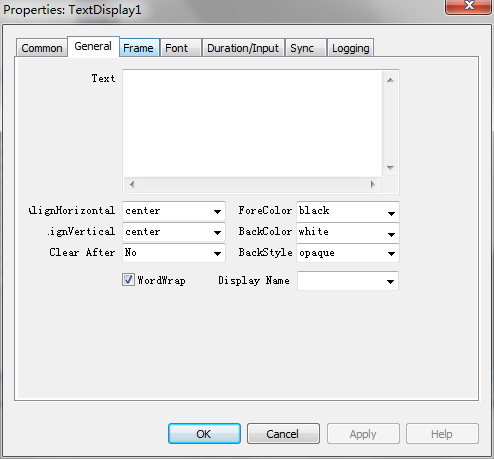
**双击**

父窗口的空白（其实是灰色）处应该会呈现下面这个子窗口。（父窗口和子窗口很好理解，像整个E-Studio的窗口就是最大的窗口，是最大的“爸爸”；其他包括在这个父窗口里的窗口，比如Toolbox、Structure这些都是“儿子”，也就是子窗口。）这是预览窗口，让你看效果的。



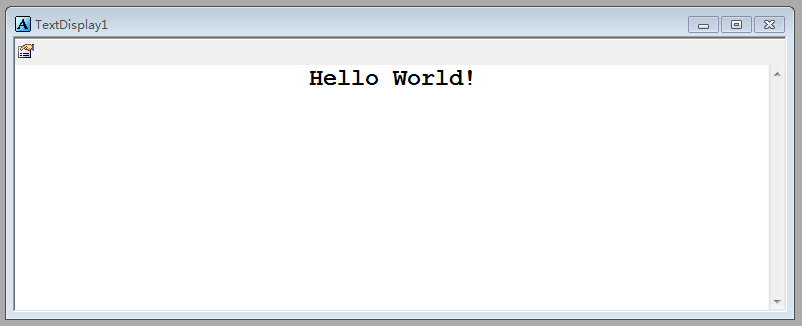
与此同时，Properties窗口的内容也会改变。如果不知道“Properties”这个单词的意思，建议你查一下。

点击上图中红框所标记的按钮，会出现下面这个窗口：



像这些Text、ForeColor、BackColor啥的都是TextDisplay1它的属性。如果不知道这些单词的意思，建议去查一下。

现在我们暂时先不管其他的内容，只在Text这一属性里面输入“Hello World！”输入完成后，预览窗口里应该会显示“Hello World！”，也就是像下面这样。

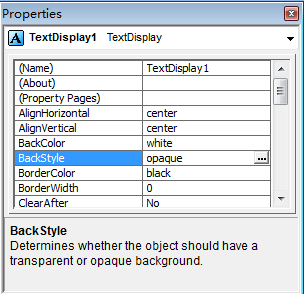


再次运行，然后看看现在效果如何？**（可以查看本教程案例中的“示例程序1-1.es2”）**

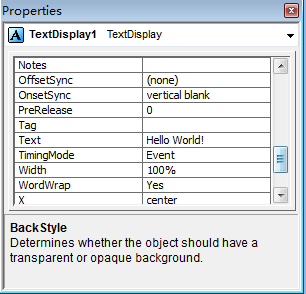
## 继续修改属性

如果上述操作正确的话，你刚才应该已经和世界问好了。

除了用上述方法改变属性以外，我们还可以用Properties窗口改变TextDisPlay1这个E-Object的属性。



在Properties窗口这个窗口里你可以看到里面有很多很多属性。下拉滚动条你会看到Text属性已经被我改成了“Hello World！”.

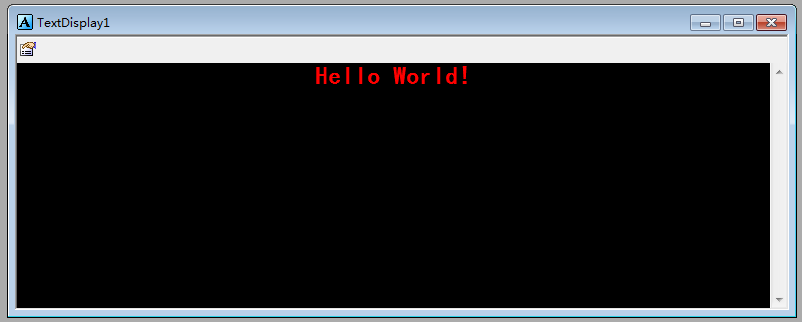


现在你尝试找到FontSize并把数值改成36，ForeColor改成”red”，FontName改成黑体，BackColor改成Black。这个时候预览窗口应该已经发生变化了，现在再运行看看会有什么效果。（提示：有些属性单击后旁边会有下拉框让你选择。）

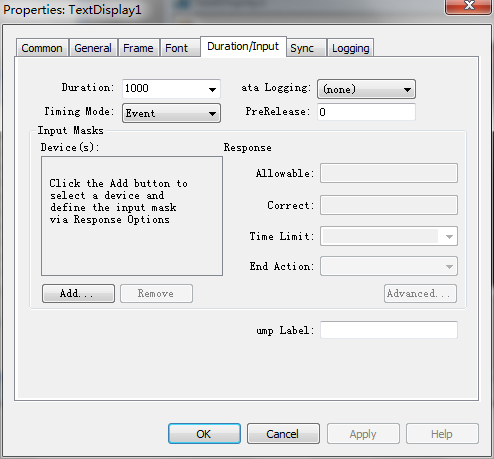
如果一切正常，你应该会看到漆黑的屏幕中央鲜红的“Hello World！”！不得不说有点诡异。**（请参考“示例程序1-2”）**

## 修改呈现的时间（这个属性）

到目前为止，我们已经会在屏幕上呈现按我们的要求呈现某行文字了，然而它就这么一瞬间就消失了。如果我们要让它呈现久一点，我们就必须修改一个叫Duration的属性。让我们用老方法来修改。



点击这个按钮。



找到这个窗口，把Duration的1000改成5000。Duration表示的是呈现的时间，按毫秒计算。因此1000就是1000毫秒，也就是1秒。

现在你再次运行，TextDisplay1的呈现将停留5秒的时间，比之前更长了。请运行试试看。

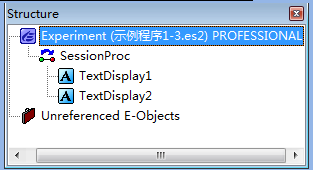
这里还有一个Infinite的选项（实际上数值是-1）。



Infinite意思是无限，所以也就是呈现无限久的时间。如果你不幸选择了这个而无法退出E-Run的话，可以按下Ctrl+Alt+Shift来强制退出E-Run。

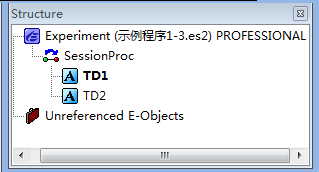
现在还是设置TextDisplay1的呈现时间为5秒，再从ToolBox中拖动一个TextDisplay到TextDisplay1的后面。它会被自动命名为TextDisplay2。

效果如下图所示：

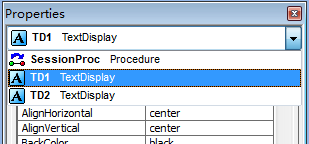
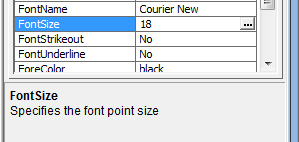


您可能会发现您的TextDisplay2在TextDisplay1的前面，而和上图中的顺序不一样。那么请自行拖动它们，尝试把顺序设置的和上图一样。

修改它们的Name属性为TD1和TD2，如下图所示：



你还会发现，Properties窗口这里有个下拉选框，里面多了一个TextDisplay，也就是我们加进去的TD2。

用Properties窗口这个地方，方便查看、选择你现在加入实验中的E-Object，并进行属性的修改。**在Properties窗口最下方有对当前选择的属性的解释！**

总之，现在用你喜欢的方式，把TD2的属性按下面进行修改：

Text：Hello E-Prime！ FontSize：72

ForeColor：green! BackColor：gray

Duration：3000

由于TD1的Duration为5000，TD2为3000，所以现在运行会呈现TD1五秒，然后呈现TD2三秒。假设你是0基础看到这里，现在！恭喜你不仅已经可以可以用E-Prime呈现文字，而且还知道如何按自己的想法来进行呈现了！**（请参考“示例程序1-3”）**

## C:\Program Files\Microsoft Office\MEDIA\CAGCAT10\j0298653.wmf阶段练习

*一个悲伤的故事：*

新建一个空白的es或es2文件（使用左上角菜单File-New来新建或者重新打开E-Studio，选择Blank(Professional)），完成如下操作：

①在默认的名为SessionProc的Procedure这个E-Object中加入一个TextDisplay，改变它的名字为“NewText”，要呈现的文本为“主试很生气，后果很严重！”，让它靠屏幕左方显示，并且字体大小为20，颜色红色，字体为黑体，有下划线。呈现时间为3秒。（关键词提示：Name，Text，AlignHorzontal，FontSize，ForeColor，FontName，FontUnderline，Duration）

②在 NewText之后再加入一个TextDisplay，改变它的名字为“OldText”，要呈现的文本为“原因是被试拿了钱乱填问卷。”，让它靠屏幕**右方**显示，并且字体大小为22，颜色绿色，斜体。呈现时间为3秒。（这次不给提示了……）

③在OldText之后再加入一个TextDisplay，改变它的名字为“Result”。要呈现的文本是主试生气后的结果，请自由发挥。字体及背景以及呈现时间都请放飞自我。

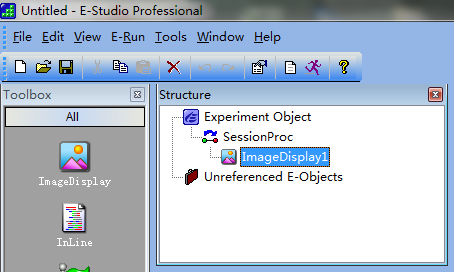
④继续放飞自我，自己玩玩TextDisplay的其他属性，研究一下有什么作用，甚至可以看一下ToolBox里其他Object的作用，玩一玩，研究研究，看看它们是用来做什么的。觉得玩得差不多了，就进入下面的学习。

## 呈现图片

到目前为止，我们已经学会了怎么样修改TextDisplay，并按照我们想要的方式呈现文字。那么接下来是另一个问题了，如何呈现我想要的图片呢？

Text的中文意思是文本。那么现在看看我们的ToolBox窗口，您觉得哪个最有可能是用来呈现图片的E-Object？（在E-Prime里，ToolBox里的这些东西都被称为E-Object，它们又都属于“控件”）

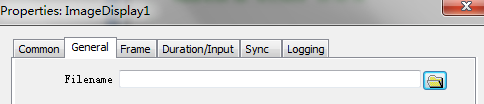
没错，Image的中文意思是图片，也就是说，呈现图片的控件名叫ImageDisplay，现在你可以拖动一个到我们的流程里，然后我们开始研究吧！



拖动完以后，应该会像上面那个样子。

现在如果你尝试运行这个程序，你会看到一秒钟的白屏，这当然也是因为我们没有设置属性的原因。

我们需要重点设置的属性是：Filename



这个属性可以手动输入也可以按下来检索文件夹自动输入。

现在您先使用检索文件夹，找到你想要的图片双击，让它自动进入到Filename属性里。比如下面是一个例子：



现在再次运行，你会发现这个图片呈现了一秒。

**同理，如果你要让它呈现更多秒，只需要设置Duration属性！**

**也就是说，请学会举一反三。ImageDisplay呈现图片的原理就是输入Filename，让E-Prime找到你存在电脑里的图片，把它呈现在屏幕上；而TextDisplay只不过是你自己输入文字，并呈现在屏幕上。其他完全除了部分属性不同以外，有很多和TextDisplay一样！呈现时间同样用Duration设置！**

**举一反三！举一反三！举一反三！重要的事说三遍！**

## C:\Program Files\Microsoft Office\MEDIA\CAGCAT10\j0298653.wmf阶段练习

*小游戏*

新建一个空白的es或es2文件，加入一个ImageDisplay：

①将其Filename属性设置为本教程的案例中的“示例教程1-4”文件夹里的“number.png”，把呈现时间改成3秒。

②在ImageDisplay1的后面增加一个TextDisplay，并让其能够呈现“翻转”这两个字，呈现时间1秒。然后在TextDisplay后面再加入一个ImageDisplay，呈现和ImageDisplay1相同的图片，呈现时间3秒。把其Mirror Up/Down属性改成Yes，看看有什么区别？如果修改Mirror Left/Right呢？

③设置ImageDisplay2的UseSourceColorKey为Yes，SourceColorKey为White，运行看看不同。自行再试试其他属性的修改。

## 相对路径和绝对路径

路径是什么？实际上Filename属性里所填的就是所谓的路径。Filename的意思是文件名。在ImageDisplay的使用中，电脑通过我们填写在Filename里的路径，找到那个文件，然后显示出来。

路径分为绝对路径和相对路径。我们接下来先举一个现实的例子来理解。

假设我们要找一个人，我们要知道他所在的城市和门牌号。我们姑且说一个人所在的地方分为相对位置和绝对位置。假设我们都在福建师范大学里，我们要说桃李园餐厅，至少有以下两种说法：

**绝对位置**：银河系太阳系地球中国福建省福州市福建师范大学桃李园餐厅

**相对位置**：桃李园餐厅

这两种说法我们都能找到桃李园餐厅，绝对位置非常清晰的描述了桃李园餐厅的位置。而只说“桃李园餐厅”五个字，我们也能找到桃李园餐厅，因为我们都在福建师范大学里，我们知道对方所指的是福建师范大学的桃李园餐厅，不是其他某个学校的“桃李园餐厅”。

同样的，电脑也有相对路径和绝对路径，以下就是例子：

**绝对路径**：D:\新建文件夹\abc.jpg

**相对路径**：abc.jpg

在Filename里输入绝对路径，无论你的es或es2文件保存在哪里，都可以找得到abc.jpg这个文件。而用相对路径的话，你的es或es2文件必须和abc.jpg**在同一个文件夹里**，也就是在“D:\新建文件夹”这个地方。就好像只有我们都在福建师范大学，我们才能理解“桃李园餐厅”指的是福建师范大学的桃李园餐厅。如果在其他大学，直接说这五个字可能造成误会。

以下是另一个相对路径的例子：

**相对路径**：Pics\abc.jpg

上面这个也是相对路径，那么假设还在“D:\新建文件夹”这个地方的话，完整的路径就应该是“D:\新建文件夹\Pics\abc.jpg”，这个时候指向的是另一个文件了。同样，你的es或es2文件也应该放在“D:\新建文件夹”。

也许上面这些让你感到难以理解，那么现在就是你实践的时候了。打开“示例程序1-4.es2”，看看里面的ImageDisplay的Filename属性，和我们在上一环节填写的用的有何不同？为什么可以这么简单的填写？尝试理解它们。

## 为什么使用相对路径？

很简单。当你要把你的实验拿给别人分享的时候，假设你填写的是绝对路径，比如说是“E:\新建文件夹\abc.jpg”，然而你朋友的电脑上根本就没有E盘，这个时候运行程序会由于E-Run找不到你所填写的文件而出错。

然而填写相对路径的话，只要在复制es或es2文件的时候，把图片一起复制过去。只要图片和es或es2的相对位置保持不变（比如还在同一文件夹），程序就还能运行。因此建议多使用相对路径。

## C:\Program Files\Microsoft Office\MEDIA\CAGCAT10\j0298653.wmf阶段练习

*理解性问题*

①假设一个Soundout里Filename中填写的路径是“Pics\abc.wav”，es或者es2文件放在“D:\Program Files\”里，那么此abc.wav的绝对路径是什么？

②上一题中，假设只把es或es2文件移动到其他文件夹，会对实验造成影响吗？如果把图片文件和相应文件夹一起移动过去呢？

③如果把Pics这个文件夹名称改为Images，那么如何修改Filename？

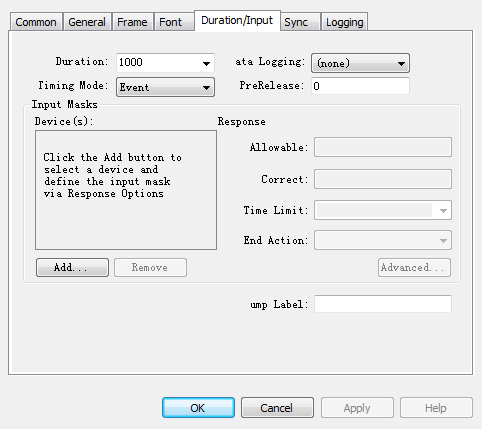
## 按键反应

目前您应该已经可以让文本或者图片按我们的要求呈现，并学会了什么是路径。

但是心理学的实验里，我们常常要进行按键反应，并获得反应时。如果您不知道这两个词是什么，建议您问一下同学或者老师，此处不解释。

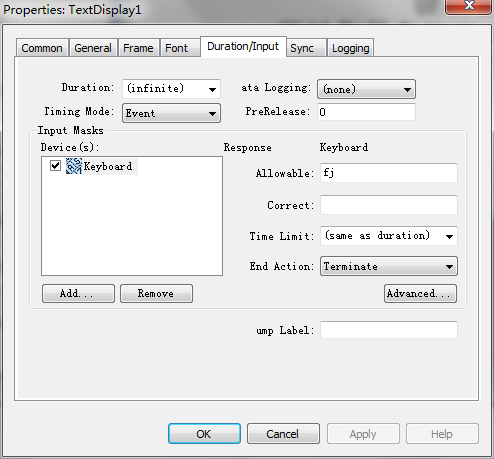
现在新建一个es或es2文件，在里面加入一个TextDisplay或者ImageDisplay，按照你喜欢的方式来呈现点什么东西。

我们假设我们的实验只允许被试按f键和j键，并且按下后才继续，那么如何设置呢？首先要到下图所展示的属性页面里。



单击图中框起来的按钮。

里面有两个选项，请你想一想，本次任务应该选择哪个？



请确认各个地方是否和上图一致，全部一致的情况下，这个时候你再运行E-Run，你会发现除非你按下了小写f或j（按键区分大小写），否则程序不会结束。（除非按下Ctrl+Alt+Shift来强制结束）**（请参考“示例程序1-5”）**

此外，建议你运行之前，设置输入法为默认的英文键盘。

上图中的各属性中：

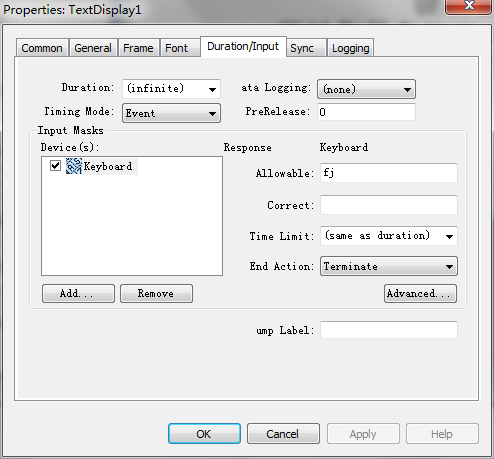
Allowable代表允许的按键，默认为{ANY}，也就是任意按键。Correct代表哪些是正确的按键，之后我们再介绍其作用。（提示：关于特殊按键可以查看附录）

## C:\Program Files\Microsoft Office\MEDIA\CAGCAT10\j0298653.wmf阶段练习

假设我现在想把允许的按键改为dk，我该怎么做？假如我想要即能使用键盘上的dk键，又能用鼠标的左键和右键呢？

## 刺激呈现的时间和允许按键的时间

继续看上面的图：



我们接下来解释一下Time Limit这个属性是做什么的。首先如果你看不懂这两个英文单词，请先去翻译。

很简单，这个属性表明了，你希望给被试按键的时间限制。例如你Duration设置了10000，但是Time Limit设置为3000，则如果被试在3秒内没有按键，再按键就无效了。**（请参考“示例程序1-6”）**

## 多次呈现——Procedure和List

现在我们学会按照想要的方式呈现图片和文字，并且进行按键反应了。理论上，我们可以开始设计实验了。

等等！问题来了！要是我们要呈现30次不同图片，难道我们要设置30个不同的ImageDisplay？天哪？那要做多久？

我们当然不需要做这样的苦力活，E-Prime提供了一种办法来方便我们多次呈现图片或文字，乃至声音、视频等等。

接下来的内容相对于前面会稍难一些，请仔细地阅读。

首先先在**Toolbox窗口**中找到**Procedure**和**List**这两个控件。

您应该也早就注意到，实验程序的树状图里总是自带SessionProc这个E-Object，这就是一个Procedure控件的实例。（可以理解成个例，个体，比如你是人类中的一个个体，也就是一个实例。对应地，也就是说这个SessionProc是一个Procedure，名字叫“SessionProc”，当然你也可以任意改它的名字。）



所以，我们可以知道，我们之前放在实验中的各种E-Object的实例（也就是之前我们所使用的TextDisplay，ImageDisplay），都是放在SessionProc这个Procedure里面的。Procedure的中文意思就是“过程”，也就是我们之前放入的各种E-Object，组成了SessionProc这个过程，我们也可以改变这个过程里的各种E-Object呈现的顺序，属性等等。

**请尽可能理解上面两个自然段的话，如果暂时无法理解，可以跳过。**

而多次呈现文本和图片刺激的关键就在于List和Procedure的结合应用。

接下来请您先将一个List放入实验程序中，然后虽然冒着被你骂的风险，然而我还是想说：请自行研究出如何在List窗口中的表格里加入行和列，以及如何删除行和列。等您自行研究完毕，才继续阅读。



## List的一行，可运行一个Procedure

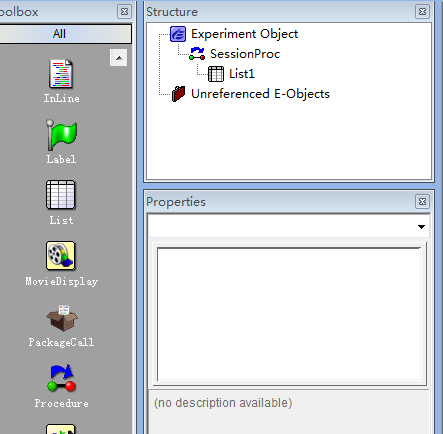
通过研究，您应该已经知道如何添加行（Level）和列（Attribute）了。Attribute是属性的意思，但你不需要管，你只要记住这是列就可以了。

List自带着3个可修改的列，分别叫Weight，Nested，Procedure，现在不用理会Weight，Nested这两个列，这两个列将在“进阶篇”中介绍。而Procedure这个列？发现了什么？这个单词是不是很眼熟？没错，就是这个：

此时你尝试修改该列的值，你会发现有一个下拉选项供你选择，如图所示：

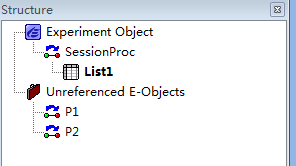


那么现在，请在Unreferenced E-Objects中加入两个Procedure（如图所示）

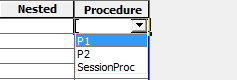


**两次**

添加完之后，分别改名为P1和P2。



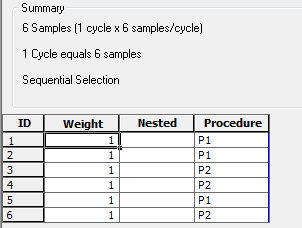
这个时候再尝试修改List1里面的Procedure，你会发现……



哈哈！所以明白这意味着什么了吧！？可这有什么用呢？

接下来，在P1中加入一个TextDisplay，呈现文本为你喜欢的话；在P2中加入一个ImageDisply，呈现你喜欢的图片。你可能会发现Unreferenced E-Objects重复出现了很多TextDisplay，但它们其实是相同的。凡是名字（Name属性）相同的一切E-Object，尽管它们可能有多个分身，但是都是相同的（你对任何一个进行修改，都会影响所有的）。至于为什么会这样，等你用多了E-Prime就会悟出来，我不解释。

然后你给List1加入6个行，并设置为如图所示：



**顺序**

设置后运行，你会发现，E-Run按照你的设置运行了6次，先是两次文字刺激（P1的过程），然后是两次图片刺激（P2的过程），然后又一次文字，然后又一次图片。**（请参考“示例程序1-7”）**

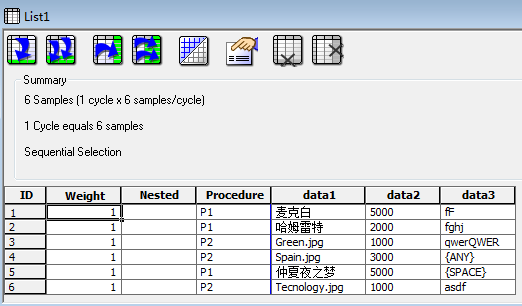
也就是说，顺序是和你设置的顺序是一样的。（如果你发现不是这个顺序，要么是写错了，要么你不小心设置了其他属性的值，我们在“进阶篇”会讲到）

然而，有时候我们在实验中需要变换我们呈现的文本和图片，我们仍然不可能设置几十个Procedure！但E-Prime当然有办法解决这个问题。

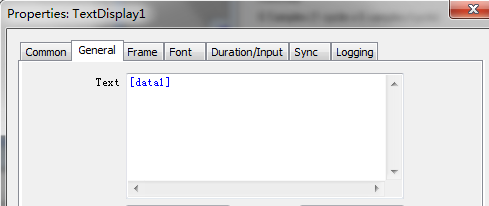
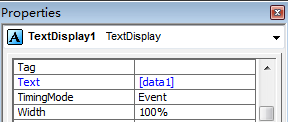
## 读取List中的Attribute

那么，如果我们要改变呈现的文字刺激中的文字，以及图片刺激中的图片，我们该怎么做呢？只要按下列方法一步一步来做就好了。

首先你需要在List1中插入三个新的列，在这次示例中，我们分别先命名它为data1，data2，data3。（双击来重命名）并将其值设置为如图所示……



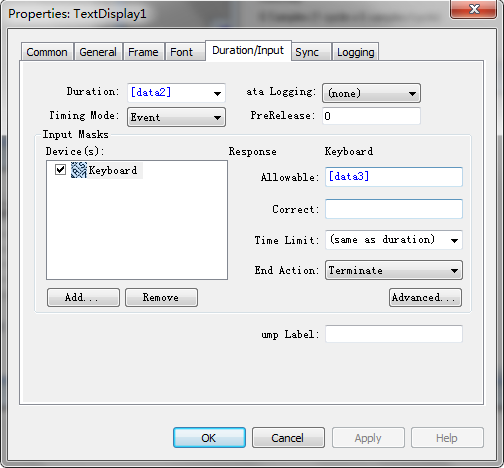
现在，我们把P1里的TextDisplay1的Text属性改成[data1]，注意包括括号，也就是如下图所示的样子。

或者

P2里的ImageDisplay1的Filename属性也改成[data1]，并从本教程自带的“示例程序1-8”里把相应的图片复制到你的es或者es2文件相同的目录里。

**然后你运行一次，看看会发生什么。**

接下来，把P1和P2里的TextDisplay1以及ImageDisplay1的Duration属性改成[data2]，把Allowable属性改成[data3]（Device使用Keyboard），想想看会发生什么？运行一下，看看是否证实了你的猜想。**（请参考“示例程序1-8”）**



## C:\Program Files\Microsoft Office\MEDIA\CAGCAT10\j0298653.wmf阶段练习

独立完成下面这个实验，**并想想有几种办法**：

在实验开始的时候，呈现指导语，指导语内容写你喜欢的东西就好。指导语字的颜色要求是白色，背景色是黑色，字体设置随你个人喜好。只要被试还没看完指导语，程序就不继续。被试看完指导语后，可以按空格键来继续程序。

实验共有四种刺激要呈现：

①刺激在中央偏左，背景颜色为红色，只允许按D（d）键

②刺激在中央偏右，背景颜色为红色，只允许按F（f）键

③刺激在中央偏左，背景颜色为蓝色，只允许按J（j）键

④刺激在中央偏右，背景颜色为蓝色，只允许按K（k）键

至于偏左和偏右的程度按你个人喜好设置，呈现什么刺激也随你的高兴。一共呈现48次刺激，每种刺激呈现次数相同，呈现顺序尽量混杂。

## 呈现正误反馈

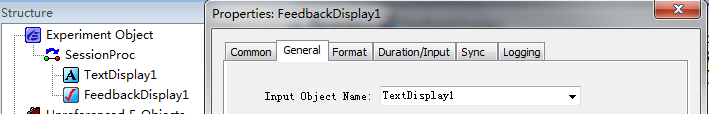
要进行正误反馈，我们需要用到FeedbackDisplay这个控件。在本小节开始前，你要先知道下列英文单词的意思：

Feedback、Correct、Incorrect、Average、Percent

接下来用一个小例子，你就能很容易地理解这个控件的作用。

新建一个es或es2文件，加入一个TextDisplay，显示的文字随你喜欢，显示时长设置为10000或无限，只允许用键盘进行按键反应，允许的按键（Allowable）设置为fj，正确的按键（Correct）设置为f。

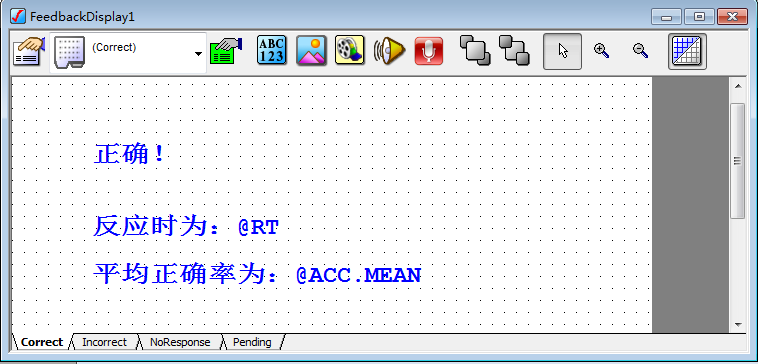
添加一个FeedbackDisplay到TextDisplay的后面，设置此FeedbackDisplay的Input Object Name为刚才加入的TextDisplay。



这个时候试着运行程序，看一下按下小写f和按下小写j，有什么样的区别？

如果你设置的都正确，你会发现，按下小写f和小写j，显示的结果是不同的。我们刚才设置的FeedbackDisplay1的Input Object Name是TextDisplay1，所以FeedbackDisplay1会检查TextDisplay1的按键是否正确，来提供反馈。因为我们设置了TextDisplay1的正确按键是f，所以按下小写f，FeedbackDisplay1便显示“正确”这个英文单词，并显示了反应时和平均正确率。

接下来看看下面这图，注意框选的地方：



你会发现左下角有四个可切换的页面，分别对应着正确时、错误时、无反应时反馈给被试的画面。当然，我修改了显示的内容（把英文改成了中文），你也可以任意地修改内容，按你想要的方式添加文字、音频、图片等等。在不同的页面可以呈现不同的东西。**（请参考“示例程序1-9”）**

上方有各种各样的按键，其作用如下：

设置整个FeedbackDisplay的属性；

设置你选择的FeedbackDisplay中的子控件（如文字、图片）的属性；

添加文本、图片、视频、音频、录音工具这些控件；

将当前选择的控件置于前方或者置于后方；

选择鼠标的类型（选择、放大、缩小）；

切换是否显示网格。

## C:\Program Files\Microsoft Office\MEDIA\CAGCAT10\j0298653.wmf阶段练习

对你上次进行的阶段练习做如下修改：

①把每个刺激允许的按键都设定为fjdk；②把“正确”的按键对应设定为每个按键在上一次允许的按键中的小写格式。③在刺激流程后增加一个反馈，要求显示反应时和平均正确率。

## 音频和视频

音频和视频的控件分别为Soundout和MovieDisplay，和ImageDisplay一样，也是用路径调用你想要的音频和视频。音视频呈现最常见的错误是格式错误，转换格式后再次尝试有时就可以了。

这两个控件一样可以进行按键反应，设置显示时间。大多数属性不需要去修改，通常只需要在正确输入Filename后，注意一下以下一些属性：

  Stop After：是否在呈现结束后继续播放。如果你想要用某首音乐当背景音乐，就把它放开头，并把这个属性设置为No吧。

 Loop：是否循环播放。

 Start Offset：从多少毫秒后开始。

 Stop Offset：在多少毫秒的地方结束。

 Start Position：从什么地方（时、分、秒、毫秒）开始。

Stop Position：在什么地方（时、分、秒、毫秒）结束。



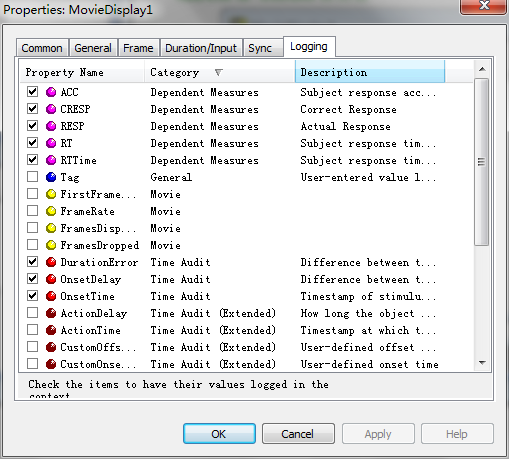
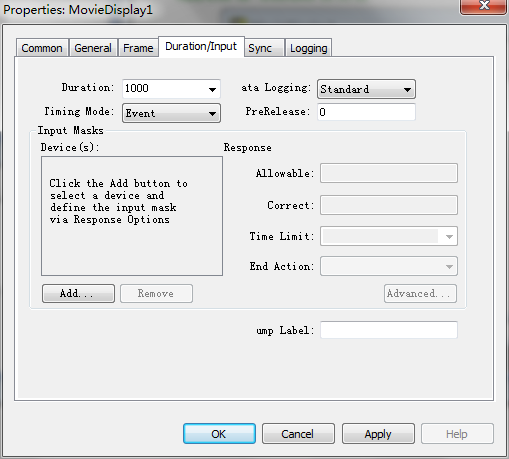
## 数据的记录

你是否已经注意到了，每次使用E-Run后，在和es或es2或ebs或ebs2相同目录下会生成一个edat或edat2文件，同时伴随一个txt文件。这些文件的文件名往往是“【你的es或es2文件的文件名】-1-1”+“.后缀名”。

这些文件就是数据文件。为什么文件的背后是1-1呢？还记得每次运行E-Run的时候，在实验开始前总是有两个问题并让你输入数字吗？你可以试着修改这两个数字，然后看看出现的文件名有什么不同。

如果你不知道这两个问题是什么？请自行查询翻译，你就可以理解为什么填这两个信息了。通过输入这两个信息，可以生成不同的数据文件。

我们暂且不管里面txt和edat文件里面记录的到底是什么，先看看我们怎么设置要记录哪些数据。请见下图：



对于任意一个可以进行按键反应并记录数据的控件：Ata Logging（显然这个属性第一个单词没显示完整）表示的是预设的记录属性的方法，选择下拉菜单其中一项后，Logging页面中的数据就会发生改变，你也可以自行对里面要记录的数据进行增减。右图列表中Dsecription一列就告诉你记录的是什么，比如说RT这个东西，就是我们平常说的反应时。

**如果你暂时不明白上一段话的意思，只要记住每次把准备要记录反应时的刺激的Ata Logging属性设置为Standard就好了。**

## 数据的查看

双击txt文件或者edat文件（如果显示要打开方式，则需要用**E-DataAid**打开），里面很详细地记录了每次刺激的数据。“.RT”记录的就是常说的反应时。

Txt文件的\*\*\* Header Start \*\*\*到\*\*\* Header End \*\*\*之间还记录了关于es或es2文件的各种信息，如日期、被试号等各种不同水平的信息。至于其他的数据，请读者们自行查看推测是什么。（逃）提示：可看上一节的Dsecription这一列。

## 数据的录入（Excel或SPSS）

我知道很多学弟学妹，甚至我见过很多学长学姐，在把数据录入到SPSS或Excel的过程中，完全是手动输入的！这简直是噩梦！看着都令人心疼。

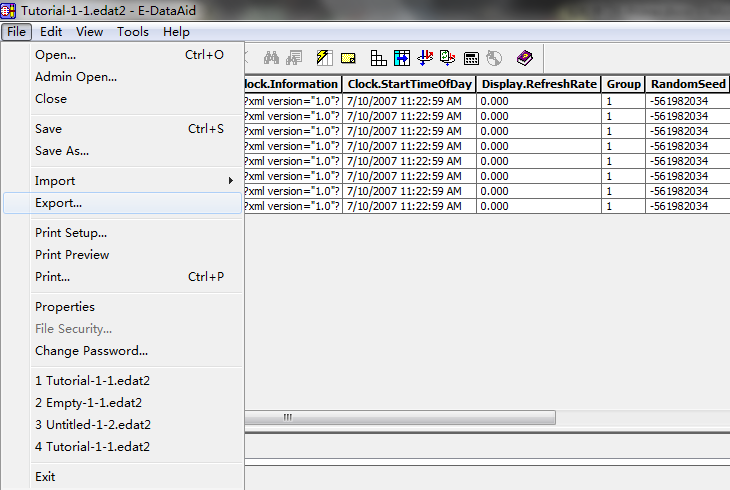
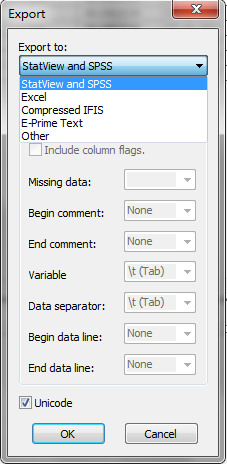
实际上，E-prime作为一个专业的软件，怎么可能让人做这样的体力活呢？这一小节将教你如何把E-Prime的结果以瞬间录入到SPSS或者Excel。

首先打开edat文件而不是txt文件（如果你只有txt，请使用E-Recovery把它转换成edat）。

对于Excel，最简单的方法就是直接用Ctrl+C复制下来，粘贴到Excel里就好了，也可以通过这种方法，再粘贴到SPSS。

但**E-DataAid**提供了另一种更好的方法来录入这些数据。

如下图，打开File这个菜单，选择Export。

点击完以后会显示如右图的对话框。在Export to里面选择你要输出的类别，另外，下方的Unicode建议你去掉，否则有可能会出错。强烈建议！

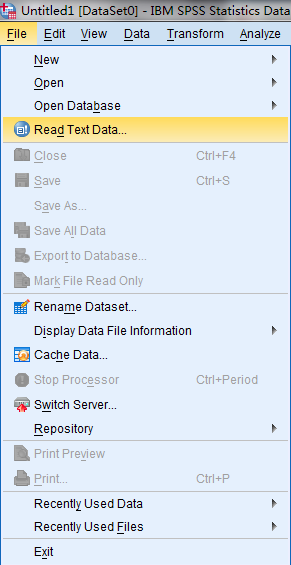
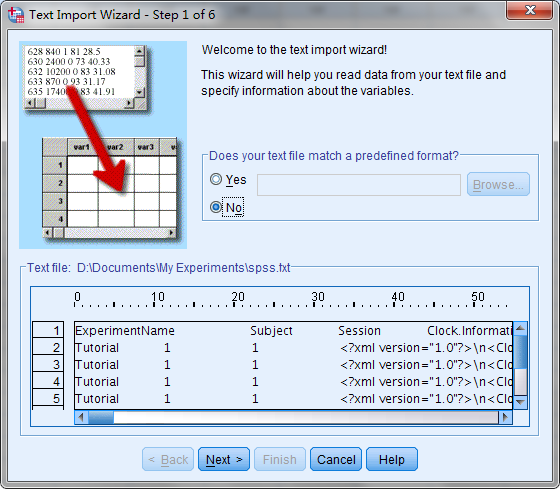
点击OK后，实际上生成的是txt文件。你可以全选里面的内容复制到Excel里，或者用Excel打开再设置录入的方式，这很简单。

接下来重点放在SPSS上，此处使用SPSS19来进行示例：

首先：按照上述方法，用**E-DataAid**的Export功能生成SPSS专用的txt文件。除非你知道该怎么做，否则一定要去掉Unicode的勾选。

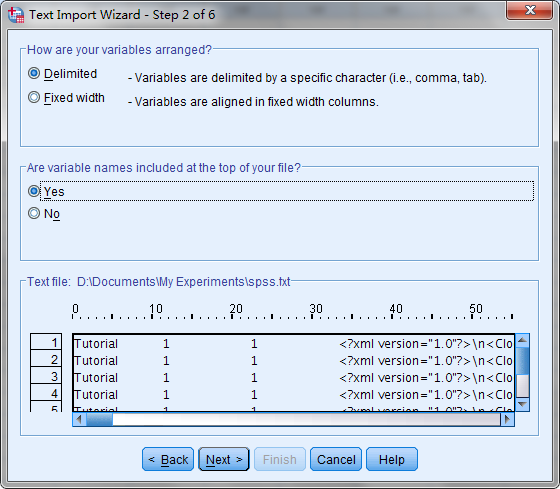
第二步：打开并耐心等完SPSS漫长的读取时间…（` v ` !）

第三步：在SPSS里的File菜单里选择“Read Text Data…”，如下左图。

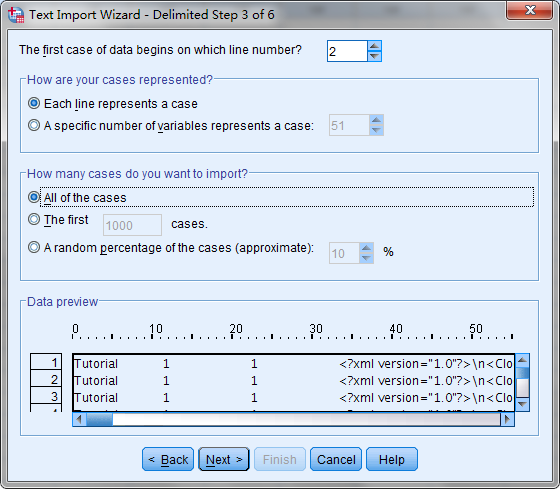
 

第四步：选择完文件后进入上方右图的对话框，直接按Next就好了。

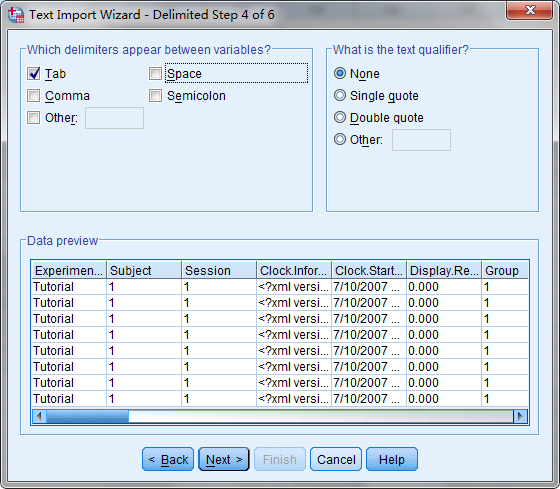
第五步：进入下面的对话框，选择Delimaited和Yes，进入下一步。



第六步：确认这里第一个框的数字是2，另外两个框的选择和下图是一致的，然后进入下一步。



第六步：确认这里左上角只框选Tab，右上角是None，然后下一步。



第七步：不用再理会弹出的任何对话框，直接下一步到最后按Finish就足够了。

以上就是SPSS以19为示例软件的录入方法，您暂时不用理会这一波操作的意义，只要知道这样就能录入就好了。中文版应该更为容易，我就不再重复说一遍了。

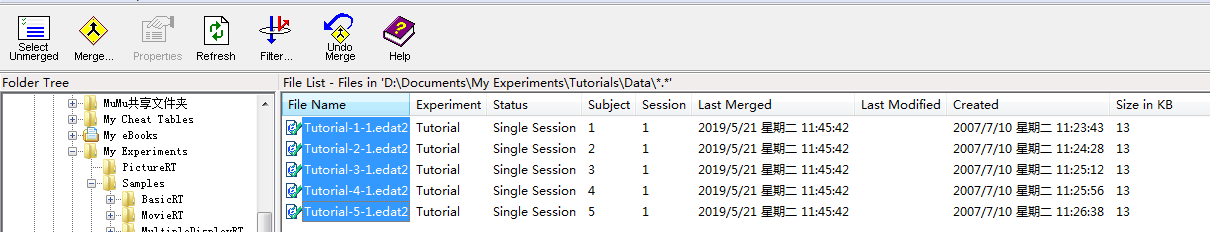
## 数据的合并

我们又遇到了一个致命的问题。我们每次进行一个实验，就生成一个文件。比如生成了“实验1-1.edat2”，“实验2-1.edat2”，“实验3-1.edat2”……“实验35-1.edat2”。我们有这么多文件，难道我们要手动将它们合并？

其实不用，E-Prime提供了合并这些文件的工具，E-Merge。



它的使用也非常的简单，如下图所示，按住Ctrl选择你要合并的文件，然后按左上角的Merge，点击OK，就可以了。生成的emrg2文件的使用方式和edat2完全一样。



## 本章结语

学到这里，你应该已经能够用E-Studio的List控件来按你想要的次序呈现文字、图片、音频、视频，并设定反应键，以及记录反应时并自由选择记录其他你想记录的数据。

你也能够自由地设置呈现的各种刺激，如它们的位置、颜色、背景颜色、呈现时间，并能够设定哪个反应键才是正确的，还能给予被试反馈。

此外，你能够把生成的数据很快地导入到Excel或者SPSS里，而不是使用手动录入这样原始、缓慢、容易出错的方法。

如果上面的一切你都已经掌握，恭喜你！在福师大心理学院本科阶段，你已经游刃有余，几乎没有什么实验程序，能够难倒你了。你可以打开书，试试自主设计一些实验程序。

在使用计算机的时候，务必记住，掌控它！使用它！凡是能够用技术快速解决的！请不要用人工来解决。电脑永远比人快速、精确，而人永远是电脑的主宰！有活请给电脑干！

下一页是本章的综合训练，用来测试你是否牢固地掌握了本章内容。

## 第二章综合练习

**带\*的题目如果无法完成，不影响进入下一个章节。**

第一关：观察前哨站

①打开E-Prime2.0官方自带案例的BasicRT、PictureRT，查看这些程序对控件属性的设置，分析它们是如何呈现的。

②\*有兴趣的话打开MovieRT和SoundRT，你可以先不用理会Inline里的内容，以及Slide是什么？你只需要研究本章所讲到的内容，查看它们如何对属性进行设置。

第二关：心理旋转测验

本关需要用到的材料在“附带材料\心理旋转测验”中。

①你需要设计一个实验程序，让被试判断呈现在屏幕中央的“R”这个字母是正向的还是经过水平翻转的，即“Я”。此外，字母R会经过旋转，共有0度、60度、120度、180度、240度和300度六种。实验的目的是，判断是否旋转的角度越大，被试判断字母“R”是正向的还是水平翻转的反应时是否变长。实验的试次、呈现时间、所按的按键等均由你自己来定。

②\*如果你使用了我为你准备的材料。那么，在实验中，我要让材料中的白色变成透明色，应该怎么做？

③运行一次你设计的实验程序，并把数据分别录入到Excel和SPSS里各一次。

第三关：女魔法师的阴与晴

本关需要用到的材料在“综合练习\女魔法师的阴与晴”中。这个程序需要进行一次练习阶段，一次正式阶段。

新建一个es或es2文件，并完成下面这个实验程序：

①练习阶段要显示希腊字母和对应的英文名字，这些材料放在“希腊字母.txt”中。每个字母至少呈现两次，每次至少3秒，不允许被试按键。

②第一次正式阶段允许被试按f和j键。这个阶段呈现练习阶段里的材料，并将顺序打乱。共呈现20次，每次5秒，每个字母至少出现1次。20次中，字母和英文名字正确对应的刺激呈现10次，错误对应的呈现10次，要求被试在正确的时候按f，错误时按j。（也就是字母与英文名字不对应时，按下j键才是正确做法）至于呈现的颜色位置等按你的喜好来定。

③\*正式阶段后面给予反馈。按键正确时，呈现附带材料中的“happy2.jpg”，并且呈现反应时和当前平均正确率，并播放“correct.wav”；按键错误和没有按键时，分别呈现“angry2.jpg”和“angry1.jpg”，并播放“incorrect.wav”。

④\*做完这一切后，请看参考答案。你会发现我的TextDisplay里用了我之前没说到的技术。从中你可以得到什么启示呢？

⑤就当玩一次小游戏，运行一次答案，并把数据分别录入到Excel和SPSS里各一次。

第四关\*：恶魔的地堡（娱乐题）

用你学过的知识来使用E-Prime设计一个打地鼠游戏，用键盘操作，一共要按8个按键。设计的越精彩，呈现的刺激越丰富越好。

# 第三章- E-Prime进阶教程—理解并进一步操作

## C:\Program Files\Microsoft Office\MEDIA\OFFICE14\Bullets\BD14752_.gif导言

感谢您读到这里，如果您是福师大心理学院本科生，但是对心理学实验或者对这本书接下去的内容**不感兴趣**，那么你可以去做别的事情了。时间如此宝贵，就不要用在你认为不重要又不感兴趣的事情上面了。因为**第二章**的内容绝对够福师大本科阶段使用了，即使偶尔不够用那就问问同学老师也能解决问题。

但是如果你对E-Prime感兴趣，或者觉得这本书有意思想继续看的话，那么很欢迎你的继续。

不过我要说的是，已经到了这里，说明你已经对E-Prime比较熟悉了。接下去的内容我讲的**不会像基础篇那么详细**，不会一步步地介绍，不会有那么多示例程序，不会提醒你要查英文单词，简单的步骤更是不会说了。

那么想继续的话！开始吧！

## C:\Program Files\Microsoft Office\MEDIA\OFFICE14\Bullets\BD14752_.gifSlide的应用

Slide和FeedbackDisplay几乎是完全一样的。实际上，FeedbackDisplay就是一个特殊的Slide。下图为工具箱中的Slide和树状图里的Slide。

如果你玩过了第二章综合训练，第三关中的③，那么你应该尝试过在FeedbackDisplay的页面中同时加入文字、图片和音频了。

Slide也是完全一样，可以在同一页中加入多个刺激。所不同的是，Slide的页面完全是由你自己控制的。您会看到，Slide比FeedbackDisplay多了三个按键。

按键如下：



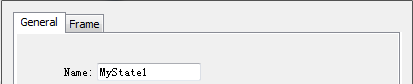
这三个按钮的作用分别是，增加一个Slide页面、删除一个Slide页面、复制一个Slide页面。**如果你忘了其他按键的功能请回到“呈现正误反馈”一节查看。**

点击了增加页面后，你就会看到窗口左下方增加了一些页面，就和FeedbackDisplay一样，可以显示不同的内容。



多个页面

选择相应页面后，按下按钮，可以设置该页面的名字，比如我现在把Default改成MyState1。

现在点击这个按钮，里面是这整个Slide的属性。除了也有呈现时间，按键反应设置外，还有一个重要的属性：



属性名称没有显示完整，很明显是ActiveState，这表明了Slide将会显示哪个页面。还记得吗？FeedbackDisplay根据按键反应正确与否显示不同页面，而Slide则是可以自己选择的。

您现在可以尝试一下，在不同页面里加入不同的刺激，然后修改ActiveState这个属性，看一下显示结果有什么不同。

此外，ActiveState和其他属性一样，可以通过方括号来调用List中的内容，这样就可以在运用List的同时按自己的要求改变Slide呈现的页面。



请打开官方自带教程的SlideRT.es2，结合本节分析里面的内容，你就会很清楚了解怎么使用Slide了。

## C:\Program Files\Microsoft Office\MEDIA\OFFICE14\Bullets\BD14752_.gifWait的应用

Wait这个控件样子如下图：

它非常简单，简单到我认为不需要讲。它的功能就是让被试等待几秒。我只提醒一点，如果你把它放在一个Procedure的开头，你会发现一轮Procedure过去后，刺激没有消失。

还是举一个具体的例子：在一个List里，运行Procedure1一共三次，Procedure1只包含一个Wait，持续时间3s，一个TextDisplay，持续时间3s。你会发现，从第二个TextDisplay开始，每个TextDisplay呈现时间变成了6s，直到最后一个TextDisplay为止。实际上，这是因为TextDisplay呈现后，并不会清除之前显示的内容。解决这个问题的方法就是把TextDisplay的Clear After属性设置为Yes。还记得我们在Soundout的时候也讲过Clear After吗？原理是一样的。

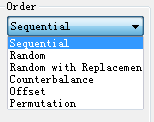
## C:\Program Files\Microsoft Office\MEDIA\OFFICE14\Bullets\BD14752_.gifList的进阶应用

**1.List的运行顺序和运行轮数**

如果我们要让List中的各个刺激的呈现顺序随机，应该如何做呢？你可以选择用其他软件来改变它们的顺序，但这意味着所有被试做的仍然是同一顺序，除非你每次都改变一次顺序。

实际上，E-Prime提供了随机的方法，也就是其他排序方法，具体如下：

在List的属性页面的Selection这一页中，有一个Order属性，里面有六种选择。



您只需要知道前三个就可以了，后三个**可以跳过**（读不懂也不要太在意）：

Sequential：也就是按列表顺序进行。

Random：选择了这个，List的顺序被随机打乱。

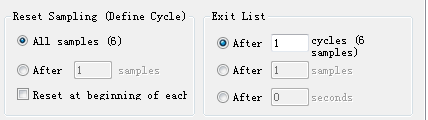
Random With Replacement：这也是随机，然而这个随机是每次都从整个List中随机挑选。也就是说，有些刺激可能呈现多次，有些可能一次都没出现。

Counterbalance\*：根据您选择的变量来确定呈现的行。比如说，你在Order By里选择Subject。那么，在实验中，如果Subject输入的是2，则此处只会运行第2行的那个Procedure。如果数字比List中的行数还大的话，那么取余数。**选择这个会改变Reset/Exit这一页中的属性，影响其他选项。**

Offset\*：根据您选择的变量来确定呈现的行。比如说，你在Order By里选择Subject。那么，在实验中，如果Subject输入的是2，则此处从第2行开始运行，一直运行完一轮，然后回到开头运行第1行，这样走完一个周期。同理，如果输入3，则运行到末尾后，回到开头运行第1行和第2行。如果Subject的数字比List中的行数还大的话，那么取余数。

Permutation\*：事先生成所有可能的组合。你输入的Order By决定了这次运行的会是哪个组合。

在List的属性页面的Reset/Exit这一页中， Reset Sampling属性和Exit List属性分别决定了怎么样算走完一轮，以及要怎么退出List。



Reset Sampling (Define Cycle) :

All samples表示所有行都呈现完算一轮。

After x samples就是走过x个行就算一轮。

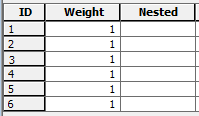
Reset at beginning of each Run则是写代码的时候用的，暂时不用理会。

至于Exit List里面的三个选项都是很简单的英文，请你自己想一想它们是什么意思，做什么用的。（逃）给个提示：After 2 cycles的意思就是运行两轮就算结束。

结合List的运行顺序和运行轮数，就可以设计出很复杂的，符合平衡的实验设计了。**请自己多多尝试，这里不再给出示例程序或者布置练习。**

**2.List的Weight和Nested**

List里面有两列一直都存在，然而在之前我们从未用过，就是下图这两列：



想要知道Weight的作用，你只需要随便选择一个已经完成的实验设计，把里面某一个List的一行的Weight属性改成3，然后运行一次就好了。

我会告诉你你会发现，Weight的作用是修改这一行的运行次数？准确来说，实际上是相当于行数。比如输入3，就相当于有相同的3行。

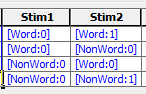
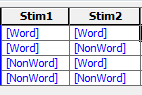
而Nested属性则允许你的这一个Sample（行）调用Nested里提到的一个或多个List里的列。这听上去很难理解，没关系，打开官方示例的NestingRT。你会看到它的TrialList里面的Nested属性用到了WordList和NonWordList这两个List，有的用到两个有的只有一个。

在后面的Stim1和Stim2中，就调用了这两个在Nested属性里中的List的相应的列。

试着运行看看，你会发现显示的文字刺激就在WordList和NonWordList这两个List中。

这样做有什么好处呢？好处就是我们做到了对这两个属性都进行了随机。发现了吗？这三个List的排序方式都是Random。

此外，这次调用还有一个不同是，调用的后面还有一个“：x”，也就是冒号加数字，如下左图所示：

而我们平常是按照右图那样来调用List里的列的。

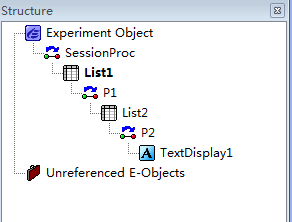
为什么要加上冒号和一个数字呢？这是因为每次随机会生成一个列表，不加上冒号和数字的话，总是默认调用该列表的第一个数据。而加上冒号和数字以后，则会调用相应数字的数据。

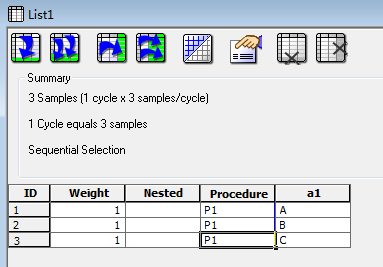
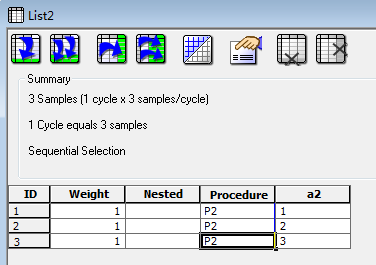
请您试着改成右边的样子，再次运行，比较一下，很快就能理解了。

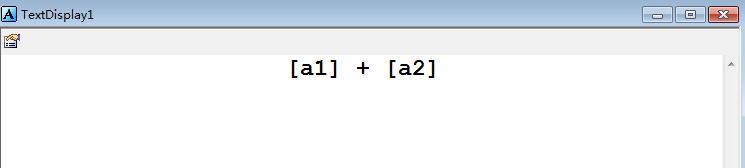
**3.List的Procedure中包含List**

这个恐怕您早就使用过了，通过这种方法，可以很容易地做不同自变量下的刺激的变化。

不多说，只要我上四张图，相信你很快就能理解。如果不能理解，就自己去尝试一下，马上就知道答案了。



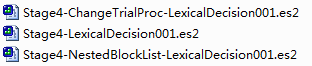
 



想想看，根据这四张图的内容，结果会是什么？刺激会呈现几次？自己尝试一下看看是不是预想的结果。

## C:\Program Files\Microsoft Office\MEDIA\CAGCAT10\j0298653.wmf阶段练习

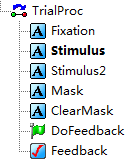
分析“E-Prime2.0官方自带案例\Tutorials\Using E-Studio Stages”中的下列这些实验程序，看看是否理解：



分析完后可以接着看，之后有实战练习。

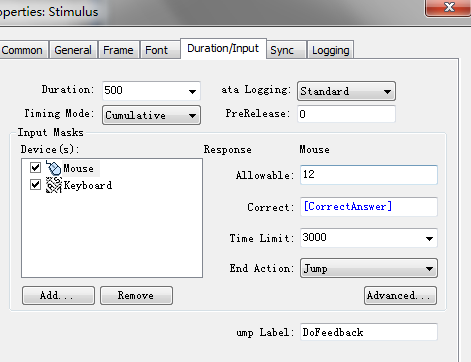
## C:\Program Files\Microsoft Office\MEDIA\OFFICE14\Bullets\BD14752_.gifTime Limit!!End Action！！Jump！！

打开官方示例中的NestingXRT，试着运行。查看控件的属性，明明只有Stimulus是有设置按键反应的，但实际上到CleatMask之前按键却都有效！



而且，你会看到，Feedback的Input Object Name只设置了Stimulus，也就是只接收来自Stimulus这一控件的结果。

这是怎么一回事呢？其原因就在于这个程序的Stimulus的Time Limit属性得到了设置：



Time Limit属性默认是Same as Duration，也就是和呈现时间一样。但它也可以比呈现时间短、也可以比呈现时间长。如这里的3000，刚好是后面几个TextDisplay的呈现时间之和。所以即使Stimulus的文字刺激已经不呈现了，按键仍然有效。

现在再注意这一点，只要你一按键，不管在Stimulus到DoFeedback之间的哪个文字刺激，都会立刻跳转到Feedback那里。

这是因为设置了End Action和Jump Label这两个属性。

把End Action设置为Jump，然后在Jump Label输入当前程序中的某个Label控件的名字，Label控件也就是下面这个东西：

这个Label控件就相当于一个标记，专门用于Inline语句的使用或者此处的End Action的Jump的使用。

设置了End Action为Jump，并输入了某个Label控件的名字后，在你按键反应结束的时候，程序就会自动跳转到Label的位置，执行它之后的东西。在NestingXRT.es2这个实验程序中，也就是执行后面的Feedback了。

## C:\Program Files\Microsoft Office\MEDIA\CAGCAT10\j0298653.wmf阶段练习

完成以下这几个任务：

①设计一个实验，呈现10次刺激，刺激是一些算式，有的是正确的，有的是错误的，被试需要进行判断。刺激呈现后的5秒内都可以按键反应，但是刺激只呈现1秒。被试按键后给予反馈。

②利用End Action的Jump功能设计一个死循环的实验。（可以用Ctrl+Alt+Shift强制退出）

③设计一个实验，呈现20次刺激。准备两种图片类型，一种是A类图片（由你来定义），一种是B类图片。每个刺激都会在屏幕上偏左位置和偏右位置呈现两个图片，当两个图片都是A类图片时，按f键；否则，按j键。要求共有5次是两个图片都为A类图片；5次是左边呈现A类图片、右边B类图片；5次右边A类图片，左边B类图片；剩下5次两个图片都为B类图片。

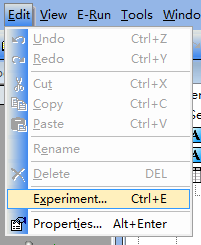
每次刺激，呈现的A类图片和B类图片都要是随机的。刺激呈现1秒，但是被试可以在3秒内做出反应。按键后有反馈。

此外，A类图片共2张，其中一张出现的概率为75%，另一张只有25%。B类图片也只有两张，呈现概率相等。

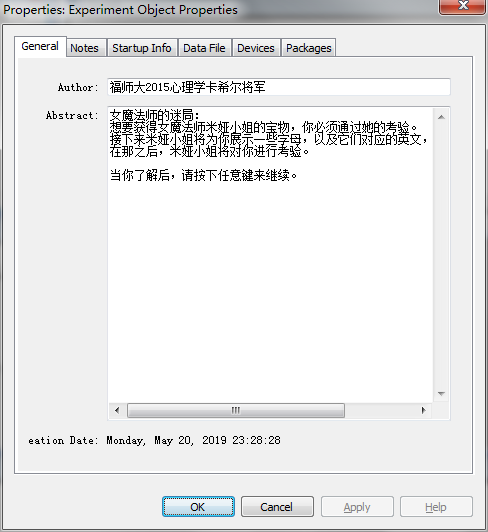
## C:\Program Files\Microsoft Office\MEDIA\OFFICE14\Bullets\BD14752_.gif实验程序的另一些设定

如果你是新手，想必上面的学习对你来说是非常不容易的，能走到这里，恭喜你已经不是一般的“福师大本科生”了！

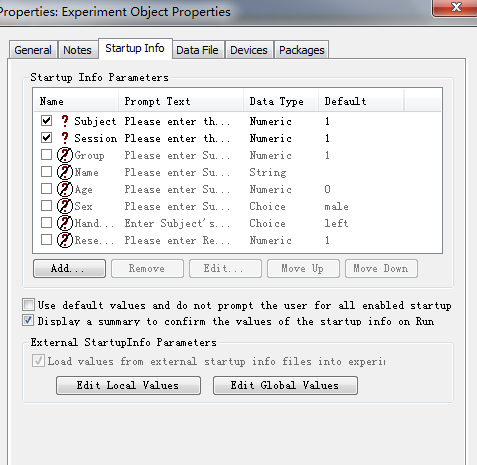
这一小节我们将讲一讲另一些有关实验的设定。

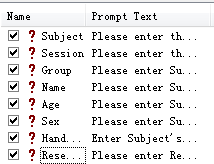
双击左图的树状图上的东西（也就是树状图的根部），或者点击右图菜单里的按钮，或者按Ctrl+E，可以打开下图的一个窗口。



如图，在这个窗口里我已经写了作者信息和摘要。这一部分对实验没有影响，如果没有特殊要求或不注重版权也可以不写。请跳转到下面那一页：

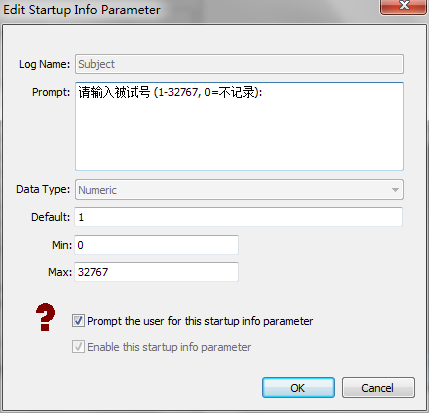
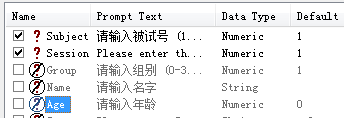


这一页上面你看到很多个问号，有很多被禁止住，也没有打勾。现在你那些问号都打勾，再单击一下问号，禁止标志就会消失。变成下面这样：



现在再运行一次程序，体会一下发生了什么，然后再继续阅读。

当然，你还可以按Add按钮来增加自己想要增加的问题，还可以按Edit把这些问题修改成中文的。比如下图所示：

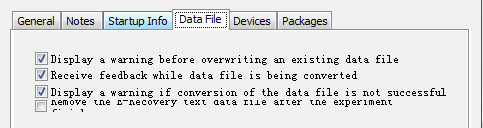
 

这样，你再运行的时候，这些问题就变成中文的了。

至于下方那两个多选按钮：



上面的英文写的非常明白了，我就不翻译了。



包括Data File的这一页。

## C:\Program Files\Microsoft Office\MEDIA\OFFICE14\Bullets\BD14752_.gif你不想让别人知道你的实验程序？

有时会下面这几种情况：

1.你希望做双盲实验。

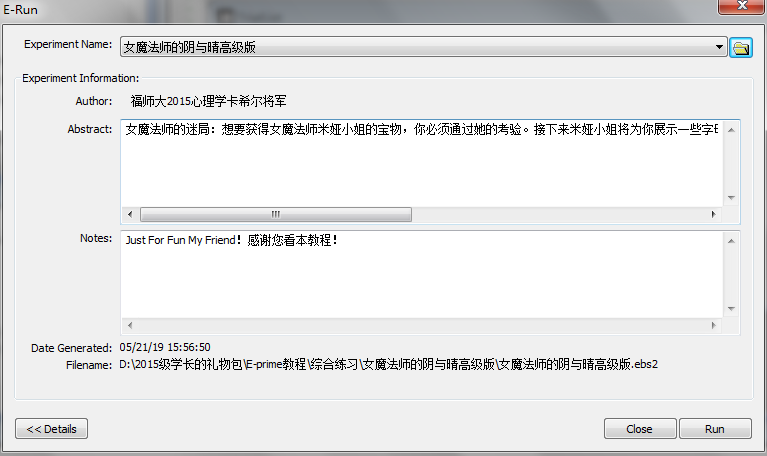
2.你想分享你的实验，需要让别人知道实验内容和作者名字。

3.你想让别人帮你找被试，帮你做实验，但是你不希望你辛辛苦苦设计的实验程序被别人抄袭。

那么，你就需要学习这一小节的内容。

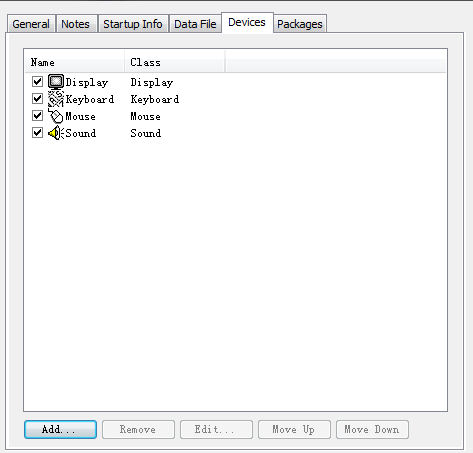
这一小节非常简单，上一小节我们介绍了怎么在你的实验程序里输入作者信息，摘要等等。这些是做什么用的呢？是给别人看的，给用你这个实验程序的人看的。

在E-Studio里，我们每次运行实验，都会在相同目录里生成一个ebs2文件。如果你遇到上面三种情况，很简单。你只要把这个文件，和相应的图片音频等文件复制给他/她就行了。这样，他/她看不到你的程序设计，但是可以用E-Run来运行，从而进行实验。下面就是一个例子：



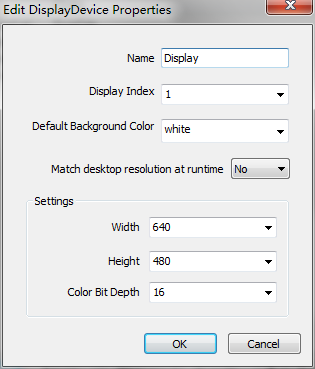
## C:\Program Files\Microsoft Office\MEDIA\OFFICE14\Bullets\BD14752_.gif修改设备（分辨率、颜色位数、增加设备）

在实验程序属性设定的这一页面中，你可看到目前接入的各种设备：



Display也就是显示器。左下角还有增加按钮（框选处），比如说你有多个显示器，或者其他各种鼠标键盘以外的输入输出设备，你可以在这里添加。

双击Display这个设备，你可以在里面设定分辨率，颜色位数等等。这些操作非常简单，就不更具体介绍了。

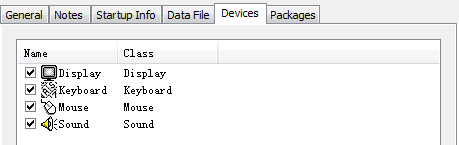
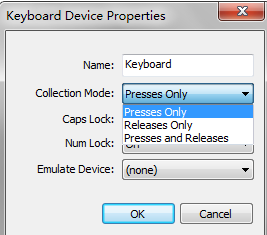


如果你有多个显示器，记得修改Display Index到相应的显示器序号上。

此外，有时候，程序的出错也有可能是分辨率和颜色位数的原因。

## C:\Program Files\Microsoft Office\MEDIA\OFFICE14\Bullets\BD14752_.gif松开按键时记录反应时

有些实验有特殊需要，需要在松开按键时才记录反应时，那么我们首先需要在实验程序的设置中设置键盘或者鼠标这两个设备（依照需要）。

你需要修改的是Collection Mode这个属性。这个属性表明了以什么模式来收集按键反应。

Presses Only就是只收集按下的反应，Releases Only是只收集松开的反应，而Presses and Releases则既收集按下的反应，也收集松开的。

默认情况下是Presses Only，所以如果你需要记录松开按键时的反应，就必须到这里来设置，设置为那种模式就按你的需要了。

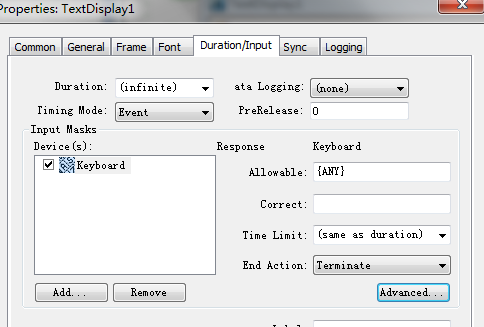
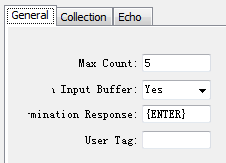
设置完毕后，打开你需要记录松开反应的控件。在允许的按键（Allowable）中把想要作为松开的记录键加上一个负号，并用花括号括起来，如：{-f}就是松开f键时记录。很简单，去试试看吧。右边是例子：

## C:\Program Files\Microsoft Office\MEDIA\CAGCAT10\j0298653.wmf阶段练习

设计一个实验，要求被试在看到刺激的时候立刻松开f键。

## C:\Program Files\Microsoft Office\MEDIA\OFFICE14\Bullets\BD14752_.gif接收多个按键与中止接收

有时候，我们需要让被试输入多个按键。这种情况，我们只需要让控件接收按键的个数变多就行了。设置方法如下：

点击左图Advanced按钮，弹出右图的对话框。其中，Max Count表示了需要记录多少个按键，当按键次数没到的时候，只要还在刺激呈现时间内，刺激就不会消失。您可以试试把一个刺激的Max Count改成5，看看会是什么结果。

Termination Response表示按下哪个键会结束这个刺激（也就是就算还有呈现时间或者还没到Max Count的数量），比如这里设置了ENTER。

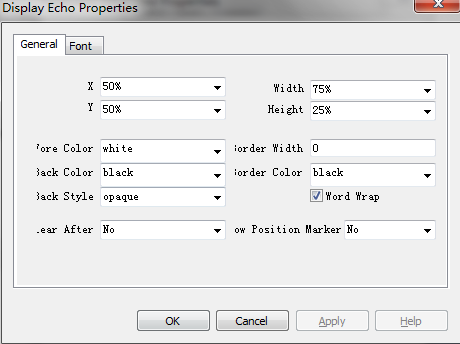
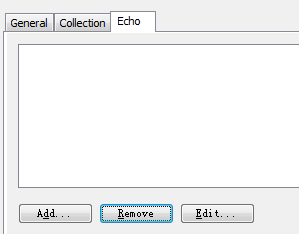
当然这样能做到的事情还很有限，只有结合了自定义代码才能让功能更加强大！

## C:\Program Files\Microsoft Office\MEDIA\OFFICE14\Bullets\BD14752_.gif输入框制作

如果你需要让被试输入某些信息，那么首先你需要知道上一部分的内容。我们接着上一小部分继续讲。现在请先按步骤进行，理解后再自由设定。

第一步：把Max Count改成大数字，如100；Termination Response为{ENTER}。

第二步：在点击上面的左图的Advanced按钮弹出的对话框后面，除了右图的页面以外，还有两个小页面，本小节看Echo页面，请进行如下操作：



首先点击Add来增加一个Display，然后双击它或者选中它点击Edit，出现右图的对话框。把X和Y设置成这样。

第四步：确认无误后，点击Apply，再点OK。

第五步：运行程序！享受胜利的喜悦或者又一次失败的苦果。

**（可参考“示例实验2-1”）**

运行正确的话，你会看到在白色屏幕上有一个黑色矩形，输入内容后，黑色矩形上显示白色的字。

这虽然可能不太令人满意，不过好歹能让被试输入信息了。你可以多尝试修改其他的属性，看看能不能让这更美观。还可以试试在Slide中使用。

建议：把Echo中的Display的Back Color改成white，Fore Color改成Black，并把X和Y的值再作调整。

## C:\Program Files\Microsoft Office\MEDIA\OFFICE14\Bullets\BD14752_.gifClear After，Echo这一切暗示了什么？

这一切暗示了E-Prime里的这些Display都是和屏幕大小一样的一块一块的东西。也就是说，这些TextDisplay或者ImageDisplay实际上是呈现一个方框覆盖了整个屏幕，并呈现你设置的内容。Echo里的Display的宽度和高度更低，所以无法遮挡整个屏幕。

无法理解？不要紧，做一个这样的测试就行了。连续用两个ImageDisplay，都读取大图片。什么都别动，把第二个ImageDisplay的X和Y都设置为0，看看会有什么结果。然后把第一个ImageDisplay的Clear After设置为Yes，再看看结果。再试试修改它们的宽度和高度，看看修改的是什么。

## C:\Program Files\Microsoft Office\MEDIA\OFFICE14\Bullets\BD14752_.gif本章结语

学完本章的内容，你在福师大本科生中已经是大神了。如果你喜欢哪个同学，和老师商量，让老师安排一下，给他/她布置困难的任务，然后让你去教他/她，哈哈！机会就来了！

学到这里，你应该已经能够很熟练的应用List达到你想要的效果，如随机、随机组合，这样，你在设计一些复杂实验的时候相比于只会第二章内容的同学可以走捷径。你能够用Slide来图文并茂地呈现刺激，知道如何根据情况需要呈现不同的Slide页面。

你还知道如何在刺激消失后仍然在规定时间内允许按键反应，知道如何用End Action来跳转当前实验程序的位置。

你还懂得了用E-Prime对实验程序的设置，来增加在实验开始前的问题，或者把它们改为中文版。以及如何让别人无法查看你的实验程序，而能够运行你的实验程序，从而做到双盲实验或者在不暴露你的实验程序编写方式的情况下分享你的实验。

你也知道了如何设置显示器的分辨率，如何添加输入输出设备，以及如何让设备在松开按键时记录反应时，甚至知道如何在一个刺激中收集多个按键，做一个相对破烂的输入框。

恭喜你现在成为了“福师大心理学本科生实验设计大神”！

## 第二章综合练习

**带\*的题目如果无法完成，不影响进入下一个章节。**

第一关：简易的障碍

如果你想要随机呈现实验中的一系列刺激，你该怎么做？

第二关：普通的指示

按下列指示来完成一个实验：

①新建一个实验文件，输入你的作者信息和实验摘要：“这是一个实验”。设置此次实验的屏幕分辨率为800x600，颜色位数16，默认背景颜色为黑色。

②把实验开头询问被试编号的问题改成中文，并添加询问年龄和询问性别的两个问题。

③在SessionProc后面添加一个TextDisplay，命名为Intro，里面输入任意的文字，颜色、位置任意，呈现时间无限，设置只有在松开空格键的时候才能继续。

④在Intro后面添加一个List，命名为BlockList，给它添加一个列，那个列命名为“Allowable”。再添加两个行，第一个行的Allowable里输入fj，第二个行里输入dk。此外，第一行的Procedure填上Block1，第二行的Procedure填上Block2。

⑤在Block1这个Procedure后面添加一个Block1Intro，里面输入任意的文字，颜色、位置任意，呈现时间无限，设置只有在按下两个任意键后才能继续。

⑥在Block1Intro后面添加一个List，名为DesignList1。在这个Block中，要求用到的TextDisplay的Allowable调用BlockList的Allowable”一列，来完成下述实验：

要求被试对呈现的词语进行判断，对于三个字的词语按下f键，两个字的按下j键。刺激呈现1秒，但是允许被试在刺激消失后2秒内进行反应。刺激呈现前显示注视点。按键结束后有反馈。

⑦在Block2这个Procedure后面添加一个Block2Intro，里面输入任意的文字，颜色、位置任意，呈现时间无限，设置只有在按下五个任意键或者按下空格键后才能继续。

⑧在Block2Intro后面添加一个List，名为DesignList2，设置其排序方式为Random With Replacement。在这个Block中，要求List的Nested功能，来完成下述实验：

实验将呈现A类词和B类词，每种词有8个。一共呈现20次刺激，每个刺激都同时呈现两个词语，如果这两个词语属于同一类词，按下d；否则按k。两个词为同一类词的刺激有10次，不为同一类词的也是10次。DesignList2中Nested的所有列表排列方式都要为Random。刺激呈现时间自定。

第三关：女魔法师的晴与雨\*

本关需要用到的材料在“综合练习\女魔法师的阴与晴高级版”中。

①如果你做了把第二章中的第三关。利用本章所学的知识，将女魔法师的反馈随机化。当按键正确时，女魔法师的表情在blink.jpg、happy.jpg、happy2.jpg之中随机，并呈现相应的文字；当按键错误时，女魔法师的表情在angry1.jpg、unhappy.jpg、cry1.jpg中随机，并呈现相应的文字；没有反应时为angry2.jpg，并呈现相应的文字。

②利用本章学的知识，将实验程序中希腊字母与英文名字的匹配随机化，并且正确的有10次，错误的也是10次。

③查看参考答案“女魔法师的阴与晴高级版.es2”，参考一下对表格的嵌套，以及关于调用变量的处理方式。

第四关：学习程序\*

用你所学的一切，设计一个学习程序。这个程序专门用来做选择题，对于同一道题，选项的位置（在ABCD上的位置）可能是随机的。

制作得越精良、越合理、越引人入胜越好。

# 第四章- E-Prime高阶教程—使用自定义脚本

## C:\Program Files\Microsoft Office\MEDIA\OFFICE14\Bullets\BD21295_.gifE-Basic

从这一章开始，你将打开一个新世界的大门。你将开始学习手动输入代码。

我仍然提醒你，如果你对心理学实验其实不感兴趣的话，你可以打住了；如果你对编程感兴趣，本章的“类”这一节涉及编程思想，你看看这个就够了，然后去学其他编程语言吧。学习编程，想要入门的话，虽然很奇怪，我推荐“魔兽3地图编辑器”，玩玩里面的触发器系统，有趣又能学习编程思想。

如果你的确对E-Prime非常感兴趣，想要更进一步地了解它的**强大功能**，那么欢迎你继续！

E-Prime的专用脚本语言称为E-Basic，语法和Visual Basic极为相似。关于它的各个Object的方法和属性，全都在它的帮助文档里，也就是E-Basic.chm。想要打开E-Basic的世界，你必须知道如何查阅这个文档，找到自己想要的内容。编程重在思维，单单靠背是不可能的，得学会查阅文档，就像打开字典。

顺便小小吐槽一下：我们学院本科生大部分都用E-Prime而不是Matlab或者Presentation，那我觉得二级还不如学VB呢！何必跟风选择C语言呢？毕竟VB又简单，又可以迁移到E-Prime上。而Presentation的语法很像C语言，然而用的人少，而且不少人使用过程中只知道固定用法，基本上都是没有理解的。

不同的人可以用不同的方式开始这一章：

0基础：接着看下去，认真写代码，模仿。

学过其他编程语言：可以看得很快。

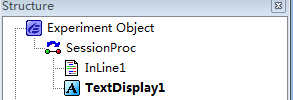
拿了C语言二级证书然而学的不清不楚，低分飘过：你是0基础。

学过VB：直接跳到“查阅以及面向对象编程”这一节。

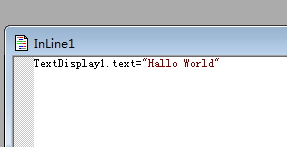
## C:\Program Files\Microsoft Office\MEDIA\OFFICE14\Bullets\BD21295_.gifHello World！

很多人恐惧输入代码，实际上只要慢慢来，一切并没有想象中的那么困难。接下来我们进行一次实际操作，您马上就会明白。

首先我们在实验程序中加入一个Inline，并加入一个TextDisplay，就像下图中那样：

将TextDisplay1的Text属性设定为“这是原来的内容。”，也就是让它能够呈现这些，呈现时间设的长一些以便你能够看完整。现在测试，TextDisplay1显示的就是我们刚才输入的话。

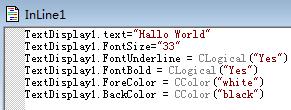
那么现在在InLine1里加入下列语句。

TextDisplay1.text="Hallo World"

如右图所示：

没错，就这么简单。然后你现在再次运行，你就会看到，TextDisplay显示了Hallo World。

所以，你应该能明白了。通过E-Basic代码，我们能对要呈现的东西进行影响。我们可以更改TextDisplay1的属性。当然，我们也可以更改它的其他属性。

比如你试试，在代码后面添加：

TextDisplay1.text="Hallo World"

TextDisplay1.FontSize="33"

TextDisplay1.FontUnderline = CLogical("Yes")

TextDisplay1.FontBold = CLogical("Yes")

TextDisplay1.ForeColor = CColor("white")

TextDisplay1.BackColor = CColor("black")

这些都是你熟悉的属性，你应该明白会有什么效果。**（可参考“示例文件3-1”）**

## C:\Program Files\Microsoft Office\MEDIA\OFFICE14\Bullets\BD21295_.gif声明并赋值一个变量

对于E-Basic的变量你可以这样理解，这是你自己可以随意改变的量。编程过程中我们时不时要声明一些变量来操作。比如，我们声明一个叫“data”的东西，他是一个字符串（你可以理解为句子），假设内容为“这是数据”。然后我们在TextDisplay里调用他，设置它为Text，TextDisplay就会显示“这是数据”了。

没错，这和用TextDisplay去调用List中的某一列是极其相似的。

如果你还是无法理解，只要继续看下去，等你完成了几个例子，你就知道是怎么回事了。

对于E-Prime，你只要掌握下面几种变量类型就行了。这里不会非常具体的介绍，如果你对VB感兴趣，直接去看有关VB的书。想要系统学习编程，更不需要在这里浪费时间了。

Integer：整数，允许的范围在-32768到+32767。

Long：更大的整数，允许的范围在-2147483648到+2147483647。

Single和Double：当做有小数点的数用就好了，一般不出现。

String：字符串，你可以理解为句子。

Boolean：布尔型。有True和False两种类型，理解成“是”和“否”就好。

尽管VB里还有多种多样的变量类型，不过对于E-Prime实验设计，差不多知道上面4种就够用了。定义变量的方式非常简单：

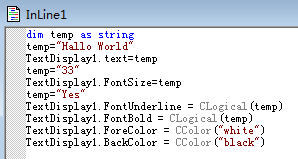
Dim x as Integer

上面的句子就定义了一个临时变量，名字为x，类型为Integer。这个时候x的数值会被初始化为0。也就是x=0。我们也可以修改它的数值，例如直接写：

x=1

这样，x的数值就为1了。

现在我们把上一节的代码修改成下面的样子：

dim temp as string

temp="Hallo World"

TextDisplay1.text=temp

temp="33"

TextDisplay1.FontSize=temp

temp="Yes"

TextDisplay1.FontUnderline = CLogical(temp)

TextDisplay1.FontBold = CLogical(temp)

TextDisplay1.ForeColor = CColor("white")

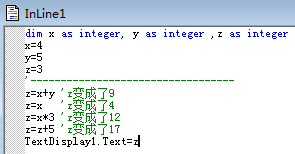
TextDisplay1.BackColor = CColor("black")

然后运行，你会看到结果没什么不同。**（可参考“示例程序3-2”）**

这就说明，我们可以用变量来替代原来的值。我们在上面的代码中，三次改变变量temp的数值，并赋给属性或者作为参数赋给函数（也就是CLogical和CColor，这两种东西称函数）。第二行中，temp变成"Hallo World"，第三行temp赋给了TextDisplay1的Text属性；接下去变成“33”，赋给FontSize；变成“Yes”，放进CLogical方法中。从这里我们也可以知道，代码的运行方向是从上往下的。

另外，变量名是可以自定的，但是不能重复。对于E-Prime，”c”这个名称是不能用的，比如：Dim c as Integer，这是不允许的，因为c已经被用走了，之后你会知道。变量名也不能以数字开头，一些特殊符号也不能包含。建议您用英文字母作为变量名就好了，可以在变量名最后用数字。

## C:\Program Files\Microsoft Office\MEDIA\OFFICE14\Bullets\BD21295_.gif变量的运算

变量之间可以进行相互运算，例如：两个Integer型，或者两个Long型，或者一个Integer和一个Long之间都可以相互加减。甚至String型也可以相加。

例如右图中：x=4,y=5

z=x+y，则z=9。

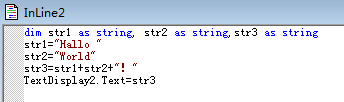
z=x，则z又变成了4。

z=x\*3，则z变成了12。

z=z+5，z又变成了17。

String型的相加如下：

dim str1 as string, str2 as string,str3 as string

str1="Hallo "

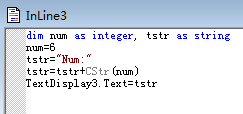
str2="World"

str3=str1+str2+"！"

TextDisplay2.Text=str3

最后的结果是str3相当于把str1和str2的内容合起来，最后加一个感叹号，也就是变成“Hallo World！”

String和Integer是不能相加的。但是可以把Integer转换成String型。

以下是一个例子：

dim num as integer, tstr as string

num=6

tstr="Num:"

tstr=tstr+CStr(num)

TextDisplay3.Text=tstr

用这种方法，最后TextDisplay3呈现的就会是“Num:6”。

请打开**示例程序3-3**来查看。并且务必完成下面的阶段练习：

## C:\Program Files\Microsoft Office\MEDIA\CAGCAT10\j0298653.wmf阶段练习

①输入一些代码，在代码中定义两个Integer，一个是a，一个是b，a赋值为10，b赋值为6。在接下去三个TextDisplay中，分别显示这两个数的和、差、积。

参考答案见“第三章阶段练习答案1”。

②把上一题的代码进行修改，使Display显示的内容，不仅显示结果，同时在前面说明是什么结果。比如显示为：“和：16”这样的结果。

参考答案见“第三章阶段练习答案2”。

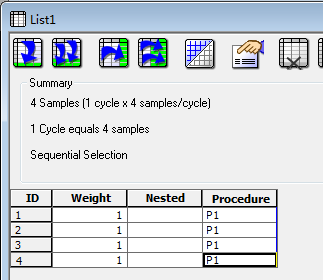
## C:\Program Files\Microsoft Office\MEDIA\OFFICE14\Bullets\BD21295_.gif记录呈现的次数——Public声明变量（全局变量）

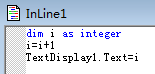
到目前为止，我们学的代码好像没什么（和谐）用！因为仅仅是设置属性的话，我们直接设置就好了，何必要写这些代码呢？

如果你有这样的疑问！首先恭喜你的质疑精神！然后我想说，因为这才刚开始，我们还有很多要学。

现在为了让你切身体会到代码的作用，我们先来做这个实验。

我们接下来准备写一段代码来记录刺激一共呈现了几次，于是我们这么做：

在Inline1里写入下面的代码：

dim i as integer

i=i+1

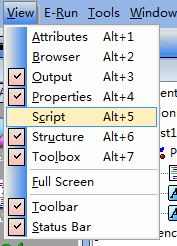
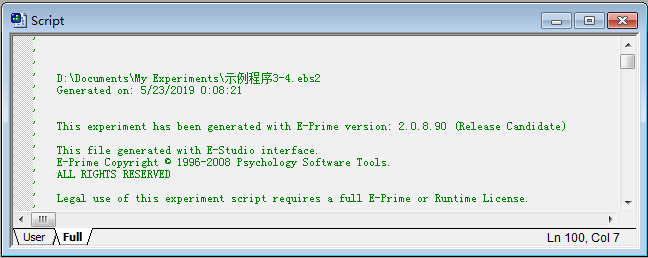
TextDisplay1.Text=i

现在我们运行。TextDisplay会呈现4次。按我们的想法，呈现的内容应该分别是1、2、3、4。

然而！运行后我们会发现！都是呈现1。这是怎么回事呢？这是因为用Dim来定义变量，定义的变量是临时的。为了节约内存，用完就清除了。

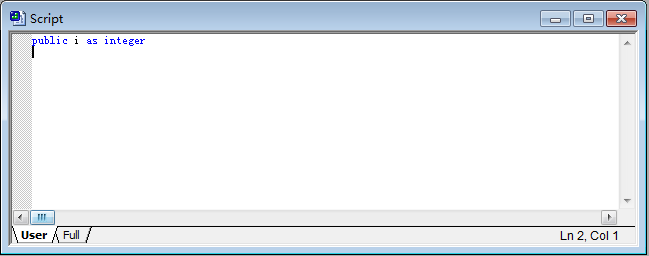
那么这里我们该怎么做呢？我们可以在User自定义脚本中用Public关键字声明变量。这样就能达到我们想要的效果。

首先在菜单中，点击Tool，再点击Script，或者直接按Alt+5来打开Script窗口。如下左图所示：

点击后，弹出右图的对话框。

点左下角，进入User页面，输入：Public i as integer



然后把Inline1中的Dim i as integer删掉。

现在运行，就是我们想要的效果了。

请对比**示例程序3-4**和**示例程序3-5**来查看，您就能明白其中的不同了。记得查看示例程序3-5的User页面的代码。

## C:\Program Files\Microsoft Office\MEDIA\CAGCAT10\j0298653.wmf阶段练习

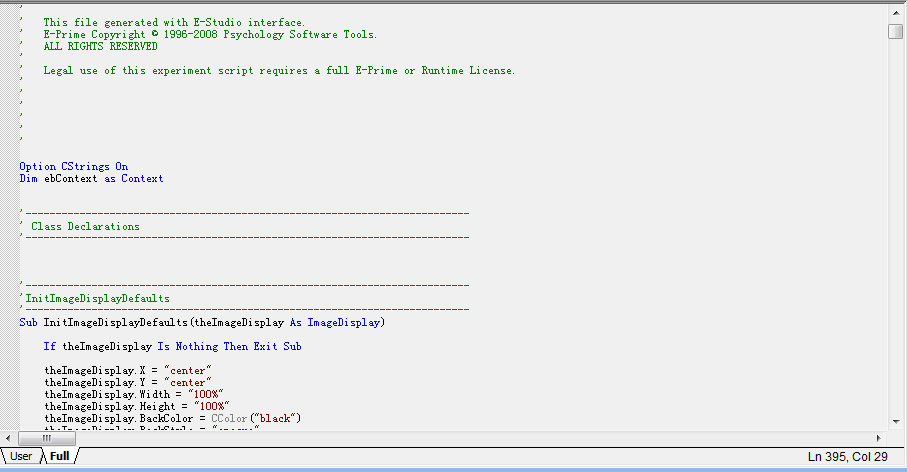
让一个TextDisplay呈现五次。呈现的内容为“呈现次数的两倍是：”这些内容，并紧跟在后面显示【呈现次数乘以2的数值】。其Duration属性也总是为【呈现次数乘以2000】。

## C:\Program Files\Microsoft Office\MEDIA\OFFICE14\Bullets\BD21295_.gif查看当前的代码

上一节我已经说过，通过在菜单中，点击Tool，再点击Script，或者直接按Alt+5，可以打开Script窗口。而且我们甚至已经在User这个页面中输入了一点点代码。

那么我现在告诉你一个秘密，其实，E-Prime的原理就是，通过你在树状图、属性列表里的各种操作，生成相应的代码，然后在E-Run运行这些代码。

明白了这点！你就知道，**只要你掌握了写自定义代码的方法，你几乎可以在E-Prime的世界里为所欲为！**



例如在这个Full页面中，展示的就是当前实验的全部代码。它在你进行Run或者Generate操作以后就会生成，你可以在这里检查哪里出错了，代码是否正确等等。

如果你学过VB，看懂这一些绝对没有问题。

不管怎么样，我建议您时不时地进来看一下，查一查这些英文单词，尝试理解这背后是怎么操作的。一时看不懂没关系，学得越多越能看得懂，且越有可能在这里得到收获。

## C:\Program Files\Microsoft Office\MEDIA\OFFICE14\Bullets\BD21295_.gif在Inline中读取List中的变量

如果你企图定义一个叫c的变量，你会发现程序就出错了。为什么呢？因为c这个变量已经被E-Prime自己拿去用了。



您很容易在很多函数后面发现c As Context这句话。您完全可以不用管这到底是什么，仅仅是E-Prime的话，基本上不需要知道。除非你已经足够强了，知道VB是怎么一回事。那你就会想到可以在User里写自定义的函数，用于Inline中。

本教程不会把你教到那么强的地步，你只能自己去学VB了。

总之，你只要知道c这个变量是不能定义的。我们再找找看它出现在了哪些地方？

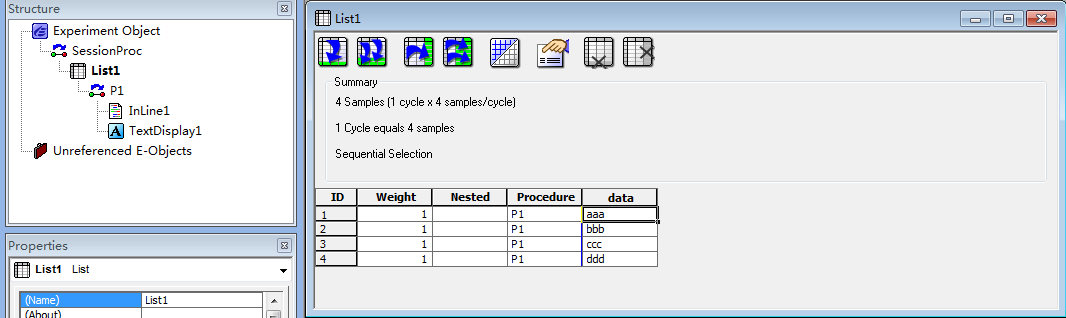
稍微看一下你就发现它随处可见：

既然c As Context，那么Context是什么呢？英文解释是“上下文、背景、环境”什么的。

你就大概这么理解吧，然后试一下，在一个List中，加入这么一个列，这个列里每一行数据都不同，比如说像下面这样。



然后在Inline1中输入下面的内容：

TextDisplay1.Text=c.GetAttrib("data")

然后运行试试看。你会发现，我们之前没有动过TextDisplay1，可是，它却读取了List1中的东西，这就说明我们读取了List1中的列，赋给了TextDisplay1。**（可参考“示例程序3-6”）**

做完这一切时候，请在Script窗口中找找看我们输入的代码在哪？（**提示：可以按右键点击Find来查找**）

再试试，如果删掉Inline1，把TextDisplay1的Text属性设置为”[data]”，运行一次，再试着查找一下这条语句，你会发现什么？

## C:\Program Files\Microsoft Office\MEDIA\CAGCAT10\j0298653.wmf阶段练习

给一个List加一个列，命名为”num”；加三个行，这样一共有四行。这四行的num分别为44,12,56,98。要求在随后的TextDisplay中总是显示num里的数字除以2后的数值。

## C:\Program Files\Microsoft Office\MEDIA\OFFICE14\Bullets\BD21295_.gif条件语句（If）

If这个单词是什么意思，应该不需要解释吧？Select Case你可以理解为“根据情况来做出反应。”

老实说我有点不太愿意写这几节，因为这都是VB的内容。而且深入解释起来又很复杂，所以我打算只写一些最基本的，在E-Prime中比较有可能用得上的。

现在你来翻译一下下面这段话：

if a>5 then

temp="a大于5"

else

temp="a小于5"

end if

里面没有什么难以理解的单词吧？表达式应该也很容易明白。

下面是我的翻译：

如果a大于5，那么就把“a大于5”这个字符串赋给temp，否则把“a小于5赋给”temp，结束if语句。

你可能不像我翻译的那么规（酸）整（臭），但是这段话的作用应该是显而易见的。现在打开**“示例程序3-7”**看一看，然后自己新建一个文件，进行操作，使其完全和“示例程序3-7”一样。

那么下面这两段，也非常好理解：

if a>5 and b>5 then if a>5 or b>5 then

temp="a和b都大于5" temp=”a和b至少一个大于5”

else else

temp="a和b不全都大于5" temp=”a和b都不大于5”

end if end if

一样地，打开**“示例程序3-8”**看一看，然后自己新建一个文件，进行操作，使其完全和“示例程序3-8”一样。尝试理解一下。

查看代码后，你会发现Chr(13)这个东西，这个意思是回车。运行看看，现在明白了什么了吗？如果不明白，试着删掉它，看看不同。

当然，我们也可以在if语句里加入if语句，比如看下面这个：

if a>5 and b>5 then

temp="a和b都大于5"

else

if a>5 then

temp="a大于5，但是b不大于5"

else

temp="a一定不大于5，b不一定。"

end if

end if

相信仔细查看后你也能明白它的意思，简单来说就是判断a和b是否都大于5，如果并没有都大于5，那就看看a是不是大于5。

仔细看，我们看到了两个if then，还有后面是end if的格式。end if的意思是声明一个if语句的结束，想想看，这两个end if 分别是对应哪个if？没错，最底下一行是对应开头的if，而倒数第三行的end if是对应中间的if。

这种if里又有if语句，被称为嵌套语句。因此内层的end if 对应的是内层的if，外层的自然对应的是外层的。

一样地，打开**“示例程序3-9”**看一看，然后自己新建一个文件，进行操作，使其完全和“示例程序3-9”一样。尝试理解一下。

那么下面这句，没有了else，应该更好理解了吧：

if a>5 then

temp="a大于5"

end if

像“a>5”这种表达式被称为条件表达式，除此之外，还有：

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 表达式 | 例子 | 例子的作用 |
| 大于 | > | a>5 | 检查a是否大于5 |
| 小于 | < | a<9 | 检查a是否小于9 |
| 大于等于 | >= | a>=7 | 检查a是否大于等于5 |
| 小于等于 | <= | a<=3 | 检查a是否小于等于5 |
| 等于 | = | str=”ss”  str=”” | 检查str是否是字符串”ss”，数字当然也可以检查。  第二个例子检查str是否是空的，如果是空的，则运行then和end if之间的语句。 |
| 不等于 | <> | a<>5  str<>”” | 检查a是不是不等于5，如果不等于5，才运行then和end if之间的语句。  第二个例子检查str是否是空的，如果不是空的，则运行then和end if之间的语句。 |

## C:\Program Files\Microsoft Office\MEDIA\CAGCAT10\j0298653.wmf阶段练习

①我们刚才上上一节的各种示例实验中，已经尝试根据比较读取的数据来显示不同的内容了。现在，如果我们要做到准确地显示a和b哪个大于5哪个没有，应该怎么做呢？如果我们删除List1中的num这一列，应该做什么才能实现相同的功能？

②自己编写一些实例，尝试使用其他的一些条件表达式，熟悉它们的作用。

③做这么一个实验：在屏幕上显示数字，当显示数字在10到50之间的时候，在显示的数字底下加上一行：“这是个重要的数字。”显示的次数由你来定，但是呈现的在10到50之间的数字，和不是在这之间的数字的数量要相同。

④编写这么一个实验：在屏幕上同时呈现两个数字，一个是事先准备好的一列数字，从当中读取其中一个数字x来呈现；另一个数字是当前刺激呈现的次数n，且当n大于10的时候，让n又变成0。当x和n同为奇数或者同为偶数，按f键；否则，按j键。

⑤随意打开一个实验程序，查看它生成的代码，分析出现的if语句的作用。

⑥上网查询VB的Select Case语句，或者到官方自带教程的“E-Basic.chm”这个文件中查询，努力理解怎么使用，并且亲自尝试使用。

## C:\Program Files\Microsoft Office\MEDIA\OFFICE14\Bullets\BD21295_.gif循环语句（仅是说明）

循环语句在任何编程语言里都是极其重要的语句。不过在E-Prime里因为有List的存在，我们很少有机会用到。

如果你是有追求，有挑战困难的勇气的学生，或者是本身有编程基础的学生，那么你可以去学一学VB的循环语句。否则的话你大可不必学习循环语句，因为在E-Prime里用到的机会不多，更因为我太懒，不愿意详说。反正网络上都有介绍。接下来您看一下“示例程序3-10”感受一下其作用就好了。

**只要做到，以后当有类似需求的时候，想起来能够用循环语句解决就OK了。**有兴趣的话，再查一查VB的For、While、Do这些循环语句。

## C:\Program Files\Microsoft Office\MEDIA\OFFICE14\Bullets\BD21295_.gif查阅帮助以及面向对象编程

这是本章的重中之重！会编程的人，这整章中几乎只要看本节就可以了。毕竟，E-Basic的代码非常之多，我不可能全部介绍。这就需要你会知道查看官方自带的帮助文件，就像你有需要的时候就查阅字典一样。

此外，本章还将简单的介绍面向对象编程。是非常简单的介绍，仅仅足够你理解E-Prime而已。请在你的大脑里用高级编程语言执行下面的语句：

**If(state==1) //0=没学过面向对象编程;1=学过。**

**{goto CheckPoint;}//在后面，很容易找到，学过的话直接跳到那里，下面这段不用看。**

你不需要管什么叫“面向对象”，接下去的一小段你知道一下，这是面向对象编程思想的一部分就行了，至于这具体是什么，有兴趣深入学编程的同学才去了解。

现在我们来这样这样定义一个概念：人类，这是一个大类，我们称之为父类；然后男人和女人，都属于人类这个大类中的小类，我们把它们称之为人类的子类。很好理解，简单来说，就是父子父子，父类是大类，子类是小类。所以我们定义了父类和子类。

我们可以看到，子类是从父类继承而来的。男人和女人都属于人类，它们拥有人类共同的特征：如正常发育情况下有手和脚、能学习、走路、睡眠等，我们可以称之为属性。这就叫继承，子类（男人和女人）从父类（人类）那里继承了它的一些特征。

而人类又是动物这个类的子类，人类从动物这个父类那里也继承了一些特征：如会进食、会睡眠、会行走等等。

现在我们弄清楚了什么叫父类和子类，什么是继承。

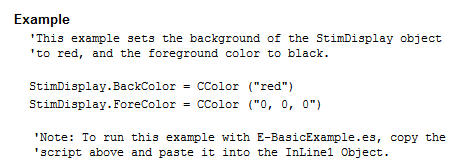
接下来我们再定义一个概念，叫做“实例”。实例就是某一类里的一个特定的对象。比如说“张三”是男人这个类对象的一个实例；“李小红”是女人这个类对象的一个实例。

现在，我们应该已经弄清楚了类、继承、实例这三个概念。那么我们为什么要学习这个呢？因为在E-Prime里也包含有这样的关系。

例如ImageDisplay、TextDisplay和SlideDisplay都是一种叫“StimDisplay”的子类。它们都从这个StimDisplay里继承了X,Y,Duration等这些属性。Stim就是英文里“刺激”这一单词的简写。所以我们看到，ImageDisplay、TextDisplay和SlideDisplay这三个类都包含相同的一些属性，正如男人、女人、人妖虽然各有不同，但是也包含一些相同的属性。

同样，StimDisplay也是”E-Object”这个类的子类。

知道了这一点之后，当你在官方教程“E-Basic.chm”中看到下面这样的示例代码：



你要知道，这里的StimDisplay就是说，凡是属于这个类都可以。

也就是，我们改成：

TextDisply.BackColor=CColor(“red”) 是可以的

ImageDisply.BackColor=CColor(“red”) 也是可以的

因为它们都属于StimDisplay，都有BackColor这个属性。当然，**要注意的是，你只能修改某一个实例的BackColor。**也就是说，比如我们有三个TextDisplay，分别叫T1,T2,T3。

我们可以修改它们的BackColor

T1.BackColor=CColor(“red”)

T2.BackColor=CColor(“red”)

T3.BackColor=CColor(“red”)

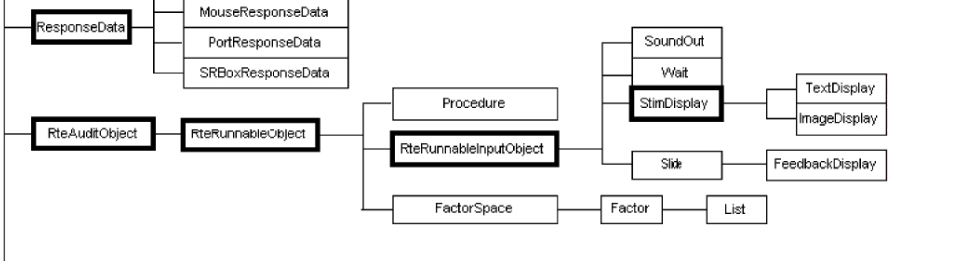
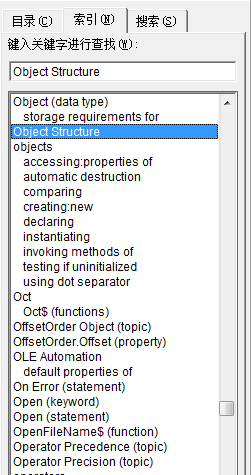
但是：

~~TextDisply.BackColor=CColor(“red”)~~

这是不行的。这就好比，我们可以说“张三有300块钱”，但我们不能说“男人有300块钱一样”；这里，我们只能说“把T1这个TextDisplay的BackColor设定为red”，而不能说“把TextDisplay的BackColor设定为red”。即使你要让一般的TextDisplay默认BackColor为red，也不是这样设定的。

**CheckPoint：-----------------------------------------------上面所说的CheckPoint在这里！！**

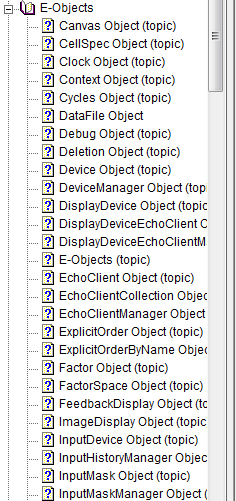
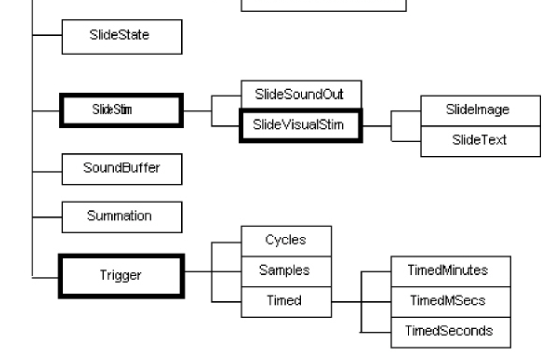
理解了这一点以后，打开“E-Basic.chm”，在索引或者目录中找到Object Tree(Object Structure)：

现在，你大概知道里面的内容是什么意思了吧？我们可以看到这是一个分支图，这些矩形后面的分支代表了它的子类。看，我们常用的TextDisplay、ImageDisplay都是一个叫StimDisplay的子类呢！而StimDisplay又是一个叫RteRunnableInputObject的子类。FeedbackDisplay居然是Slide的子类呢！想想看是不是那样？FeedbackDisplay是不是和Slide很像？而Slide是不是更高级？想想看，如果我们加入自定义代码，是不是能让Slide达到和FeedbackDisplay一样的效果？

另外，建议你如果看不懂这些英文单词，请查一查。这有助于理解。

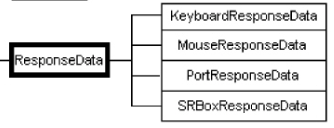
通过下面左图的树状图，或者右图的分支图看到，还有很多E-Object是我们之前不曾用过的。在之后的学习中会接触一些，但无论如何，您要有自主意识。要知道，以后遇到困难，或者想要更深入地了解某个E-Object，您要知道打开这个帮助文件自查。知道了这点，本书的意义就不仅限于学习了E-Prime了：

## C:\Program Files\Microsoft Office\MEDIA\OFFICE14\Bullets\BD21295_.gif读取按键记录（ResponseData）

我们虽然已经能够Advanced按钮设定来显示被试的按键，不过这终究不够灵活。

通过查看帮助文件，我们会发现一个叫ResponseData的类，这个类下面还有好几个子类，如KeyboardResponseData,MouseResponseData。



这些类，都是用来记录相应的设备的反应数据的。如KeyboardResponseData是用户对键盘的按键的各方面数据记录。

那接下来我们首先先看一下在帮助文档里的东西。

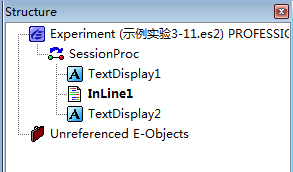
ResponseData有这些属性：

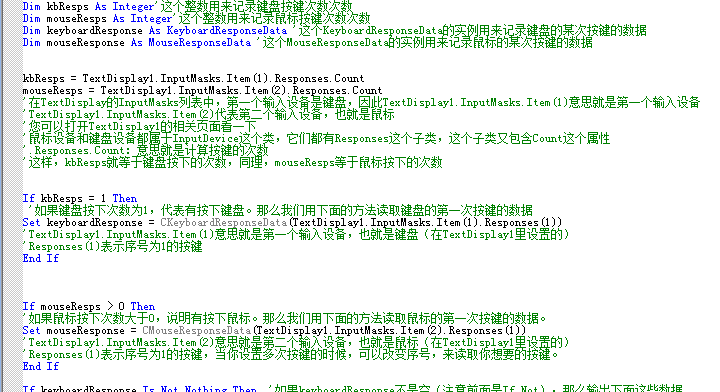


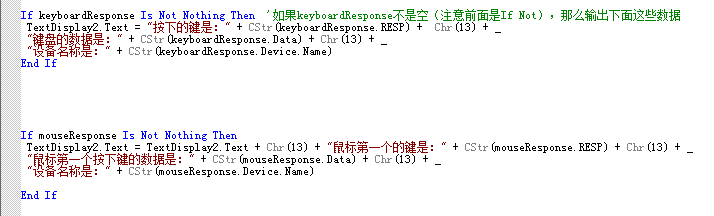
首先明白一点：一个ResponseData，对应一次反应。其中RESP就是按下的按键，如”A”键，”Shift”键等等，总之是你在刺激控件中按下的按键。

听上去很抽象，那么请打开“**示例程序3-11**”，请查看里面的Inline1，看看代码和注释。

这些代码是根据官方自带教程里的代码改编而成的。







其中，你会注意到后面的代码里，有些代码在行的最后有下划线。

这个下划线其实前面和后面各有一个空格，其作用是当一行代码太长的时候，可以用这个方式换行。注意空格是不可少的。

## C:\Program Files\Microsoft Office\MEDIA\CAGCAT10\j0298653.wmf阶段练习

①设计一个程序，一个StimDisplay接收一次按键，这次按键可以是鼠标也可以是键盘。按键后，在屏幕上显示按下的是什么键，是鼠标还是键盘。并且，如果按下的是空格键，显示：“你刚才按了空格键。”

提示：用If [ResponseData] .RESP = “{SPACE}” then 来检查。

②设计一个程序，一个StimDisplay接收两次键盘按键。检查两次按键是否是相同的，如果是相同的，那么显示“你刚才按了两个相同的按键”；否则，分别把两个按键在屏幕上显示出来。

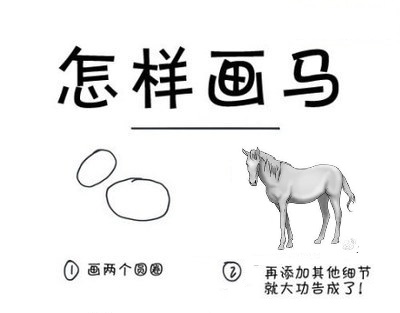
③设计一个程序，一个StimDisplay只接收松开的按键。此外，如果是松开了空格键，那么显示“你刚才松开了空格键”，否则显示“你没有松开空格键”，然后重新显示该StimDisplay，直到被试按下空格键为止。

## C:\Program Files\Microsoft Office\MEDIA\OFFICE14\Bullets\BD21295_.gif控制Slide及其各页面控件（SlideDisplay Sub-Objects），VB数组

你会发现，要使用代码修改Slide中的控件的时候，有点无从下手。实际上这并不难。“示例程序3-12”将为您展示如何修改Slide中的子控件，还有如何控制Slide当前显示的页面。**请首先运行体验一下这个程序。**

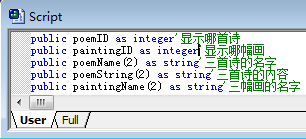
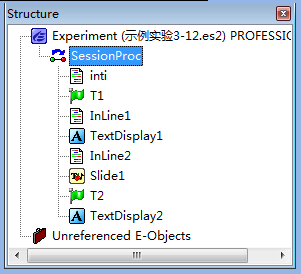
这个程序，我们将在Script的User页面中，使用VB数组来设定内容，我们要做到以用户的按键来决定显示Slide的哪个页面，以及显示什么内容。

**如果您没学过编程，这个示例程序对您而言可能非常困难，甚至不亚于下图：**

****

**那么您没必要要求自己完全理解，您只要知道，如何修改Slide将要显示的页面，以及其子控件的内容就好了。VB数组在实验中也是极少用到，可以不掌握。**

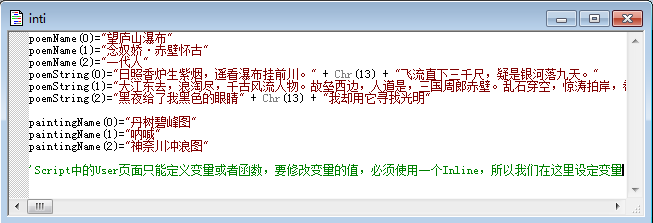
我们简单看一下这个示例程序：



首先这是树状图，以及我在Script的User页面里定义的变量。

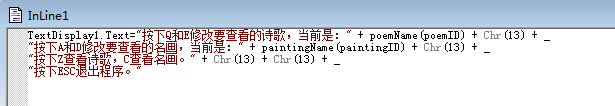
你会看到有public poemName(2) as string这个东西。

这就是VB数组的定义，它的作用是什么呢？请看下图：



我在inti这个Inline里定义了poemName(0)还有poemName(1)还有序号为2的这些poemName。没错，我想你猜到了，public poemName(2) as string这句话，按序号定义了三个变量，它们分别是poemName(0)、poemName(1)、poemName(2)。而我在inti中分别给这三个变量赋值。

现在我们看Inline1：



这里我们修改了TextDisplay1的Text属性，其中你会看到这个：

poemName(poemID)

我们前面定义过，poemID是integer，也就是整数，这样我们可以利用poemID来调用我们想要的poemName的其中某个元素。比如说：

当poemID的值为0时：

poemName(poemID)= poemName(0)= "望庐山瀑布"

同理，poemID的值为0时：

poemName(poemID)= poemName(1)= "念奴娇·赤壁怀古"

这样，我们可以在程序中按意愿修改poemID，从而调用我们想调用的那个的poemName元素。这就相当于，按学号找到对应的同学。

**下面这段小字给学过C语言的同学看，其他人可跳过**：

要注意VB数组和C语言数组是不一样的，如果是C语言，我用下列方法定义一个名叫poemName 的字符串数组：

String poemName[2];

那么一共只有两个元素，下标是0和1，而这里，我用

public poemName(2) as string

定义出的是三个元素，下标分别是0、1、2。

注意不要弄混两者。

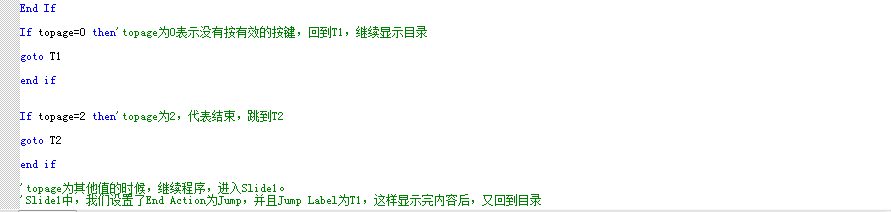
**如果您看完上面的内容仍然无法理解VB数组，那么就算了！算了！别纠缠于此，实验中很少用到。**

**接下来你也只需要尽可能理解该程序，理解不了不用强求，最后我会告诉你知道到什么程度就够了。**

接下来看Inline2：







上面就是Inline2中的全文，那么接下来讲一下思路。我们首先要用一个标记，来代表被试是否退出，这个标记就是topage这个变量。该变量为0时，代表不显示Slide页面，仍然停留在当前的选择菜单中；为1时，按照其要求显示Slide中的页面；为2时，代表退出程序。

然后，我们接收被试的按键，通过按键判断行为。如果按下ESCAPE键，那么就代表要退出，我们将topage设置为2。这样，在最后的代码中，if语句发现topage为2后，会“goto”T2，这样就跳过了Slide1，显示TextDisplay2，然后结束程序。

如果被试按下Z和C，那么代表他想看诗歌或者想看名画，那么我们通过设定Slide1的ActiveState并修改其中的内容，来展示他要看的东西。你可能暂时不理解里面的语句，先不必管它。

如果按下其他键，则停留在这个选择页面中。其中，如果按了Q，则让poemID减1，为了防止poemID小于0，所以每当poemID小于0时，我们将其赋值为2；同理，按了E则让poemID加1，为了防止poemID大于2（即大于poemName或poemString的元素下标最大值），每当poemID大于2时，我们将其赋值为0，从而形成一个循环。A键和D键也类似，只不过更改了PaintingID的值。

我们来看按下C的时候，我们是如何操作的：

If CStr(keyboardResponse.RESP) = "C" or CStr(keyboardResponse.RESP) = "c" then

Slide1.ActiveState = "State2"

Dim name2 as SlideText

Dim pic1 as SlideImage

Set name2 = CSlideText(Slide1.States(Slide1.ActiveState).Objects("Text1"))

Set pic1 = CSlideImage(Slide1.States(Slide1.ActiveState).Objects("Image1"))

name2.Text=paintingName(paintingID)

pic1.Filename="img" + CStr(paintingID+1) + ".jpg"

pic1.load

topage=1

End if

首先看第一句：Slide1.ActiveState = "State2"

这就设定了Slide1即将显示State2这个页面。

Dim name2 as SlideText

Dim pic1 as SlideImage

这两句我们设定了两个变量，name2和pic1，他们分别是SlideText这个类和SlideImage这个类的实例。

Set name2 = CSlideText(Slide1.States(Slide1.ActiveState).Objects("Text1"))

Set pic1 = CSlideImage(Slide1.States(Slide1.ActiveState).Objects("Image1"))

这里，我们设置了name2和pic1分别为Slide1中的State2这个页面的Text1这个Object和Image1这个Object。这样，name2就代表了Slide1中的State2这个页面的Text1；同理，pic1代表Slide1中的State2这个页面的Image1。我们对name2和pic1的操作，将相当于对Slide1中的State2页面里的相应的Object进行操作。

我们知道，刚才我们设置了Slide1.ActiveState为”State2”，所以这里这两句相当于下面的样子：

Set name2 = CSlideText(Slide1.States("State2").Objects("Text1"))

Set pic1 = CSlideImage(Slide1.States("State2").Objects("Image1"))

它们在含义上有所不同前者意思相当于让name2和pic1分别代表Slide1当前即将显示的页面里面的物体Text1和物体Image1；而后者，含义是让name2和pic1分别代表Slide1的State2页面里面的物体Text1和物体Image1。

之后我们对其进行了修改：

name2.Text=paintingName(paintingID)

pic1.Filename="img" + CStr(paintingID+1) + ".jpg"

pic1.load

这两句就会影响随后Slide1中的State2页面的内容。注意pic1由于是图片，修改了Filename后，一定要调用load方法（包括音频也需要调用），也就是像第三句那样pic1.load，来读取图片，否则显示的只会是默认的图片。

最后我们设定topage为1，代表等一下将显示Slide1。

那么我就将这个示例程序讲解到这里，不知您理解了没有。

**即使没有理解，请您不用再回头看，请继续往下阅读。**

这个程序只是为了更好的展示，掌握了对Slide1进行操作有什么好处，不然的话，我们本来就能用其他方法对Slide1的内容进行设定，何苦费这么大劲儿呢？这个程序就告诉你为什么我们要努力写这么多代码。

总而言之，您大可不必硬要理解“示例程序3-12”，你只需要理解接下来这两段代码，并至少完成阶段练习中的①②就可以了。

看下面：

Slide1.ActiveState = "State1"

Dim SText as SlideText

Set SText = CSlideText(Slide1.States("State1").Objects("Text1"))

SText.Text=”这是页面1”

这段话的意思是：

将Slide1将要显示的页面设置为State1；

定义一个名叫SText的SlideText；

设置SText为Slide1中State1页面里的名叫Text1的Object；

设置SText的Text属性为“这是页面1”。

这就是第一段，理解完这一段后，再看下面这一段：

Slide1.ActiveState = "State2"

Dim SImage as SlideImage

Set SImage = CSlideImage(Slide1.States("State2").Objects("Image1"))

SImage.Filename=”pic2.jpg”

SImage.load

这段话的意思是：

将Slide1将要显示的页面设置为State2；

定义一个名叫SImage的SlideImage；

设置SImage为Slide1中State2页面里的名叫Image 1的Object；

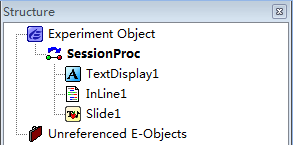
设置SImage的Filename属性为“pic2.jpg”；

读取SImage的Filename中所设定的图片。

**只要理解了上面的代码，能够独立完成阶段练习中的①②，你就已经算是完成了本章的学习了。至于VB数组的运用，不懂也无妨。**

## C:\Program Files\Microsoft Office\MEDIA\CAGCAT10\j0298653.wmf阶段练习

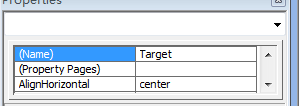
①拖动控件到树状图里，和下图一致：



设定TextDisplay1的Duration为无限，呈现内容由你自己决定。设定TextDisplay1接收键盘输入，允许的按键为{ANY}。

给Slide1添加三个页面，分别命名为Stage1,Stage2,Stage3。

此时Slide1一共四个页面：Default, Stage1,Stage2,Stage3，让它们呈现你喜欢的内容，不同页面呈现内容必须不一致。此外，每个页面至少有一个SlideText，并且名字叫”Target”（也就是Name属性设置为”Target”），如图：



把下面的代码复制到InLine1里，运行，思考其作用：

Dim kbResps As Integer

Dim keyboardResponse As KeyboardResponseData

kbResps = TextDisplay1.InputMasks.Item(1).Responses.Count

If kbResps = 1 Then

Set keyboardResponse = CKeyboardResponseData(TextDisplay1.InputMasks.Item(1).Responses(1))

End If

Dim temptext as SlideText

Dim thekey as string

If keyboardResponse Is Not Nothing Then

If CStr(keyboardResponse.RESP) = "Q" or CStr(keyboardResponse.RESP) = "q" then

Slide1.ActiveState = "State1"

thekey="Q"

End if

'-------------------------------------------------

If CStr(keyboardResponse.RESP) = "W" or CStr(keyboardResponse.RESP) = "w" then

Slide1.ActiveState = "State2"

thekey="W"

End if

'-------------------------------------------------

If CStr(keyboardResponse.RESP) = "E" or CStr(keyboardResponse.RESP) = "e" then

Slide1.ActiveState = "State3"

thekey="E"

End if

'-------------------------------------------------

End If

Set temptext = CSlideText(Slide1.States(Slide1.ActiveState).Objects("Target"))

if thekey<>"" then

temptext.Text=temptext.Text + Chr(13) + "您刚才按下了" + thekey + "键"

end if

②这次不看提示，新建一个文件，设法独立设计①中的内容，并且按自己的喜好进行改造，变式。如果做不到的话，回到①获取你想知道的内容，然后重新进行②。

③设计一个程序，里面必须包含一个Slide，并且这个Slide有多个Stage，每个里面有都有排版方式不同图文信息。根据之前被试所按的按键来决定所呈现的内容（提前准备好图文），及其排版方式。被试先通过按QWER四个字母选择要呈现什么内容，然后按1234来选择排版方式。

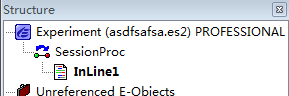
④设计一个程序或者打开一个任意程序。在这个程序中使用到Slide控件，让这个控件起到和FeedbackDisplay几乎相同的作用。（也就是用于正负反馈）

⑤设计一个程序，里面包含20个试次。每个试次前，被试首先通过按键选择两种刺激呈现方式，一种是“声音”，一种是“图片”。选择完毕后，呈现一个词语，比如“猫”，根据被试先前的选择，如果选择“声音”，则呈现词语时**同时**给予声音刺激，让被试判断声音是否是“猫”；如果选择“图片”，则呈现词语时**同时**给予图片刺激，让被试判断图片是否是“猫”。如果词语和刺激一致则按f，反之按j。词语与刺激一致和不一致的情况各呈现十次。每个词语最多只能出现两次。

## C:\Program Files\Microsoft Office\MEDIA\OFFICE14\Bullets\BD21295_.gifCanvas Object \*

这一节不太重要，但是很有趣，算是给您放松用的。Canvas能实现一些很特殊的功能，Canvas意思是画布，Canvas Object允许你在屏幕上画图，如画矩形、圆形、或者显示文字。。假设你想用E-Prime制作游戏的话，最好知道怎么用Canvas。

为了看到它的作用，首先您得像我这么做，先像我如此安排树状图里的内容：



没错，就这么简单，一个Inline就够了。然后输入下面的代码：

Dim cnvs As Canvas

Set cnvs = Display.Canvas

Dim x, y, w, h As Integer

h = 100

w = 200

x = 100

y = 100

cnvs.Rectangle x, y, w, h

Sleep 3000

**运行，没错，就这样运行。**这样，你就会看到我们在屏幕x为100，y为100的地方画了一个矩形（**注意一下，显示器x轴正方向向右，y轴正方向向下**），它高为100，宽为200。没错，就是我们上面设置的数值。

现在我解释一下上面的代码，它们的意思是：

定义一个名叫cnvs的Canvas的实例；

设置它为Display（也就是我们的显示器，默认名字Display，除非你有多个显示器，否则一般不改动）这个显示器的Canvas；（记得Canvas意为画布）

设置四个整数x,y,w,h，下面四行给它们相应赋值；

画一个矩形，位置在x,y，宽为w，高为h；（这四个字母是代表我们刚才刚定义的四个整数的变量名，. Rectangle表示画矩形，你可以看看后面的代码，像.Pie，就表示画扇形）

等待3000ms。（Sleep的作用是等待，如果没有这句，我们就没法看到结果）

不知您理解了没有，如果没有理解，试着修改x,y,w,h的值，这样就能帮助您理解。

上面的代码也可以替换成如下的这段：

Dim cnvs As Canvas

Set cnvs = Display.Canvas

cnvs.Rectangle 100, 100, 200, 100

Sleep 3000

效果是完全一样的。

注意一下，Rectangle后面的四个参数（理解成数字就好）一定是按格式的，它们分别代表x轴位置，y轴位置，宽，高，你只能输入，不能随意调换位置。

当然，我也可以改变其边框颜色，或者填充颜色。例如，我们这样修改：

Dim cnvs As Canvas

Set cnvs = Display.Canvas

Dim x, y, w, h As Integer

h = 100

w = 200

x = Display.XRes/2-w/2

y = Display.YRes/2-h/2

cnvs.FillColor = CColor("yellow")

cnvs.PenColor = CColor("green")

cnvs.PenWidth = "10"

cnvs.Rectangle x, y, w, h

Sleep 3000

这样，就会在屏幕正中央呈现一个矩形，填充颜色是黄色，边框颜色是黄色。

语句中，Display.XRes和Display.YRes的作用是获取显示屏的宽度和高度。一个矩形的x轴位置在显示器的宽度的二分之一，减去w（其宽度）的二分之一；y轴位置在显示器的高度的二分之一，减去h（其高度）的二分之一。那么这个矩形就能正好呈现在屏幕正中央。

而FillColor决定了矩形的填充色，PenColor决定了矩形的边框色，PenWidth决定了矩形边框的宽度。您不妨试着修改FillColor为”red”，PenColor为”black”，PenWidth为20，看看有什么不同，这样能方便您理解。颜色还可以用RGB值来表示，如CColor("0,255,0")代表绿色。里面的三个数值就是对应RGB值。

当然，我们也可以画圆形甚至扇形，比如你现在再试试下面这段：

Dim cnvs As Canvas

Set cnvs = Display.Canvas

Dim r, startAngle, endAngle As Integer

r = 50

startAngle = 25

endAngle = 60

cnvs.PenWidth = "4"

cnvs.Pie Display.XRes/2, Display.YRes/2, r, startAngle, endAngle

Sleep 3000

其中的startAngle决定了扇形起始角度，endAngle决定了终止角度，r代表的是半径。同样，Pie后面的五个参数也是固定位置的，分别代表x轴位置，y轴位置，半径，起始角度，终止角度。

如果您打算画圆形，startAngle设为0，endAngle设为360就好了。当然，这会存在一个问题，有一个黑色的线段在角度为0的位置。

其实，画圆形的话E-Prime有其专门的方法，如下面是其例子：

Dim cnvs As Canvas

Set cnvs = Display.Canvas

Dim rad As Integer

rad = 150

cnvs.PenWidth = 5

cnvs.PenColor = CColor("0,255,0")

cnvs.Circle Display.XRes/2, Display.YRes/2, rad

Sleep 3000

还可以显示字，如下面的例子：

Dim cnvs As Canvas

Set cnvs = Display.Canvas

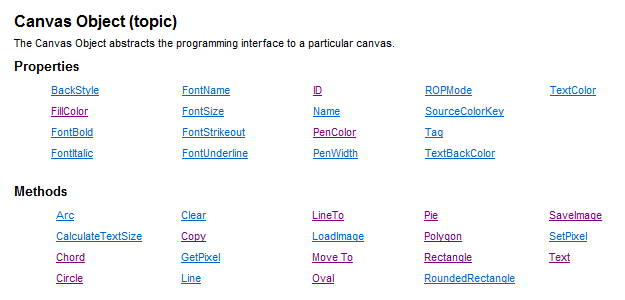
cnvs.TextBackColor = CColor ("red")

cnvs.TextColor = CColor ("blue")

cnvs.Text Display.XRes/2, Display.YRes/2, "Get Ready!"

Sleep 3000

除此之外，还有很多很多的功能，比如画线、通过MoveTo和LineTo勾勒特殊图形、通过Polygon和数组来画多边形、保存或显示图片等等等等。由于数量多，学习难度等原因，这里不一一做介绍。有兴趣的话请大家自己打开帮助文档查看，每个条目里都有示例代码。



假设能学好这个，那几乎可以替代ImageDisplay和TextDisplay了。

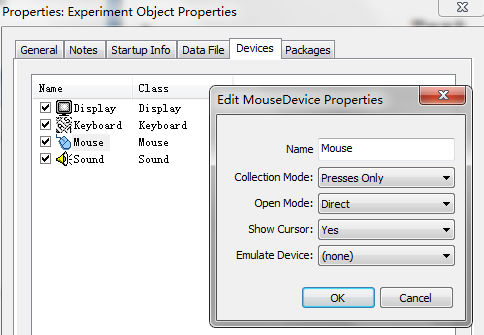
如果您试图把这一节学到的东西用在ImageDisplay和TextDisplay上，就要设置ImageDisplay和TextDisplay的Duration为较低的数值，按键中的Limit Time则较高，Clear After一定不要勾选（设置为No）。因为我在“**Clear After，Echo这一切暗示了什么？**”这一节说过，各个Display实际上是一块一块的东西，它们会相互覆盖。Clear After决定了它们在呈现后是否清除前面的内容。

可以查看“示例程序3-13”，然后试着修改上面我说过的东西，试着自己去尝试发现。由于Canvas里的内容至少在福师大本科阶段乃至研究生阶段的实验中本身极少用到，（实际上从来就不会用到）我也不打算详细说明。不过可以说，Canvas的功能是非常强大的，如果能学会，是很不错的一件事。

## C:\Program Files\Microsoft Office\MEDIA\OFFICE14\Bullets\BD21295_.gif制作按钮？读取MouseResponseData中被试点击的位置

这一节我们准备制作按钮，也就是让用户能用鼠标来点击的按钮。比如说做选择题，我们让被试操纵鼠标来点击。

首先，我们要设置鼠标为显示的，我们可以在Device中设置：



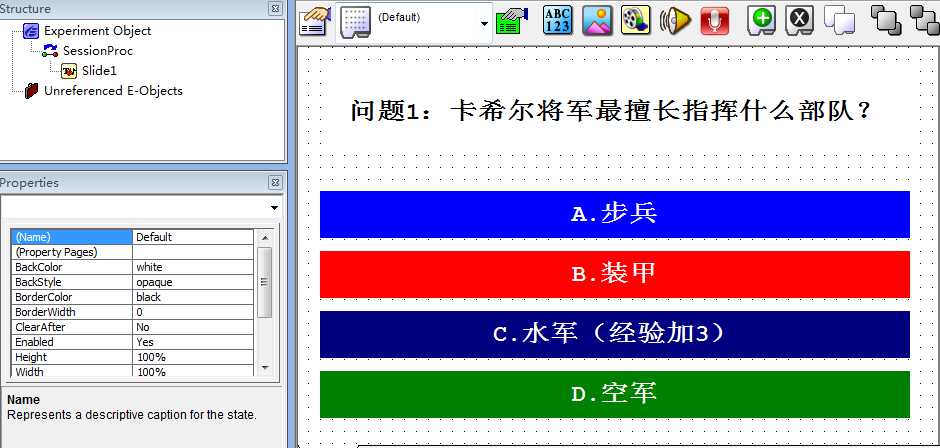
当然啦，我们也可以在实验过程中通过代码进行设置。如果您希望在实验过程中展示或隐藏鼠标，就必须用代码，方法如下：

Mouse.ShowCursor True

加入这一句到你想要显示鼠标的地方就可以了。相应的，隐藏自然就是把True改为False。

这一步很容易，接下来，问题的关键在于我们要接收被试点击鼠标的地方，判断是否点击到了物体。这一点，在E-Prime中提供了方法。

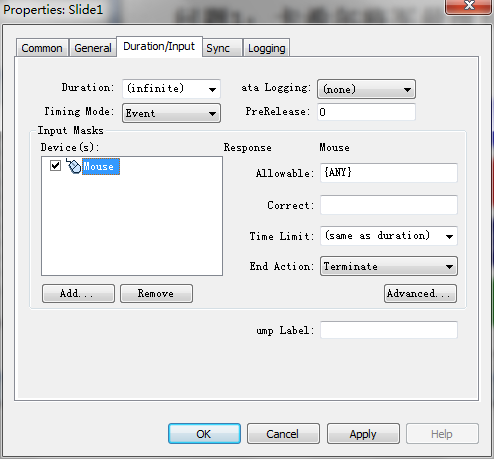
我们首先加入一个Slide1控件，在里面加入我们想要问的问题，以及相应的选项。比如说下面是我的“示例程序”里的无聊问题：



**正确答案是C。**

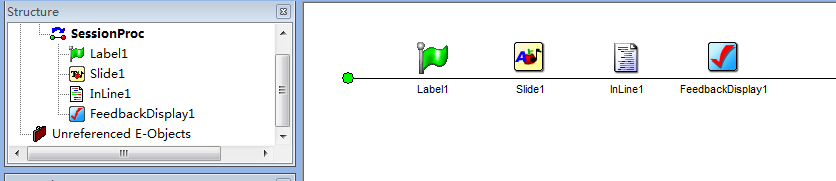
那么我们要达到的效果就是，当被试用鼠标点击的时候，点到ABC三个选项框就提示错误，点到C提示正确，点到其他地方不做任何反应。

首先，我们要在Slide1中设置接收鼠标按键。



如果你想只允许被试按左键，就设定Allowable为1。

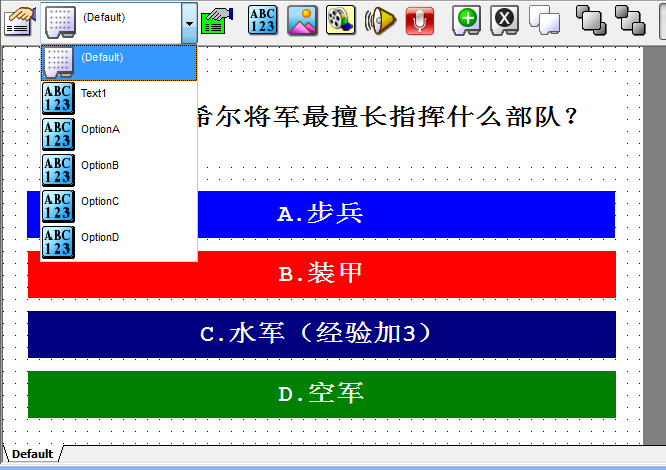
接下来先安排一下要用到的控件：



然后按步骤完成：

1.把FeedbackDisplay1的Input Object Name改为Slide1。

2为了方便代码的书写，我修改了Slide1中各控件的Name属性：



Name属性为“Text1”的SlideText对应的是显示问题的那个SlideText。自然，“OptionA”对应A选项，“OptionB”对应B选项，依次类推。

3.在Inline1中输入下列代码：

Mouse.ShowCursor True

Dim theState as SlideState

Set theState = Slide1.States ("Default")

Dim ptMouse As Point

Dim strHit As String

Mouse.GetCursorPos ptMouse.x, ptMouse.y

strHit = theState.HitTest(ptMouse.x, ptMouse.y)

if strHit = "Default" or strHit = "Text1" then

goto Label1

end if

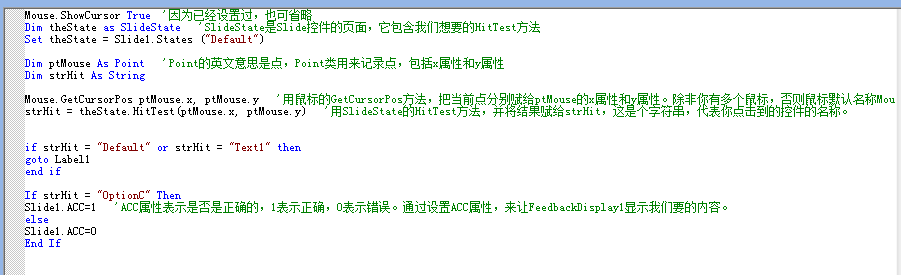
If strHit = "OptionC" Then

Slide1.ACC=1

else

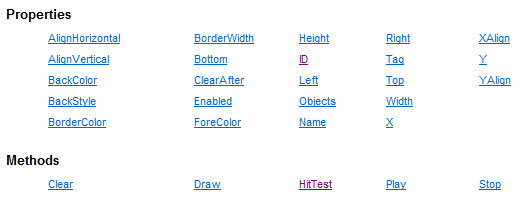
Slide1.ACC=0

End If



Mouse.ShowCurser前面说过，用来设定鼠标是否显示。

接下来我们定义了一个名为theState的SlideState，SlideState就是我常说的Slide的页面，并设定它为Slide1中的Default页面。我们这次主要要用到的是其HitTest方法。**该方法能检测某个点是否包含某个控件，并返回包含的控件的Name属性（如果没学过编程，“返回”一词你可以姑且可以这么理解。就是会会变成一个变量，然后你可以赋值给另一个变量）。**当然它还有其他属性和方法，见下方。



SlideState类中包含的属性和方法

我们又定义了一个Point类的实例，名为ptMouse；还定义了一个string名为strHit。

Point类用来表示一个点，它有x属性和y属性。

接下来这句：

Mouse.GetCursorPos ptMouse.x, ptMouse.y

我们使用了鼠标的GetCursorPos方法，该方法的作用是，把鼠标当前的点分别赋给后面两个参数。在这个句子里，我们赋给了ptMouse的x属性和y属性。除非你有多个鼠标，否则鼠标默认名称Mouse。

当然，你也可以另外定义两个整数，比如：

Dim px as integer, py as integer

Mouse.GetCursorPos px, py

然后你调用GetCursorPos方法的时候，px和py的值就变成你鼠标当前坐标的x值和y值。比如你的鼠标这是在坐标为198,201的地方。那么px就变成198，py就变成201。同理，我们这次写的代码，会让ptMouse的x属性和y属性为鼠标点击时候的坐标的x值和y值。因为我们是在被试按下鼠标按键的瞬间，调用了GetCursorPos方法。

下一句：

strHit = theState.HitTest(ptMouse.x, ptMouse.y)

非常简单，我们调用了theState这个SlideState的HitTest方法，检查了ptMouse.x, ptMouse.y这一屏幕上的坐标是否有控件。如果有，就把控件的名字赋给strHit。

比如说被试点了Text1，strHit就是“Text1”；被试点了选项A，strHit就是“OptionA”（因为我设置其Name属性为OptionA）；用户点了空白处，strHit就是“Default”。

因此接下来的这段语句：

if strHit = "Default" or strHit = "Text1" then

goto Label1

end if

我们判断了被试是不是点击了空白处或者点击了我们显示问题用的SlideText。如果是，就goto Label1，重来；如果不是，就说明用户点击了选项，程序继续。

然后接下来我们判断用户是否点击了正确的选项：

If strHit = "OptionC" Then

Slide1.ACC=1

else

Slide1.ACC=0

End If

ACC属性表示是否是正确的，1表示正确，0表示错误。通过设置ACC属性，来让FeedbackDisplay1显示我们要的内容。各个能记录反应的控件都有这个属性。

所以，如果strHit是“OptionC”的话，说明被试点击了选项C，那么我们设置Slide1的ACC属性为1，表示正确，FeedbackDisplay1就会显示“Correct”；否则说明点击了其它选项，ACC设为0，FeedbackDisplay1显示“Incorrect”。

看到这里你明白了吗？原来FeedbackDisplay是通过检查Input Object Name所表示的Object的ACC属性来判断自己要显示什么内容的。只不过过去我们没有使用代码来亲自设置，而是E-Prime通过我们设置的接收反应按键的控件的Correct属性来自动进行了判断。（本节可参考“**示例实验3-14**”）

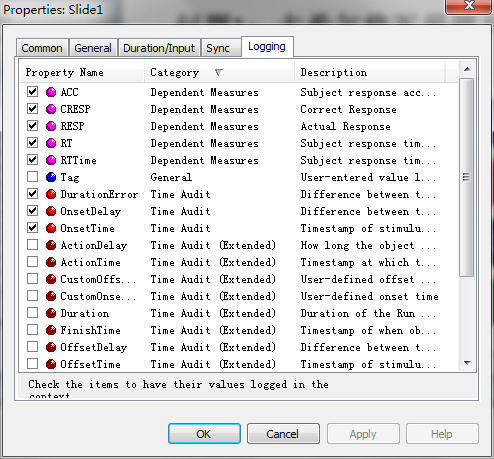
## C:\Program Files\Microsoft Office\MEDIA\CAGCAT10\j0298653.wmf阶段练习

①设计一个实验。实验一共会在屏幕上显示10次刺激，每次刺激，屏幕上都显示一个问题和四个选项。四个选项里只有一个是正确的，被试必须移动鼠标，点击正确的选项。点击之后给予选择的结果被试是否正确的反馈。

②将上一题做如下修改：四个选项可能有多个是正确的，也可能没有正确选项。被试必须选择出他/她认为正确的那些选项，并点击“确定”按钮，之后给予正负反馈。被选中的Object通过把背景色变成红色来提示被选中。

## C:\Program Files\Microsoft Office\MEDIA\OFFICE14\Bullets\BD21295_.gif自定义记录数据

我们已经知道通过修改Logging页面中的内容来设定在一次刺激中要记录哪些数据，并且我们可以按自己意愿勾选想要记录的数据：



但有时候我们可能处于特殊要求，要记录另外一些菜单中没有的数据。

方法就是，通过写代码自己来记录，比如下面的代码：

c.SetAttrib "D-RT", a

这个语句的作用是在数据记录文件中，给**当前试次（也就是前面的c，那是一个叫Context的类的一个实例，总之其实你不需要理解，你就这么写就对了）**新增加一个记录的项目叫D-RT，记录的东西为a。

这样看显然很不直观，接下来我们通过修改“示例实验3-14”来向您展示具体的作用。

很简单，我们在Inline1后面加这么几句：

c.SetAttrib "Who", "卡希尔将军"

c.SetAttrib "Choose", strHit

c.SetAttrib "ThePoint", "(" + Cstr(ptMouse.x) + "," + Cstr(ptMouse.y) + ")"

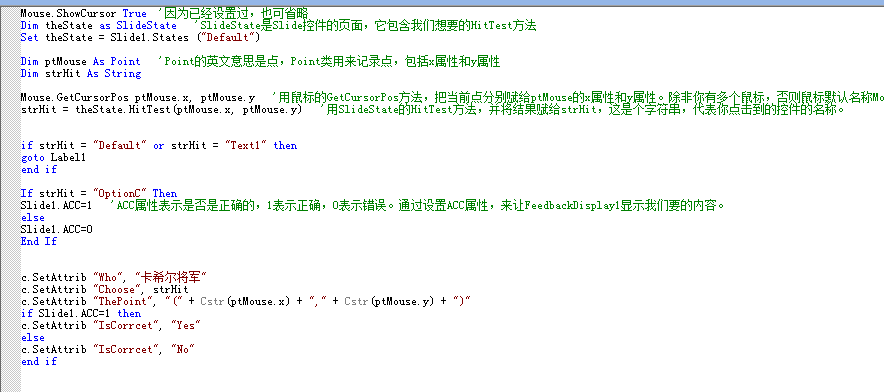
if Slide1.ACC=1 then

c.SetAttrib "IsCorrcet", "Yes"

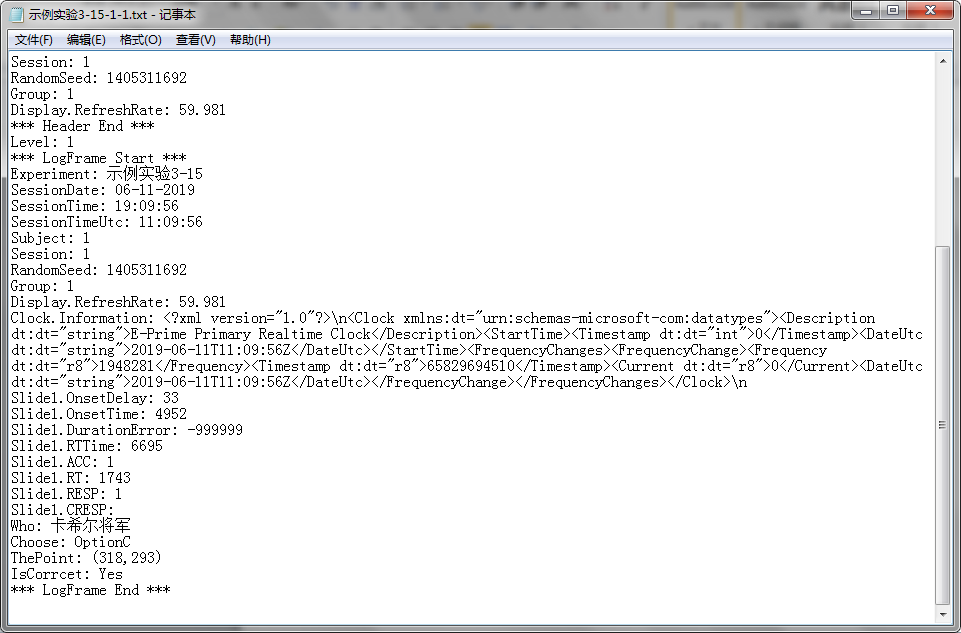
else

c.SetAttrib "IsCorrcet", "No"

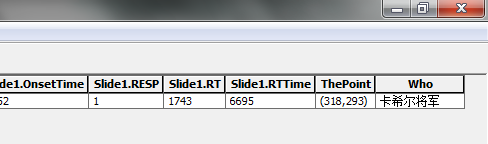
end if

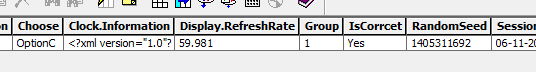


这样，等到做完实验后，我们来看一下结果文件：



结果的txt文件





结果的edat2文件

怎样？我们刚才输进去的那些东西都被记录下来了。

发现了吗？我们给记录添加了数据：

Who: 卡希尔将军

Choose: OptionC

ThePoint: (318,293)

IsCorrcet: Yes

这些都是我们通过语句自己添加的。很容易就能看出这其中的关联。（可参考“**示例实验3-15**”）

**这背后的原理接下来我会介绍，但你不需要掌握，随意看看。如果不明白，反正以后只要知道记录数据的时候在inline中输入：**

**“c.SetAttrib [数据名], [数据内容]”**

**就够了。**

**另外要注意的是，数据名和变量名一样，不能以数字开头，不能包含空格、特殊符号、中文字符、日文字符等等。**

这背后的原理是这样的。我们给当前正在运行的刺激设置了一个新的属性，其名字是我们自己输入的属性名，并赋值为我们输入的内容。

实际上，SetAttrib的真正作用是设置一个属性。与之相对应的还有GetAttrib用来获取属性的值。我们前面用到过c.GetAttrib，我们用这个读取了List中的值。

虽然SetAttrib算是一个用来自定义记录数据的办法，不过不算是一个非常好的办法。真正的好办法是用VB的文件系统，这样你可以为所欲为地记录数据了。这个我们这个教程不做介绍，因为难度较大。反正，用c.SetAttrib的方法已经完全足够您使用了。有兴趣的话，您可以学学VB，那操作E-Prime必然是游刃有余。

## C:\Program Files\Microsoft Office\MEDIA\OFFICE14\Bullets\BD21295_.gif娱乐：枪战游戏\*

现在你可以查看本教程目录里的“E-Prime能有多强大”里面的“枪战游戏”，试着理解了。

这可能很困难，但通过这么久以来的学习，希望您能够看懂一些了。

当然这游戏有不少BUG，要是E-Prime再稳定一点，就是个很好的游戏引擎了。

## C:\Program Files\Microsoft Office\MEDIA\OFFICE14\Bullets\BD21295_.gif本章结语

恭喜你，如果你看到这里，除去带有\*符号的小节外，能熟练掌握基础篇，以及进阶篇+高级篇的2/3的内容，那么你至少在福师大圈子里，可以自豪的说：“我是个E-Prime至尊高手”了。

学完本章，你至少要知道如何用代码修改控件的属性，如何自定义变量和使用If语句判断按键，如何用自定义代码读取List里的列。知道这些，结合前面学的内容，你已经几乎能完成所有实验了。

如果能跟进一步，知道怎么制作按钮，甚至能用E-Prime制作游戏，则你已经上天入地无所不能了。

当然，更重要的，你要学会怎么查看E-Prime自带的帮助文档。**如果只学会个别语句，那么相当于只会看字罢了，只有学会了使用这个帮助文档，你才真正能够自主进步，最终超越这本书所写的内容，直至超越我。**

此外，你还要注意的是，学完本书还不值得你得意。因为“人外有人，天外有天”，就算你已经和卡希尔将军一样能够用E-Prime制作游戏了，且先不说作者仍然还有很多内容没讲，就连将军也深刻感受到还有很多地方尚未精通之处，还有许许多多的大神远远胜于将军。要注意这条道路是没有止境的，除非你对E-Prime非常感兴趣，或者是为了娱乐，我还是建议你，学到够用就好，然后把有限的生命投入到其他更重要的地方去。

当然不管怎样，能学通我这本书，我还是要恭喜你已经能够在福师大圈子里当个E-Prime方面的大神了。

## C:\Users\Administrator\AppData\Local\Microsoft\Windows\Temporary Internet Files\Content.IE5\MW09DSRM\1200px-FelisAurataKeulemans[1].jpg第四章综合练习

C:\Program Files\Microsoft Office\MEDIA\OFFICE14\Bullets\BD14578_.gif第一关：停止信号任务

本关需要用到的材料在“综合练习\停止信号实验”中。

设计如下实验：一共呈现20次刺激，这20次刺激分为“Go信号”和“Stop信号”。“Go信号”刺激会首先呈现一个空心的圆500ms，然后呈现一个向左或向右的黑色箭头。如果箭头向左，被试需要按下f键，反之按下j键头。箭头呈现1000ms，但是允许被试在3000ms内按键；而“Stop信号”与“Go信号”不同的是，同心圆呈现结束后，呈现的黑色箭头会在SSD毫秒后变成红色，要求被试抑制冲动，不进行按键。SSD毫秒默认为200ms，但是如果被试判断失误而按键了，则SSD变成SSD当前数值减50ms；若被试判断成功，则SSD变成SSD当前数值加50ms。且SSD不小于150ms，不大于300ms。

红色箭头的显示时间变成1000 减去SSD(ms)。

如果被试在箭头变成红色之前就按键，且按键正确。则认为被试对这次试次的判断仍然是正确的。

C:\Program Files\Microsoft Office\MEDIA\OFFICE14\Bullets\BD14578_.gif第二关：快速反应！

设计任意的一个有关反应时的实验，一共呈现20次刺激，刺激分为两种，一个是A刺激，一个是B刺激（自行决定呈现什么，以及呈现时间等），且A和B都需要被试判断正误。在呈现一个先前刺激A后，被试按键来判断A刺激是否正确，然后立刻呈现B刺激。要求是：只有当被试按下正确按键，并且当反应时不超过500的情况下，才算判断正确。只有判断正确，刺激B才呈现，且被试要立刻对刺激B进行判断，反应时也不得超过500秒。只有当A和B都判断正确，反应时都不超过500秒的情况下，才算一个试次判断正确。

此外，在正式实验前有一个10试次的练习阶段。这个练习阶段会给被试正误反馈，且如果被试觉得练习不足，还可以按下B键重新进行练习阶段。

在最后的数据记录中，要求添加“TrueACC”和“RTDiff”这两个数据，它们分别记录一个试次是否真正正确以及每个试次A刺激和B刺激的反应时之差。

C:\Program Files\Microsoft Office\MEDIA\OFFICE14\Bullets\BD14578_.gif第三关：问卷\*

用“制作按钮？读取MouseResponseData中被试点击的位置”这章节中学到的知识，制作一个问卷程序。

C:\Program Files\Microsoft Office\MEDIA\OFFICE14\Bullets\BD14578_.gif第四关：前奏曲\*

用E-Prime制作一个文字RPG游戏。

C:\Program Files\Microsoft Office\MEDIA\OFFICE14\Bullets\BD14578_.gif第五关：究极测验\*

用E-Prime制作一个贪吃蛇游戏。

# 第五章- 后记

## C:\Program Files\Microsoft Office\MEDIA\OFFICE14\Bullets\BD21300_.gif本书的不足

写这部分会让我想到写论文里的“研究展望”这部分。不过这部分还是很重要的，不管你学到什么程度，你都有必要，有权利知道还有什么作者没有写的，还有什么作为读者可以改进的。以下是我目前考虑到的可能作为疏漏之处的地方：

**①时间精度**

这一部分诸如TimingMode（你可能会注意到的一个各种控件都出现的属性）、屏幕刷新率等等。这部分难度不小。由于作者忘记写了，写完本书又懒得再码字排版，所以也就算了。

当然，仅仅我在福师大的这几年发现，福师大里几乎没有人注意这个问题。况且这是个很小的问题，影响不大。如果你只是想用E-Prime搞定本科生甚至研究生实验的话，根本不用想这个问题。即使是要发表论文的重要实验，只有你的实验对时间精度要求特别高的时候，才需要考虑这个问题。

**②本书对于实验程序设计框架方面介绍很少**

用不同的实验设计框架，适合不同的实验。

本书主要讲了各种应用，分散的各种技能。至于大体上的实验设计框架，本书未曾提到，只能由读者自己在做题的时候，或者平常设计实验的时候自己思考如何实现自己想要的效果。

虽然说“实验设计框架”这方面属于理论，也属于要思考的部分、问题解决的部分，只要能达到效果，不知道也没什么要紧的。但是如果能花点篇幅讲讲，则可能对读者快速想出实验设计会有不少好处。

**③本教程不是很正式，有些词语可能有些和其他教程不同**

**这一点非常重要！**

重点提醒大家！

因为作者为了写教程方便，因此很多时候是按自己的方式来翻译词汇的，因此我很难保证和其他更正式教程翻译的一致（并且我根本不知道别人是怎么翻译的）。当然，这并不会给你的学习造成多大影响，这点放心。但是你必须知道这一点：那就是作者翻译词汇时是随心所欲的，还是记英文可靠。

**④示例程序还不够丰富**

拜托……作者精力也是有限的……这点大家也可以相互交流，共享的示例程序越多，无疑对大家的进步都有好处。

**⑤自定义代码仍然还有很多没讲的地方**

虽然这一条写在这里，但这不算是什么不足之处，因为作者本身也不打算讲。进阶篇实际上已经满足大部分实验需求了，更深的很多功能其实本来就很少用。如果有哪位大神有兴趣，何不扩展一个“究极篇”？

**⑥很多内容要求读者不求甚解，理论知识讲的不够多**

这也不算什么不足之处，我本来就只要读者会使用就够了，不需要理解，或者用简单的其他方式来理解。我不希望理论知识让这份教程变得冗长难懂。

**⑦出于作者本人能力问题，有些地方还讲不到**

作者也说了，人外有人，天外有天。作者虽然在福师大本科生里敢自称“大神”，不过在其他地方作者可不了解情况了。但是我知道，即使是在E-Prime上，比作者强的人又何止千万呢？

**⑧教程中或有语言不当之处**

没啥可说，恳请谅解。

## C:\Program Files\Microsoft Office\MEDIA\OFFICE14\Bullets\BD21300_.gif给学弟学妹以及同学们的一段话

光阴荏苒，转眼之间，四年已过。

这四年可谓是意义非凡的四年，在这四年里，无论课内课外，我都学到了很多。即便这样，我仍然感叹时光易逝，不少时间就这样从指尖划过，浪费掉了。

我写这份教程，就是为了纪念在学院生活的四年，想为学院留点什么，以表感谢。写这教程费了不少精力，我希望这份教程能派上用场，帮助到学弟学妹，乃至更多的心理学本科生，让他们当中想认真做心理学研究的有志之士有所收获；也让他们当中另有志向的同志们轻松愉快地学习E-Prime，以便于应付任务，不至于在不感兴趣的地方上浪费更多时间。

虽然这段话看上去不好听，不过我通过四年学习，还是深刻体会到：志向不在这里，硬是搞一些无法对科学研究做出贡献的实验，实际上是没什么意义的。我想大家都承认，很多学生做实验无非是为了应付任务，这自然不可能对学科有什么贡献，对他自身如果志不在此也没什么太大帮助。

教程自然也有很多疏漏之处，也希望更厉害的“大神”们能够对教程做出修改完善，编辑出更高级的版本。

我想对学弟学妹们说，时间是非常宝贵的，请好好使用。无论你志在何方，请用在你认为不会后悔的事情上吧。时光不会倒退，这一点是最令人惋惜的。但我不会为浪费的时光而后悔，无论如何，它们都是我生命的一部分，希望你们也不会。我希望，我们都能够，抬头往前看。

另一点我想说的是，我在大学的四年里，听到太多大学生，太多太多人。他们的志向仅仅只是一辈子吃喝玩乐，一辈子不用做太多事，简单的度过就行了。我并不是要批判什么，我也没资格这么做。可我希望我们大家能不能参考另外一种人生态度：**那就是，我要为世界进步做出点什么，哪怕是再小的一步。**

毕竟，我们还年轻，人生才刚刚开始，如果就把自己定位成咸鱼，定位在吃喝玩乐上，未免显得太单调了。

难道，短暂的一生，我的同学们，你们就愿意这样简简单单就过了吗？还有那么多可能性，那么多值得我们去探索的地方。充满神秘的量子力学、严谨而功能强大的计算机、蕴含着不同民族文化的外语、让我们感受多种人生的文学、包括我们自己所学的学科，混杂着科学色彩又不愿彻底舍弃哲学讨论，还在冉冉升起的新星——心理学，这么多选择……这么多选择！为什么不试试呢？如果我们只是工作，赚钱，吃饭睡觉，朋友，你真的不觉得太简单了吗？

何妨一试呢？假设最终我们没有获得什么，没有什么成就，我们终究试过，我们可以问心无愧地说，我们曾经尝试。我们也比那些度过简单人生的其他人们，有了更丰富的经历。

我们也不需要过于担心什么，不需要为做不出大贡献而惭愧。毕竟，勤勤恳恳地工作，获得自己劳动所得，这就是再为世界做贡献了。

我非常喜欢费恩曼这位物理学家在他书中说的：“物理学是一杯红酒，我们应该好好欣赏，一饮而尽，然后全部忘掉。” （大意）

自然，我没资格替任何人决定什么，我所说的话仅供参考。如果你觉得吃吃喝喝才是人生真谛，我欢迎你保留看法，并且祝你如愿以偿。因为看了很多大学生持这种观点，颇有感悟，故作此文。

祝各位读者学习工作顺利，永远快乐，一生平安。

By 卡希尔将军

2019毕业季

# 附录

## 本教程的组成部分

轻松学习E-Prime.docx：教程的主要部分

E-Prime2.0官方自带案例：这是E-Prime2.0自带的案例，一般能够在你电脑上的“我的文档/My Experiments”里找到。

E-Prime2.0官方自带教程：E-Prime2.0自带的帮助文件，一般在E-Prime的目录下能找到。

本教程的案例：作者编写的用来说明的案例。

综合练习：各个综合练习的答案或者案例。里面的内容不会包括所有题目的。

## 各个按键在按键反应中输入的方法

在官方帮助文档的E-Basic里有非常清楚的说明，建议您如果英文过关直接去那里看。这里只是从那里照搬过来，也很方便复制。所有符号都为英文键盘输入格式。

**键盘：**

数字0-9：直接在Allowable中输入相应数字。如可以按1和2，就输入：12。



**输入{NUMBER}则设定所有数字键都允许。**

字母：直接输入字母，大小写区分。如要按fjdk，不计大小写，则如下：



**输入{ALPHA}则为允许所有字母。**

特殊：

{ANY}：所有按键都允许。

{SPACR}：空格键。

{ENTER}：回车键。

{ALT}：换挡键（Alt键）。

{CAPSLOCK}：大写锁定键。

{CONTROL}：控制键（Ctrl键）。

{SHIFT}：上档键（SHIFT键）。

{TAB}：制表符。

{DELETE}：删除键。

{DOWNARROW}：下箭头。

{UPARROW}：上箭头。

{LEFTARROW}：左箭头。

{RIGHTARROW}：右箭头。

{END}

{ESCAPE}：退出键（ESC键）。

{HOME}

{INSERT}

{LEFTBRACE}：左花括号。

{RIGHTBRACE}：右花括号。

{LEFTBRACKET}：左方括号。

{RIGHTBRACKET}：右方括号。

{NUMLOCK}：数字锁定键。

{NUMPAD5}：小键盘5号键。

{PGDN}：Page Down。

{PGUP}：Page Up。

{QUOTE}：引用（esc下方的键）

{SCROLLLOCK}：滚动锁定键。

{F1}：F1。

{F2}：F2。

{!}：感叹号。

{@}：@号。

{#}：井号。

{$}：美元符号。

{%}：百分号。

{^}：乘方。

{&}：和。

{\*}：星号。

{(}：左括号。

{)}：右括号。

{\_}：下划线。

{+}：加号。

{|}：分隔符。

{<}：小于号。

{>}：大于号。

{?}：问号。

{-}：负号。

{=}：等于号。

{;}：分号。

{’}：单引号。

{,}：英文逗号。

{.}：英文句号。

{/}：斜杠。

{BACKSLASH}：反斜杠。

**鼠标：**

鼠标用1和2分别代表左键和右键，此外还有3-8也是按键。

**松开按键：**

以上所有按键都可以设定松开时记录，只需要在其前面加一个负号，如：

{-S}松开大写S时记录。

{- DOWNARROW }松开下箭头时记录。

要注意的是，如果要设定松开按键记录，您需要先去实验设置中设定。详见第三章的“松开按键时记录反应时”这一节。