

**认知心理学进阶第三讲：
认知实验与PsychoPy程序编制**

严超轶
Chao-Gan Yan, Ph.D.

ycg.yan@gmail.com
http://rfmri.org/yan

清华大学心理与认知科学系

1

认知实验

RDoC Framework

NIH Toolbox - for Assessment of
Neurological and Behavioral
Function

2

2

认知实验

CEST / 1_2_2_DCCS /

Chaogen-Yan CEST Init

- Name
- resources
- DS_Store
- DCCS-legacy
- DCCS.js
- DCCS.pyexp**
- DCCS_lastrun.py
- index.html**
- readme.md

3

3

前言：关于PsychoPy

- PsychoPy 是一个用 Python 编程语言编写的开源软件包。PsychoPy 结合了 OpenGL 的图形优势和简单的 Python 语法，为科学家提供了一个免费和简单的刺激呈现和控制的软件包。它被世界各地的许多实验室用于心理物理学、认知神经科学和实验心理学。

>30k users >4k citations >190 github contributors >150 pavlovian licences

4

4

开发者

Jonathan Pierce

Jeremy Gray

Additional: Yaroslav Halchenko, Erik Kastman, Mike MacAskill, William Hogman, Jonas Lindelav, Ariel Rokem, Dave Britton, Gary Strangman, C Lüthmann, Sol Simpson, Hiroyuki Sogo

5

5

为什么选择PsychoPy?

工具名	上手难度	可迁移性	自由度	在线实验
MATLAB	高(纯代码环境)	中(需配置环境)	高	难以实现
E-Prime	中(可视化编辑器+代码编辑器)	低(换个机器就炸了)	低	几乎无法实现
Inquisit	高(纯代码环境)	高(仅需浏览器插件)	中	完全支持
jsPsych	高(纯代码环境)	高(基于网页开发)	高	完全支持
PsychoPy	中(可视化编辑器+代码编辑器)	高(需配置环境)	高	完全支持

表格来源: <https://zhuanlan.zhihu.com/p/547122997>

6

6

为什么选择PsychoPy?

- 简单的安装过程(独立环境)
- 各种各样的学习资源
- 一个活跃的用户论坛提供帮助和支持
- 精确计时(Bridges et al., 2021)

Easy

enough for teaching

Precise

enough for psychophysics

Flexible

enough for everything else

Online
or lab-based, your choice

- 各种各样的刺激(图像, 文本, 光栅, 随机点, 噪音, 电影, 声音)
- 输入从键盘, 鼠标, 麦克风或按钮框
- 兼容一系列硬件(EEG, 眼动设备, fMRI.....)
- 多显示器支持
- 显示器自动校准

7

下载

8

学习资料推荐

- 书籍: 《实验编程: PsychoPy从入门到精通》
- 视频: [Youtube tutorials](#)
- 论坛: [user forum](#)
- 一个最为快速的学习方式: 边做边学, 使用ChatGPT。遇到报错, 将报错信息复制到浏览器搜索引擎中, 看看网上有没有解决方法(通常在论坛上可以找到)。

9

认知实验与PsychoPy程序编制

- 1 Builder 界面介绍
- 2 常用组件及其属性
- 3 设置循环
- 4 提供反馈
- 5 PsychoPy 输出结果文件
- 6 例: 编写一个 2-back 实验

10

认知实验与PsychoPy程序编制

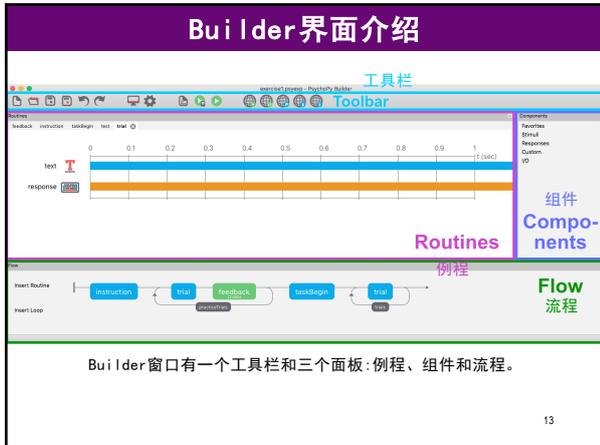
- 1 Builder 界面介绍
- 2 常用组件及其属性
- 3 设置循环
- 4 提供反馈
- 5 PsychoPy 输出结果文件
- 6 例: 编写一个 2-back 实验

11

Builder界面介绍

- 双击安装好的 PsychoPy 软件图标, 将会打开三个窗口: Builder, Coder 和 Runner
- Coder 界面为比较有力的程序员提供强大的灵活性和齐全的功能, 与此相比, Builder 界面则设计得更为简单, 即使是不熟悉编程的用户也可以在其中编写和运行自己的实验
- 通常的做法是: 首先在Builder 界面下编写大部分实验程序, 然后简单地通过添加代码组件或进入Coder 界面来处理不容易用图形用户界面建构的内容

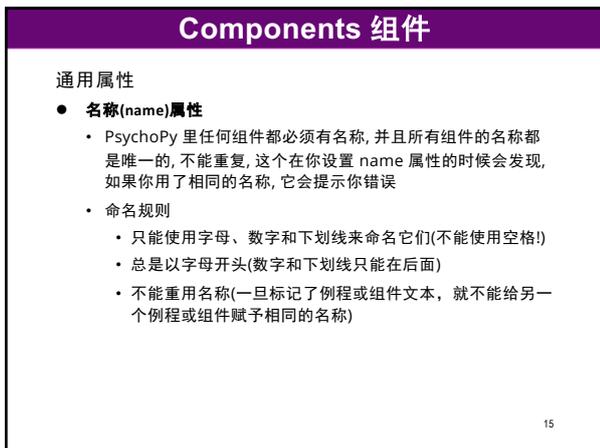
12



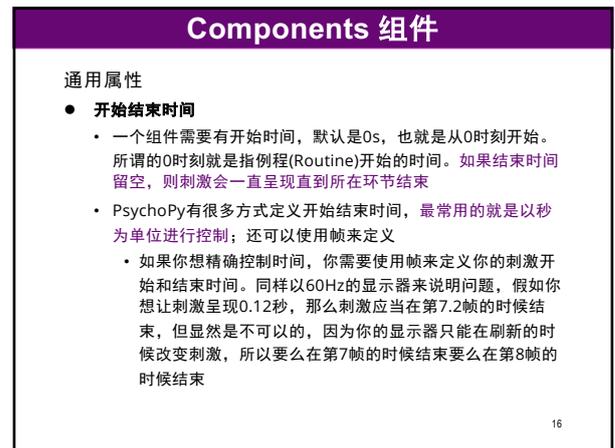
13



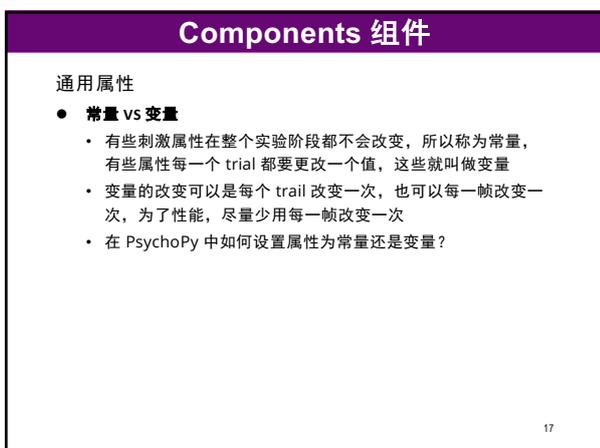
14



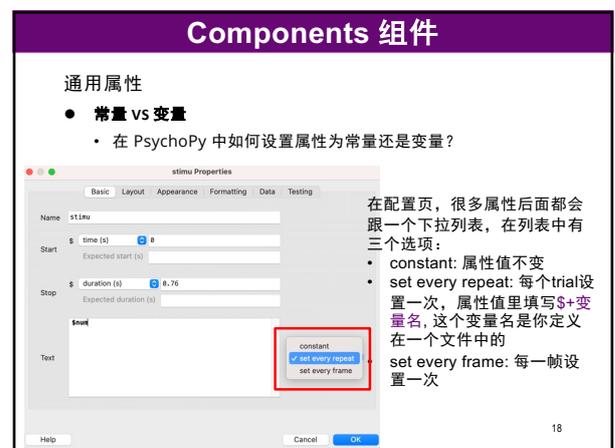
15



16



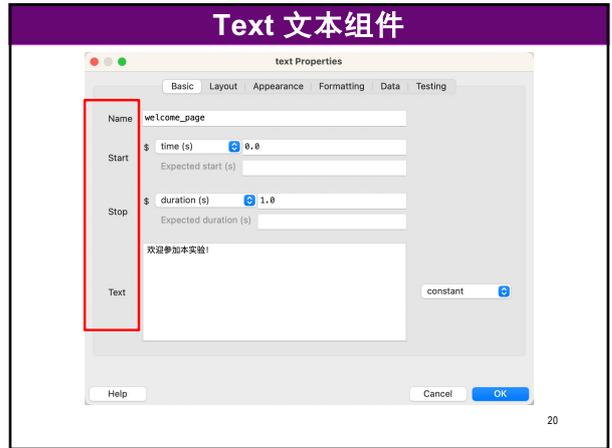
17



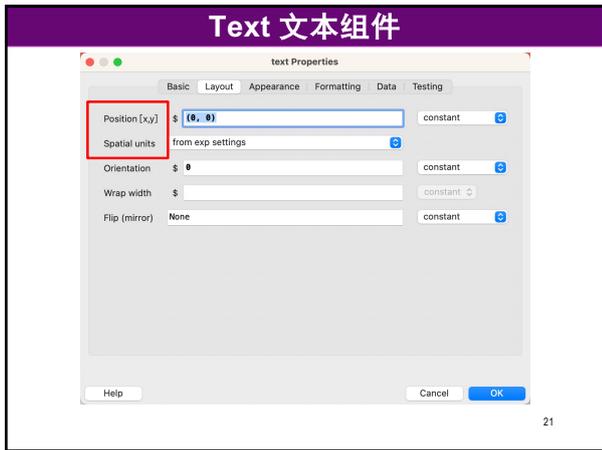
18



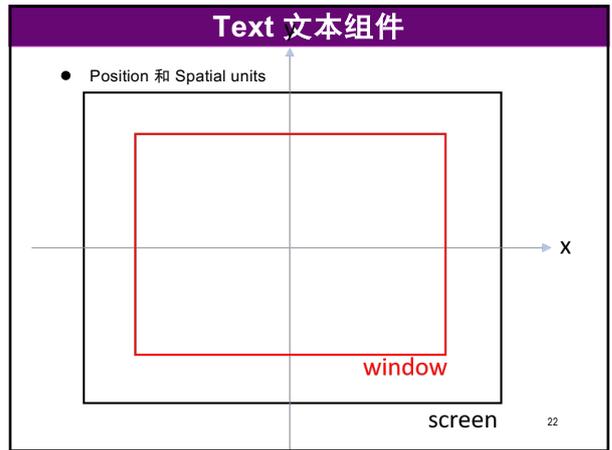
19



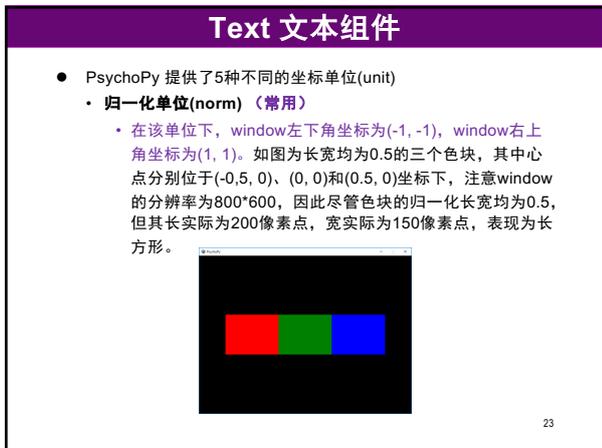
20



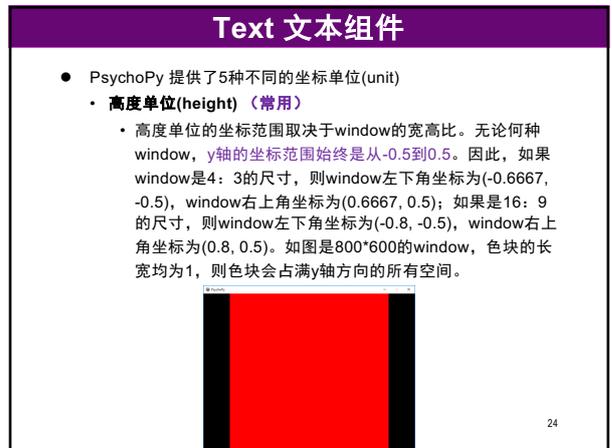
21



22



23



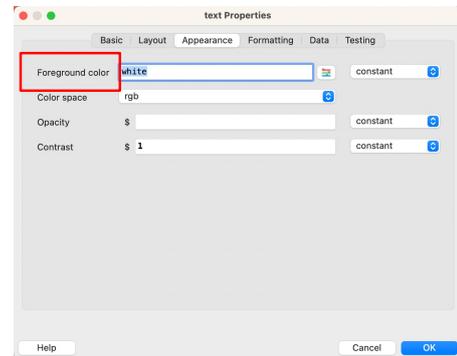
24

Text 文本组件

- PsychoPy 提供了5种不同的坐标单位(unit)
 - **像素单位(pixel)**
 - 像素单位的坐标范围取决于screen的宽、高像素点数, 假设screen宽度为w个像素点, 高度为h个像素点, 则screen左下角坐标为(-w/2, -h/2), 右上角坐标为(w/2, h/2)。
 - **厘米单位(cm)**
 - 厘米单位的坐标范围取决于screen的宽度和高度, 假设screen宽度为w厘米, 高度为h厘米, 则screen左下角坐标为(-w/2, -h/2), 右上角坐标为(w/2, h/2), 每cm所代表的像素长度则由screen的像素点数确定。
 - **视角(deg, degFlatPos, degFlat)**
 - 视角单位是五种单位中最复杂的坐标单位, 使用该单位, 不仅要知道屏幕的大小、像素点的多少, 还要知道被试距离屏幕的距离, PsychoPy 提供三种不同的视角单位deg, degFlatPos和degFlat。

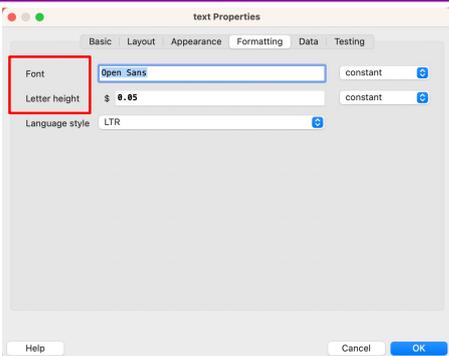
25

Text 文本组件



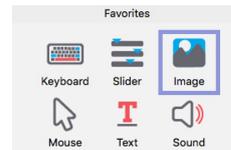
26

Text 文本组件



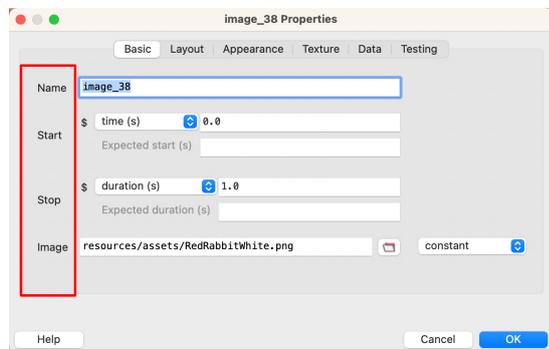
27

Image 图片组件



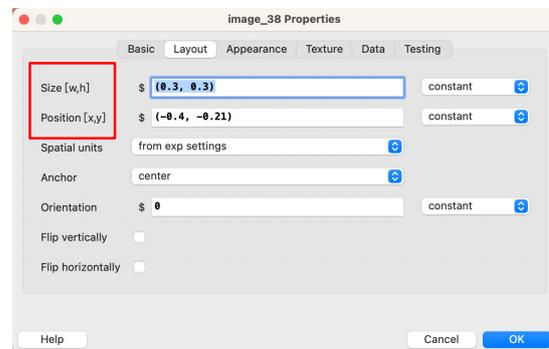
28

Image 图片组件



29

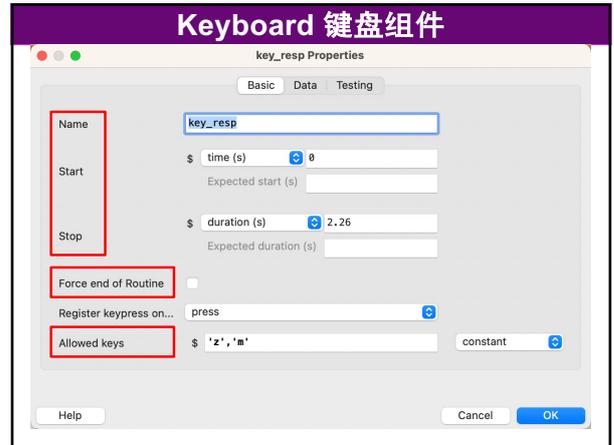
Image 图片组件



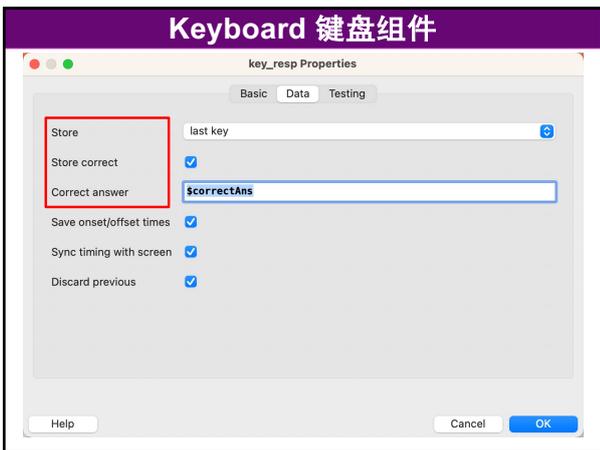
30



31



32



33

Keyboard 键盘组件

- Force end of Routine
 - 如果勾选此项，则响应将结束试次
- Allowed keys
 - 通过插入一系列按键名称来设置按键。每个按键都应使用引号（单引号或双引号）并用逗号隔开。若要允许键盘上的所有键，请将Allowed keys字段保留为空
 - 例如如果你已经告知被试要使用 z 键和 m 键来进行实验，但他们按了别的键，那么这些按键就无法被视为“反应”
- Store
 - 存储响应的默认选项是“最后一个键”。然而，“第一个键”可能是更好的选择(如果响应不正确，你将存储不正确的键，从而知道响应是不正确的)。但是，请注意，如果选择了“Force end of Routine”，两个设置的结果将会是相同的（因为第一个响应将会结束试次）。还要注意，你可以选择“nothing”，即不存储响应，这适用于指导语的例程中

34

Keyboard 键盘组件

- Store correct
 - 如果勾选，PsychoPy将存储试次的答案是否正确，1=正确，0=错误
- Correct answer
 - 当然，PsychoPy无法知道给定试次的正确答案是什么。因此，你必须向PsychoPy提供此信息。你可以通过在输入文件中设置一系列答案列表来实现，例如\$corrAns
- 对于其余属性，只需保留默认值

35

Routine 例程

The image shows the PsychoPy routine editor with a timeline from 0 to 1 second. A 'test' task is scheduled at the beginning, followed by a 'response' task. Below the editor is the 'New Routine' dialog box, which has 'New Routine name:' and 'Routine Template:' fields. The 'Routine Template' dropdown is set to 'Basic:blank'.

36

新建例程

Insert Routine

(new)
trial
taskBegin
instruction
feedback

New Routine

New Routine name:

Routine Template:

Cancel OK

- 通过单击流程中的 Insert Routine 和 (new) 向实验添加新例程。
- 新创建的例程将为空，等待你向其中添加组件。
- Insert Routine 也可以用来向流程中添加已有的例程。

37

37

删除例程

Routines

feedback instruction **newRoutine** taskBegin trial

- 将例程从整个实验中删除（完全删除）：点击例程名称旁边的

instruction

remove
rename

- 将例程从流程中删除（部分删除）：右键单击例程并选择 remove
 - 注意，这只是从流程中的特定位置删除例程，位于流程中其他位置的例程不会被删去。

38

38

复制和粘贴例程

Experiment	Demos	Pavlo
New Routine		⌘N
Copy Routine		⌘C
Paste Routine		⌘V
Rename Routine		⌘R
Paste Component		⌘V

- 选择菜单栏 Experiment >> Copy Routine 和 Paste Routine
- 在 Insert Routine 中选择添加已复制的例程

复制粘贴例程
可以减少重复
劳动，节省时间！

39

39

The toolbar 工具栏

- 工具栏的按钮分为四组

文档的基础操作：新建、保存、撤销等

显示器设置和实验设置

编译Python/JS脚本、运行实验

运行在线实验

40

40

The toolbar 工具栏

- 显示器设置和实验设置
- 你不需要更改显示器设置的任何内容
- 然而，实验设置很重要。在实验设置中最相关的是 Basic 选项卡和 Screen 选项卡
 - Basic 选项卡是你更改实验信息的地方，这些信息是在实验开始时呈现的启动对话框的一部分。“实验信息”的默认设置询问 participant 和 session

	Field	Default		
Experiment info	participant	<input style="width: 80%;" type="text"/>	+	-
	session	001	+	-

41

41

The toolbar 工具栏

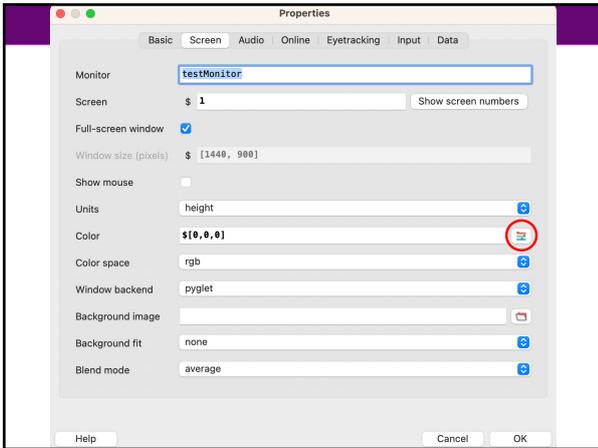
- 显示器设置和实验设置
- 你不需要更改显示器设置的任何内容
- 然而，实验设置很重要。在实验设置中最相关的是 Basic 选项卡和 Screen 选项卡
 - Screen 选项卡：设置背景颜色。默认的背景颜色是灰色
 - 修改背景颜色可以通过点击右侧按钮选择相应颜色或输入 RGB 值

常用颜色的RGB值：

- 默认: \$[0, 0, 0]
- black: \$[-1, -1, -1]
- white: \$[1, 1, 1]
- red: \$[1, -1, -1]
- green: \$[-1, 1, -1]

42

42



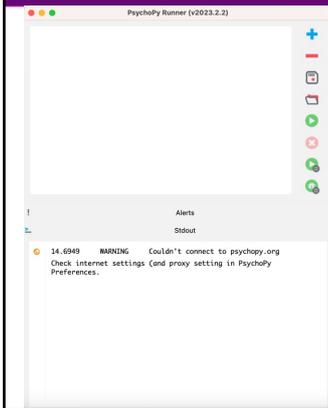
43

The toolbar 工具栏

- 编译Python/JS脚本、运行实验 
- 点击 Compile to Python/JS script 图标将把 Builder 实验编译成 Python/JavaScript 代码
- Run experiment 图标（最右边的那个）直接运行实验（常用）
- Send the experiment to Runner 图标首先将实验发送到一个单独的 Runner 窗口，在那里单击 Run 按钮运行实验（不常用）

44

Runner 界面



- Runner 窗口的下部是 PsychoPy 显示错误消息的地方。Runner 窗口的这一部分称为标准输出 (Stdout)。
- 也可能会有警报、警告或被称为 PsychHID-ERROR 的东西，但这些不会阻止你的实验运行。
- 如果出现错误导致实验停止运行，则会在##### Experiment ended. #####之后立即显示错误信息。

45

退出实验

- 可以随时按 Escape 键退出正在运行的实验
- 如果运行的是macOS，按 Escape 没有效果，试试 Cmd + Alt + Escape

46

认知实验与PsychoPy程序编制

- 1 Builder 界面介绍
- 2 常用组件及其属性
- 3 设置循环
- 4 提供反馈
- 5 PsychoPy 输出结果文件
- 6 例：编写一个 2-back 实验

47

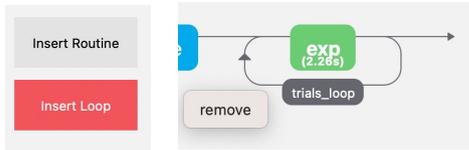
设置循环

- 在一个典型的实验中，往往会有几十甚至上百个试次。这时就需要用到循环。使用循环，你可以定义一个试次模板，并根据需要多次重复该模板，同时从一个试次到下一个试次更新一些信息。例如，在 Nback 任务中，从一个试次到下一个试次改变所呈现的数字



48

新建和删除循环

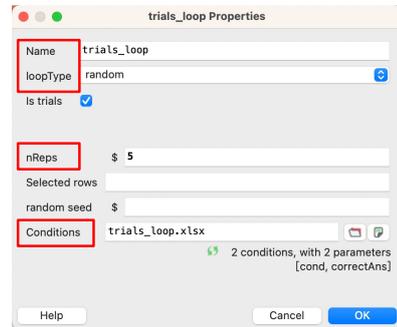


- 新建循环：在流程中单击“Insert Loop”。然后，点击你想要循环开始的实验时间轴。最后，点击你想要循环结束的时间轴
- 删除循环：右键单击循环，然后单击“remove”

49

49

循环的常用属性



50

50

循环的常用属性

- loopType
 - 决定 PsychoPy 是保持输入文件中非标题行的顺序，还是将它们随机化。以下是几个常用的选项：
 - sequential: 输入文件不是随机化的。试次以与输入文件完全相同的顺序呈现
 - random: 输入文件中的行在每次循环重复时随机
 - fullRandom: 输入文件中的行乘以重复的次数，然后随机化
 - 区别？

condition
A
B
C
D

51

51

循环的常用属性

- | | |
|--|---|
| <p>random</p> <ul style="list-style-type: none"> ● 第一次重复 <ul style="list-style-type: none"> ● 把这四种情况随机化 ● 示例结果: C, D, A, B ● 第二次重复 <ul style="list-style-type: none"> ● 把这四种情况随机化 ● 示例结果: A, D, B, C ● 向参与者展示的试验顺序: C, D, A, B, A, D, B, C ● 注意条件是以四个为一组进行随机的 | <p>fullRandom</p> <ul style="list-style-type: none"> ● 将条件乘以重复次数: A, B, C, D * 2 = A, B, C, D, A, B, C, D ● 随机化所有条件 ● 示例结果: C, D, A, A, B, C, D, B ● 请注意, 这四个条件不再组合在一起 |
|--|---|

52

52

循环的常用属性

- nReps
 - 输入文件应该重复多少次。在我们的 Nback 任务中，输入文件共包含2种情况，重复5次，所以总共有10个试次
- Conditions
 - 输入文件的路径。要将输入文件添加到循环中，请单击右侧按钮，导航到文件并将其添加到循环中

Conditions trials_loop.xlsx

说明：
建议将输入文件位于与实验本身(.psyexp文件)相同的目录中(否则，必须指定输入文件的路径)

53

53

认知实验与PsychoPy程序编制

- 1 Builder 界面介绍
- 2 常用组件及其属性
- 3 设置循环
- 4 提供反馈
- 5 PsychoPy 输出结果文件
- 6 例：编写一个 2-back 实验

54

54

提供反馈

- 有时你可能想给被试的反应提供反馈。特别是，当他们还在练习任务时，就可能出现这种情况
- 假设你希望被试在每个试次后都会得到一个反馈。问题是你无法提前知道需要什么类型的反馈(例如，正确或错误)，因为反馈将取决于被试的反应。因此，在实验进行的过程中，你需要弄清楚要给出什么样的反馈。要实现这一点，你需要在反馈例程中添加一个 Code 组件(位于 Custom 组件中)

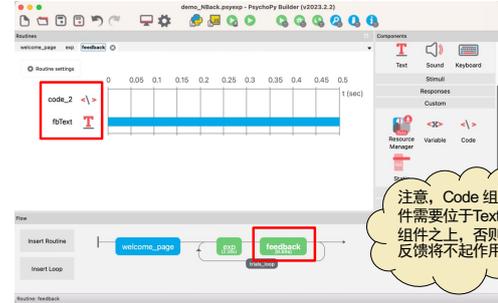


55

55

提供反馈

- 新建一个反馈例程，并向其中添加 Code 组件和 Text 文本组件

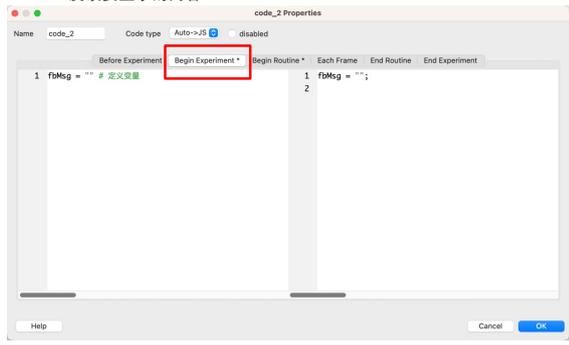


56

56

提供反馈

- Code 组件检查响应是否正确。根据准确性，Code 组件将更新反馈要显示的内容

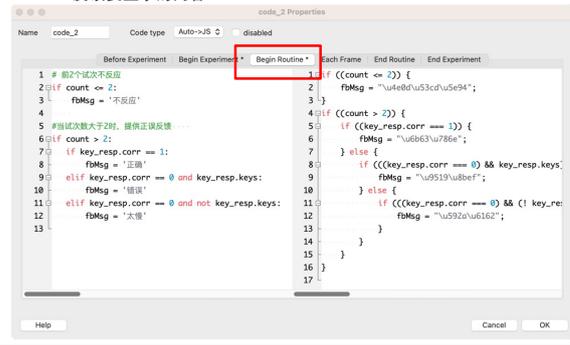


57

57

提供反馈

- Code 组件检查响应是否正确。根据准确性，Code 组件将更新反馈要显示的内容

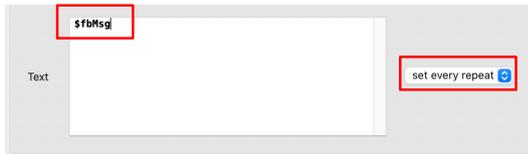


58

58

提供反馈

- Text 文本组件用来呈现反馈内容



59

59

认知实验与PsychoPy程序编制

- 1 Builder 界面介绍
- 2 常用组件及其属性
- 3 设置循环
- 4 提供反馈
- 5 PsychoPy 输出结果文件
- 6 例：编写一个 2-back 实验

60

60

PsychoPy 输出结果文件

- PsychoPy 将输出文件放在一个名为 data 的文件夹中
- 此文件夹是在你第一次运行实验时自动创建的
- 该文件夹位于与实验.psyexp文件相同的目录中
- 对于每个被试，该文件夹中通常有三种类型的文件
 - .csv文件
 - .log文件
 - .psydat文件
- 我们重点关注.csv文件，其默认的命名结构为
<participant_ID>_<experiment_name>_<date>_<time>.csv

61

PsychoPy 输出结果文件

输出结果文件的列包含哪些信息？

	A	B	C	D	E	F	G	H	I	J
1	pracTrials.thisRepN	pracTrials.thisTrialN	pracTrials.thisN	pracTrials.thisIndex	trials.thisRepN	trials.thisTrialN	trials.thisN	trials.thisIndex	response.rt	response.com
2	0	0	0	5					0.425	1
3	0	1	1	9					0.368	0
4	0	2	2	12					0.521	1
5	0	3	3	7					0.475	1
6	etc.	etc.	etc.	etc.					0.423	1
7					0	0	0	4	0.569	1
8					0	1	1	3	0.523	1
9					0	2	2	11	0.463	0
10					0	3	3	9	0.487	1
11					etc.	etc.	etc.	etc.	etc.	etc.

62

61

62

PsychoPy 输出结果文件

输出结果文件的列包含哪些信息？

- 输入文件中的所有信息
- 来自启动对话框的信息(例如， participant ID)
- 与响应组件相关的信息：
 - 按下的键: <name_of_your_response_component>.keys
 - 反应时: <name_of_your_response_component>.rt
 - 正确率: <name_of_your_response_component>.corr (只有当你选择“Store correct”时，这一列才会存在)

63

63

PsychoPy 输出结果文件

输出结果文件的列包含哪些信息？

- 关于循环的信息（不常用）：
 - 每个循环都有自己的几列信息
 - 这些列的列名将以循环的名称开始，例如“trials.+与循环相关的列”
 - 各种与循环相关的列是：
 - .thisRepN: 这个循环运行的频率是多少？
 - .thisTrialN: 这个循环的试次数是多少?(当循环重复时重置)
 - .thisN: 总试次数是多少?(当循环重复时不会重置)
 - .thisindex: 当前试次位于输入文件中的行号

64

64

PsychoPy 输出结果文件

输出结果文件的列包含哪些信息？

- 有关组件的开始和结束时间的信息（不常用）：
 - .started: 开始时间，以秒为单位（相对于实验开始）
 - .stopped: 结束时间，以秒为单位（相对于实验开始）
- 此外，还有日期，帧率，实验名称和 PsychoPy 版本（不常用）

- 在 PsychoPy 中计数从0开始；
- 所有时间都以秒为单位；
- 最相关的列通常是反应时、正确率和条件(有关条件的信息通常来自你的输入文件)。

65

65

认知实验与PsychoPy程序编制

- 1 Builder 界面介绍
- 2 常用组件及其属性
- 3 设置循环
- 4 提供反馈
- 5 PsychoPy 输出结果文件
- 6 例：编写一个 2-back 实验

66

66

例：编写一个 2-back 实验

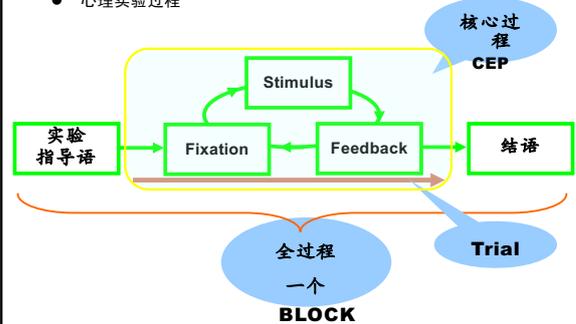
- 编写实验的一般步骤
 - 第一步：画出实验流程图
 - 第二步：新建一个目录，存储本实验程序的相关文件
 - 第三步：在总流程中建立指导语
 - 第四步：在总流程中创建核心实验过程
 - 第五步：在核心实验过程中创建实验各个组成部分，并设置相应的呈现时间、反应方式、需要记录的数据等实验控制的要素。
 - 第六步：在总流程中创建结语
 - 第七步：运行、调试和修改实验程序（多次）直至符合实验
 - 第八步：修改指导语，定稿。运行实验
 - 第九步：查看结果，合并数据，提取数据

67

67

例：编写一个 2-back 实验

- 心理实验过程



68

68

例：编写一个 2-back 实验

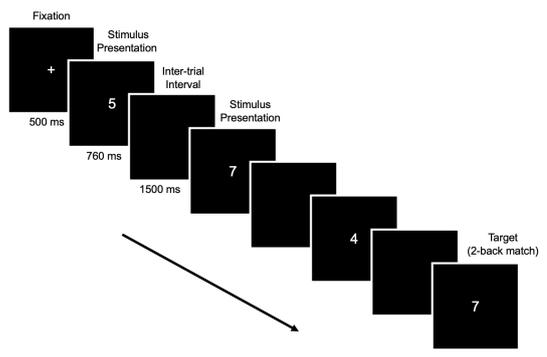
- n-back任务是一种用来评估工作记忆的认知任务。在这个任务中，参与者需要记住一系列呈现的刺激（如字母、数字或图形），并判断当前刺激是否与之前n个刺激中的一个相同。例如，在2-back任务中，参与者需要判断当前刺激是否与两步之前的刺激相同。任务的难度通过调整“n”的大小来改变，n越大，任务难度越高。这种任务有助于探究工作记忆的容量和更新能力。
- 先来运行一下这个实验
 - https://experiment.psych.ac.cn/vanlab/demo_NBack

69

69

例：编写一个 2-back 实验

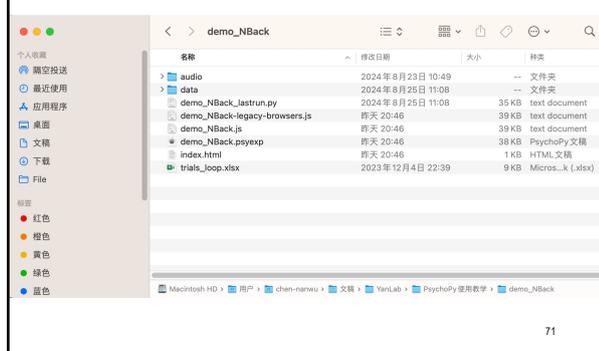
- 首先，画出实验流程图



70

例：编写一个 2-back 实验

- 然后，在你的电脑上新建一个目录，存放实验程序(.psyexp文件)

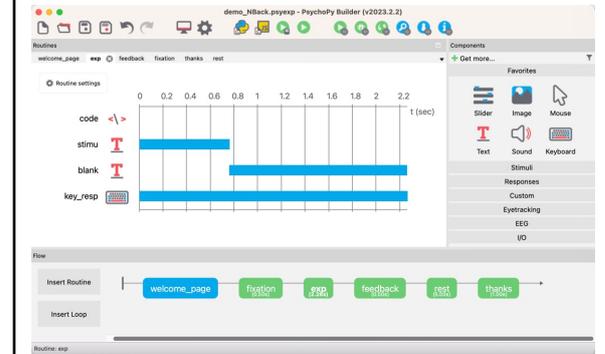


71

71

例：编写一个 2-back 实验

- 接着，在总流程中建立指导语、核心实验过程、结语



72

例：编写一个 2-back 实验

选择Auto->JS可以将python代码自动转写成JS代码，便于跑在线实验

73

例：编写一个 2-back 实验

Begin Experiment

定义变量

```
count = 0
tempnum = []
num = 0
```

count 用来记录试次数
tempnum 是一个列表，用来存放实验刺激
num 是屏幕上所呈现的实验刺激

74

例：编写一个 2-back 实验

Begin Routine

创建一个包含从0到9的整数的列表

```
numlist = [0,1,2,3,4,5,6,7,8,9]
```

打乱 numlist 列表中的元素顺序

```
shuffle(numlist)
```

从 numlist 列表中弹出（并移除）最后一个元素，将其赋值给变量 num

```
num = numlist.pop()
```

count = count + 1

当试次数大于2时，执行以下代码块

```
if count > 2:
    if cond == 'same':
        num = tempnum[1]
        tempnum.pop(1)
    if cond == 'different':
        while num == tempnum[1]:
            num = numlist.pop(0)
            tempnum.pop(1)
```

tempnum中始终只存放2个数

3-5-7-5-6.....

75

例：编写一个 2-back 实验

End Routine

```
tempnum.insert(0,num)
```

将每个试次呈现的刺激放入 tempnum列表最前面的位置

#在输出结果文件中增加列

```
thisExp.addData('stimuli',num)
thisExp.addData('tempnum',tempnum)
thisExp.addData('numlist',numlist)
```

将每次呈现的刺激存入输出结果文件中，便于检查实验运行是否正确

76

例：编写一个 2-back 实验

● 提供反馈、设置循环

	A	B
cond	correctAns	
same	z	
different	m	

77

例：编写一个 2-back 实验

● 提供反馈、设置循环

78

例：编写一个 2-back 实验

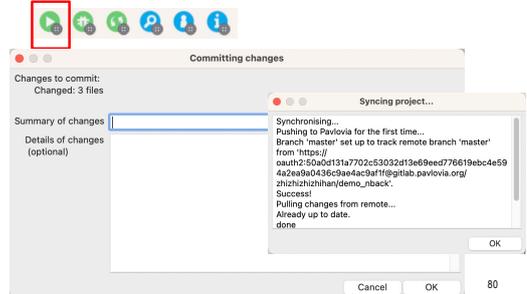
- 设置中途休息
在我们的demo_NBack.psyexp 实验程序中，共包含两个block，只需要在第一个block结束后设置中途休息，第二个block结束后则跳过rest例程
- ```
if block_loop.thisN == 1:
 continueRoutine = False
else:
 continueRoutine = True
```
- count = 0  
tempnum = []
- 在一个block结束后重置实验参数

79

79

## 例：编写一个 2-back 实验

- 上传在线实验
  - 实验编写完成后，在本地测试无误，如有需要，还可以上传至 Pavlovia.org 上进行在线运行



80

80

## 认知实验与PsychoPy程序编制

- 1 Builder 界面介绍
- 2 常用组件及其属性
- 3 设置循环
- 4 提供反馈
- 5 PsychoPy 输出结果文件
- 6 例：编写一个 2-back 实验

81

81



82