

一、写在前面

psychopy去年考试中设计一道填空与一道判断，分值约为4分，如果复习时间充裕的同学可以去官网查看具体的API定义，网址如下：<https://www.psychopy.org/api/index.html>

该资料为 非AI学组官方资料，不代表学组观点，如有错误，欢迎指正！

written by LYY

二、个人认为容易考察的内容

1.core类（感觉和clock.Clock里面内容用法是一致的）

作为psychopy的核心功能，感觉应该比较重要。其中考过的便是时钟：

代码实例如下：

```
timer=core.Clock() #倒计时请用core.CountdownTimer(start=0)，其他用法保持一致
timer.reset(newT=8)
timer.addtime(5)
while timer.getTime()>0:
    ...
```

注意：这个设置的timer是采用正计时的方式来计时的

其他core用法：

```
core.wait(sec) #等待一段时间
core.getAbsTime () #得到unix绝对时间
core.getTime () #得到实例化core之后的时间
```

2.Visual类（提供视觉刺激）

```
from psychopy import visual, core

win = visual.Window(size= (700, 800) ) #构建一个可供显示的窗口

txt = visual.TextStim(win, text='Hello', pos=(0, 100), color='white')#文本
rect = visual.Rect(win, width=200, height=120, pos=(0, -50), fillColor='red')#矩形
#圆形: Circle 多边形: Polygon 线: Line
img = visual.ImageStim(win, image="my_picture.png", pos=(3.0,7.0))#图片
txt.draw() # 先把文字画到后台
rect.draw() # 再把矩形画到后台
win.flip() # 一次 flip 同时显示两者
#要更改属性可以用: img.pos=(9,9) 这种格式
core.wait(1)
win.close()
```

3.sound类

```
import psychtoolbox as ptb
from psychopy import sound

mySound = sound.Sound('A')
now = ptb.GetSecs() #得到高精度时间戳
mySound.play(when=now+0.5) # play in EXACTLY 0.5s
```

4.如何判断摠键或者鼠标摠下

```
from psychopy import visual, event, core

win = visual.Window(size=(800, 600), units='pix')
mouse = event.Mouse(win=win)

event.clearEvents() # 清空之前的键盘事件，避免误触

while True:
    # 1) 键盘：检测是否有按键
    keys = event.getKeys(keyList=['space', 'escape'])
    if keys: # 列表非空，说明按下了
        break

    # 2) 鼠标：检测左键是否按下（按钮0=左键）
    buttons = mouse.getPressed() # [left, middle, right]
    if buttons[0]:
        break

    win.flip() # 刷新一帧（不画东西也可以flip）

win.close()
core.quit()
```

以上只是最简单版本，不考虑摠下时间长短以及是否连击，去抖版本不会写（）

5.保存数据

```

from psychopy import data

exp = data.ExperimentHandler()

exp.addData('resp.key', 'space') #前一个参数是列名，后一个是写入的值
exp.addData('resp.rt', 0.52)
exp.nextEntry()#换行

exp.saveAsPickle('data/demo.psydat') #保存
exp.saveAsWideText('data/demo.csv')

```

```

from psychopy import data

conds = [
    {'color': 'red', 'word': 'RED'},
    {'color': 'green', 'word': 'GREEN'},
] #给一个条件列表
trials = data.TrialHandler(trialList=conds, nReps=3, method='random')#重复3次

for t in trials: # t 就是当前 trial 的 dict
    # 这里呈现刺激...
    # 假装得到反应:
    resp_key = 'space'
    rt = 0.42

    trials.addData('resp.key', resp_key)
    trials.addData('resp.rt', rt)

# 保存 (TrialHandler 自己就能存)
trials.saveAsWideText('data/trials.csv')

```

最后，利用这些应该可以装模作样的写出一个根本跑不通的程序了（）