

# PSYSCRIPT3 TUTORIAL #3

- PROCEDURES
- TABLES
- THE WHOLE EXPERIMENT
- OTHER RESOURCES





# **SECTION 1: PROCEDURES**



# Procedures

```
3  define logMethod localStorage
4
5  proc main
6      move cell F to (0,0)
7      load cell F with image fixationPoint.png
8      move cell C to (0,0)
9      load cell C with image FEN_Carl.png from
10
11     move cell H to (300,200)
12     load cell H with image happy.png
13     move cell S to (300,-200)
14     load cell S with image sad.png
15
16     wait for 2 seconds
17     show cell F
18     wait for 1 second
19     hide cell F
20     show cell C
21     show cell H
22     show cell S
23     wait for a click in HS
24     hide cell C
25     hide cell C
26     hide cell S
27
28     log $lastEventEnder
29     log $lastClick
30     log $lastEventTime
31 end proc
```

- \* By the end of Tutorial #2 you have a script like that shown on the left
- \* It does what you want but it is one long messy list of commands and it's hard to see what part of it does what
- \* We can improve it a bit and this will help when we add even more to the script



# Procedures

```
3  define logMethod localStorage
4
5  proc main
6      move cell H to (300,200)
7      load cell H with image happy.png
8      move cell S to (300,-200)
9      load cell S with image sad.png
10     move cell F to (0,0)
11     load cell F with image fixationPoi
12
13     move cell C to (0,0)
14     load cell C with image FEN_Carl.pr
15
16     wait for 2 seconds
17     show cell F
18     wait for 1 second
19     hide cell F
20     show cell C
21     show cell H
22     show cell S
23     wait for a click in HS
24     hide cell C
25     hide cell H
26     hide cell S
27
28     log $lastEventEnder
29     log $lastClick
30     log $lastEventTime
31 end proc
```

- \* First decide on what you think the sections of **proc main** are
- \* In this case I'm going to say that the first seven lines are involved in setting up the display and the lines after those run one trial and log the results



# Procedures

```
5  proc setUpDisplay
6      move cell H to (300,200)
7      load cell H with image happy.png
8      move cell S to (300,-200)
9      load cell S with image sad.png
10     move cell F to (0,0)
11     load cell F with image fixationPoint.png
12
13     move cell C to (0,0)
14 end proc
15
16 proc runOneTrial
17     load cell C with image FEN_Carl.png for 1 second
18
19     wait for 2 seconds
20     show cell F
21     wait for 1 second
22     hide cell F
23     show cell C
24     show cell H
25     show cell S
26     wait for a click in HS
27     hide cell C
28     hide cell H
29     hide cell S
30
31     log $lastEventEnder
32     log $lastClick
33     log $lastEventTime
34 end proc
```

- \* Use **proc** and **end proc** lines to split up **proc main** up into separate procedures

- \* Notice that the editor now complains that your script fails its check:

Error at line 34:  
Can't find a 'proc main'.

- \* insert a new **proc main** that calls your two new procedures in the right order



# Procedures

```
3  define logMethod localStorage
4
5  proc main
6      call setUpDisplay
7      call runOneTrial
8  end proc
9
10 proc setUpDisplay
11     move cell H to (300,200)
12     load cell H with image ha
13     move cell S to (300,-200)
14     load cell S with image sa
15     move cell F to (0,0)
16     load cell F with image fi
17
18     move cell C to (0,0)
19 end proc
20
21 proc runOneTrial
22     load cell C with image FE
23
24     wait for 2 seconds
25     show cell F
26     wait for 1 second
```

- \* Hit the 'neat' button
- \* Run the script, confirm it does exactly what it did before.
- \* See how simple **proc main** is now, and how you can easily identify which part of the script controls which part of the experiment
- \* Imagine how much this would improve things for a long experiment script instead of just our little tutorial example



# **SECTION 2: TABLES**



# Tables

```
3  define logMethod localStorage
4
5  proc main
6      call setUpDisplay
7      call runOneTrial
8  end proc
9
10 proc setUpDisplay
11     move cell H to (300,200)
12     load cell H with image ha
13     move cell S to (300,-200)
14     load cell S with image sa
15     move cell F to (0,0)
16     load cell F with image fi
17
18     move cell C to (0,0)
19 end proc
20
21 proc runOneTrial
22     load cell C with image FE
23
24     wait for 2 seconds
25     show cell F
26     wait for 1 second
```

- \* At the moment we have an experiment that runs one trial and presents the same stimulus to each participant
- \* This is probably not what you want to do. It's more usual to present a number of stimuli to each participant
- \* And you probably want the stimuli presented in a different order to each participant to prevent priming effects.
- \* Here's how to do that



# Tables

```
3  define logMethod localStorage
4
5  proc main
6      call setUpDisplay
7      call runOneTrial
8  end proc
9
10 proc setUpDisplay
11     move cell H to (300,200)
12     load cell H with image ha
13     move cell S to (300,-200)
14     load cell S with image sa
15     move cell F to (0,0)
16     load cell F with image fi
17
18     move cell C to (0,0)
19 end proc
20
21 proc runOneTrial
22     load cell C with image FE
23
24     wait for 2 seconds
25     show cell F
26     wait for 1 second
```

- \* First, you're going to make the table.
- \* Put all the stimulus files (images in this case, but they can be movies or sounds or text) into a folder
- \* You should already have a folder called **faces** in your project folder. If not, get it from the PsyScript download package
- \* Then you need to make a table which contains all the file names



# Tables

```
22     load cell C with image FEN_Carl.png
23
24     wait for 2 seconds
25     show cell F
26     wait for 1 second
27     hide cell F
28     show cell C
29     show cell H
30     show cell S
31     wait for a click in HS
32     hide cell C
33     hide cell H
34     hide cell S
35
36     log $lastEventEnder
37     log $lastClick
38     log $lastEventTime
39 end proc
40
41 -- ----- stimuli -----
42
43 table facefiles
44     FEN_A_bit_surprised_face.png
45     FEN_Bad_hair_day.png
46     FEN_bedroom_eyes.png
47     FEN_Carl.png
48     FEN_frank.png
49     FEN_Ponytail.png
50     FEN_Young_man.png
51 end table
```

- \* You can type the whole table out yourself as you see on the left
- \* Or type just **table facefiles** and **end table** lines at the top and bottom, then copy the filenames and paste them between the lines you typed
- \* hit the 'neat' button and watch PsyScript indent the table contents. If it doesn't indent correctly, you've done something wrong.



# Tables

```
22     load cell C with image FEN_Carl.png
23
24     wait for 2 seconds
25     show cell F
26     wait for 1 second
27     hide cell F
28     show cell C
29     show cell H
30     show cell S
31     wait for a click in HS
32     hide cell C
33     hide cell H
34     hide cell S
35
36     log $lastEventEnder
37     log $lastClick
38     log $lastEventTime
39 end proc
40
41 -- ----- stimuli -----
42
43 table facefiles
44     FEN_A_bit_surprised_face.png
45     FEN_Bad_hair_day.png
46     FEN_bedroom_eyes.png
47     FEN_Carl.png
48     FEN_frank.png
49     FEN_Ponytail.png
50     FEN_Young_man.png
51 end table
```

- \* Now we have to change the script. Although you've added a table to it it has no idea what to do with the table.
- \* At the moment the script runs just one trial, and it does this by calling the **runOneTrial** procedure
- \* We're going to add a new procedure called **runAllTrials** which calls the **runOneTrial** procedure for each filename in the new table



# Tables

```
3      move cell C to (0,0)
4  end proc
5
6  proc runAllTrials
7      declare $varStimulusFilename
8      repeat 7 times
9          read $varStimulusFilename from table
10         call runOneTrial
11     end repeat
12 end proc
13
14 proc runOneTrial
15     load cell C with image FEN_Carl.png fro
16
17     wait for 2 seconds
18     show cell F
19     wait for 1 second
20     hide cell F
21     show cell C
22     show cell H
23     show cell S
24     wait for a click in HS
25     hide cell C
26     hide cell H
27     hide cell S
28
29     log $lastEventEnder
30     log $lastClick
31     log $lastEventTime
32 end proc
```

- \* Add the new procedure runAllTrials at line 21

```
proc runAllTrials
    declare $varStimulusFilename
    repeat 7 times
        read $varStimulusFilename from
table facefiles
        call runOneTrial
    end repeat
end proc
```

- \* Now we've added a new table *and* a new procedure but the script doesn't know what to do with either of them !



# Tables

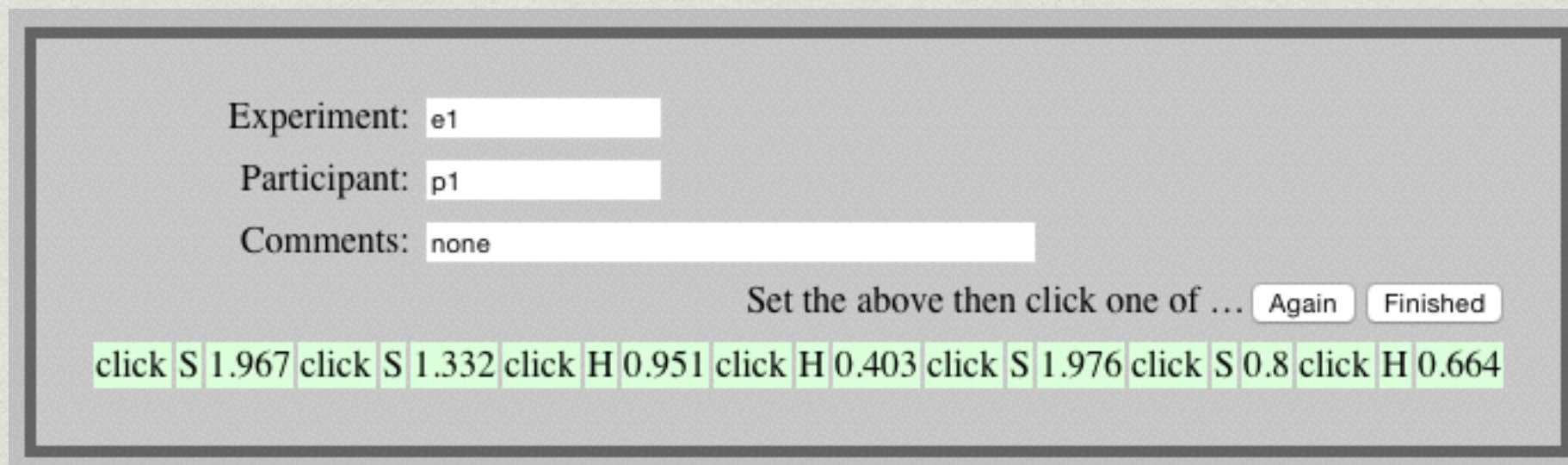
```
18      move cell C to (0,0)
19  end proc
20
21  proc runAllTrials
22      declare $varStimulusFilename
23      repeat 7 times
24          read $varStimulusFilename from table
25          call runOneTrial
26      end repeat
27  end proc
28
29  proc runOneTrial
30      load cell C with image FEN_Carl.png
31
32      wait for 2 seconds
33      show cell F
34      wait for 1 second
35      hide cell F
36      show cell C
37      show cell H
38      show cell S
39      wait for a click in HS
40      hide cell C
41      hide cell H
42      hide cell S
43
44      log $lastEventEnder
45      log $lastClick
46      log $lastEventTime
47  end proc
```

- \* Replace line 7  
**call runOneTrial** with  
**call runAllTrials**
- \* Replace the filename in line 30 with the  
**\$varStimulusFilename**  
variable declared in the  
runAllTrials procedure



# Tables

- \* Run the script. Look at the results.



The screenshot shows a data entry form with the following fields:

- Experiment: e1
- Participant: p1
- Comments: none

Below the fields, there is a text prompt: "Set the above then click one of ...". To the right of this prompt are two buttons: "Again" and "Finished".

Below the buttons, there is a single line of text representing a sequence of stimuli and responses, each in a green box:

click	S	1.967	click	S	1.332	click	H	0.951	click	H	0.403	click	S	1.976	click	S	0.8	click	H	0.664
-------	---	-------	-------	---	-------	-------	---	-------	-------	---	-------	-------	---	-------	-------	---	-----	-------	---	-------

- \* Two obvious problems: first it's all one long line; second, you can't tell which stimulus is leading to which response.
- \* Insert at line 44: `log $varStimulusFilename`
- \* Insert at line 48: `log $return`
- \* Run the script again. Look at the results now. Far better.



# Tables

```
32      wait for 2 seconds
33      show cell F
34      wait for 1 second
35      hide cell F
36      show cell C
37      show cell H
38      show cell S
39      wait for a click in HS
40      hide cell C
41      hide cell H
42      hide cell S
43
44      log $varStimulusFilename
45      log $lastEventEnder
46      log $lastClick
47      log $lastEventTime
48      log $return
49      end proc
50
51      -- ----- stimuli -----
52
53      table facefiles in random order
54          FEN_A_bit_surprised_face.png
55          FEN_Bad_hair_day.png
56          FEN_bedroom_eyes.png
57          FEN_Carl.png
58          FEN_frank.png
59          FEN_Ponytail.png
60          FEN_Young_man.png
61      end table
```

- \* One remaining problem: every time you run the script it will present the stimuli in the same order: the order they were listed in the table. You want a random order to prevent priming effects.
- \* You do this by changing the table definition, not by changing any of the programming.
- \* Append **in random order** to the table definition on line 53
- \* Now run the script again



# **SECTION 3: THE WHOLE EXPERIMENT**



# The whole experiment

```
32      wait for 2 seconds
33      show cell F
34      wait for 1 second
35      hide cell F
36      show cell C
37      show cell H
38      show cell S
39      wait for a click in HS
40      hide cell C
41      hide cell H
42      hide cell S
43
44      log $varStimulusFilename
45      log $lastEventEnder
46      log $lastClick
47      log $lastEventTime
48      log $return
49      end proc
50
51      -- ----- stimuli -----
52
53      table facefiles in random order
54          FEN_A_bit_surprised_face.png
55          FEN_Bad_hair_day.png
56          FEN_bedroom_eyes.png
57          FEN_Carl.png
58          FEN_frank.png
59          FEN_Ponytail.png
60          FEN_Young_man.png
61      end table
```

- \* Now we have a script which presents a number of stimuli to the user in random order and logs each stimulus, the reaction to it, and how long the participant took to react to it. That's a proper experiment !
- \* But the procedure is a bit raw. It would be nice if you had some introductory text explaining what the experimental procedure, and it would be nice to tell the participant that the experiment is finished and thank them.



# The whole experiment

- \* Add two text blocks to you script as follows:

```
49     log $lasteventtime
50     log $return
51 end proc
52
53 textblock instructions
54 During the experiment you will see a number of faces, one by one, with the words 'happy' and
55 'sad' shown next to each face. We are interested in whether you think each face is showing a
56 happy or sad expression. Please click on the word you think more closely describes what the
57 face is showing.
58
59 We are interested in your initial reaction to the face rather than a considered decision.
60 Please pick either 'happy' or 'sad' quickly, even if you might normally spend some time trying
61 to look for detailed clues.
62
63 When you are ready for the experiment to begin, click the 'OK' button.
64 end textblock
65
66 textblock thankyou
67 The experiment is now over ! Thank you for participating.
68
69 Please don't click or type anything more on this computer. The experimenter will now take
70 over.
71 end textblock
72
73 -- ----- stimuli -----
74
75 table facefiles in random order
```



# The whole experiment

```
3 define logMethod localstorage
4
5 proc main
6     display text from textblock instructions
7     call setUpDisplay
8     call runAllTrials
9     display text from textblock thankyou
10 end proc
11
12 proc setUpDisplay
```

- \* Add lines 6 and 9 as shown to the left
- \* The main procedure is still simple and easy to understand. Someone (your reviewer, your co-authors) is going to have to read your script and you want to make life easy for them.



# The whole experiment

```
3 define logmethod localstorage
4
5 proc main
6     display text from textblock inst:
7     call setUpDisplay
8     call runAllTrials
9     display text from textblock than:
10 end proc
11
12 proc setUpDisplay
13     move cell H to (300,200)
14     load cell H with image happy.png
15     move cell S to (300,-200)
16     load cell S with image sad.png
17     move cell F to (0,0)
18     load cell F with image fixationP
19
20     move cell C to (0,0)
21 end proc
22
23 proc runAllTrials
24     declare $varStimulusFilename
25     repeat 1 times
26         read $varStimulusFilename from
27         call runOneTrial
28     end repeat
29 end proc
30
```

- \* Run the script again and see the whole thing in action.
- \* Try responding in various ways and with various timings to the stimuli, examine the logs and see how the reflect what you did.
- \* In the unlikely event that the logs show something else than what you did, see if you can work out why.



# **SECTION 4: OTHER RESOURCES**



# More ?

- \* The PsyScript editor comes with a cheatsheet and full documentation for each command. See the yellow area in the top left corner of your editor window.
- \* The PsyScript package you downloaded comes with many sample scripts demonstrating various script commands and scripting techniques. Look in the folder that the 'editor' folder is in.
- \* For more information about PsyScript 3 please see <https://open.psych.lancs.ac.uk/software/PsyScript3/>