Opensesame Semlink script guide

Summary

[2 How to use 1](#_Toc96442068)

[3 Sequence Description 2](#_Toc96442069)

[3.0 Intro 2](#_Toc96442070)

[3.1 Phase 1: preconditionning 3](#_Toc96442071)

[3.1.1 Higher structure 3](#_Toc96442072)

[3.1.2 Block structure 3](#_Toc96442073)

[3.2 Phase 2: Conditioning 5](#_Toc96442074)

[3.2.1 Higher structure 5](#_Toc96442075)

[3.2.2 Block structure 6](#_Toc96442076)

[3.3 Phase 3: Distractor/localizer 7](#_Toc96442077)

[3.4 Phase 4: Inference 8](#_Toc96442078)

[3.4.1 Higher structure 8](#_Toc96442079)

[3.4.2 Block structure 8](#_Toc96442080)

[3.5 End 9](#_Toc96442081)

[4 Object types 9](#_Toc96442082)

[4.1.1 Sequence  9](#_Toc96442083)

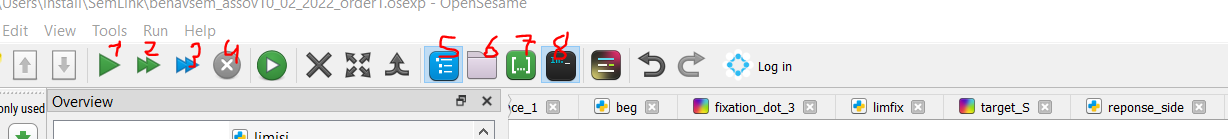
[4.1.2 Loop  10](#_Toc96442084)

[4.1.3 Sketchpad  and Feedback  11](#_Toc96442085)

[4.1.4 Script  13](#_Toc96442086)

[4.1.5 Keyboard\_response  13](#_Toc96442087)

# How to use



**Running xp**

(1) launches the experiment fullscreen. A window opens that ask for the participant number, and then asks you where you want to save it

(2) same but launch in a window, usefull for crashtesting

(3) quickrun, launch in window directly with random subject number, store in quickrun.csv

(4) stops experiment when running

**Display**

(5) shows the overvoiew on the left side

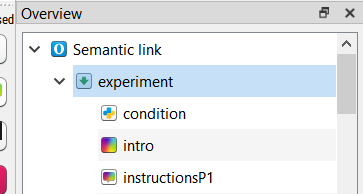
(6) shows the filepool on the right side

(7) shows the variables on the right side

(8) shows the python console at the bottom

# Sequence Description

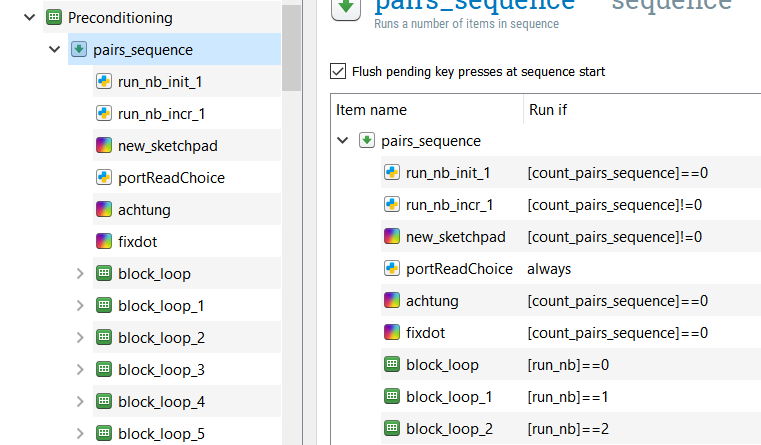
## Intro



* **Semantic link** is the name of the experiment
* **experiment** is the sequence in which all the experiment is stored
* **condition**: script that computes and implement the specific condition depending on subject number
  + It will assign the correct text for the instruction
  + and create several variable used later
    - **Pp**: gives the condition for P2 and P4 (extracted from subject number); 1 is left for reward and right for no reward, 2 the opposite
    - **d\_cond**: gives the condition for P1 and P3 (extracted from subject number); 1 is left and 2 is right)
    - **r\_cond**: gives the reward condition (extracted from subject number); 1 is as seen in the files, 2 is reversed (but taken into account in the log files)
* **intro**: sketchpad that shows welcome and general instruction, you can remove it if you want
* **instructionP1**: feedback that shows specific instructions (it is a feedback rather than a sketchpad because it depends on a newly created variable)

## Phase 1: preconditionning

### Higher structure



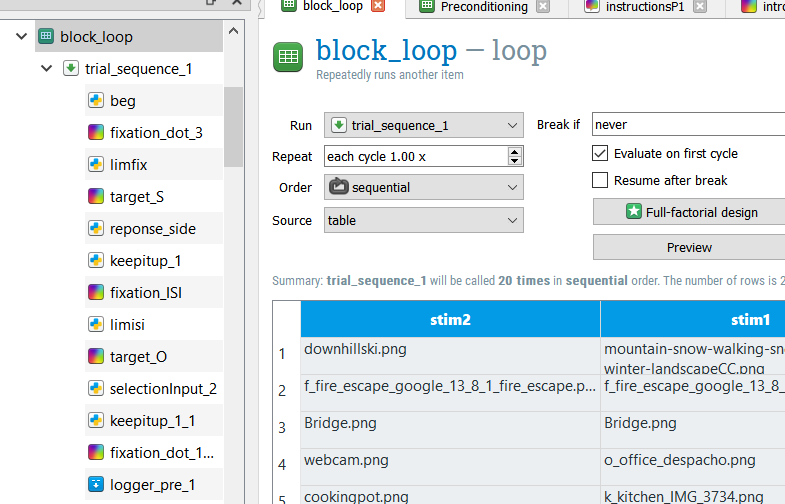
* **Preconditioning**: empty loop object that allows to rerun the following sequence 6 times (one for each run)
* **pairs\_sequence**: sequence of the next elements, replayed before each run. Each time it is repeated it plays one (and only one) different run because of the conditional statements above (under Run if).
* **run\_nb\_init\_1**: initialise a variable called run\_nb at 0, this variable is used for the conditional statement. Only in the runs in the 1st repetition of the sequence
* **run\_nb\_incr\_1:** adds 1 to the variable run\_nb; doesn’t run the 1st repetition of the sequence

Note: I might have used [count\_pairs\_sequence] directly rather than creating a new variable run\_nb, but I think to remember I had issue that way. In any case, it works now even if it is not optimal.

* **new**\_**sketchpad**: blank sketchpad with duration 0, displayed while the next script runs
* **portReadChoice**: script that reads a port and waits for 3 scanner triggers to go the next object in the sequence.
* **achtung**: sketchpad that shows “achtung” for 500ms
* **fixdot**: fixation cross for 500ms
* **block\_loop\_X**: loop that contains the stimuli to be played (as well as ISI, ITI and other parameters), and the sequence to play a trial

### Block structure

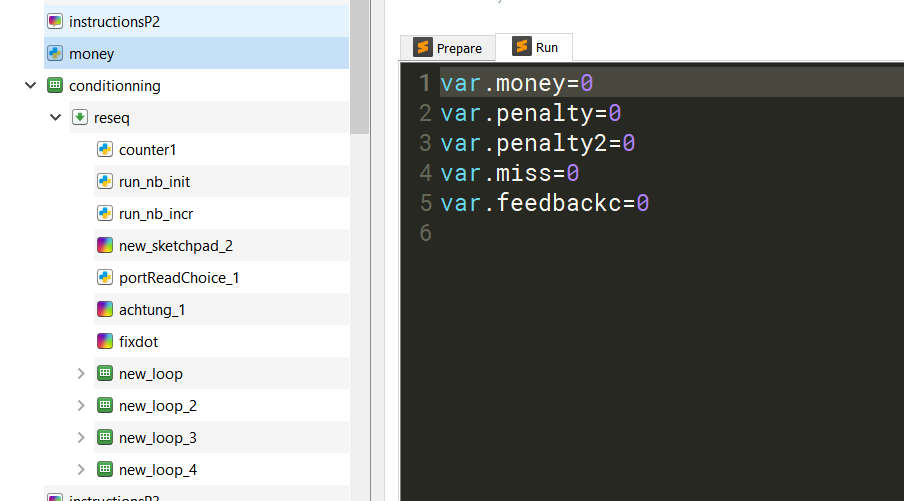
Only the first block is explained here, as they all share the same sequence, only the table containing the parameters change



* **block\_loop**: loop that contains the parameters of the 20 trials
  + stim2: object filename
  + stim1: scene filename
  + correct\_response: 1 if SO pair, 0 if SS (THIS VARIABLE IS CHANGED LATER TO REFLECT THE REAL EXPECTED RESPONSE)
  + paired: 1 or 0
  + Reward: 1 or 0
  + catnb: ignore, just leftover from neurodesignscript, not reliable
  + dur: ITI duration in seconds
  + ISI: ISI duration in seconds
* trial\_sequence\_1 : sequence trat contains all the trials elements
* **beg** : script that note the time at the beginning of the trial, to control for drifts (only run in the first repetition of the sequence, after that, the end of the last trial is used as the beginning of the current one)
* **fixation\_dot\_3**: fixation cross sketchpad with duration 0, the duration is controlled in the next script object
* **lim\_fix**: script that keeps the previous sketchpad up until the ITI duration has passed
* **target\_S**: sketchpad that shows the scene image, duration 0
* **response\_side**: script that assign a new value to **correct\_response** depending on the trial parameters and the conditions of the subject
* **keep\_it\_up\_1**: script that keeps the previous sketchpad up until the stimulus duration has passed
* **fixation\_ISI**: blank sketchpad, duration 0
* **limisi**: script that keeps the previous sketchpad up until the ISI duration has passed
* **target\_O**: sketchpad that shows the object image, duration 0
* **selectionInput2**: scripts that reads port to see if there is a trigger from the button box, and assigns the correct values to response and RT
* **keep\_it\_up\_1\_1**: script that keeps the previous sketchpad up until the stimulus duration has passed.
* **fixation\_dot\_1\_1\_1**: interrun fixation dot, only runs at the end of each run, to capture a response in its entirety
* **logger\_pre\_1**: create a new line in the log file and write the current value of each variable

## Phase 2: Conditioning

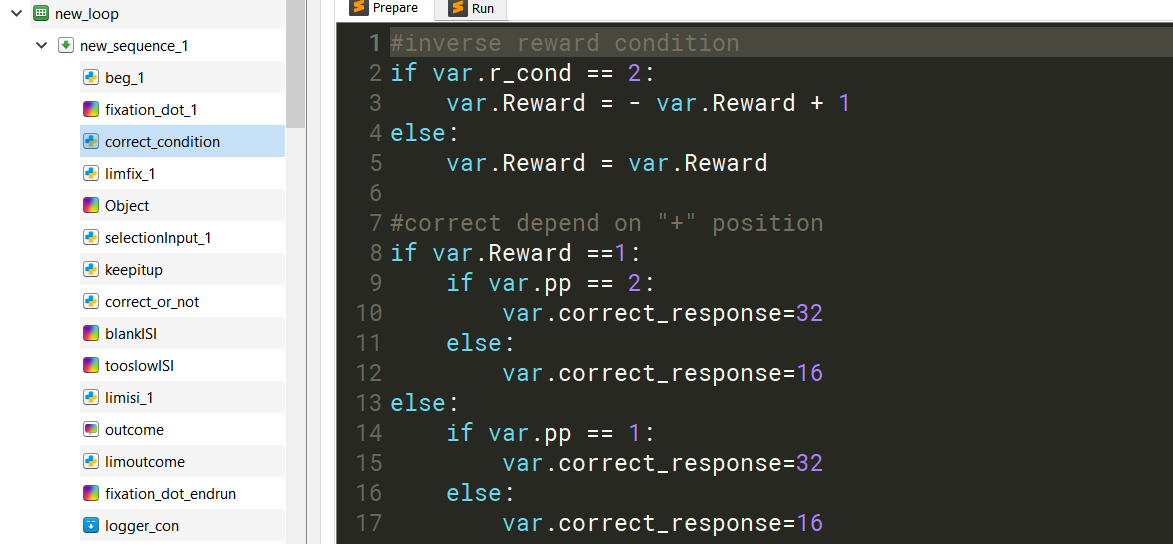
### Higher structure



* **instructionsP2**: as instructionsP1, displays instructions depending on subject condition
* **money**: initialize variables used to give the proper feedback to predictions (var.feedbackc), to tracks number of correct predictions of reward (var.money), of misses (var.miss), and of false alarm in the conditioning phase (var.penalty) and inference phase (var.penalty2)
* **conditioning**: empty loop object that allows to rerun the following sequence 4 times (one for each run)
* **reseq**: sequence of the next elements, replayed before each run. Each time it is repeated it plays one (and only one) different run because of the conditional statements as in Preconditioning.
* **counter1**: count runs but actually not used I believe, could probably be removed
* See 2.1 Preconditioning for the following:
  + **run\_nb\_init\_1**
  + **run\_nb\_incr\_1:**
  + **new\_sketchpad:**
  + **portReadChoice:**
  + **achtung:**
  + **fixdot:**
  + **new\_loop\_X:**

### Block structure

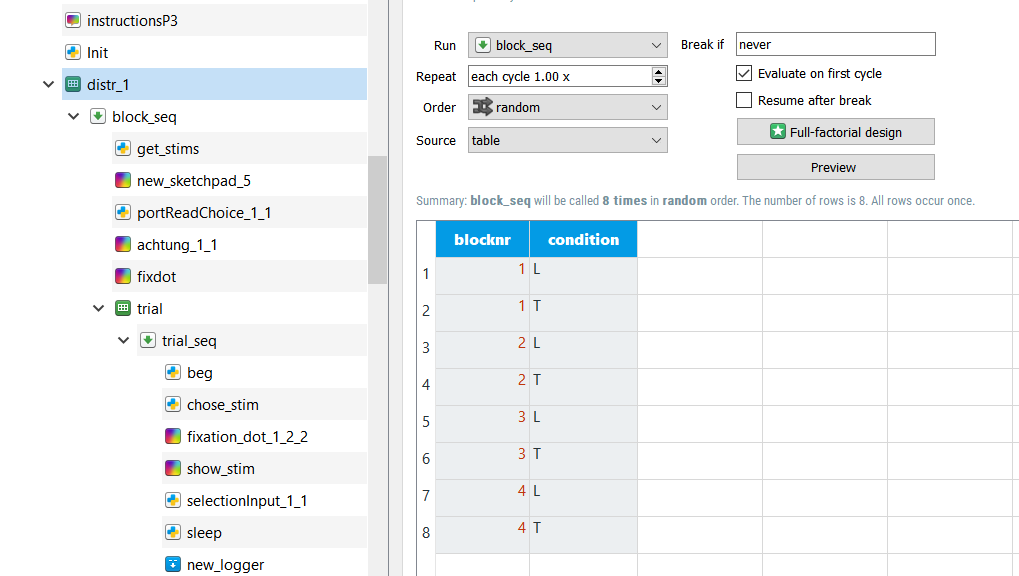
As for preconditioning, only the first block is explained here, as all blocks share the same sequence, only the table containing the parameters change



* **new\_loop**: loop that contains the parameters of the 20 trials
  + stim2: object filename
  + paired: 1 or 0
  + Reward: 1 or 0
  + catnb: ignore, just leftover from neurodesignscript, not reliable
  + dur: ITI duration in seconds
  + ISI: ISI duration in seconds
* **new\_sequence\_1** : sequence that contains all the trials elements
* **beg**\_1: See 2.1 Preconditioning
* **fixation\_dot\_1**: See 2.1 Preconditioning
* **correct\_condition**: script that assigns the correct answer expected (var.correct\_response), depending on the subject conditions (rewards as in list or reversed; left or right for reward prediction) and of course the trial parameters (rewarded or not),
* **lim\_fix**: See 2.1 Preconditioning
* **Object**: sketchpad that shows the object image, duration 0
* **selectionInput\_1**: scripts that reads port to see if there is a trigger from the button box, and assigns the correct values to response and RT.
* **keep\_it\_up**: script that keeps the previous sketchpad up until the stim duration has passed
* **correct\_or\_not**: script that assigns value to var.miss, var.money, var.penalty, and var.feedbackc depending on the participant’s response and the condition.
* **blankISI**: blank sketchpad, duration 0; played only if the subject answered
* **tooslowISI**: blank sketchpad, duration 0; played only if the subject DIDN’T answer
* **limisi\_1**: script that keeps the previous sketchpad up until the ISI duration has passed
* **outcome**: feedback that shows the correct outcome using var.feedbackc (coin, scrambled coin…), duration 0
* **limoutcome**: script that keeps the previous feedback up until the stim duration has passed.
* **fixation\_dot\_endrun**: interrun fixation dot, only runs at the end of each run, to capture a response in its entirety
* **logger\_con**: create a new line in the log file and write the current value of each variable

## Phase 3: Distractor/localizer

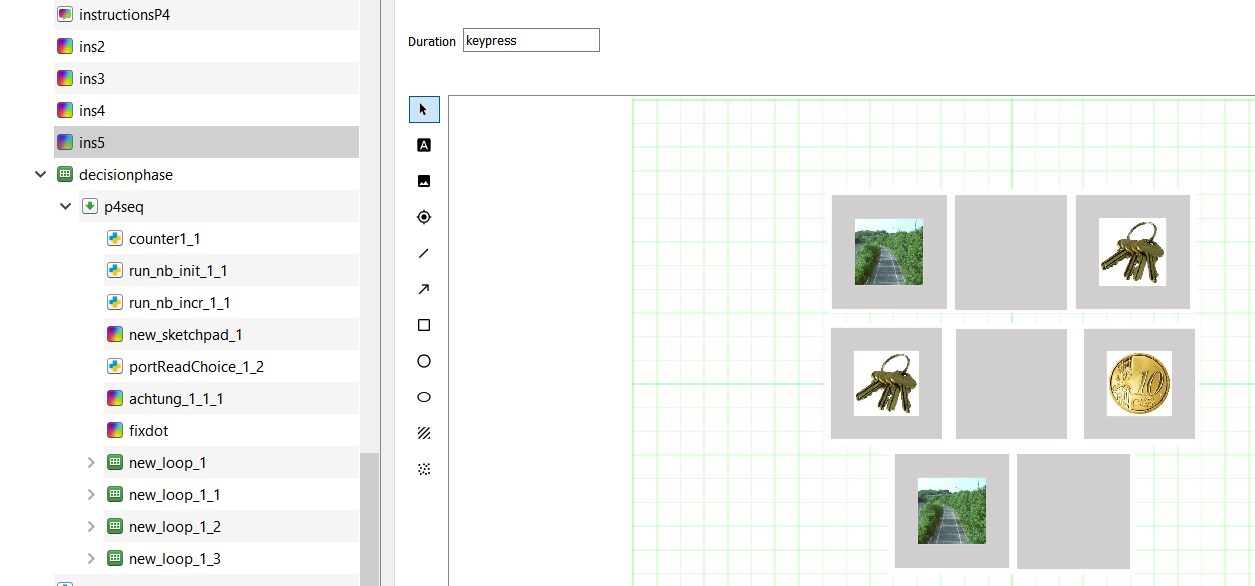
Here the organisation is different, this is from Elias.



* **instructionsP3**: as instructionsP1, displays instructions depending on subject condition
* **Init**: script that creates pseudorandom lists of 10 images of the same type (object [5 brooms and 5 buckets] or scene [5 insides and 5 outsides]) used in the following blocks. The lists are named L1 to L4 and T1 to T4 (for Location and Tool)
* **distr\_1**: loop that contains the 8 different blocks, the list corresponding to the letter and number is used in each block
* **block seq**: sequence of a block
* **get\_stims**: scripts that get the stimulus list corresponding to the block and add 3 repetitions randomly
* **newsketchpad\_5**: blank sketchpad of duration 0, displayed while the next script runs
* **portReadChoice\_1\_1**: script that reads a port and waits for 3 scanner triggers to go the next object in the sequence. ONLY RUNS IN THE FIRST BLOCK.
* **achtung:** 500ms
* **fixdot:** 500ms
* **trial:** loop that contains in the table only a list of “trialnr” from 1 to 13 to indicate 13 repetitions of the next sequence
* **trial\_seq:** trial sequence to be repeated 13 times
* **beg:** See 2.1 Preconditioning
* **chose\_stim:** use var.trialnr to extract the element of the list corresponding and assign it to var.image
* **fixationdot\_1\_2\_2:** fixation cross, duration 300ms
* **show\_stim:** sketchpad that shows the image which filename was in var.image, duration 0
* **selectionInput\_1\_1:** See 2.1 Preconditioning
* **sleep:** script that keeps the previous sketchpad up until the stim duration has passed.
* **new\_logger:** See 2.1 Preconditioning.

## Phase 4: Inference

### Higher structure

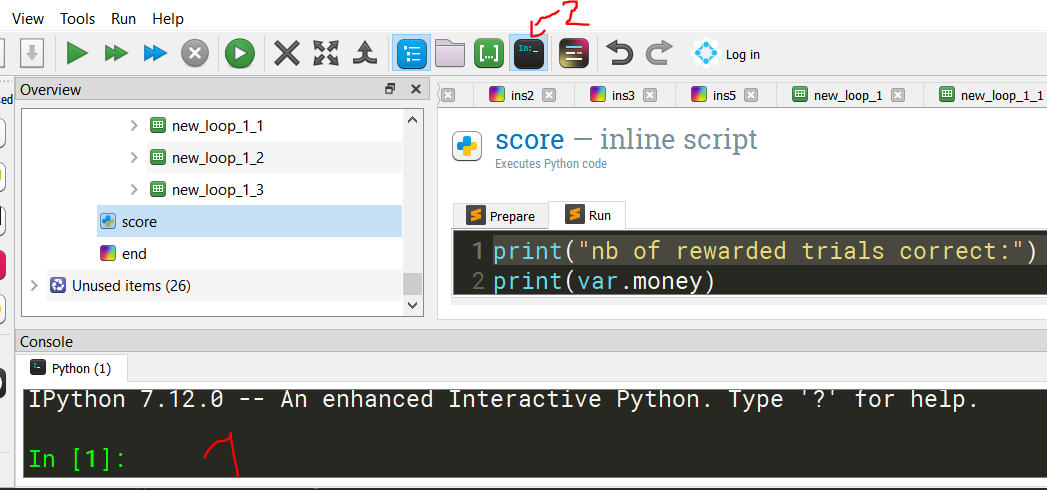


* **instructionsP4**: as instructionsP1, displays instructions depending on subject condition
* **ins2 to ins5**: make the 3 images lines shown above appear one by one, can be removed.
* **decision phase:** empty loop object that allows to rerun the following sequence 4 times (one for each run)
* **p4seq**: sequence of the next elements, replayed before each run. Each time it is repeated it plays one (and only one) different run because of the conditional statements as in Preconditioning.
* See 2.1 Preconditioning and 2.2 conditioning for the following:
  + **counter1**:
  + **run\_nb\_init\_1**
  + **run\_nb\_incr\_1:**
  + **new\_sketchpad:**
  + **portReadChoice:**
  + **achtung:**
  + **fixdot:**
  + **new\_loop\_X:**

### Block structure

* **new\_loop\_1**: loop that contains the parameters of the 20 trials
  + stim1: scene or object filename
  + paired: 1 or 0
  + Reward: 1 or 0
  + catnb: ignore, just leftover from neurodesignscript, not reliable
  + dur: ITI duration in seconds
  + doesn’t contain ISI, fixed to always 1 in the next seq
* **new\_sequence\_1\_2** : sequence that contains all the trials elements
* **beg**\_2: See 2.1 Preconditioning
* **fixation\_dot\_1\_1**: See 2.1 Preconditioning
* **correct\_condition**: script that assigns the correct answer expected (var.correct\_response), depending on the subject conditions (rewards as in list or reversed; left or right for reward prediction) and of course the trial parameters (rewarded or not),
* **lim\_fix**: See 2.1 Preconditioning
* **cond\_3**: sketchpad that shows the image, duration 0
* **selectionInput\_1**: scripts that reads port to see if there is a trigger from the button box, and assigns the correct values to response and RT.
* **keep\_it\_up**: script that keeps the previous sketchpad up until the stim duration has passed
* **counter2\_2**: script that assigns value to var.miss, var.money, and var.penalty2 depending on the participant’s response and the condition
* **isiscreen**: blank sketchpad, duration 0
* **limisi\_2**: script that keeps the previous sketchpad up until the ISI duration has passed, always 1s here
* **fixation\_dot\_1\_3\_1**: interrun fixation dot, only runs at the end of each run, to capture a response in its entirety
* **logger\_con\_1**: create a new line in the log file and write the current value of each variable

## End



* score: scripts that prints the number of correctly guessed rewarded trials in P2 and P4, and the number of false alarm P2, in the (1) area. To see the (1) area, one must first click on 2
* end: sketchpad that displays “end of experiment” for 5s, can be removed.

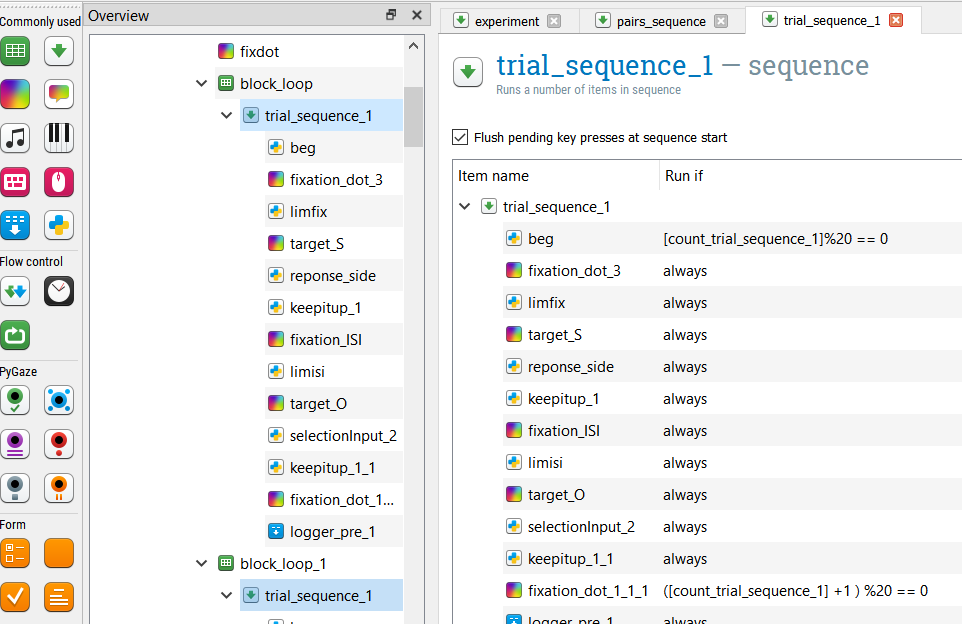
# Object types

There are different object types in open sesame, to add one to your experiment, just click on the corresponding icon on the left and drag it to the position you want it to be

### Sequence

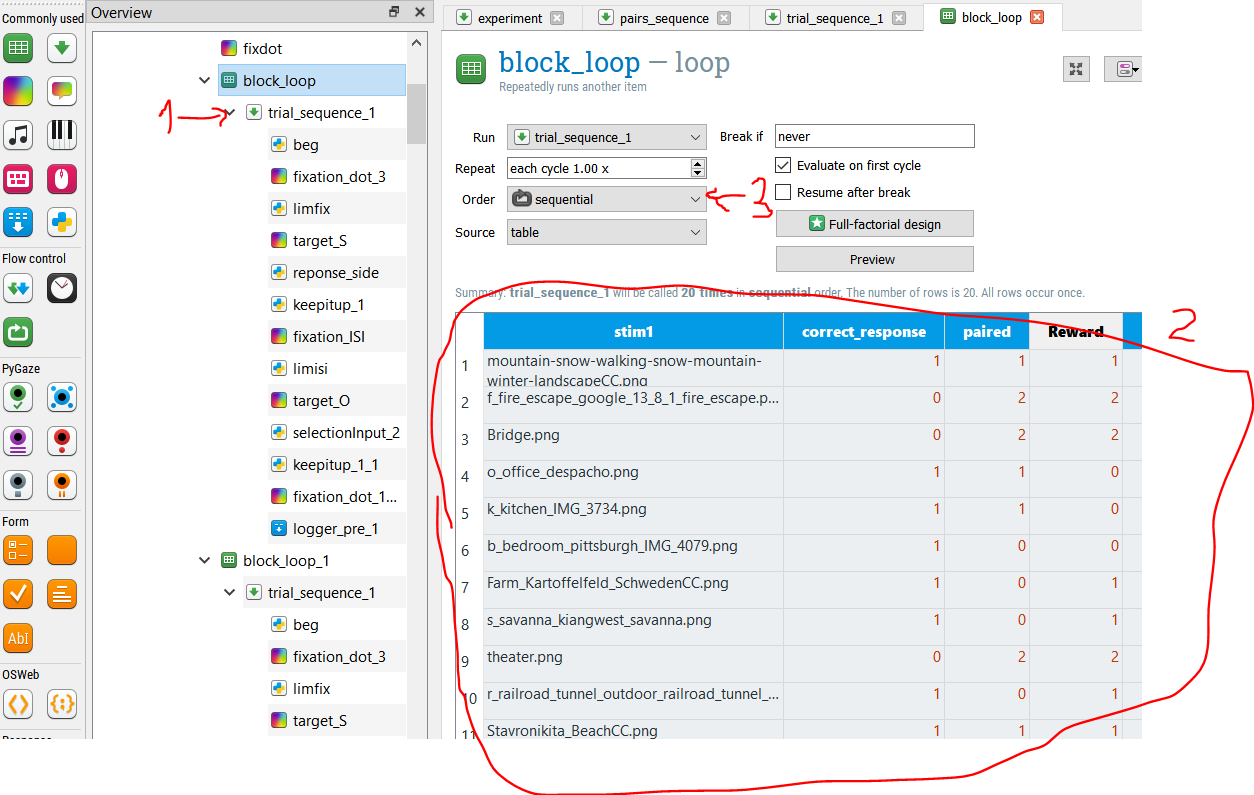
Basically just the structure, but one can add conditions to run elements.

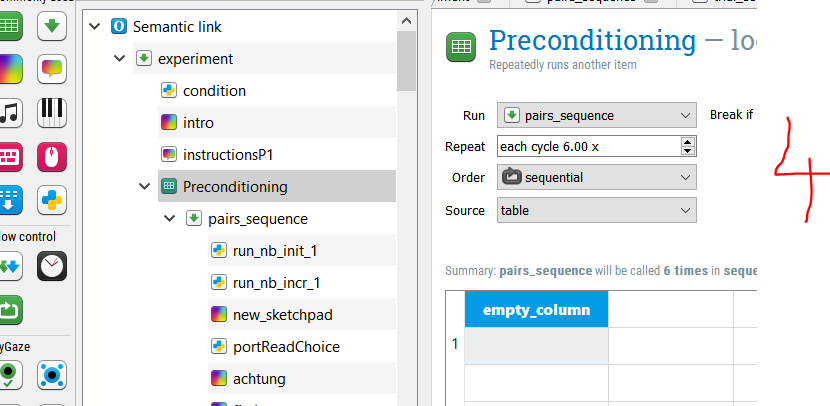
In the example below, **beg** will be run at the beginning of each run (ie when the sequence has been repeated a multiple of 20 starting at 0 : 0,19,39,…), and **fixation\_dot\_1\_1\_1** at the end of each run (ie when the sequence has been repeated a multiple of 20 starting at 1 : 1,20,40,…)



### Loop

A loop repeats the sequence that is into it (1) , for each trial in the table (2), taking the values in the line for the trial. You can choose the order of the trials to be as in the table (sequential), or random (3). You can also choose to repeat a block several time, in (4), it is done so that the waiting for the scanner trigger is repeated each time.





### Sketchpad and Feedback

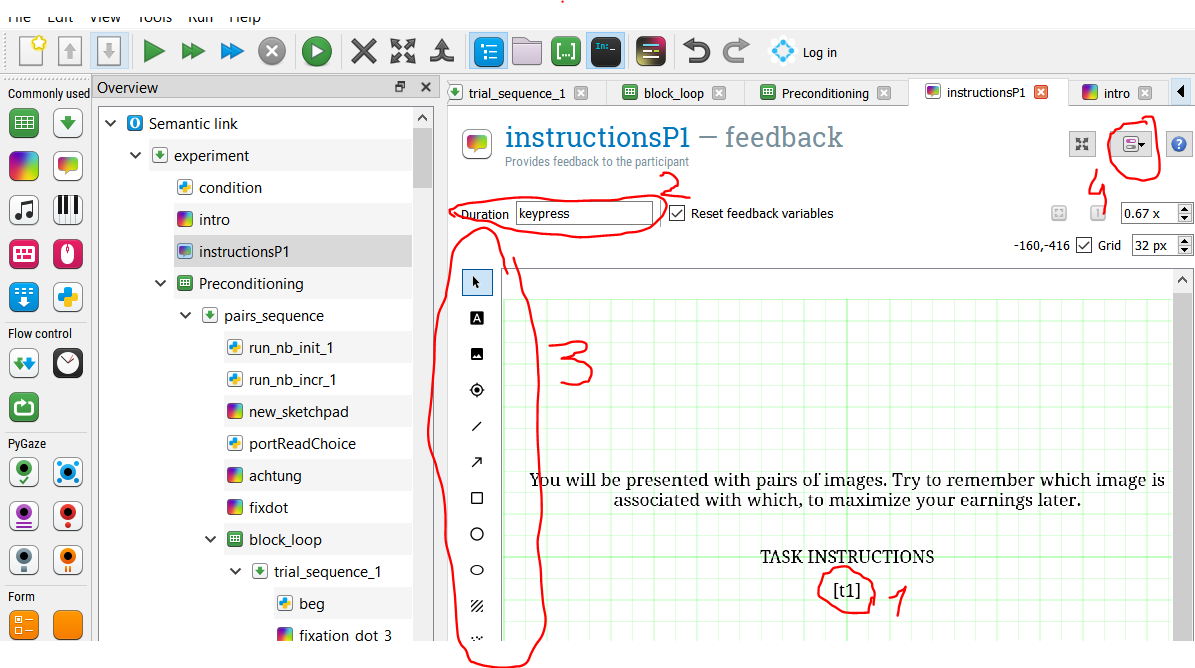
Sketchpad and Feedback objects allow to display text or images. However, sketchpads are loaded in advance, so they cannot be changed by a variable recently set. That is why some instructions are on feedback, because they depend on the subject number.

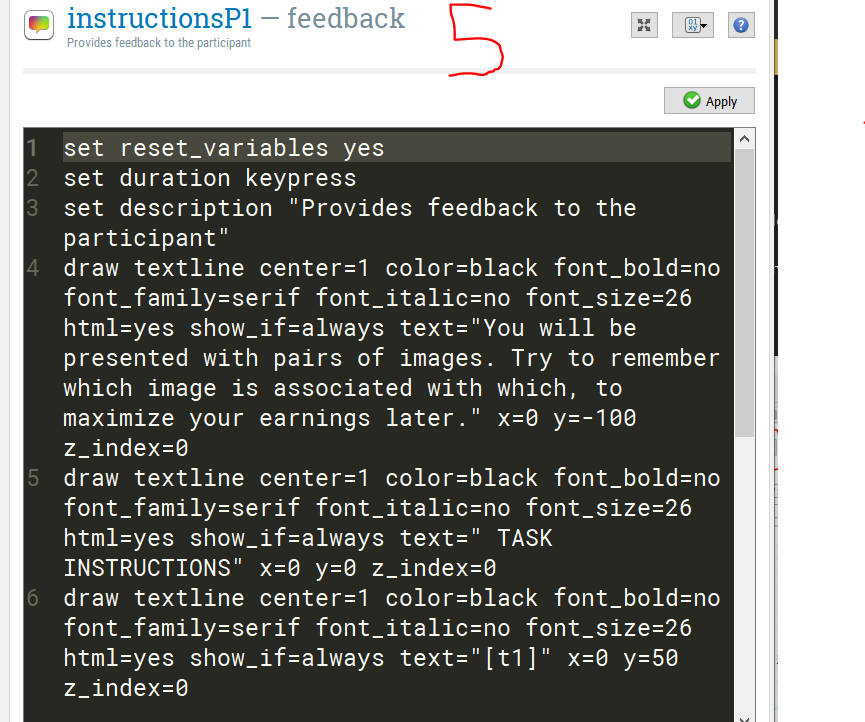
In the example below, the variable t1 (1) is created in the condition script, 2 objects above.

The duration of the sketchpad/feedback can be determined in several manners: either a keypress, with no timeout like in the example below (2). Alternatively, the duration should be set to zero, and the next object will control the time it is displayed. The object can be either a **keyboard\_response** object (where one can set a timeout, and/or the correct\_response if it is not done elsewhere), or a **script** waiting for a duration (predetermined or eg depending on the number of scanner triggers read) and/or a button press.

To edit a sketchpad or feedback, you there is two ways. First, you can click on the object you would like to add on the left (3), and then clock on the grid where you want it: this will open a window where you can write your text or choose your image. Alternatively, you should clock on the button (4), and then click, on “View script” in the dropdown menu.

On the script (5), you can edit the text, the position, the font and font size, etc more in details. Once you are done, you should click on apply on the upper right and you will see the changes.



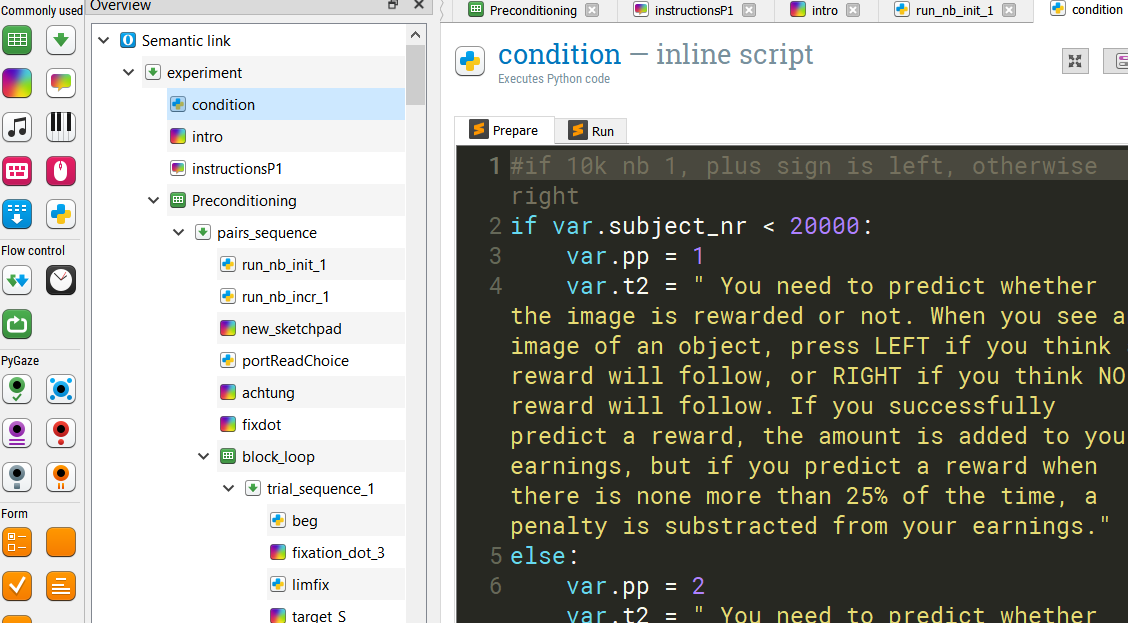


### Script

A script object is used to write in python, there can be different reasons for that. Create a variable, create a random number, read a port to wait for a trigger, etc.

If you want to use an existing variable (eg response) you need to write var.response. Similarly, if you want to create a variable to be in the logfile, and accessible elsewhere in the sequence, you should write var.variable\_name when you use it

In the example below **condition** assigns different values to newly created variables depending on the subject number:



### Keyboard\_response

Only used in behavioral experiments outside of scanner, replace the script that waits for a button box trigger. You can define the correct response here, but assigning a value to the variable correct\_response has the same effect.

You can put a time out, and thus it keeps the previous sketchpad (of duration 0) up until a key is pressed or the timeout has passed.

