# How To: Randomise Conditions (2 methods)

If you have more than one condition and you want to present them independently you may want to randomise the blocks. You will need to use several conditions files to complete this task.

### Case 1 Example

You have three blocks where the stimuli are the same, they just need to behave differently e.g they all show image stimuli, but they are different image categories.

You are examining the difference in recognition between faces, buildings and abstract patterns. Each trial should look the same, you want to present a fixation, a stimulus and a target response. The images of faces/buildings/patterns should be the same size and on screen for the same duration. You could simply create three separate routines for each condition, however this is not the most efficient approach.

#### Why not?

1. You are essentially making duplications of the same trial
2. Your data file will be a mess with many extra variables, thus harder to analyse

Conditions

Faces/Building/Pattern

+

250ms

stimPres

250ms

Target

keypress

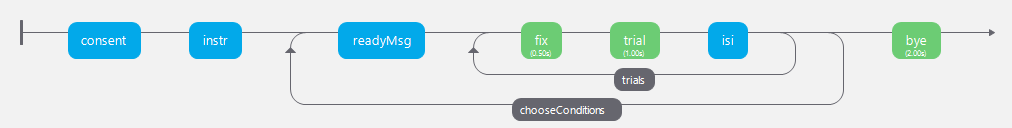
#### Caveat

Randomising in this way is not always necessary and not always useful, as conditions can be randomised within a single conditions file using the Loop (interleaved). Using this entirely depends upon your study design. The disadvantage of this technique is that you cannot control the order of randomisation as PsychoPy will randomly choose the order of presentation. If you want to counterbalance the presentation order see the How to: Counterbalance document.

### How to randomise blocks: Case 1

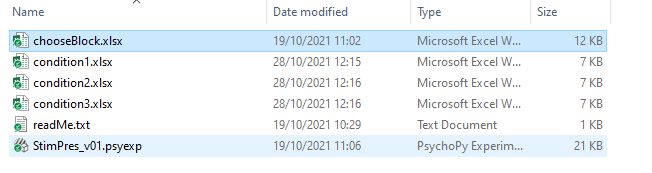
There is a template available for you to use, but you must appreciate some of the complexities to apply this to your trials/blocks

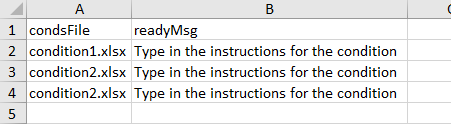
Here is the flow of the template:



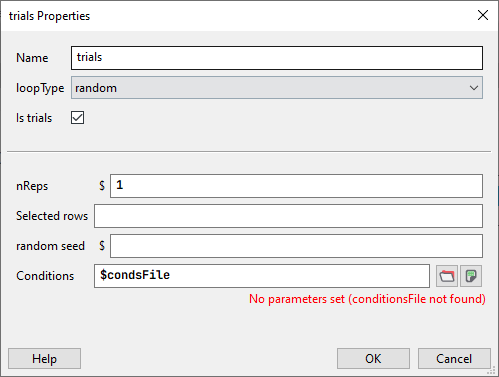
You can see in the flow above that we have 1 trial set with a fixation, the trial and an inter stimulus interval. The trials have a loop around them (**trials**), the instructions for each condition block (**readyMsg**) are included in a second loop called **chooseConditions**

When this program runs, the readyMsg will change dynamically depending upon the condition chosen to run. This is important if you have different instructions for the conditions.

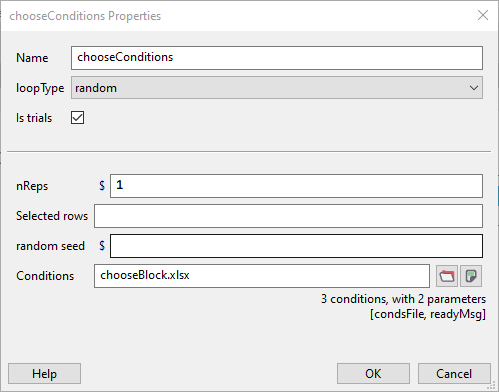
1. Create individual conditions files for each condition, your folder may look something like this (it makes sense to name the conditions files with something that is more useful/recognisable than 1, 2, &3 for data analysis purposes)
2. Each conds file should have the EXACT same column headings. Populate the conditions files as necessary with the path to stimuli/words/numbers/correct responses etc.
3. Create a new conditions file in Excel called **chooseBlock**
4. Call the first column **condsFile** and type in the file path for the different conditions that you are applying



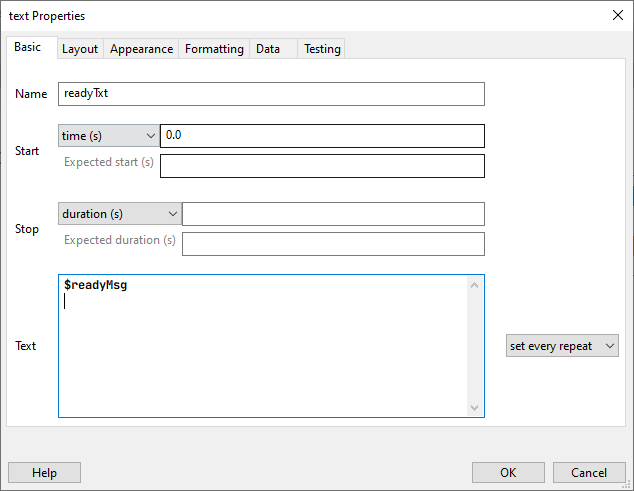
1. In the next column, name it **readyMsg** (if you require more than one text component on the readyMsg routine you can suffix readyMsg with numbers)
2. Place a **Loop** around the trials and in the **Conditions** box type in **$condsFile and** you can ignore the red error message here. You can change other parameters such as the number of repetitions (nReps) as per usual here too



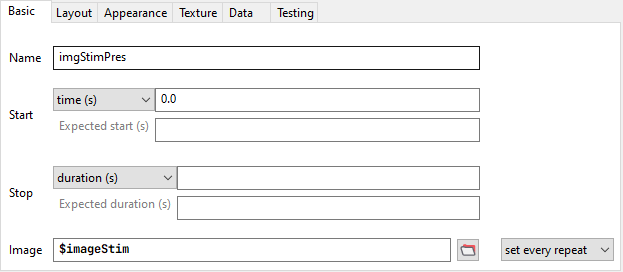
1. Place a second **Loop** around the **readyMsg routine** and the **trials Loop** (as per the Flow above) and in the **Conditions** box navigate to your **chooseBlocks.xlsx** file



1. In the **readyMsg routine** you will require a **Text** component to display the individual instructions for each of the conditions. You should have typed these into the **chooseBlocks** conditions file.
2. Now you need to click in the **text properties** and change the text in the Text box to **$readyMsg** and select **set every repeat,** which will pull the information from the chooseBlock file that corresponds with the condition to be displayed



1. Similarly, in the components that you have in the trial, you need to reference the column/s in the individual conditions file e.g **imgStimPres** component **$imageStim** from conditions file and in **imgStimResp** keyboard component **$corrAns** to collect the correct response data



### Case 2 Example

You have two blocks where the stimuli are totally different. e.g one block consists of audio presentation, and the other is text on screen.

You are examining the difference in digit recall with two presentation modalities. The visualisation below shows the differences in presentation duration, and that the modalities require different component types

+

audioPres

duration of stimulus determined by the spoken length of digits

Response

Keypress

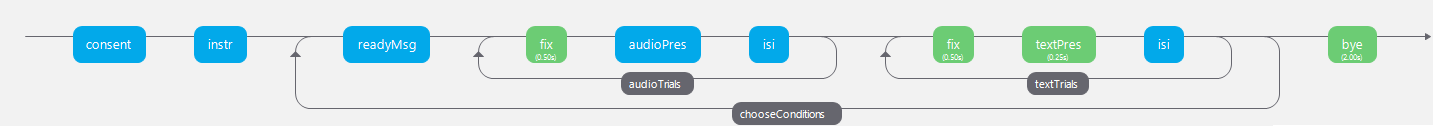
+

textPres

300ms

Response

Keypress

You can see in the flow below that we have 2 trial blocks each with a fixation, the trial and an inter stimulus interval, but the component in the audioPres routine is a Sound comp[onent, as opposed to a text component for the visually presented trial block. The blocks have a loop around them (**audioTrials** and **textTrials**), the instructions for each condition block (**readyMsg**) are included in a second loop called **chooseConditions**

The process is the same as for the example above, but this represent the idea that the trial blocks need to behave differently, where the trial blocks in Case 1 needed to behave the same.