

# Feedback syntax

---



[main website](http://www.psychtoolkit.org) (<http://www.psychtoolkit.org>) | [documentation 2.0.6](http://www.psychtoolkit.org/doc2.0.6) (<http://www.psychtoolkit.org/doc2.0.6>)

---

## Background

What you need to be able to give feedback

## Syntax

text

text attributes

set

mean

min

max

sum

count

perc

Just set a variable to the value of another variable

Show calculated values to participant

Set in summary

select

lineplot and xyplot

wait\_for\_key

Full example of a task with feedback

---

Feedback will give a participant feedback about performance, for example about the average response times in a block. Feedback can be programmed in a "block" section only (and not on a trial by trial basis).



Feedback is not for PsyToolkit beginners. Feedback is an advanced feature of PsyToolkit experiment coding, and you should only use this once you have mastered the programming of more basic experiment

aspects. There are good examples of it in the downloadable experiments in the [experiment library](http://www.psychtoolkit.org/experiment-library/) (<http://www.psychtoolkit.org/experiment-library/>).

## Background

Researchers might want to give participants feedback about their performance. Here are some reasons for feedback:

- Feedback can motivate participants to respond more accurately or faster
- Researchers might want to use the feedback when designing the task themselves (yet not give it in the end to the participants)

## What you need to be able to give feedback

In order to use feedback for participants, you will need to make sure you have thought about the following:

- Which font would you want to use to present the feedback information in? Feedback will use the first specified font.
- Which data do you want to give feedback about? You need to make sure you have saved this data using the "save" instruction in the task description.
- What sort of feedback do you want to give? Most common is just to tell the user the number of mistakes they made or the average response speed.

## Syntax



The feedback instructions are given within the "block" statement. In the examples below, the feedback part of blocks will be shown, though. There is one full example at the bottom of this page. Look for complete examples in the [experiment library](http://www.psychtoolkit.org/experiment-library/) (<http://www.psychtoolkit.org/experiment-library/>).

There are currently 4 main components to give feedback. It is probably good to look first at the example below to get a quick impression (coding is often learned most quickly from looking at examples).

- set
- text
- lineplot
- xyplot

These are described in more detail below.

## text

Text allows you to present variable values (the "set" command allows you to calculate averages, min, max, and percentages of participant responses), and more (each function described below).



Quotes around literal text in feedback is currently not required, although recommended. In the task **show text** commands, quotes **must** be used, though.

### *Example of the text*

```
feedback
  text 0 -150 "Press space bar to continue"
end
```

### *Example of the text*

```
feedback
  set &MyVar perc 10 ; select c4 == 1
  text 0 -150 &MyVar ; prefix "Percentage errors (should really be below 10%):" ;
  postfix "%"
end
```

You need to refer to the column of the datafile (in the example about we select the 4th column, the prefix c indicates the column. These are column of the datafile and not of the table!

The select keyword only works with numbers, not with names. Thus, if you want to select only the trials of a certain block, don't check that with



its name but with its number. You can use BLOCKNUMBER in the **save** line. Also, note that a standalone message increases the BLOCKNUMBER counter.

## text attributes

You can align the text to the left if needed. You can use the same text attributes in experiments. For example:

### *Example of the text*

```
feedback
  text color yellow
  text align left
  text 0 0 "Line 1"
  text 0 100 "Line 2 with some more text"
end
```

## set

You can set global variables (use the & sign). These are most likely variables you have not used in the tasks or blocks, although you can show those here too!

There are a number of functions you can use, each is explained with an example below. The "select" part of a "set" line in the feedback section allows you to only use the trials of certain lines (see below for further explanation)

## mean

Calculates the mean.

### *Example 1 of mean*

```
feedback
  set &MyVar mean c6 ; select c4 == 1
  text 0 -150 &MyVar ; prefix "Mean response time:" ; postfix "ms."
end
```

In the above example, imagine that column 6 of the data contains the reaction time and column 4 the status (with value 1 being correct, as is the default in PsyToolkit). In this case, &MyVar will be set to the mean response time, but only for those trials in which no error was made (because only the trials for which the fourth column is 1 are considered).

## min

Calculates the lowest value

### *Example 1 of min*

```
feedback
  set &MyVar min c6 ; select c4 == 1
  text 0 -150 &MyVar ; prefix "Shortest response time:" ; postfix "ms."
end
```

In the above example, imagine that column 6 of the data contains the reaction time and column 4 the status (with value 1 being correct, as is the default in PsyToolkit). In this case, &MyVar will be set to the minimum (i.e., shortest) response time, but only for those trials in which no error was made (because only the trials for which the fourth column is 1 are considered).

## max

Calculates the highest value

### *Example 1 of max*

```
feedback
  set &MyVar max c6 ; select c4 == 1
  text 0 -150 &MyVar ; prefix "Longest response time:" ; postfix "ms."
end
```

In the above example, imagine that column 6 of the data contains the reaction time and column 4 the status (with value 1 being correct, as is the default in PsyToolkit). In this case, &MyVar will be set to the maximum (i.e., longest) response time, but only for those trials in which no error was made (because only the trials for which the fourth column is 1 are considered).

## sum

Calculates the sum of all values.

### *Example 1 of **sum***

```
feedback
  set &MyVar sum c6 ; select c4 == 1
  text 0 -150 &MyVar ; prefix "Total response time:" ; postfix "ms."
end
```

In the above example, imagine that column 6 of the data contains the reaction time and column 4 the status (with value 1 being correct, as is the default in PsyToolkit). In this case, &MyVar will be set to the total sum of all response times (across all trials), but only for those trials in which no error was made (because only the trials for which the fourth column is 1 are considered). Of course, this would rarely be used, but sum can be a handy function for further calculations.

## count

Calculates the number of trials.

### *Example 1 of **count***

```
feedback
  set &MyVar count ; select c4 != 1
  text 0 -150 &MyVar ; prefix "Number of errors:"
end
```

In the above example, imagine that column 4 contains the status (with value 1 being correct, as is the default in PsyToolkit). In this case, &MyVar will be set to the number of errors (only trials for which the fourth column is not correct are counted).

## perc

This calculates the percentage (i.e., value from 0 to 100) of occurrence. This only makes sense in combination with the **select** option in the same line (see below). By default, the percentage is taken of all trials in the dataset so far, unless a specific number is given (this must be an integer value, it cannot be a variable).

The example below calculates the percentage of trials for which the value in column 4 equals 1. Imagine there are a total of 100 trials, and in 90 of those, the fourth column contains a 1, and in 10 of those a 2. In that case, MyVar would contain 90, based on  $90/100*100$ .

### *Example 1 of perc*

```
feedback
  set &MyVar perc ; select c4 == 1
  text 0 -150 &MyVar ; prefix "Percentage errors:" ; postfix "%"
end
```

The example below calculates the percentage of trials for which the value in column 4 equals 1. Imagine there are a total of 100 trials, and in 90 of those, the fourth column contains a 1, and in 10 of those a 2. In the example, the total number of trials is ignored, and instead the value 200 is used in the percentage calculation. Thus in this case, MyVar would contain 45, based on  $90/200*100$ . This is often useful when you want to calculate a percentage for a subset of trials of which you know the number of trials.

### *Example 2 of perc*

```
feedback
  set &MyVar perc 200 ; select c4 == 1
  text 0 -150 &MyVar ; prefix "Percentage errors:" ; postfix "%"
end
```

## Just set a variable to the value of another variable

### *Example of setting a variable to the value of another variable*

```
feedback
  set &current_variable &old_variable
end
```

Or you can do this after the feedback:

### *Example of setting a variable to the value of another variable*

```
feedback
  ...code...
end
set &current_variable &old_variable
```

## Show calculated values to participant

### *Example of set and text*

```
feedback
  set &StroopCompatible mean c5 ; select c6 == 1
  set &StroopIncompatible mean c5 ; select c6 == 2
  set &MyStroopEffect expression &StroopIncompatible - &StroopCompatible
  text 0 -150 &MyStroopEffect ; prefix "Your Stroop compatibility effect:" ;
  postfix " ms"
end
```

## Set in summary

In short, there are two types of "set" in the feedback:

- 1) The set with an expression to calculate new values to be shown. This is handy to show the difference between conditions like in the example above.
- 2) Set a global variable to the average, minimum, maximum, or percentage of errors of the value of a column in your datafile. Imagine that the 5th column of your datafile contains the response times, then you can do the following:

### *Example of set*

```
feedback
  set &MyAverage mean c5
  set &MyAverage min c5
  set &MyAverage max c5
  set &MyAverage perc ; select c6 == 1
  set &MyAverage perc 100 ; select c6 == 1
  set &MyAverage &some_other_variable
  set &MyAverage expression &MyAverage * 100
  set &MyAverage expression 100
end
```

Now imagine that column 6 contains the status, with 1 being correct, and anything else incorrect. Also imagine there are 200 trials in total. Now you can look for the percentage of correct trials, and you can calculate for correct trials only too:

### *Example of set*



feedback

```
set &MyAverage mean c5 ; select c6 == 1
set &MySlowest min c5 ; select c6 == 1
set &MyFastest max c5 ; select c6 == 1
set &MyErrorRate perc ; select c6 != 1
text 0 -50 "Some feedback about your performance:"
text 0 0 &MyAverage ; prefix "Average response time" ; postfix "ms"
text 0 50 &MySlowest ; prefix "Slowest response time" ; postfix "ms"
text 0 100 &MySlowest ; prefix "Slowest response time" ; postfix "ms"
text 0 150 &MyErrorRate ; prefix "Error percentage" ; postfix "ms"
```

end



The feedback "perc" function can take an additional number, or you can leave that out. If you give the additional number, it will use that as the total, rather than the total number of trials. There are situations where this is useful.

## select

The select part of a set line allows you to only select the rows of your datafile you are interested in. This select statement uses its own syntax. Every select statement starts obviously with "select", followed by a number of comparasions. You can use && for logical and || for logical or.

You can ask if the values of a specific column in your datafile (as created by your "save" statements) have a specific numeric value, or the value of a global variable, or the BLOCKNUMBER.



If you only want to select the trials of the last block you can do this as follows. First of all, determine which line of your datafile represents the BLOCKNUMBER.

Imagine that the 5th column has the RT, the sixth column the STATUS, and the first column the BLOCKNUMBER. Then you can easily use the following to only take the averages in the block you are giving feedback over:

*Example of set with values only on the last block*

feedback

```
set &MyAverage mean c5 ; select c6 == 1 && c1 == BLOCKNUMBER
set &MySlowest min c5 ; select c6 == 1 && c1 == BLOCKNUMBER
set &MyFastest max c5 ; select c6 == 1 && c1 == BLOCKNUMBER
set &MyErrorRate perc ; select c6 != 1 && c1 == BLOCKNUMBER
text 0 -50 "Some feedback about your performance:"
text 0 0 &MyAverage ; prefix "Average response time" ; postfix "ms"
text 0 50 &MySlowest ; prefix "Slowest response time" ; postfix "ms"
text 0 100 &MySlowest ; prefix "Slowest response time" ; postfix "ms"
text 0 150 &MyErrorRate ; prefix "Error percentage" ; postfix "ms"
```

end

## lineplot and xyplot

The line and xyplot allows you to present the data points as a line plot on the screen. Currently, no examples available yet.



The semi-colon to separate sections of feedback commands **must** be surrounded by spaces (on both sides). Without this, it will be ignored.

## wait\_for\_key

By the default, the feedback stays on the screen until the space bar is being pressed. There is **no** need to include a "wait\_for\_key" within the feedback section. But if you want that a different key than space is being used, you can specify this with "wait\_for\_key".



This can be useful if you use an external keyboard, like the BlackBox keyboard.

*Example of set with values only on the last block*

feedback

```

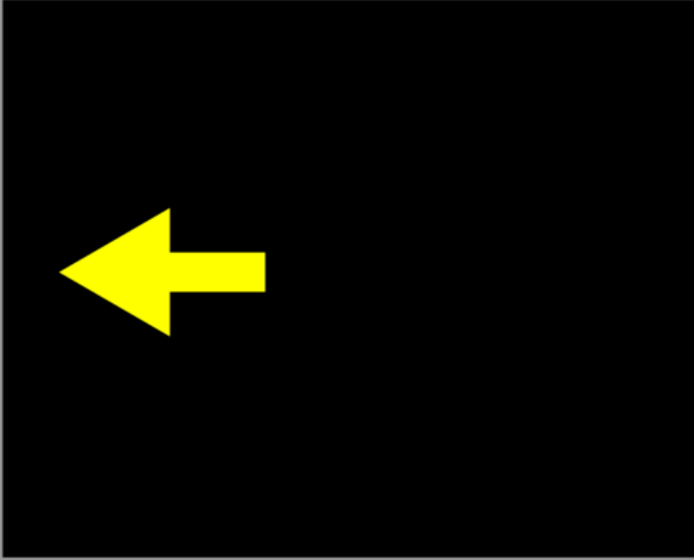
set &MyAverage mean c5 ; select c6 == 1 && c1 == BLOCKNUMBER
set &MySlowest min c5 ; select c6 == 1 && c1 == BLOCKNUMBER
set &MyFastest max c5 ; select c6 == 1 && c1 == BLOCKNUMBER
set &MyErrorRate perc ; select c6 != 1 && c1 == BLOCKNUMBER
text 0 -50 "Some feedback about your performance:"
text 0 0 &MyAverage ; prefix "Average response time" ; postfix "ms"
text 0 50 &MySlowest ; prefix "Slowest response time" ; postfix "ms"
text 0 100 &MySlowest ; prefix "Slowest response time" ; postfix "ms"
text 0 150 &MyErrorRate ; prefix "Error percentage" ; postfix "ms"
wait_for_key d
end

```

## Full example of a task with feedback



In this example participants respond with the "a" or "l" key to a left or right pointing arrow (a trial with a left-pointing arrow on the left in the image below). The feedback tells the participant how fast they were on average in correct trials, and what their error percentage was (example on the right in the image below).

Screen during experiment. Participant responds to left and right pointing arrows	This is what the screen looks like when participant gets feedback.
	<p>Error percentage and response times in training block (in ms = milliseconds = there are 1000 ms in one second)</p> <p>Percentage errors (should really be below 10%): 10%</p> <p>Average speed of ALL CORRECT RESPONSES: 436ms</p> <p>This was just training. Now press space bar of your keyboard with the REAL data collection (will take about 5 minutes)</p>



Feedback scripts look relatively cryptic at first, but they are very straightforward.

*Example of how to use "feedback" in PsyToolkit script*

```

# the following is an example of a Simon task with feedback about
# error rates and response times

bitmaps
    leftarrow
    rightarrow
    fixpoint

fonts
    arial arial.ttf 18

table simontasktable
    "L_pos L_arrow" -200 1 leftarrow 1
    "R_pos R_arrow" 200 2 rightarrow 2
    "R_pos L_arrow" 200 3 leftarrow 1
    "L_pos R_arrow" -200 4 rightarrow 2

task simon
    table simontasktable
    keys a l
    show bitmap fixpoint
    delay 100
    show bitmap @4 @2 0
    readkey @5 2000
    delay 5000
    save BLOCKNAME @1 @3 TABLEROW KEY STATUS RT

block mytrainingblock
    tasklist
        simon 20
    end
    feedback
        set &MyPerc perc      ; select c7 != 1
        set &MyMean mean c8   ; select c7 == 1
        set &MyCount count    ; select c7 == 1
        text 0 -250 "Error percentage and response times in training block"
        text 0 -200 "(in ms = milliseconds = there are 1000 ms in one second)"
        text 0 -150 &MyPerc ; prefix "Percentage errors (should really be below
10%):" ; postfix "%"
        text 0 -100 &MyMean ; prefix "Average speed of ALL CORRECT RESPONSES:" ;
postfix "ms"
        text 0 -50 &MyCount ; prefix "Total number of correct trials:"
        text 0 200 "This was just training. Now press space bar of your keyboard"
        text 0 250 "with the REAL data collection (will take about 5 minutes)"
    end

```

Last updated 2016-01-09 13:49:56 GMT