

# Lesim-mode: Edit Learning Simulator scripts in Emacs

Stefano Ghirlanda

May 29, 2023

Lesim-mode is an Emacs major mode to write and run [Learning Simulator](#) scripts. To learn more about the Learning Simulator, see this [short article](#) and the [online documentation](#).

Lesim-mode helps to ensure that a script is correct by highlighting common errors, like misspelled keywords and stimulus elements.<sup>1</sup>

You can run Learning Simulator scripts directly from lesim-mode buffers. Errors are reported in the minibuffer and highlighted in the script buffer.

For integration with org-mode, see [ob-lesim mode](#).

## 1 Installing

Install from here for now, hopefully from MELPA later.

Open the lesim-mode directory in Emacs and do `M-x package-install-from-buffer`.

Add `(require 'lesim-mode)` to your Emacs init file (most often, `.emacs.d/init.el`).

## 2 Using

Syntax highlighting is enabled automatically.

`C-c C-c` runs the script in the Learning Simulator, highlighting any error.

Even without running a script, some common errors are highlighted (not 100% comprehensive yet).

Hovering on an error with the mouse pops up a description. `C-c C-e` moves to the next error, displaying a description in the minibuffer.

`TAB` and `Shift-TAB` move between fields in phase blocks, and by word elsewhere. They also align phase blocks at `|` signs and parameter blocks at `=` signs.

`C-c C-t` inserts a template script (see [3](#)).

`C-c C-h` re-highlights the whole script, in case something looks funny.

---

<sup>1</sup>However, lesim-mode does not understand the full Learning Simulator syntax. That is a job for Superman, that is, Learning Simulator chief software architect [Markus Jonsson](#).

```

# title: A test script for lesim-mode
# author: Stefano Ghirlanda
# date: 2023-05-03
# summary: Just to show off. Contains errors and won't run.

n_subjects      = 1 # end of line comment
mechanism       = sr
behaviors       = response, no_response
stimulus_elements = background, stimulus, reward
start_v         = 0.1
start_w         = 0.1
alpha_v         = 0.1
u               = reward:10, default:0
beta           = 1x # invalid value

###
Highlighted typos in @phase blocks:
###
@phase training stop: stimulus==500
START_TRIAL stimulus,background | reponse: REWARD(.5) | NO_REWARD
REWARD rewar | START_TRIAL
NO_REWARD background | START_TRIA

# these are wrong:
@run xtraining
@run my_run xtraining

# these are correct:
@run training
@run my_run training

@figure v(stimulus->response)
@subplot 111 {'xlabel':'exposures to stimulus', 'ylabel':'v-value'}
@vplot stimulus->response

-:--- test-err.les All of 960 L14 Git:main (Les)
Error on line 14: Expected 'x->y:value' or 'default:value' in 'beta', got '1x'.

```

Figure 1: Sample screenshot

### 3 Configuration

Using M-x `customize-group lesim` you can configure:

- The command used to run the Learning Simulator (default: `lesim.py`).
- The key binding to run scripts.
- The key binding to insert a template script.
- The key binding to re-highlight the buffer.
- The key binding to move to the next validation error.
- Whether to insert a template script when opening an empty `.les` file (default: `nil`).

### 4 Bugs and planned features

Please use [issues on Github](#) for bug reports and feature requests.