
LuaFormatter

Reformats your Lua source code.



Editor support

- VS Code: <https://github.com/Koihik/vscode-lua-format>
- Sublime: <https://github.com/Koihik/sublime-lua-format>
- Vim: <https://github.com/andrejlevkovitch/vim-lua-format>

Install

LuaRocks

Requirements

- c++ 17 compiler
- luarocks 3.3.0+

The easiest way to install is to use LuaRocks.

```
1 luarocks install --server=https://luarocks.org/dev luaformatter
```

Build from source

Requirements

- cmake 3.9+
- c++ 17 compiler

Steps

```
1 git clone --recurse-submodules https://github.com/Koihik/  
  LuaFormatter.git  
2 cd LuaFormatter  
3 cmake .  
4 make  
5 make install
```

Usage

```
1  ./lua-format [Lua scripts...] {OPTIONS}
2
3  Reformats your Lua source code.
4
5  OPTIONS:
6
7      -h, --help                Display this help menu
8      -v, --verbose            Turn on verbose mode
9      -i                        Reformats in-place
10     --dump-config             Dumps the default style used to
11                               stdout
12     -c[file], --config=[file] Style config file
13     --column-limit=[column limit] Column limit of one line
14     --indent-width=[indentation width] Number of spaces used for
15                                     indentation
16     --tab-width=[tab width]      Number of spaces used per tab
17     --continuation-indent-width=[Continuation indentation width] Indent width for continuations
18                                     line
19     --spaces-before-call=[spaces before call] Space on function calls
20     --column-table-limit=[column table limit] Column limit of each line of a
21                                     table
22     --table-sep=[table separator] Character to separate table
23                                     fields
24     --use-tab                  Use tab for indentation
25     --no-use-tab               Do not use tab for indentation
26     --keep-simple-control-block-one-line keep block in one line
27     --no-keep-simple-control-block-one-line Do not keep block in one line
28     --keep-simple-function-one-line keep function in one line
29     --no-keep-simple-function-one-line Do not keep function in one
30                                     line
31     --align-args               Align the arguments
32     --no-align-args            Do not align the arguments
33     --break-after-functioncall-lp Break after '(' of function
34                                     call
35     --no-break-after-functioncall-lp Do not break after '(' of
36                                     function call
37     --break-before-functioncall-rp Break before ')' of function
38                                     call
39     --no-break-before-functioncall-rp Do not break before ')' of
40                                     function call
41     --align-parameter          Align the parameters
42     --no-align-parameter       Do not align the parameters
```

40	--chop-down-parameter	Chop down all parameters
41	--no-chop-down-parameter	Do not chop down all parameters
42	-- break -after-functiondef-lp	Break after '(' of function def
43	--no- break -after-functiondef-lp	Do not break after '(' of function def
44	-- break -before-functiondef-rp	Break before ')' of function def
45	--no- break -before-functiondef-rp	Do not break before ')' of function def
46	--align-table-field	Align fields of table
47	--no-align-table-field	Do not align fields of table
48	-- break -after-table-lb	Break after '{' of table
49	--no- break -after-table-lb	Do not break after '{' of table
50	-- break -before-table-rb	Break before '}' of table
51	--no- break -before-table-rb	Do not break before '}' of table
52	--chop-down-table	Chop down any table
53	--no-chop-down-table	Do not chop down any table
54	--chop-down-kv-table	Chop down table if table contains key
55	--no-chop-down-kv-table	Do not chop down table if table contains
56		key
57	--extra-sep-at-table-end	Add extra field separator at end of
58		table
59	--no-extra-sep-at-table-end	Do not add extra field separator at end
60		of table
61	-- break -after-operator	Put break after operators
62	--no- break -after-operator	Do not put break after operators
63	-- double -quote-to-single-quote	Transform string literals to use single
64		quote
65	--no- double -quote-to-single-quote	Do not transform string literals to use
66		single quote
67	--single-quote-to- double -quote	Transform string literals to use double
68		quote
69	--no-single-quote-to- double -quote	Do not transform string literals to use
70		double quote
71	--spaces-inside-functiondef-parens	Put spaces on the inside of
72		parens in
73		function headers
74	--no-spaces-inside-functiondef-parens	Do not put spaces on the inside
75		of

```

76                                     parens in function headers
77     --spaces-inside-functioncall-parens
78                                     Put spaces on the inside of
79                                     parens in
80                                     function calls
81     --no-spaces-inside-functioncall-parens
82                                     Do not put spaces on the inside
83                                     of
84                                     parens in function calls
85     --spaces-inside-table-braces      Put spaces on the inside of
86                                     braces in
87                                     table constructors
88     --no-spaces-inside-table-braces    Do not put spaces on the inside
89                                     of
90                                     braces in table constructors
91     --spaces-around-equals-in-field    Put spaces around the equal
92                                     sign in
93                                     key/value fields
94     --no-spaces-around-equals-in-field Do not put spaces around the
95                                     equal sign
96                                     in key/value fields
97     --line-breaks-after-function-body  Line breakes after function
98                                     body
99     --line-separator=[line separator] input(determined by the input
100                                     content),
101                                     os(Use line ending of the
102                                     current
103                                     Operating system), lf(Unix
104                                     style "\n"),
105                                     crlf(Windows style "\r\n"), cr(
106                                     classic
107                                     Max style "\r")
108     Lua scripts...                   Lua scripts to format
109     "--" can be used to terminate flag options and force all
110     following
111     arguments to be treated as positional options

```

The program will attempt to automatically use the current directory's `.lua-format` file if no config file is passed in the command line. If none is found, it will try to locate a `.lua-format` file in a parent directory recursively. On Linux it will use `$XDG_CONFIG_HOME/luaformatter/config.yaml` if `.lua-format` does not exist. In case there's no file, it will fallback to the default configuration. The program will give the top priority to the configuration values given in the command-line, then to the configuration files and finally to the hard-coded default values.

Style configure file

Configuration parameters not specified fallback to their default values.

See this file

Default configuration

```
1 column_limit: 80
2 indent_width: 4
3 use_tab: false
4 tab_width: 4
5 continuation_indent_width: 4
6 spaces_before_call: 1
7 keep_simple_control_block_one_line: true
8 keep_simple_function_one_line: true
9 align_args: true
10 break_after_functioncall_lp: false
11 break_before_functioncall_rp: false
12 spaces_inside_functioncall_parens: false
13 spaces_inside_functiondef_parens: false
14 align_parameter: true
15 chop_down_parameter: false
16 break_after_functiondef_lp: false
17 break_before_functiondef_rp: false
18 align_table_field: true
19 break_after_table_lb: true
20 break_before_table_rb: true
21 chop_down_table: false
22 chop_down_kv_table: true
23 table_sep: ","
24 column_table_limit: 0
25 extra_sep_at_table_end: false
26 spaces_inside_table_braces: false
27 break_after_operator: true
28 double_quote_to_single_quote: false
29 single_quote_to_double_quote: false
30 spaces_around_equals_in_field: true
31 line_breaks_after_function_body: 1
32 line_separator: input
```

Disable formatting for a line or block

Sometimes it may be useful to disable automatic formatting. This is done by putting the code between `LuaFormatter off` and `LuaFormatter on` tags:

```
1 -- LuaFormatter off
```

```
2 matrix = {  
3     {1, 0, 0, 0},  
4     {1, 1, 0, 0},  
5     {1, 1, 1, 0},  
6     {1, 1, 1, 1}  
7 }  
8 -- LuaFormatter on
```

Limitations

- Do not work when source file contains syntax error
- Do not support 'Format selection'