doitlive

Release 5.1.0

unknown

Feb 02, 2024

CONTENTS

1 Get it now 3

2 Quickstart 5

3 Examples 7

4 Stealth mode 9

5 Using the recorder 11

6 Themes 13

7 Comment magic (configuration) 15

8 Python mode 19

9 IPython mode 21

10 Shell completion 23

11 More 25

12 Project info 27

Because sometimes you need to do it live

Current version: v5.1.0.

doitlive is a tool for live presentations in the terminal. It reads a file of shell commands and replays the commands in a fake terminal session as you type random characters.

- *Get it now*
- *Quickstart*
- *Examples*
- *Stealth mode*
- *Using the recorder*
- *Themes*
- *Comment magic (configuration)*
- *Python mode*
- *IPython mode*
- *Shell completion*
- *More*
- *Project info*

CHAPTER

ONE

GET IT NOW

1.1 macOS with Homebrew:

```
$ brew update
$ brew install doitlive
```

1.2 With pip:

```
$ pip install doitlive
```

CHAPTER

TWO

QUICKSTART

1. Create a file called `session.sh`. Fill it with bash commands.
2. Run `doitlive play session.sh`.

```
$ doitlive play session.sh
```

3. Type like a madman.

CHAPTER

THREE

EXAMPLES

```
# Use the "sorin" prompt theme
$ doitlive play session.sh -p sorin

# Increase speed
$ doitlive play session.sh -s 3

# Use zsh
$ doitlive play session.sh --shell /bin/zsh
```

CHAPTER

FOUR

STEALTH MODE

Hit the TAB key to enter “stealth mode.” While in stealth mode, you can enter commands normally. Hit TAB again to exit stealth mode.

CHAPTER

FIVE

USING THE RECORDER

You can record session files using the built-in recorder command.

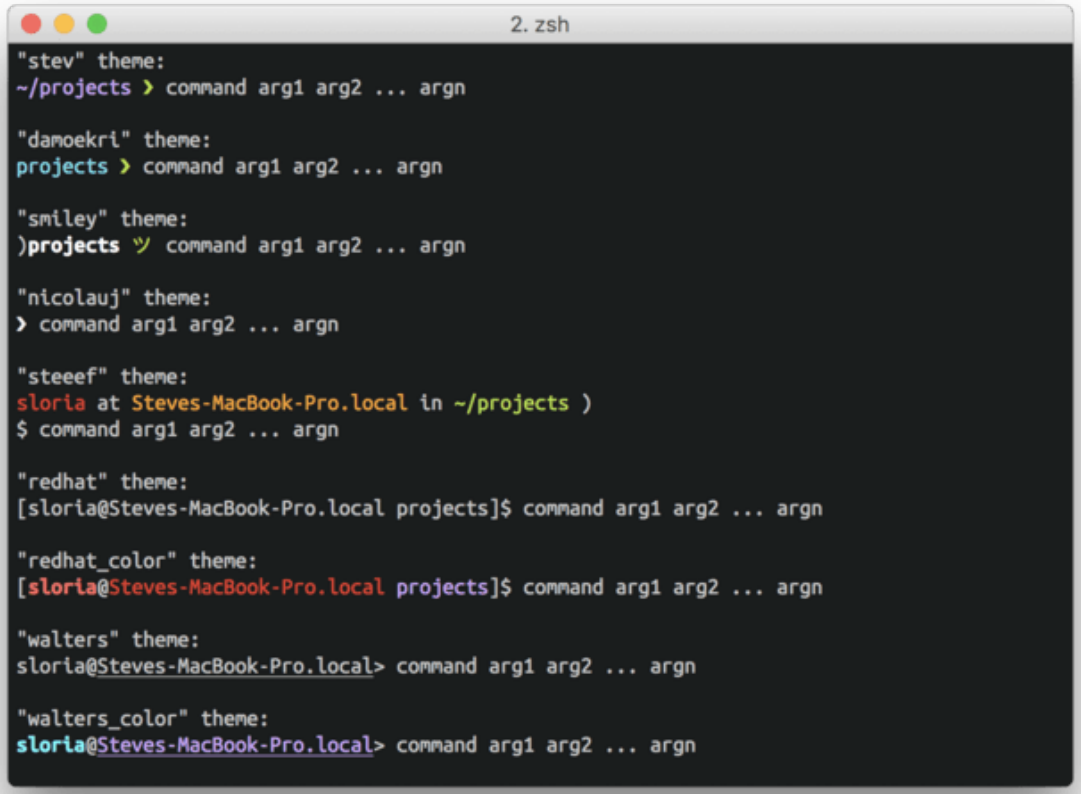
```
$ doitlive record
```

This will start a recording session. When you are finished recording, run the `stop` command. All commands will be written to a `session.sh` file.

CHAPTER

SIX

THEMES



doitlive comes with many prompt themes. To use a theme:

```
$ doitlive play session.sh -p <theme_name>
```

You can also change a session's theme by using a comment directive (see *Comment magic* below).

To view a list of available themes, run `doitlive themes` or `doitlive themes --preview`.

CHAPTER

SEVEN

COMMENT MAGIC (CONFIGURATION)

Any line in a session file that begins with # is a comment. Comments are ignored unless they begin with `#doitlive`, in which case they are used to configure the session.

The following options are available (all are optional).

7.1 #doitlive speed: <int>

configures "typing" speed. Defaults to 1.

Example:

```
#doitlive speed: 3
```

7.2 #doitlive prompt: <theme_name_or_template>

configures the prompt. Can be any of the built-in themes or a custom prompt template.

Example:

```
#doitlive prompt: stev
```

Using a custom template:

You can provide the `prompt` option with a custom template. To include the user, hostname, current directory, current path to working directory, current datetime, or vcs branch (git or Mercurial), use `{user}`, `{hostname}`, `{dir}`, `{cwd}`, `{now}`, and `{vcs_branch}`, respectively.

For git, `{vcs_branch}` just shows the branch. For Mercurial, this shows the branch name + the bookmark, except it omits the default branch name if there is a bookmark. This is equivalent to `{git_branch}{hg_id}`. There are also specialised `{hg_branch}`, and `{hg_bookmark}` keywords that only show that information, without the combined logic of `{hg_id}`.

Example:

```
#doitlive prompt: {user} is at {cwd} $
```

Any of the prompt variables can be formatted with ANSI styles, like so:

Example:

```
#doitlive prompt: {user.cyan}@{hostname.green}:{dir.bold.magenta} $
```

Newlines can be included in prompts using `{nl}`.

Example:

```
#doitlive prompt: {user}:{dir}{nl}$
```

Available styles: blue, magenta, red, white, green, black, yellow, cyan, bold, blink, underlined, dim, paren, square, curly, inverse, git, and hg.

7.3 #doitlive shell: <shell>

configures which shell is use. Defaults to the *$SHELL* environment variable.

Example:

```
#doitlive shell: /bin/zsh
```

7.4 #doitlive alias: <alias>=<command>

adds an alias to the session.

Example:

```
#doitlive alias: du="du -ach | sort -h"
```

7.5 #doitlive env: <envvar>=<value>

sets an environment variable.

Example:

```
#doitlive env: EDITOR=vim
```

7.6 #doitlive unalias: <alias>

removes an alias.

7.7 #doitlive unset: <envvar>

unsets an environment variable.

7.8 #doitlive commentecho: [true|false]

Whether to echo comments or not. If enabled, non-magic comments will be echoed back in bold yellow before each prompt. This can be useful for providing some annotations for yourself and the audience.

CHAPTER

EIGHT

PYTHON MODE

doitlive supports autotyping in a Python console. You can enter Python mode in a session by enclosing Python code in triple-backticks, like so:

```
# in session.sh

echo "And now for something completely different"

```python
list = [2, 4, 6, 8]
sum = 0
for num in list:
    sum = sum + num

print("The sum is: {sum}".format(sum=sum))
```
```

CHAPTER

NINE

IPYTHON MODE

If you have IPython installed, you can run doitlive in `ipython` mode. Enclose your Python code in triple-backticks, like so:

```
# in session.sh

```ipython
def fib(n):
   a, b = 0, 1
   while a < n:
      print(a, end=' ')
      a, b = b, a+b
   print()

# Magic!
%time fib(100)

```
```

CHAPTER

TEN

SHELL COMPLETION

Note: If you installed doitlive with Homebrew, bash and zsh completion are already installed.

Shell completion is available for `bash`, `zsh`, and `fish`.

For bash or zsh, add the following to your `.bashrc` or `.zshrc`:

```
eval "$(doitlive completion)"
```

For fish, add the following to `~/.config/fish/completions/doitlive.fish`:

```
eval (doitlive completion)
```

CHAPTER

ELEVEN

MORE

For more options, run

```
$ doitlive --help
```

You can also get help with subcommands.

```
$ doitlive play --help
```

CHAPTER

TWELVE

PROJECT INFO

12.1 Changelog

12.1.1 5.1.0 (2024-02-02)

Features:

- Add `pure` theme (#264).

Other changes:

- Remove `packaging` and `colorama` as explicit dependencies (#262).

12.1.2 5.0.0 (2024-01-09)

Features:

- Ctrl-l clears the screen (#217, #237). Thanks @Kaligule for the suggestion and @DvirB10 for the PR.

Support:

- Support Python 3.11 and 3.12.
- Drop support for Python 2 (#202), Python 3.6, Python 3.7, IPython 5, IPython 6, IPython 7, and click<8.

12.1.3 4.4.0 (2022-08-14)

Support:

- Support IPython 7 and 8 (#91). Thanks @matan129 for the PR.
- Test against Python 3.8, 3.9, and 3.10.

12.1.4 4.3.0 (2019-04-07)

Features:

- Enter and exit "stealth mode" by pressing TAB (#113). Thanks @ThinkBriK for the suggestion and thanks @jlehett for the implementation.

12.1.5 4.2.1 (2018-12-22)

Bug fixes:

- Fix behavior of `commentecho` when a comment includes quote characters.

12.1.6 4.2.0 (2018-11-08)

- Add `deadsimple` theme (#97). Thanks @Aversiste.

Other changes:

- Test against Python 3.7.

12.1.7 4.1.0 (2018-10-25)

Features:

- Support bright colors in prompt templates.

Bug fixes:

- Fix compatibility with click>=7.0.

12.1.8 4.0.1 (2018-05-22)

Bug fixes:

- Ctrl-Z suspends doitlive (#44). Thanks @emanuelhouse for reporting.
- Fix help text for `--shell` option (#43).

12.1.9 4.0.0.post0 (2018-05-14)

- Distribute a universal wheel.
- Minor docs updates.

12.1.10 4.0.0 (2018-05-13)

Features:

- Add shell completion for bash, zsh, and fish (#3).
- Add "Did you mean" suggestions.
- Support setting environment variables with `export` commands (#32). Thanks @asmacdo for the suggestion.
- Support setting aliases with `alias` commands (#40).

Bug fixes:

- Fix exiting a command such as `watch` with `ctrl-c` during a session (#29). Thanks @zigarn for the catch and patch.

Other changes:

- Drop official support for Python 3.3 and 3.4. Python 2.7 and >=3.5 are supported.

- Lots of internal re-organization of modules.

12.1.11 3.0.3 (2017-11-08)

Bug fixes:

- *–quiet* options supresses ending message (#26). Thanks @technovangelist for reporting and thanks @PandaWhoCodes for the PR.
- Fix installation issue on Windows (#4). Thanks @eXigentCoder for reporting.

12.1.12 3.0.2 (2017-10-17)

Bug fixes:

- Fix 'cd-ing' to paths with an envvar (#24). Thanks @utdrmac for reporting.
- Fix behavior of `cd -`
- Fix behavior of `Ctrl-C` after all commands have finished.

12.1.13 3.0.1 (2017-10-16)

Bug fixes:

- Fix behavior of Backspace key when speed > 1.
- Handle KeyError when `$HOME` is unset (#10). Thanks @Stefan-Code for reporting.

12.1.14 3.0.0 (2017-10-15)

- Support IPython>=5.0 (#20). Drop support for IPython<5.0. Thanks @rplevka for reporting.
- Use `$SHELL` as the default interpreter for commands if not explicitly specified.
- Remove invalid import in `ipython` module. Thanks @axocomm.
- Fix exiting a session with Ctrl-C in Python 3.

12.1.15 2.8.0 (2017-10-08)

Bug fixes:

- Don't allow passing a *–speed* that is < 1 (#17). Thanks @mblhaunted for reporting and thanks @Stefan-Code for the implementation suggestion.

12.1.16 2.7.0 (2017-03-07)

Features:

- Add `stev`, `damoekri`, and `smiley` themes.
- Modify `sorin` theme to be more like the original prezto theme.

Bug fixes:

- Prevent extra spacing when using `{vcs_branch}`, `{git_branch}`, or `{hg_branch}` in a custom prompt.

12.1.17 2.6.0 (2017-01-07)

Features:

- Prompt template variables can be styled with `.inverse`, e.g. `{user.inverse}`.
- Prompt templates can include `{nl}` for displaying new-lines. Thanks @andredias.

Other changes:

- Test against Python 3.6.

12.1.18 2.5.0 (2016-05-02)

Features

- Add ipython mode (#8).

12.1.19 2.4.0 (2015-10-18)

Features:

- Backspace key works during playback.

Bug fixes:

- Prevent unicode_literals import warning from click on Python 2 (#12, #13).
- Fix bug that caused some keystrokes to get echoed instead of swallowed (#6). Thanks @jordigh for reporting.

Other changes:

- Reorganized as a package. Added `termutils` and `version_control` modules.

Big thanks to @Stefan-Code for implementing these changes.

12.1.20 2.3.1 (2015-02-08)

- Fix bug that showed the incorrect prompt on the last slide if the theme was set using the `#doitlive prompt:` directive.

12.1.21 2.3.0 (2014-11-16)

- Add support for displaying Mercurial VCS info (current branch, bookmark)
- Add `commentecho` CLI option and magic comment.
- Add `--quiet` CLI option for suppressing the startup message.

12.1.22 2.2.1 (2014-08-02)

- Fix display of git branches on Python 3 (don't show `b` prefix).

12.1.23 2.2.0 (2014-07-13)

- Add `{TTY}` prompt variable that contains named constants for ANSI escape sequences.
- Add "giddie" theme.
- Add `help/H` command to the recorder console.

12.1.24 2.1.0 (2014-06-25)

- Python mode: Fenced code blocks can be played back in a fake Python console.
- Added ability to preview and undo commands during a recorder session.
- Current datetime (`{now}`) can be included in prompt.
- Added 'pws' theme.
- Added `--envvar` and `--alias` options to `record` command.
- Added `unalias` and `unset` comment directives.

12.1.25 2.0 (2014-06-21)

- Added session recorder (`doitlive record`).
- Improved interface.
- Sessions are played with `doitlive play <session_file>`.
- Deprecated `doitlive-demo`. Run `doitlive demo` instead.
- Deprecated `doitlive --themes` and `doitlive --themes-preview`. Run `doitlive themes` and `doitlive themes --preview` instead.
- Fix bug that raised an error when cd'ing into a non-existent directory.
- Remove extra spacing in prompt when not in a git directory.
- Added 'robbyrussell' theme.

12.1.26 1.0 (2014-06-18)

- Added themes!
- Prompt variables can have ANSI colors and styles.
- `{hostname}` can be included in prompt.
- `{git_branch}` can be included in prompt.
- Prompt variable `{full_cwd}` renamed to `{cwd}`.
- Prompt variable `{cwd}` renamed to `{dir}`.
- Short option for `--speed` is now `-s`.
- Short option for `--shell` is now `-S`.
- Changed default prompt.
- `run` and `magictype` receive prompt_template instead of a prompt function.
- Remove unnecessary `PromptState` class.

12.1.27 0.2.0 (2014-06-16)

- Add "speed" config option.
- Fix short option for "–shell".
- Custom prompts are colored.
- Remove unnecessary –check-output option, which was only used for testing.
- Fix bug where cwd would not update in custom prompts.

12.1.28 0.1.0 (2014-06-15)

- Initial release.

12.2 License

```
Copyright Steven Loria and contributors

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
```

(continues on next page)

(continued from previous page)

```
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
THE SOFTWARE.
```

12.3 Authors

12.3.1 Lead

- Steven Loria @sloria

12.3.2 Contributors (chronological)

- Josh Carp @jmcarp
- Austin Macdonald @asmacdo
- Jordi Hermoso @jordigh
- Stefan-Code @Stefan-Code
- André Felipe Dias @andredias
- @axocomm @axocomm
- Thomas Ashish Cherian @PandaWhoCodes
- Alexandre Garnier @zigarn
- Tristan Le Guern @Aversiste
- John Lehett @jlehett
- Tim Gates @timgates42
- Matan Rosenberg @matan129
- Matthias Maennich @metti
- Dvir Bechor @DvirB10

12.4 Kudos

- Idea came from Jordi Hermoso's "Revsets" talk at PyCon 2014.
- Armin Ronacher's click library made this quick to implement.
- Themes inspired by Sorin Ionescu's prezto zsh themes.
- Hat tip to related projects HackerTyper and PlayerPiano.