NAME

dof − (do-for) executes commands for each item of a list

SYNOPSIS

dof [**−efpr**] [**−s** first..last[..step]] [**−l**] [**−h**] [**−v**] [item ... | -] [**−x** ...] [**−g** ...] [**do** [command | -]]

DESCRIPTION

dof executes commands for each item (file or string) of a list

```
# convert all mp3s to ogg
dof -e *.mp3 do ffmpeg -i "%f" "%b.ogg"

# display colors on terminal
dof -e -s 0..15 do 'tput setaf %f; echo colour%f; tput op;'

# remove all files except *.pdf and *.tex
dof -e * -x *.{pdf,tex} do 'rm %f'
```

OPTIONS

− Read from stdin.

−e Execute; *dof* displays what commands would be run, this option executes them.

−x Exclude list; after **-x** are following glob's patterns to exclude items from the previous list. When the pattern is unquoted it will expanded by the shell otherwise 'glob()' will be used in each directory. See**glob**(3)

−X Exclude directories list; after **-X** are following glob's patterns to exclude directories on recursive (*-r*) run. When the pattern is unquoted it will expanded by the shell otherwise 'fnmatch()' will be used in each directory. See**fnmatch**(3)

−g Exclude list; after **-g** are following regex's patterns to exclude items from the previous list. Since the *dof* parameters can be anything, this is no glob file-patterns but regex's patterns. Example '*.c' file-pattern should be written as '.*\.c'.

−G Exclude directories list for recursive run by using POSIX regular expressions.

−r Recursive execution into subdirectories. You have to be sure that you type wildcard patterns quoted, otherwise the wildcards will expanded by the shell in the first directory.

```
# prints all directories except .git related
dof -rd '*' -X '*/.git*' do %cwd/%f

# the previous but using regex
dof -rd '*' -G '/.git($|/)' do %cwd/%f
```

−f Force non-stop; *dof* stops on error (exit code != 0), this option forces *dof* to ignore them.

−p Plain files only; directories, devices, etc are ignored.

−d Directories only; plain-files, devices, etc are ignored.

−s *first*..*last*[..*step*]
Adds a sequence of numbers (float or integer). This option can be repeated many times.

−l Print recipes (~/.dofrc)

−−*recipe*
Execute recipe (ex: dof --to-ogg)

−h Help screen.

−v Version and license information.

VARIABLES

In the 'commands' you can use several program's variables. Variables are recognized by prefix '%'. Variables and its modifiers can be included inside '{}', '()', '[]' blocks or any other '*ispunct()*' character block that follows the '%'.

%f The full string (or filename).

%b The basename (no directory, no extension).

%d The dirname (only the directory without the '/' suffix).

%e The extension without '.' prefix.

%h The home directory.

%cwd The current working directory.

%date The current date in YYYY-MM-DD format.

%time The current time in HH-MM-SS format.

%% The character '%'.

%q The apostrophe character (').

You can get a list of all variables with '*--vars*' option.

MODIFIERS

Modifiers defined by ':' that follows a variable and modifies the result string. You can have unlimited number of modifiers.

l[{f|l|s}]*c*

Returns the left part of the string until the **f**irst or **l**ast occurrence of character *c*. The**s** uses the rest of the modifier's parameter as string to search.

r[{f|l|s}]*c*

Returns the right part of the string until the **f**irst or **l**ast occurrence of character *c*. The**s** uses the rest of the modifier's parameter as string to search.

t*ab* Replace all *a* characters with *b*.

```
# displays the time (hour and minutes only)
# %time = 09-15-25 (t-:) 09:15:25 (ll:) 09:15
dof nil do %{time:t-::ll:}

# the same by using date command
dof "‘date -R‘" do '%{f:ll :rl :ll:}'
```

s/*p*/*r*/[g]

Replace regular expression match (or matches) of pattern *p* with the string *r*. By default only the first match will replaced; use the **g** to replace all matches.

FILES

dof reads '*~/.dofrc*' file. This file contains recipes in form 'name: parameters'. These recipes can later executed with *--name* option.

Example:

```
to-ogg: *.wav *.mp3 do 'ffmpeg -i "%f" "%b.ogg" && rm "%f"'
to-png: *.jpg *.gif do 'convert "%f" "%b.png" && rm "%f"'
```

NOTES

Items '.' and '..' are ignored.

SEE ALSO

glob(3), **fnmatch**(3), **regex**(7)

AUTHOR

Nicholas Christopoulos ⟨nereus@freemail.gr⟩
Project page: ⟨https://github.com/nereusx/unix-utils⟩