fish – The friendly interactive shell

Jonas Weissensel

June 12, 2015

Was ist eine Shell?

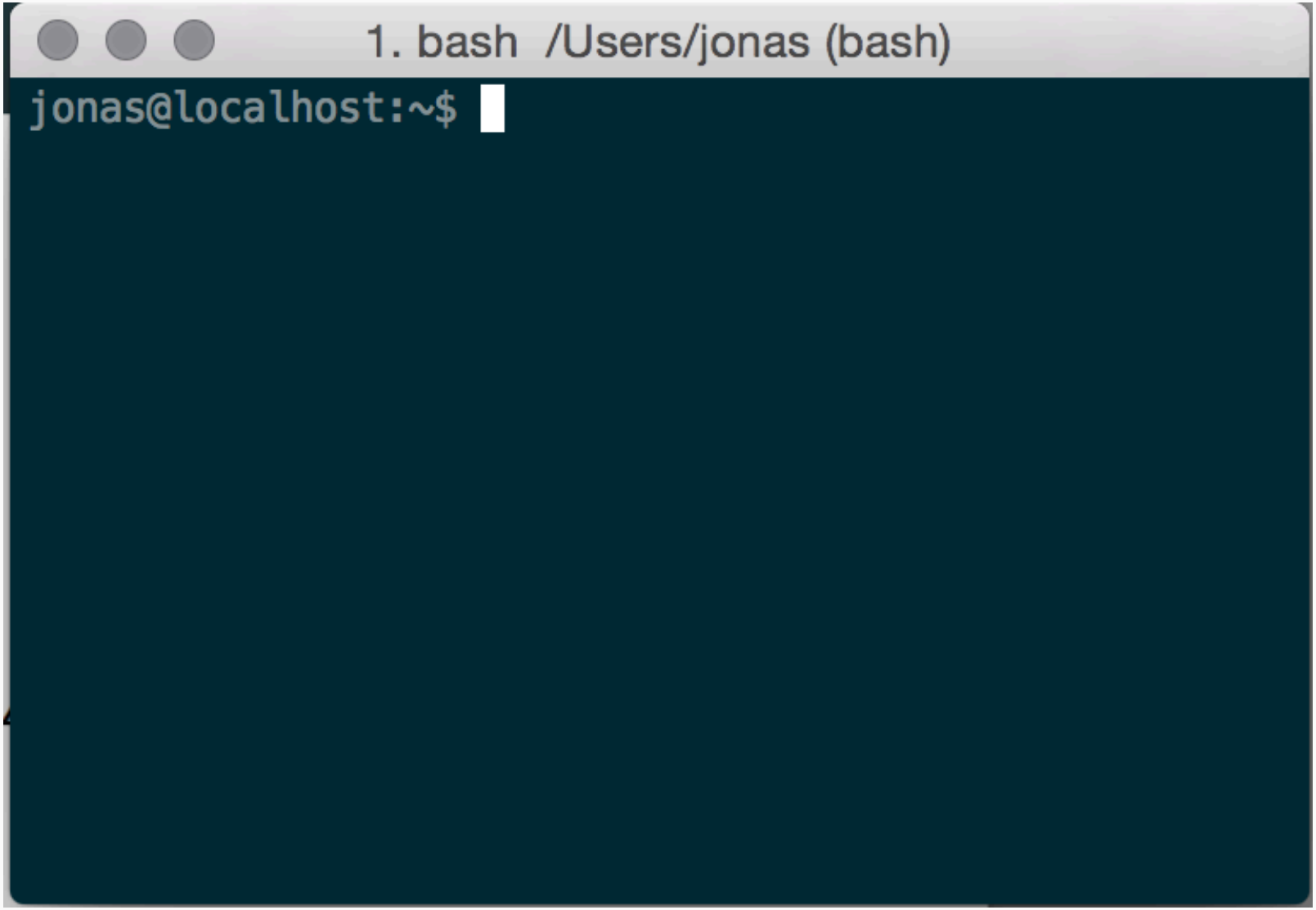
Die *friendly interactive shell*

Main features

HOW-TO fish

Was ist eine Shell?

Was ist eine Shell?



Was ist eine Shell?

- Kommandozeileninterpreter – CLI
- Textbasiert
- für „Power User”

Die *friendly interactive shell*

Information

- Initial release: 13.Feb. 2005
- Last stable release: v.2.1.2, 24.Feb. 2015
- License: GPL v2
- Binary packages for Linux & OS X

Main features

Features Out-Of-The-Box

- Syntax highlighting
- Tab complete anything
  - Command-specific tab completions
  - Descriptions for completions
  - man-pages based completion
- Browser based `help` and `fish_config` commands
- Parameter expansion (Globbing)
- Command-line editor

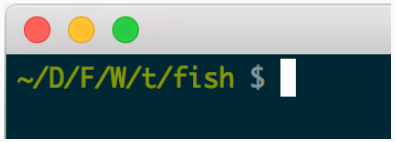


Figure 1: My prompt

```
function fish_prompt --description 'Write out the prompt'
    set -l last_status $status
    if not set -q __fish_prompt_normal
        set -g __fish_prompt_normal (set_color normal)
    end
    # PWD
    set_color $fish_color_cwd
    echo -n (prompt_pwd)
    set_color normal
    printf '%s ' (__fish_git_prompt)
    if not test $last_status -eq 0
        set_color $fish_color_error
    end
    echo -n '$ '
end
```

HOW-TO FISH

Familiar things

- Changing directories
- Copying/Moving files
- Running programs (in background)

Differences

- Changing `STDOUT` and `STDERR`
- Running commands in sub-shells
- Usage of `and` and `or` instead of `&&` and `||`

Bash – Fish

Table 1: Some differences in everyday use

Command	Bash	Fish
Move STDOUT	`>`	`>`
Move STDERR	`2>`	`^`
Run cmd in subshell	`$(cmd)`	`(cmd)`
	`cmd1 && cmd2`	`cmd1; and cmd2`
Combine commands	`cmd1 || cmd2`	`cmd1; or cmd2`

No aliases - just functions

```
$ functions
., N_, VL, alias, cask, cd, ...
```

```
$ functions ls
function ls --description 'List contents of directory'
    command ls -G $argv
end
```

```
$ alias ge 'grep $argv | less'
$ functions ge
function ge
    grep $argv | less $argv;
end
```

```
$ echo $PATH
/Users/jonas/.rbenv/shims /usr/local/sbin /Users/jonas/Library/Haskell/bin
/usr/local/bin /usr/bin /bin /usr/sbin /sbin /opt/X11/bin /usr/local/MacGPG2/bin
/usr/texbin
```

Pitfalls

- fish is non-POSIX compliant

Pitfalls

- fish is non-POSIX compliant
- Vim needs a sh compatible shell
  → add `set shell=sh` to .vimrc

Pitfalls

- fish is non-POSIX compliant
- Vim needs a sh compatible shell
  → add `set shell=sh` to .vimrc
- Setting an environment variable for one command:

```
$ env LC_ALL=en_US.UTF-8 LANG=en ls++
```

Pitfalls

- fish is non-POSIX compliant
- Vim needs a sh compatible shell
  → add `set shell=sh` to .vimrc
- Setting an environment variable for one command:

```
$ env LC_ALL=en_US.UTF-8 LANG=en ls++
```

- Exporting an environment variable:

```
$ set -x PATH ~/bin $PATH
```

Documentation

Official resources

`https://www.fishshell.com`
`https://github.com/fish-shell/fish-shell`
`corydoras@ridiculousfish.com`

oh-my-fish

`https://github.com/oh-my-fish/oh-my-fish`