

hi i'm @llkats



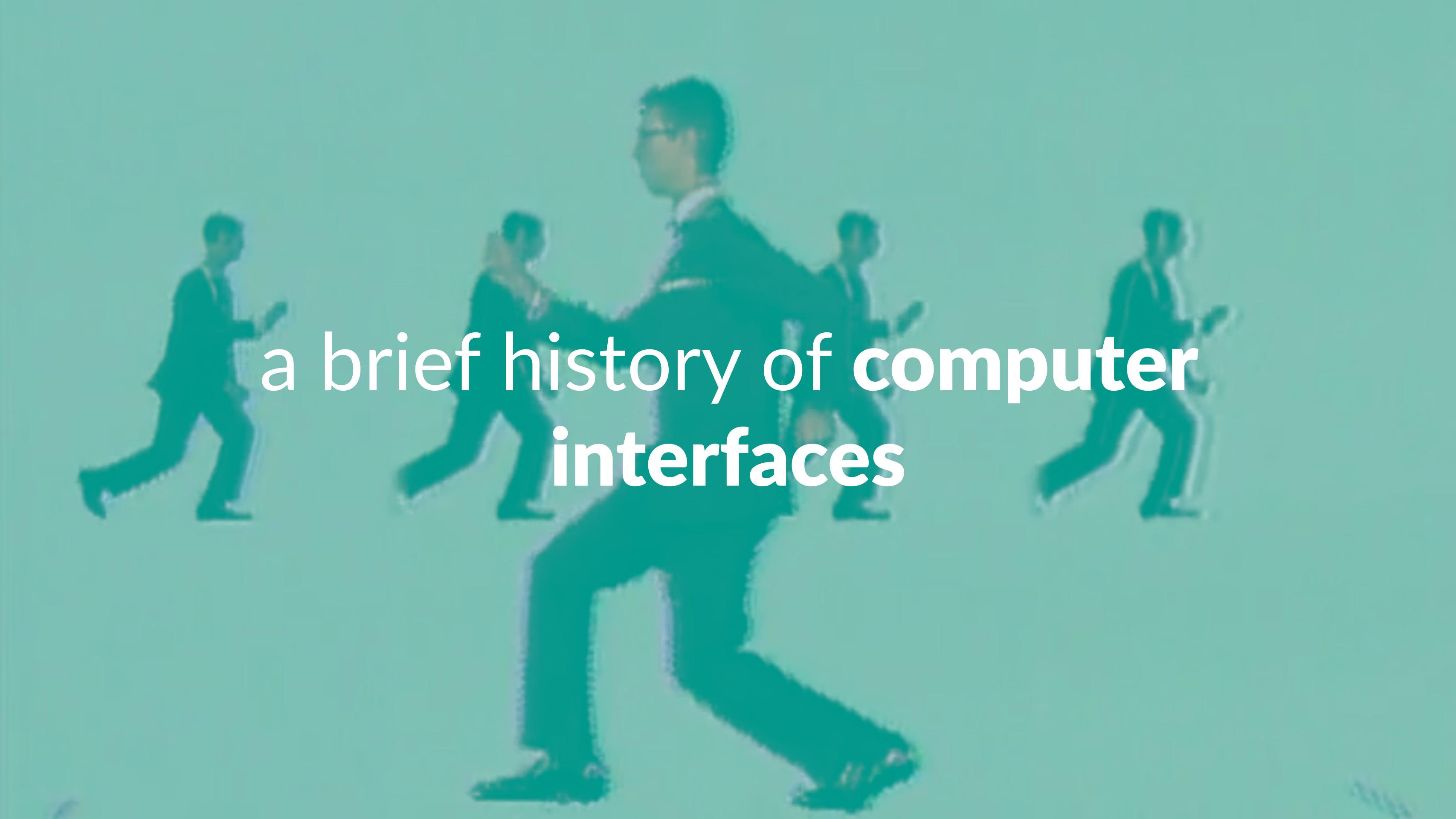
# Trick Out Your Terminal

# Game of Shells



3. bash /Users/llkittens (bash)

bash-3.2\$



# a brief history of computer interfaces



the punch card era



# the punch card era



# the punch card era



"Users had to accommodate  
computers rather than the other  
way around"

– *Raymond and Landley*







```
bash-3.2$ ls
Applications   Developer   Downloads   HOME       Movies      Pictures    libpng-1.6.13  out.txt    pubkey.txt
Desktop        Documents   Dropbox    Library     Music       Public     npm-debug.log perl5
bash-3.2$ clear
bash-3.2$ pwd
/Users/llkittens
bash-3.2$ cd ~/Developer/Projects/meatdelay/
bash-3.2$ ls
Gruntfile.js      bookmarklet.js      delay.js      meatdelaylink.html  npm-debug.log
README.md         build.js          favicon.ico  node_modules    package.json
bash-3.2$ grep -ri "meat" *.js
delay.js: // press escape key to dismiss meatdelay without submitting
bash-3.2$ git status
On branch master
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working directory)

    modified:   bookmarklet.js
    modified:   delay.js
    modified:   package.json

Untracked files:
  (use "git add <file>..." to include in what will be committed)

    build.js
    npm-debug.log

no changes added to commit (use "git add" and/or "git commit -a")
bash-3.2$ git diff
diff --git a/bookmarklet.js b/bookmarklet.js
index 6a74fa4..37f37f9 100644
--- a/bookmarklet.js
+++ b/bookmarklet.js
@@ -1 +1 @@
-javascript:(function(){;var numDependencies=0,loadedDependencies=0;function scriptLoaded() loadedDependencies++;if(numDependencies==loadedDependencies ) afterDepsLoaded()%7D%7Dfunction afterDepsLoaded()%7B(function()%7Bvar WAITFORIT,num;var divstyles=%7Bwidth:%22300px%22,height:%22150px%22,background Color:%22#fff%22,position:%22fixed%22,top:%22200px%22,left:%2250%25%22,marginLeft:%22-150px%22,border:%227px%20solid%20#f181ad%22,zIndex:%22100000%22%7D;var buttonstyles=%7Bborder:%222px%20solid%20pink%22,background:%22white%22,color:%22#333%22,cursor:%22pointer%22,height:%2240px%22,fontSize:%2216px%22,margin:%22n
```

```
bash-3.2$ ls
Applications   Developer   Downloads   HOME       Movies      Pictures    libpng-1.6.13  out.txt    pubkey.txt
Desktop        Documents  Dropbox    Library     Music       Public     npm-debug.log perl5      tmp
bash-3.2$ clear
bash-3.2$ pwd
/Users/llkittens
bash-3.2$ cd ~/Developer/Projects/meatdelay/
bash-3.2$ ls
Gruntfile.js   bookmarklet.js  delay.js   meatdelaylink.html  npm-debug.log
README.md      build.js       favicon.ico node_modules  package.json
bash-3.2$ grep -ri "meat" *.js
delay.js: // press escape key to dismiss meatdelay without submitting
bash-3.2$ git status
On branch master
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git checkout -- <file>" to discard changes in working directory)

    modified:   bookmarklet.js
    modified:   delay.js
    modified:   package.json

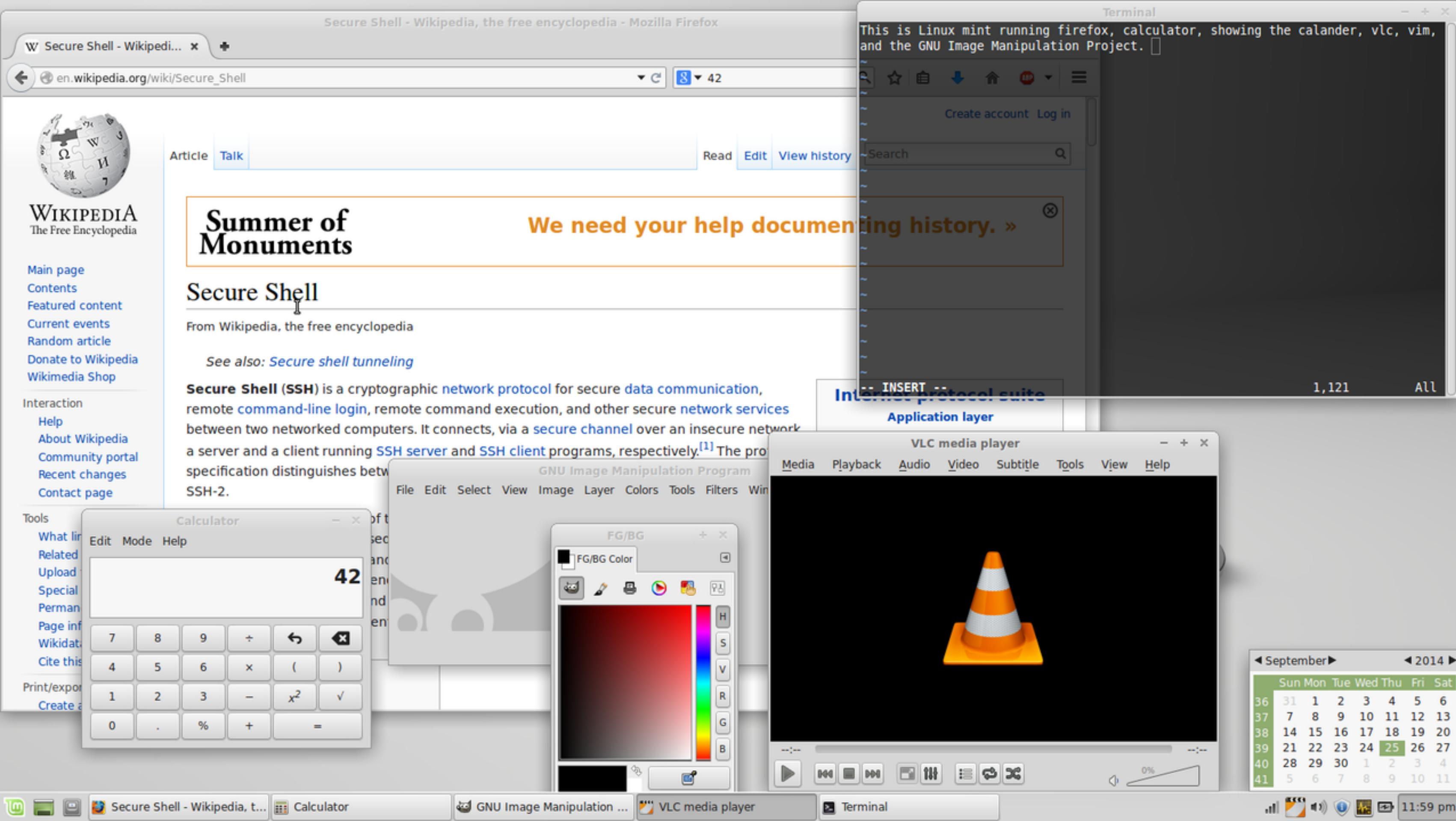
Untracked files:
  (use "git add <file>..." to include in what will be committed)

    build.js
    npm-debug.log

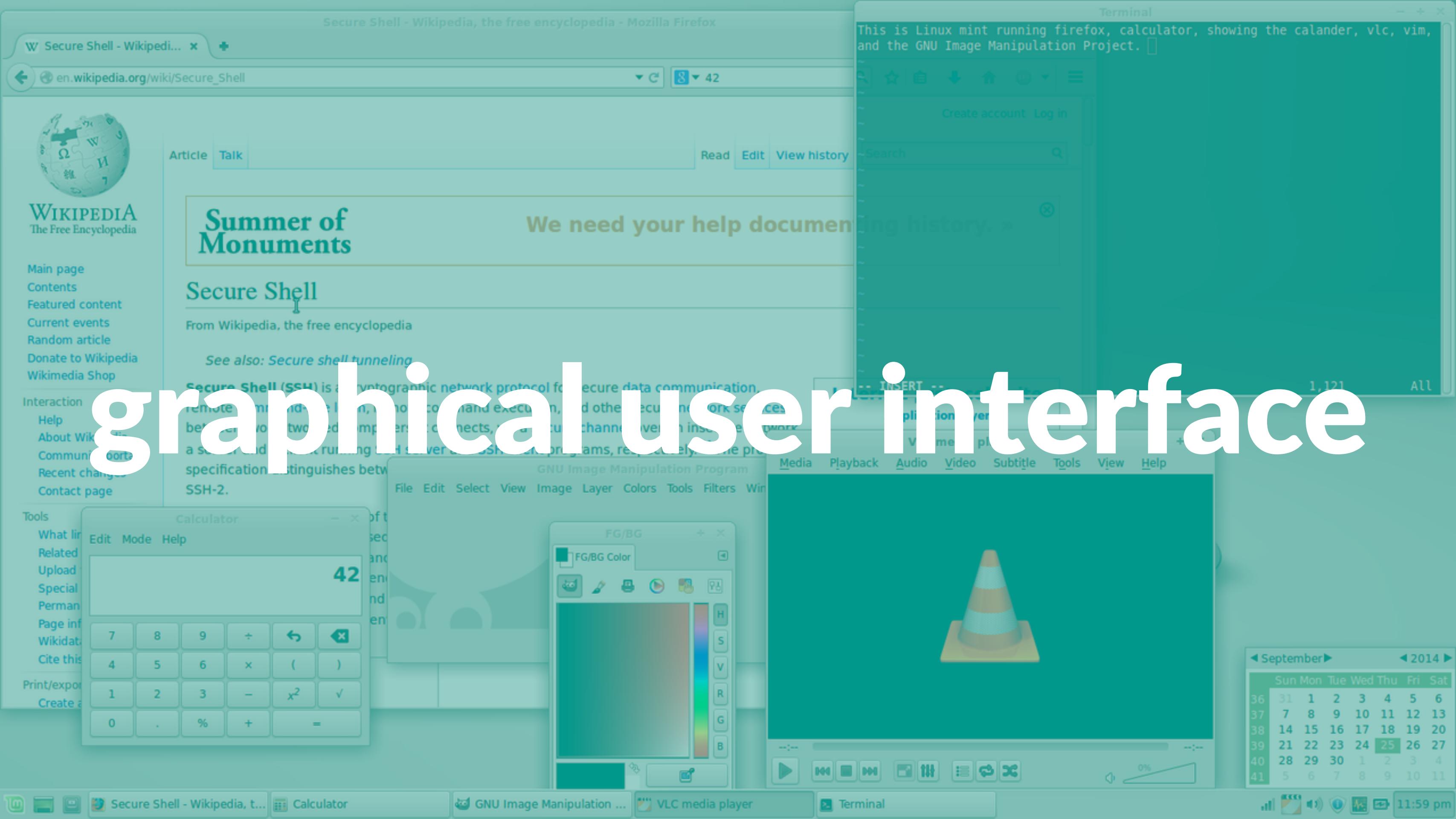
no changes added to commit (use "git add" and/or "git commit -a")
bash-3.2$ git diff
diff --git a/bookmarklet.js b/bookmarklet.js
index 6a74fa4..37f37f9 100644
--- a/bookmarklet.js
+++ b/bookmarklet.js
@@ -1 +1 @@
-javascript:(function(){;var numDependencies=0,loadedDependencies=0;function scriptLoaded(){loadedDependencies++;if(numDependencies==loadedDependencies)afterDepsLoaded()}function afterDepsLoaded(){function(){var WAITFORIT,num;var divstyles={width:'2300px',height:'150px',backgroundColor:'fff',position:'fixed',top:'20px',left:'25%',marginLeft:'-150px',border:'1px solid #f181ad',zIndex:100000};var buttonstyles={border:'2px solid pink'};background:'white' color:'#333' cursor:'pointer' height:'240px',fontSize:'16px',margin:'2px'}();}});
```

"These interfaces... required a serious investment of effort and learning time to master."

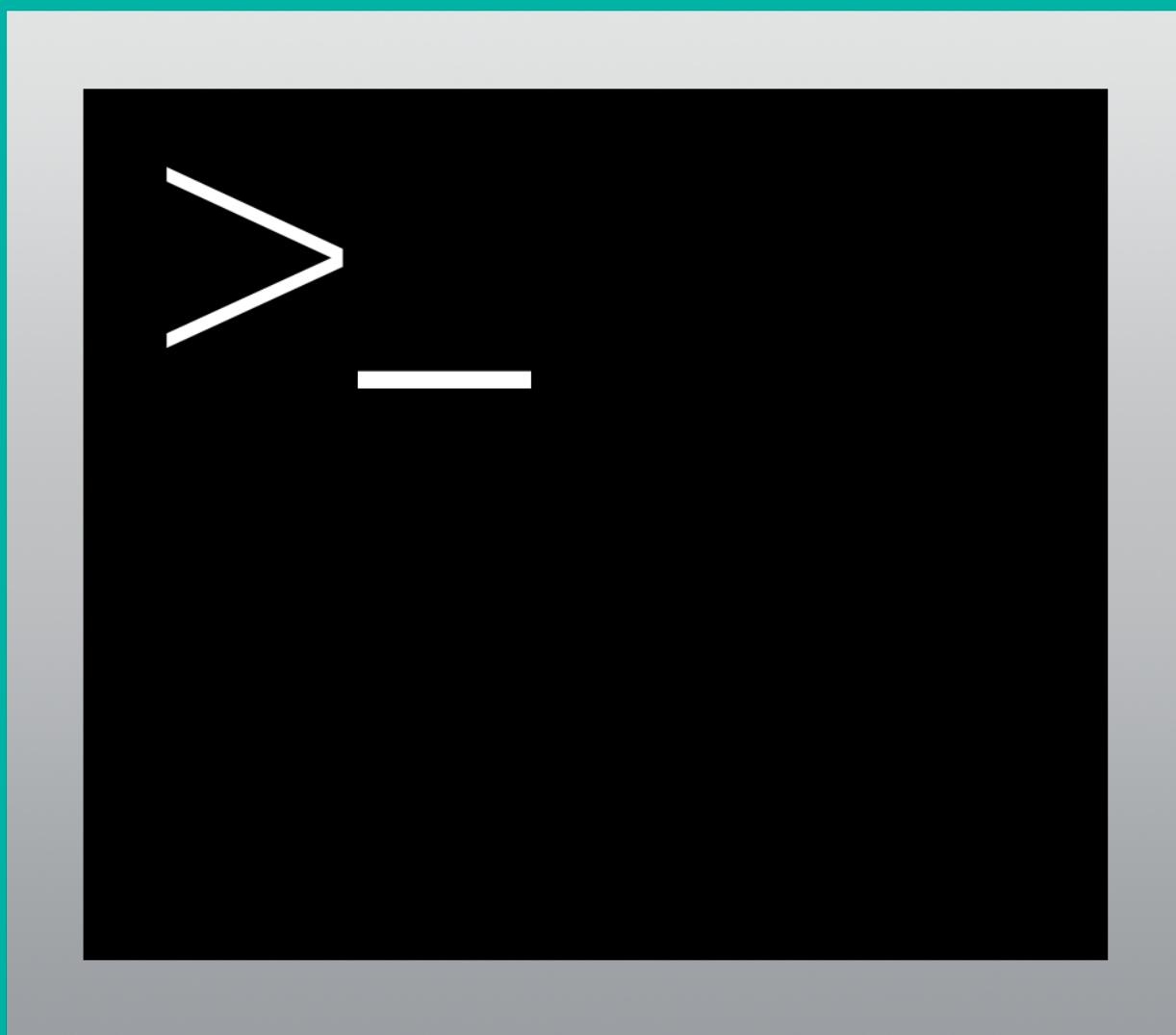
– Raymond and Landley



# graphical user interface



# emulation station



**so but why**

```
\[...\]/ - looks like you are already using zsh. woot!
linking .bowerrc...[ok]
linking .crontab...[ok]
linking .gemrc...[ok]
linking .gitconfig...[ok]
linking .gitignore...[ok]
linking .hgrc...[ok]
linking .npmrc...[ok]
linking .profile...[ok]
linking .rvmrc...[ok]
linking .screenrc...[ok]
linking .shellaliases...[ok]
linking .shellfn...[ok]
linking .shellpaths...[ok]
linking .shellvars...[ok]
linking .vim...[ok]
linking .vim/autoload...[ok]
linking .vim/backup...[ok]
linking .vim/bundle...[ok]
linking .vim/colors...[ok]
linking .vim/temp...[ok]
linking .vim/.netrwhist...[ok]
linking .vimrc...[ok]
linking .zlogout...[ok]
linking .zprofile...[ok]
linking .zshenv...[ok]
linking .zshrc...[ok]
```



<http://brew.sh/>

aesthetics

love the environment you're in

```
" Press ? for help
.. (up a dir)
/Users/brianl/Repo/quick-vim/
> backup/
> vim/
bundles.txt
quick-vim*
README.md
vimrc

1 #!/bin/bash
2 set -e
3
4 help () {
5     cat<<EOF
6
7 Usage: quick-vim [install|uninstall|upgrade|help]
8
9 Commands:
10
11     install    installs from the git repo
12     uninstall  restores original system vim config
13     upgrade    clobbers existing setup and installs fresh from git
14     help       print this msg
15
16 EOF
17 }
18
19 # backs default home vim up
20 backup () {
21     if [ -e ./backup ]
22     then
23         echo 'skipping backup'
24     else
25         mkdir -p ./backup
26         mv ~/.vimrc ./backup/.vimrc
27         mv ~/.vim ./backup/.vim
28     fi
29 }
30
31 # copy in pathogen and then clone in bundles
32 install () {
33     backup
34
35     # copy in vimrc and pathogen
36     cp ./vimrc ~/.vimrc
37     cp -r ./vim ~/.vim
38
39     # add in bundles from bundle.txt
40     mkdir -p ~/.vim/bundle
41     bundles=$(cat ./bundles.txt)
42     cd ~/.vim/bundle
43
44     for bundle in $bundles
45     do
46         git clone https://github.com/$bundle.git
47     done
48 }

quick-vim
```

@brianleroux, <https://github.com/brianleroux/quick-vim>



Last login: Sun Jun 14 20:47:09 on ttys001  
»@jlord ~: cd github/atom  
master ✨ @jlord ~/github/atom:

@jllord

A screenshot of a tmux session showing a terminal window with a complex command history and status bar.

The terminal window displays a series of commands run by user 'randall' in a directory 'Projects/node-nma'. The commands include:

- c node-nma
- touch dirty-repo
- rm dirty-repo
- git checkout HEAD~1 -q
- git checkout master -q
- top &

After running 'top &', the user runs 'kill % && (exit 42)'. This results in:

- [1] 10021 + suspended (tty output) top
- [1] + terminated top

Following this, the user runs 'workon virtualenv'. This results in:

- 42 + v0.10.35 2.0.0p481 ① 16:30:12
- virtualenv 2.0.0p481 ① 16:32:22
- with vi-mode indication on the last segment (green:insert/white:normal)
- ...and the right side segments auto-hide on long commands!

The status bar at the bottom of the terminal window shows:

- tmux
- 1 > zsh
- >
- ↑ 3d 20h 44m < 2.4 2.0 1.9
- 2015-02-08
- twenty-five to five
- ® Randalls-MBP.mk

@randallagordon

The screenshot shows a macOS terminal window with the following details:

- Title Bar:** jennmoneydollars – bash – Solarized Dark ansi – 127x41
- User and Host:** vagrant@vagrant-ubuntu-trusty-64: /mnt
- Shell:** bash

The terminal history is as follows:

```
Last login: Thu Jun 11 11:54:03 on ttys002
-bash: rbenv: command not found

pumajef:~/Sites/droplets/basedLAMP/stuff/meetups/irlclub 💜 ~*princess jenn*~ on master[!]
$ cd ~

pumajef:~ 💜 ~*princess jenn*~
$ clear
```

@jennschiffer

Share this

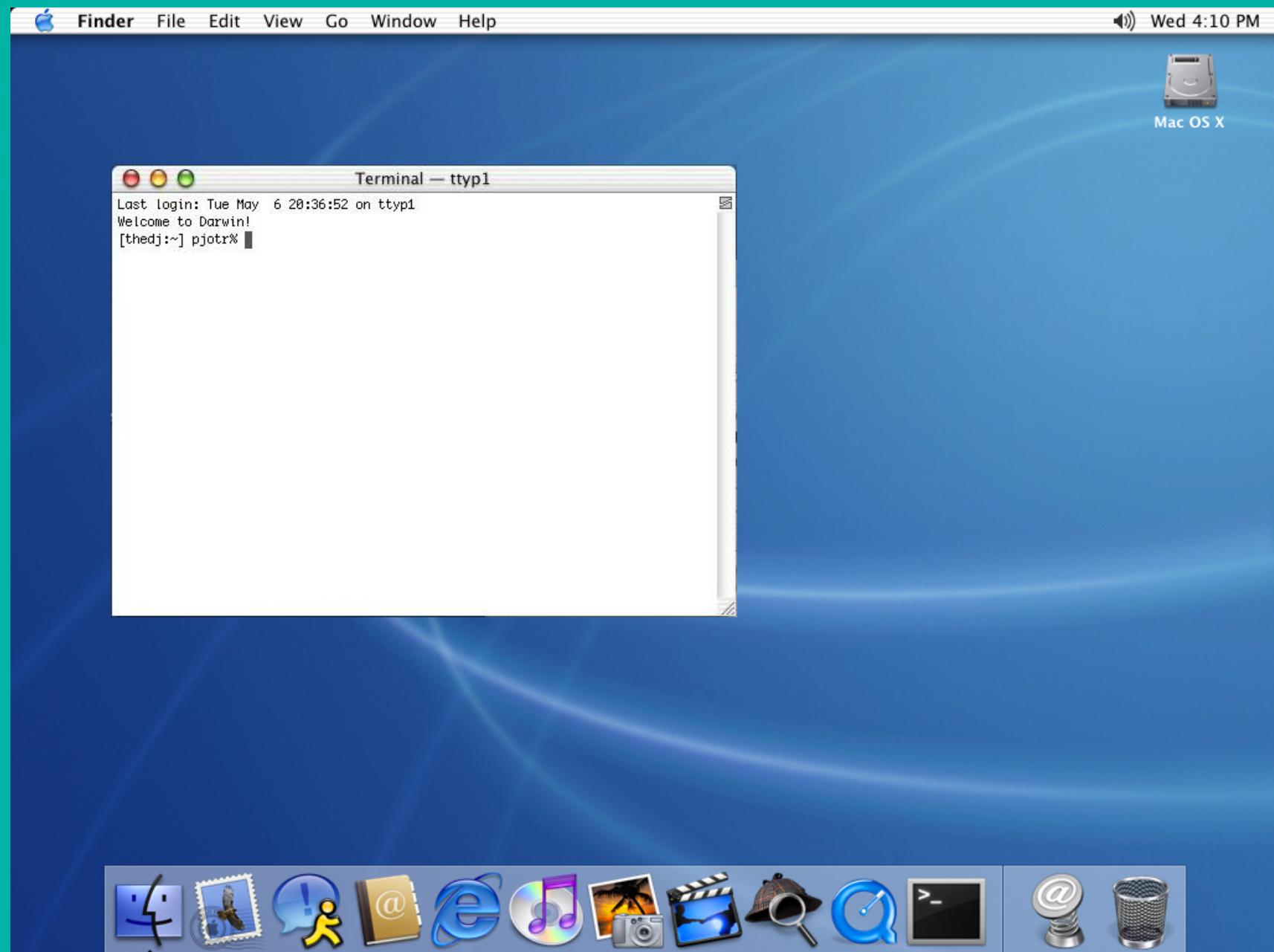
# terminals and command lines and shells

oh my

# what is a shell



# what is a shell



beginning bash

 caution 

commands do exactly what you tell them to  
only download from trusted sources

# man

bash-3.2\$ man echo

ECHO(1)

BSD General Commands Manual

ECHO(1)

## NAME

echo -- write arguments to the standard output

## SYNOPSIS

echo [-n] [string ...]

## DESCRIPTION

The echo utility writes any specified operands, separated by single blank (' ') characters and followed by a new-line ('\n') character, to the standard output.

The following option is available:

-n     Do not print the trailing newline character.

[...]

# beginning bash

# some useful commands

an incomplete list

pwd cd ls  
touch mkdir cp mv  
vi vim emacs nano pico  
rm rmdir chmod chown kill  
# ; . .. \ / \* \$ > >> | & - --  
man cat (meow) sudo find grep ps aux echo scp ssh ping



don't panic

# customizing the bash prompt

# customizing the bash prompt

# customization: editing .bashrc

# customization

ls -A ~

# customization

a Mac OS X gotcha

~~.bashrc~~

.bash\_profile or .profile

**.bash\_profile, .profile, and .bashrc conventions**

*<http://dghubble.com/blog/posts/.bashprofile-.profile-and-.bashrc-conventions/>*

# a list of common text editors

vim

emacs

nano

pico

# customization: editing .bashrc

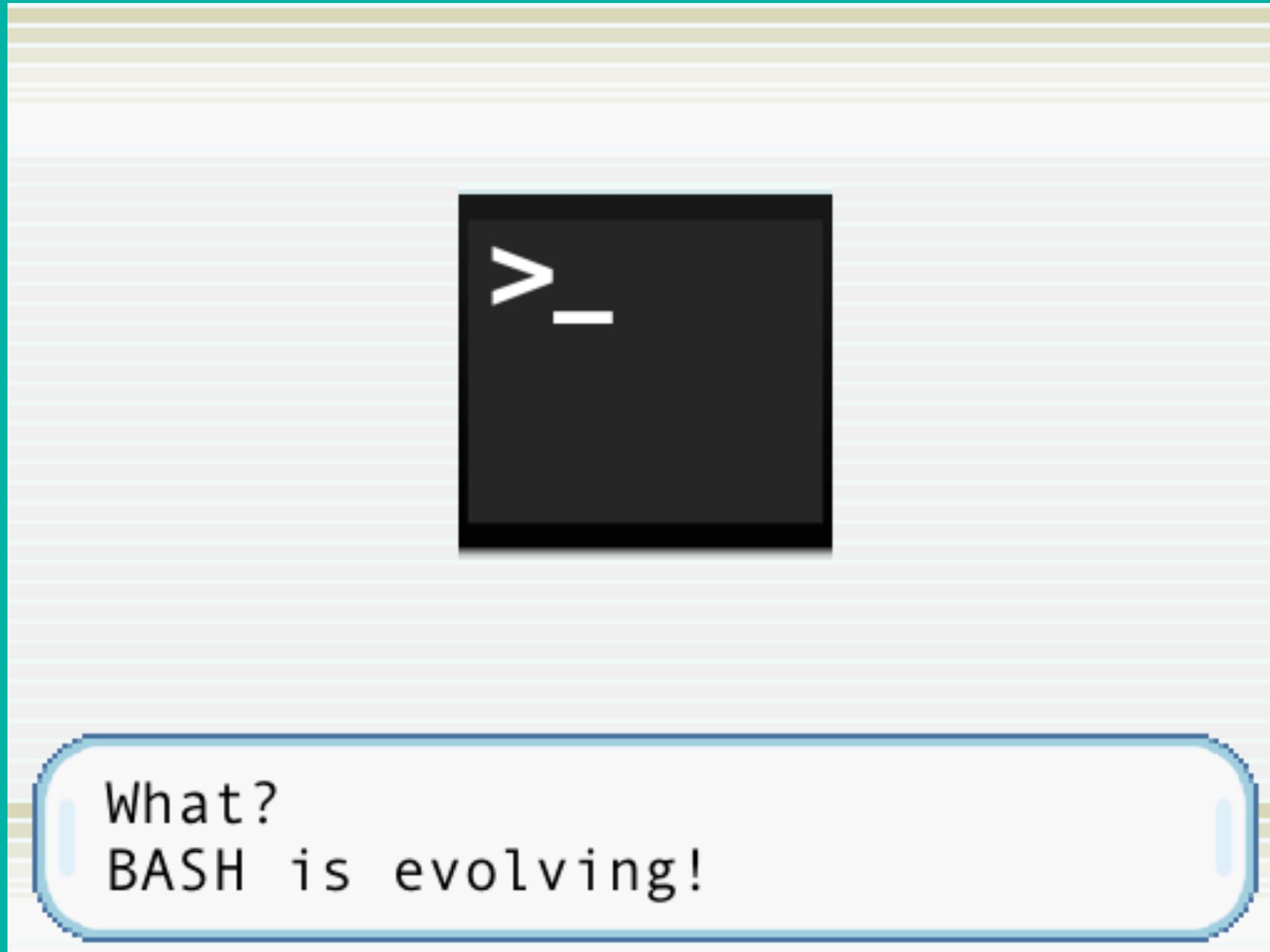
# How to Customize your Bash Prompt on a Linux VPS

– *<https://www.digitalocean.com/community/tutorials/how-to-customize-your-bash-prompt-on-a-linux-vps>*

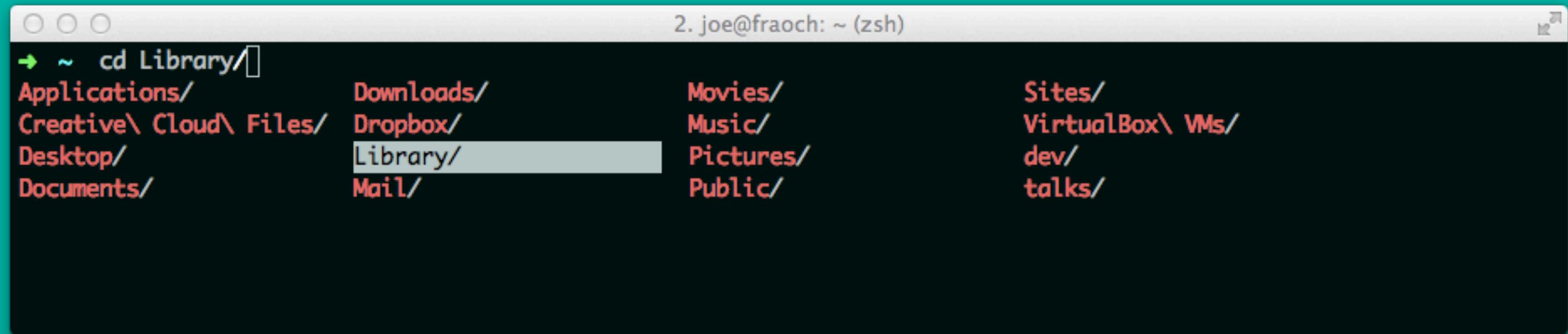


<http://www.zsh.org>

# zsh vs. bash



# zsh: tab completions



A screenshot of a terminal window titled "2. joe@fraoch: ~ (zsh)". The command entered is "cd Library/". The terminal shows a list of directory completions:

- Applications/
- Creative\ Cloud\ Files/
- Desktop/
- Documents/
- Downloads/
- Dropbox/
- Library/
- Mail/
- Movies/
- Music/
- Pictures/
- Public/
- Sites/
- VirtualBox\ VMs/
- dev/
- talks/

The directory "Library/" is highlighted with a light gray rectangle.

<http://code.joejag.com/2014/why-zsh.html>

# zsh: command history



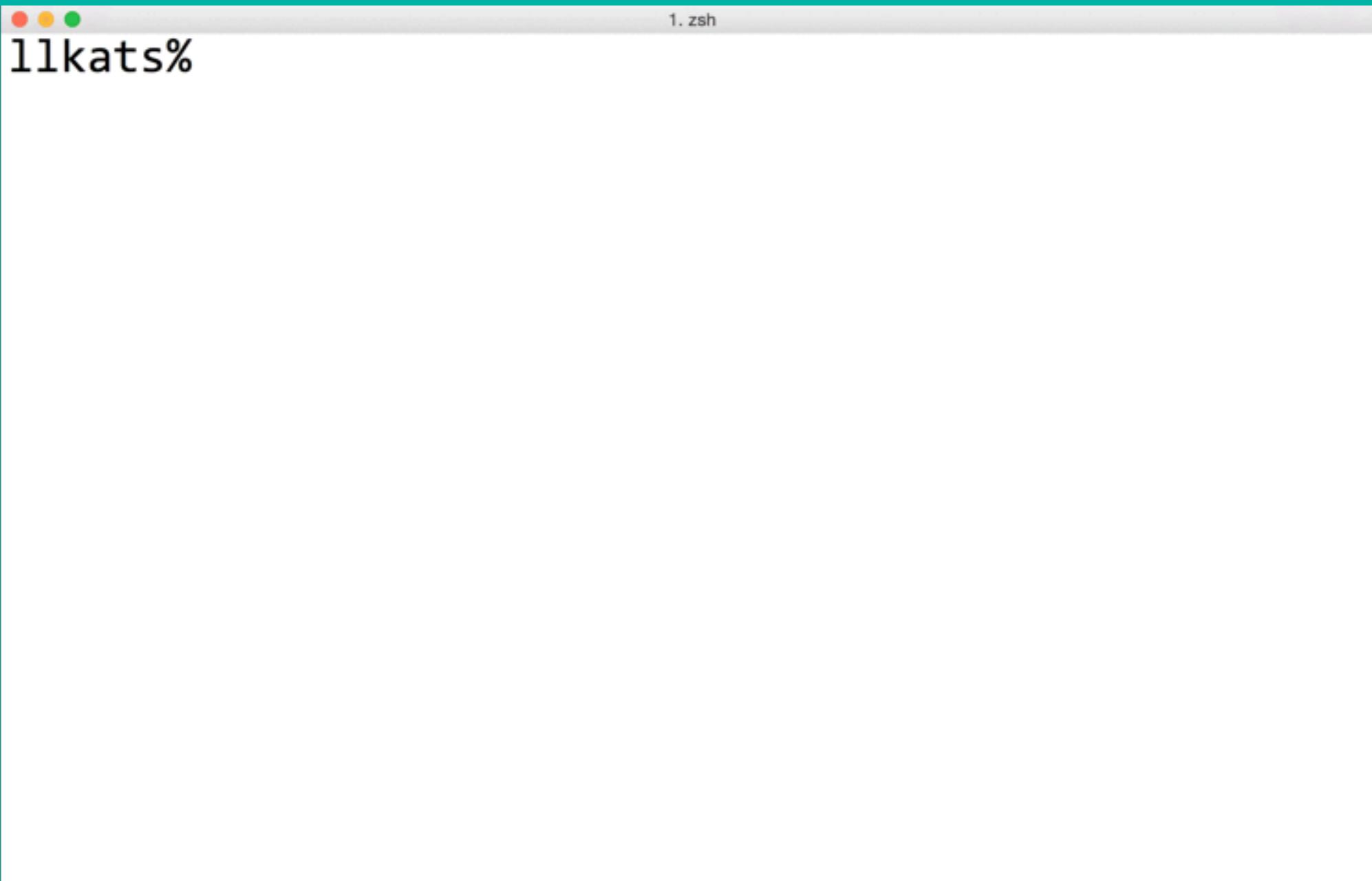
# oh my zsh

## the shell framework

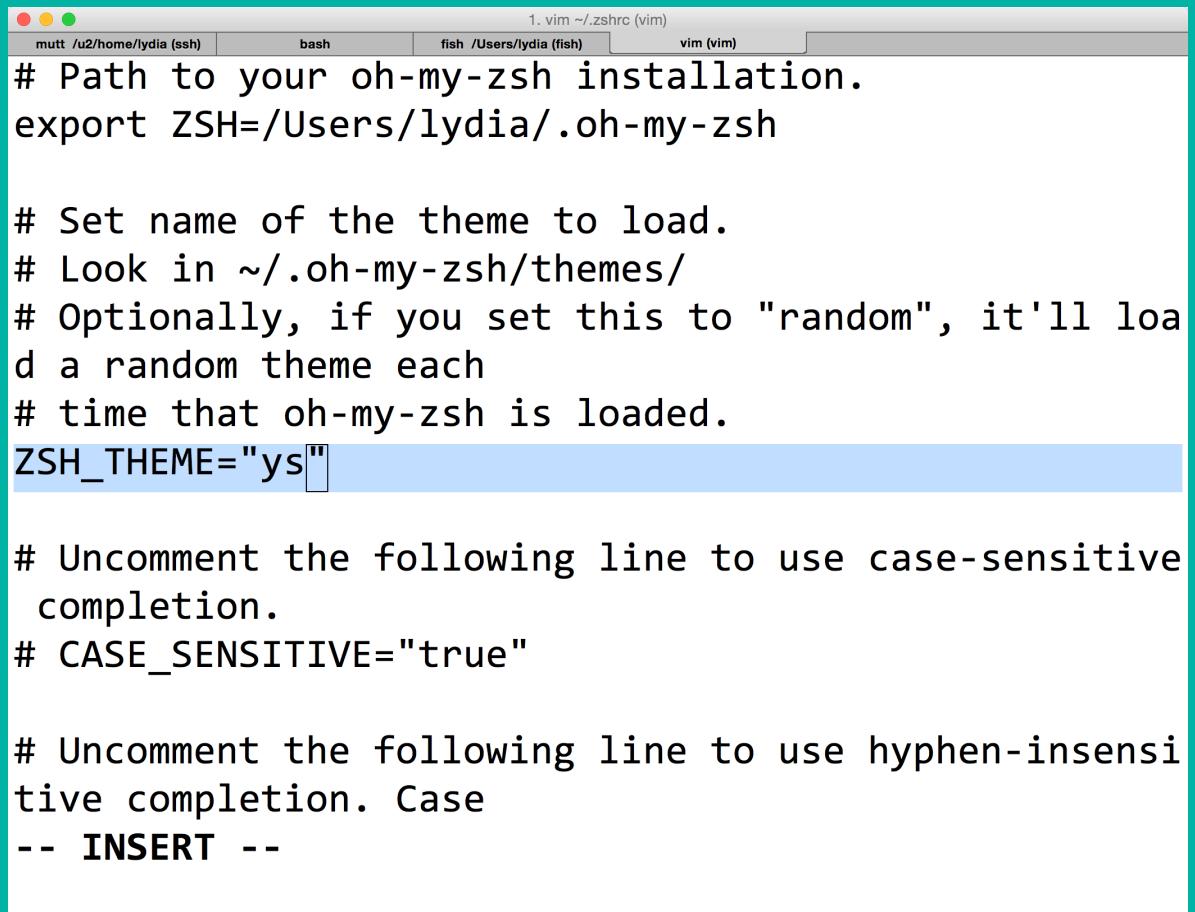
<http://ohmyz.sh/>

<https://github.com/robbyrussell/oh-my-zsh>

# installing oh my zsh



# customizing zsh with oh my zsh



```
# Path to your oh-my-zsh installation.  
export ZSH=/Users/lydia/.oh-my-zsh  
  
# Set name of the theme to load.  
# Look in ~/.oh-my-zsh/themes/  
# Optionally, if you set this to "random", it'll loa  
d a random theme each  
# time that oh-my-zsh is loaded.  
ZSH_THEME="ys"  
  
# Uncomment the following line to use case-sensitive  
completion.  
# CASE_SENSITIVE="true"  
  
# Uncomment the following line to use hyphen-insensi  
tive completion. Case  
-- INSERT --
```

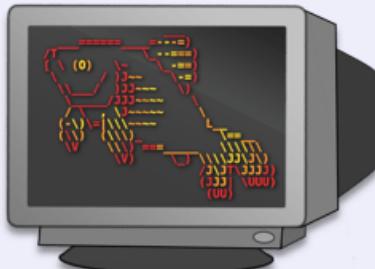
.zshrc

fish

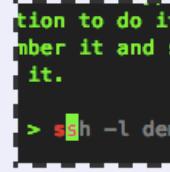
# finally a command line for the 90's

## Finally, a command line shell for the 90s

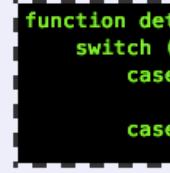
fish is a smart and user-friendly command line shell for OS X, Linux, and the rest of the family.



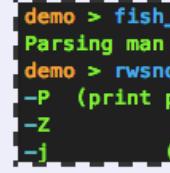
**Autosuggestions**  
fish suggests commands as you type based on history and completions, just like a web browser. Watch out, Netscape Navigator 4.0!



**Sane Scripting**  
fish is fully scriptable, and its syntax is simple, clean, and consistent. You'll never write esac again.



**Man Page Completions**  
Other shells support programmable completions, but only fish generates them automatically by parsing your installed man pages.



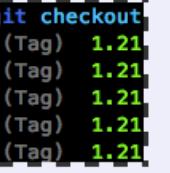
**Glorious VGA Color**  
fish natively supports term256, the state of the art in terminal technology. You'll have an astonishing 256 colors available for use!



**Web Based configuration**  
For those lucky few with a graphical computer, you can set your colors and view functions, variables, and history all from a web page.



**Works Out Of The Box**  
fish will delight you with features like tab completions and syntax highlighting that just work, with nothing new to learn or configure.



# installing fish

## Go fish

[OS X](#) [Linux](#) [Windows](#) [tarball](#)

[Installer](#)



10.6+

Installs to `/usr/local/`

[Homebrew](#)



`brew install fish`

[App](#)



10.6+

[No installation required](#)

# installing fish

OS X   Linux   Windows   tarball

The screenshot shows a grid of six boxes, each representing a different Linux distribution with its logo and a link to either subscribe or download the fish shell. The distributions listed are Ubuntu, Debian, Fedora, openSUSE, CentOS, and gentoo linux.

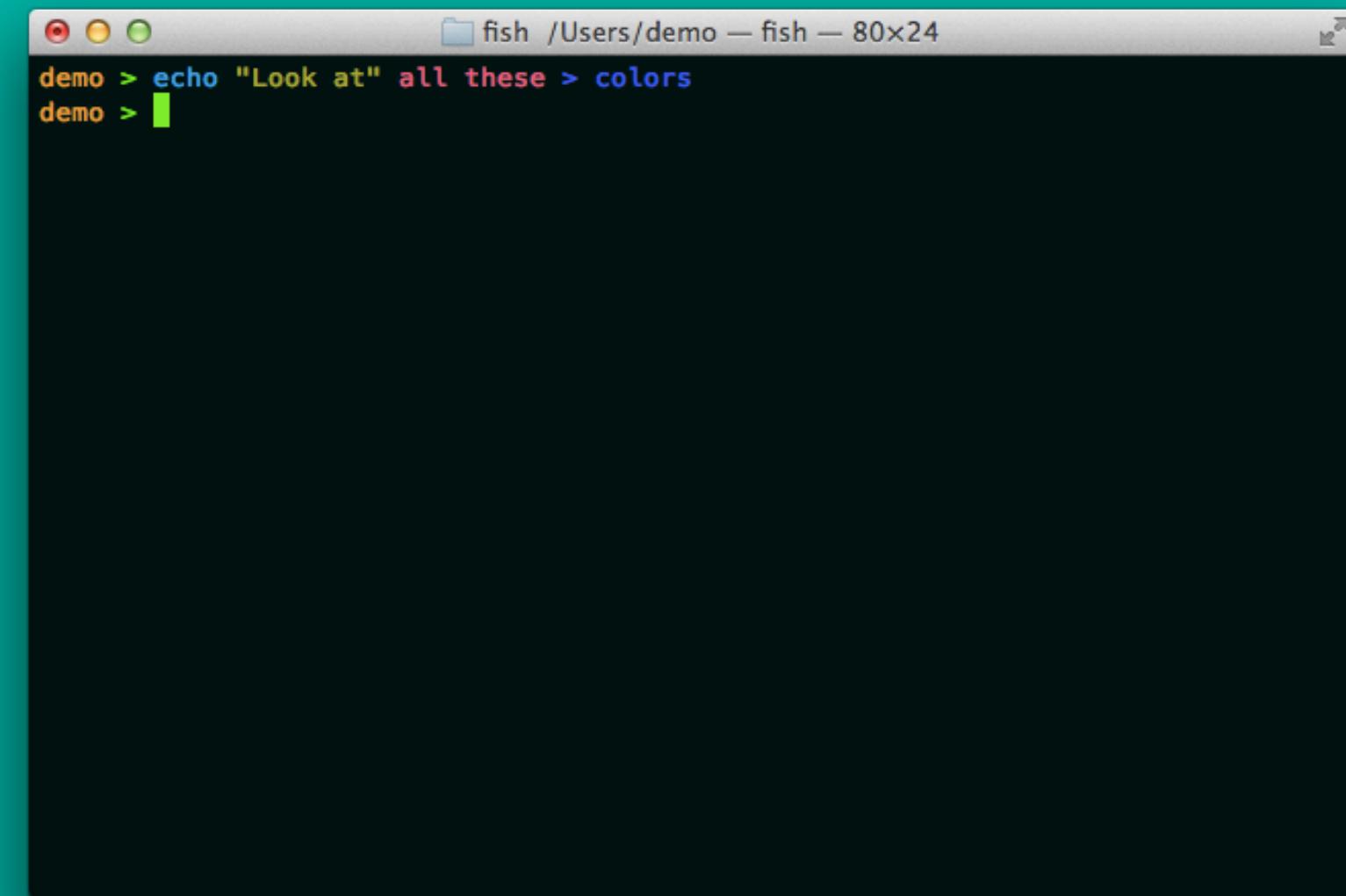
Distribution	Action
ubuntu	<a href="#">Subscribe or Download</a>
debian	<a href="#">Subscribe or Download</a>
fedora	<a href="#">Subscribe or Download</a>
openSUSE	<a href="#">Subscribe or Download</a>
CentOS	<a href="#">Subscribe or Download</a>
gentoo linux	<a href="#">Packages emerge fish</a>

OS X   Linux   Windows   tarball

The screenshot shows a single box for the Windows platform. It contains a note indicating that fish is available in the setup, specifically in the Shells category, along with the Cygwin logo.

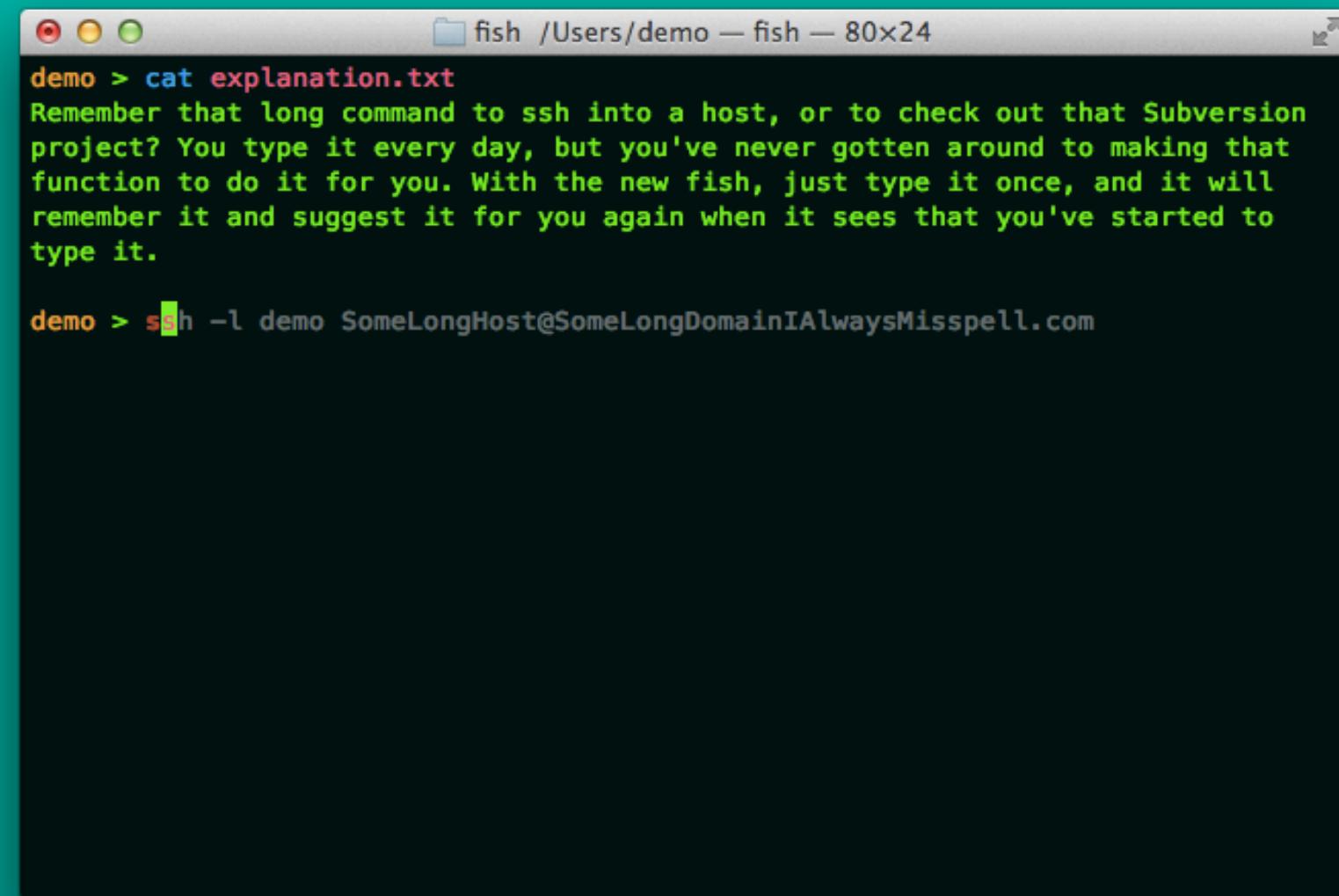
Cygwin  
fish is available in setup,  
in the Shells category

# fish: colors



```
demo > echo "Look at" all these > colors
demo > █
```

# fish: command history

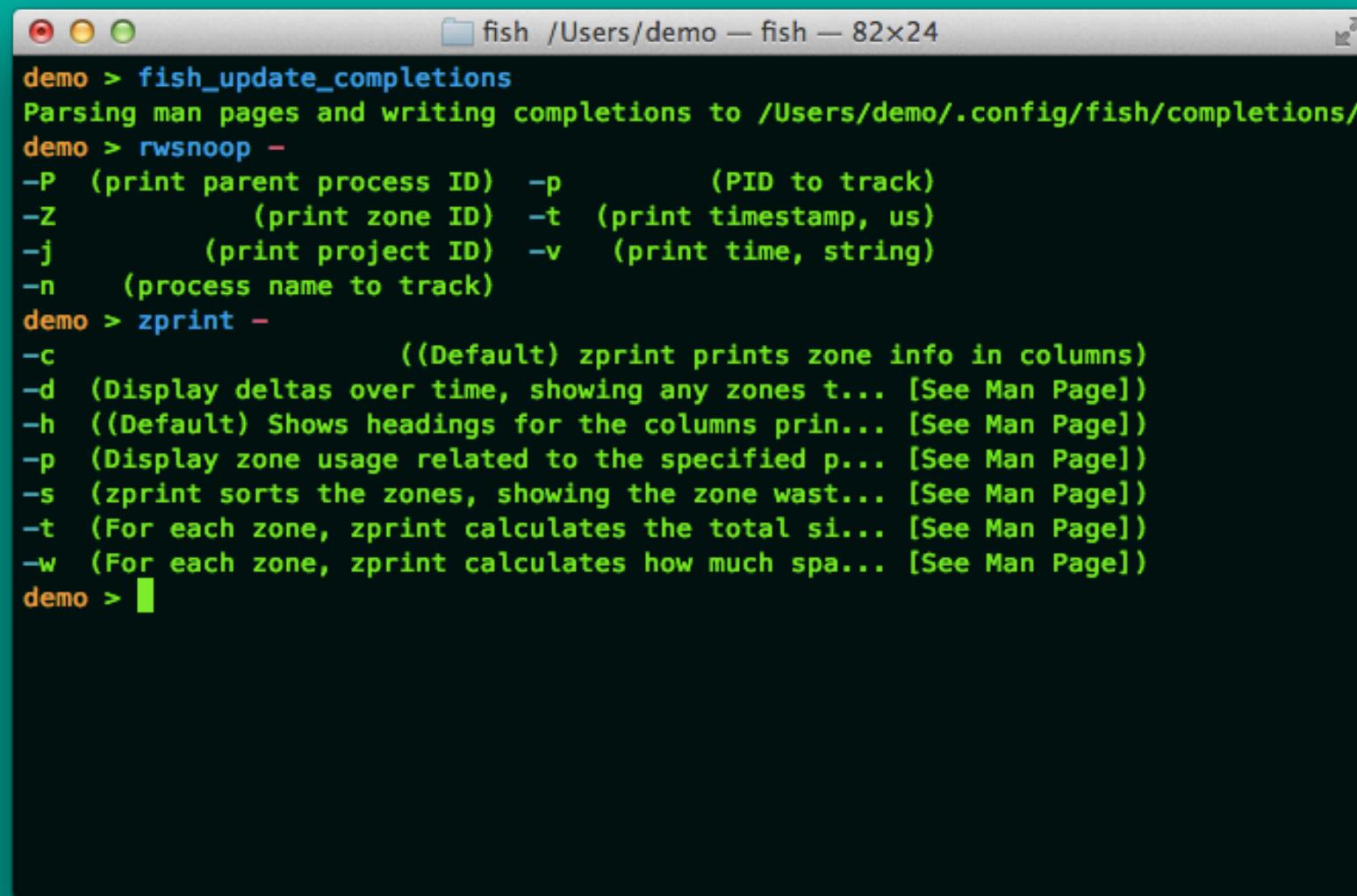


A screenshot of a terminal window titled "fish /Users/demo — fish — 80x24". The window shows the following command history:

```
demo > cat explanation.txt
Remember that long command to ssh into a host, or to check out that Subversion
project? You type it every day, but you've never gotten around to making that
function to do it for you. With the new fish, just type it once, and it will
remember it and suggest it for you again when it sees that you've started to
type it.

demo > ssh -l demo SomeLongHost@SomeLongDomainIAAlwaysMisspell.com
```

# fish: man page completions



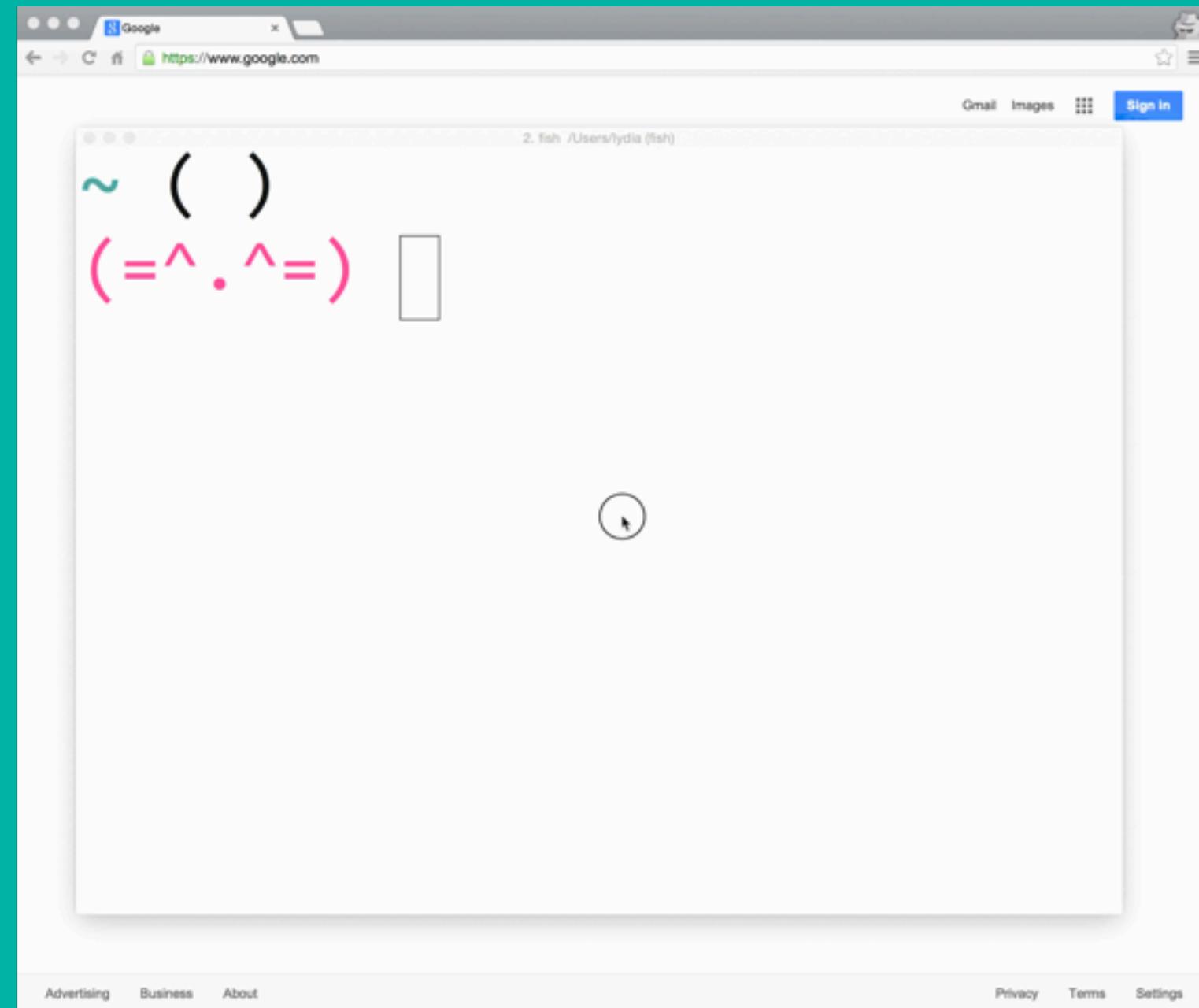
The screenshot shows a terminal window titled "fish /Users/demo — fish — 82x24". The terminal displays the following session:

```
demo > fish_update_completions
Parsing man pages and writing completions to /Users/demo/.config/fish/completions/
demo > rwsnoop -
-P (print parent process ID) -p (PID to track)
-Z (print zone ID) -t (print timestamp, us)
-j (print project ID) -v (print time, string)
-n (process name to track)
demo > zprint -
-c ((Default) zprint prints zone info in columns)
-d (Display deltas over time, showing any zones t... [See Man Page])
-h ((Default) Shows headings for the columns prin... [See Man Page])
-p (Display zone usage related to the specified p... [See Man Page])
-s (zprint sorts the zones, showing the zone wast... [See Man Page])
-t (For each zone, zprint calculates the total si... [See Man Page])
-w (For each zone, zprint calculates how much spa... [See Man Page])
demo > █
```

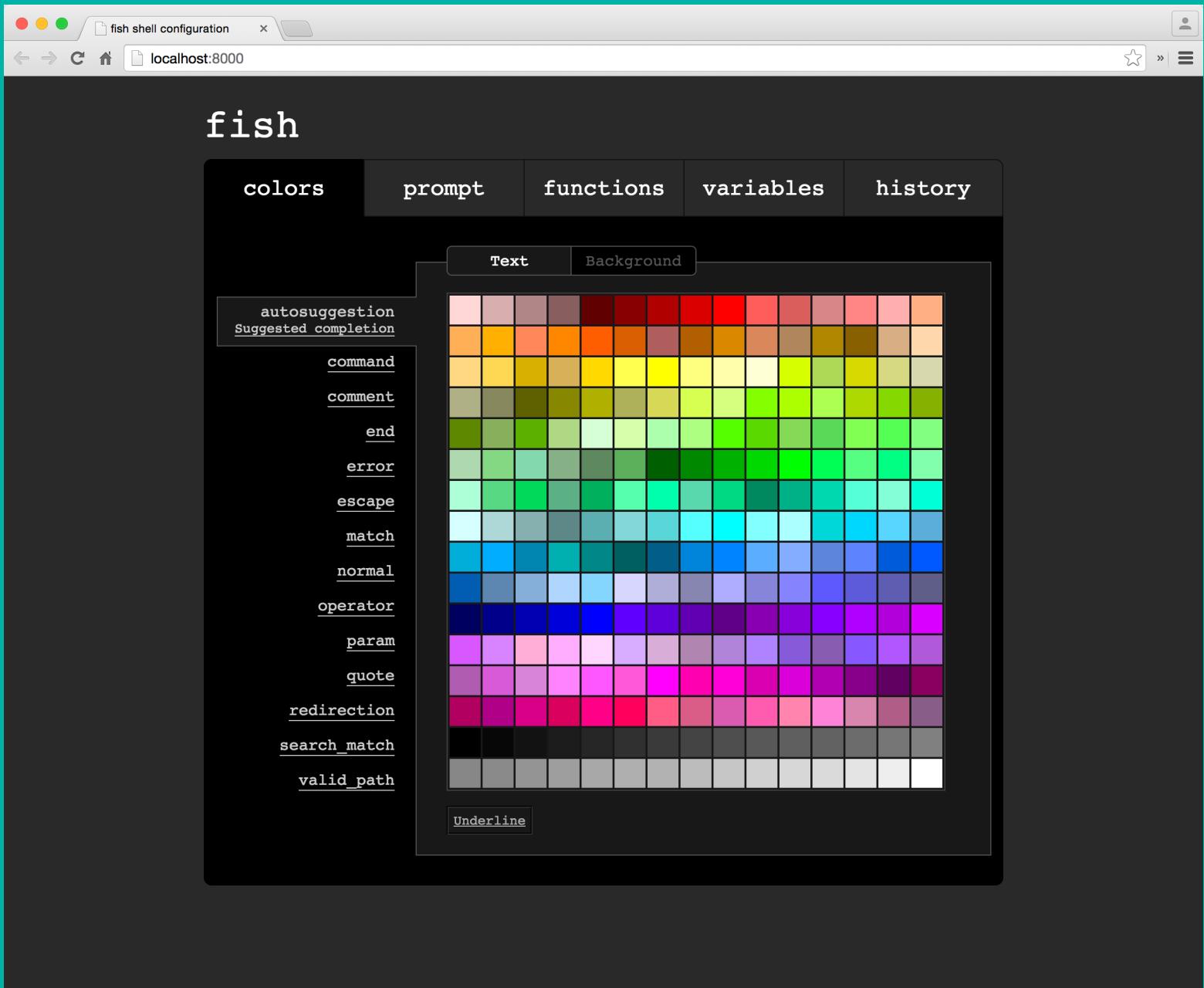
# fish: usage



# customizing fish

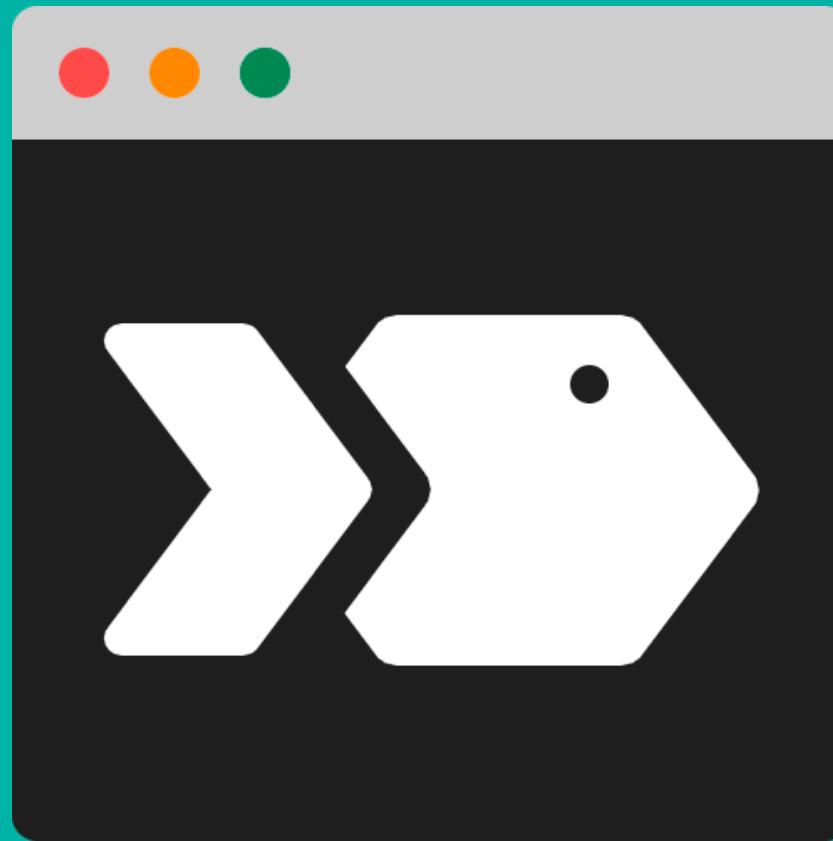


# customizing fish



# fish frameworks

oh my fish



<https://github.com/oh-my-fish/oh-my-fish>

# fish gotchas

## sublist separators

- bash\$: git clone https://github.com/llkats/sample; && cd sample
- fish\$: git clone https://github.com/llkats/sample; and cd sample

# hyperpolyglot

<http://hyperpolyglot.org/unix-shells>

## Grammar

	<a href="#">bash</a>	<a href="#">fish</a>	<a href="#">ksh</a>	<a href="#">tcsh</a>	<a href="#">zsh</a>
<a href="#">simple command</a>	ls	ls	ls	ls	ls
<a href="#">simple command with argument</a>	echo hi	echo hi	echo hi	echo hi	echo hi
<a href="#">simple command with redirect</a>	ls > /tmp/ls.out	ls > /tmp/ls.out	ls > /tmp/ls.out	ls > /tmp/ls.out	ls > /tmp/ls.out
<a href="#">simple command with environment variable</a>	EDITOR=vi git commit	env EDITOR=vi git commit	EDITOR=vi git commit	env EDITOR=vi git commit	EDITOR=vi git commit
<a href="#">pipeline</a>	ls   wc	ls   wc	ls   wc	ls   wc	ls   wc
<a href="#">sublist separators</a>	&&	<i>none</i>	&&	&&	&&
<a href="#">list terminators</a>	; &	; &	; &	; &	; &
<a href="#">group command</a>	{ ls; ls; }   wc	begin; ls; ls; end   wc	{ ls; ls; }   wc	<i>none</i>	{ ls; ls; }   wc
<a href="#">subshell</a>	(ls; ls)   wc	fish -c 'ls; ls'   wc	(ls; ls)   wc	(ls; ls)   wc	(ls; ls)   wc

A screenshot of a terminal window titled "fish /Users/lydia/Dropbox/osbridge (fish)". The window shows a file listing in the current directory (~Dropbox/osbridge) and a command entry field at the bottom.

The file listing includes:

- screen (Executable, 290kB)
- screencapture (Capture images from the screen an...)
- screengif (Executable, 527B)

The command entry field shows the user has typed "clear" and is currently typing "screen".

```
fish /Users/lydia/Dropbox/... ~ (zsh)
1. fish /Users/lydia/Dropbox/osbridge (fish)

~/Dropbox/osbridge ( master* )
(=^.^=) screen
screen                                (Executable, 290kB)
screencapture (Capture images from the screen an...)
screengif                               (Executable, 527B)
~/Dropbox/osbridge ( master* )
(=^.^=) clear
```

# use the tool for the job

bash: hella customizable and super prevalent

zsh: all that bash plus customizability and more

fish: helpful out of the pkg

# next steps

# concepts: redirection and pipes

stdin, stdout, stderr

```
grep -Po 'filename=\S*' /sample/path/example.log |  
cut -d= -f2- | sed -r 's/.*(\.[^.]+)$/\1/' |  
tr A-Z a-z | sort | uniq -c |  
sort -rn | less
```

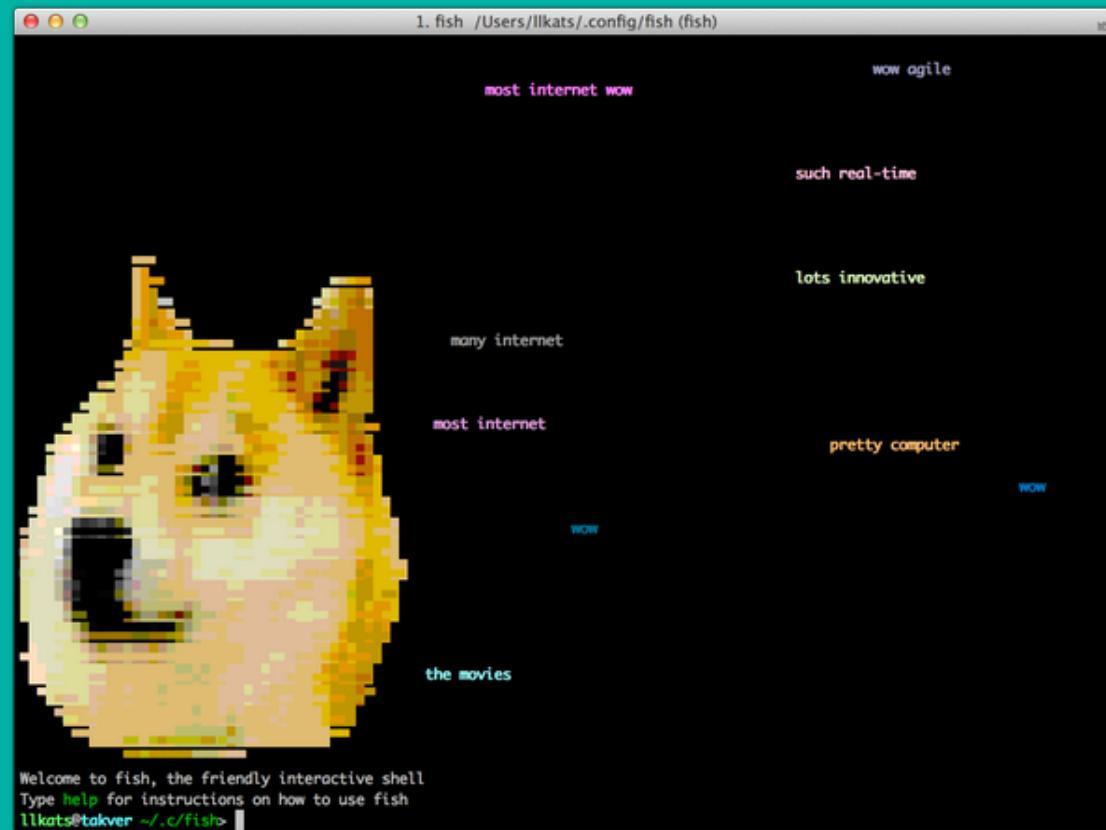
# dotfiles on github

The screenshot shows a GitHub search results page for the query "dotfiles". The search bar at the top contains "dotfiles". On the left, there's a sidebar with navigation links: "Repositories" (51,255), "Code" (213,623), "Issues" (7,429), and "Users" (5). Below that is a "Languages" section showing counts for VimL (20,315), Shell (16,851), Emacs Lisp (2,943), Python (1,278), Lua (893), Perl (840), Ruby (737), JavaScript (543), Haskell (372), and CSS (190). At the bottom of the sidebar are links for "Advanced search" and "Cheat sheet". The main area displays a list of repository results. The first result is "justone/dotfiles" (Perl, 153 stars, 180 forks, updated Dec 25, 2014). The second is "suztomo/dotfiles" (Emacs Lisp, 12 stars, 67 forks, updated Jun 30, 2013). The third is "henrik/dotfiles" (Shell, 158 stars, 31 forks, updated Apr 9). The fourth is "holman/dotfiles" (@holman does dotfiles, 3,096 stars, 1,996 forks, updated May 6). The fifth is "mathiasbynens/dotfiles" (VimL, 11,709 stars, 4,668 forks, updated 7 days ago). The sixth is "dreadatour/dotfiles" (VimL, 57 stars, 11 forks, updated Apr 15). The seventh is "gf3/dotfiles" (VimL, 649 stars, 96 forks, updated 19 days ago). The eighth is "joedicastro/dotfiles" (Emacs Lisp, 486 stars, 70 forks, updated 23 days ago).

Repository	Language	Stars	Forks	Last Update
justone/dotfiles	Perl	153	180	Dec 25, 2014
suztomo/dotfiles	Emacs Lisp	12	67	Jun 30, 2013
henrik/dotfiles	Shell	158	31	Apr 9
holman/dotfiles	Shell	3,096	1,996	May 6
mathiasbynens/dotfiles	VimL	11,709	4,668	7 days ago
dreadatour/dotfiles	VimL	57	11	Apr 15
gf3/dotfiles	VimL	649	96	19 days ago
joedicastro/dotfiles	Emacs Lisp	486	70	23 days ago

# fun open source shell projects

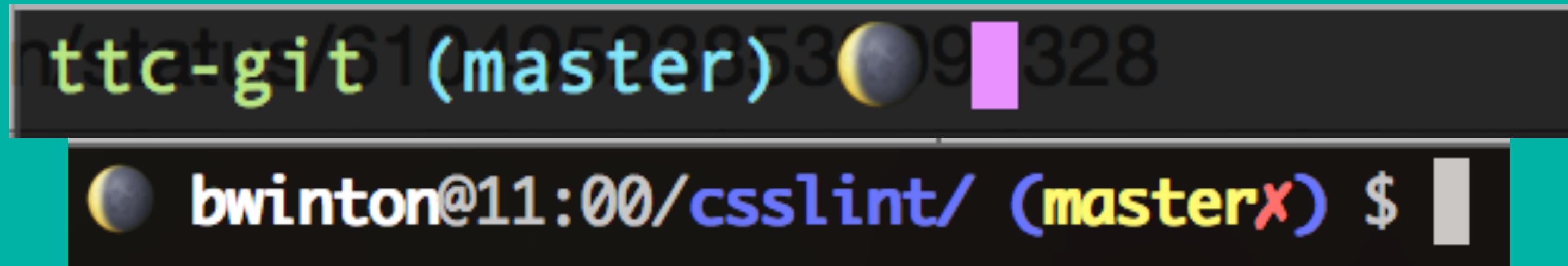
## doge shell



<https://github.com/thiderman/doge>

# fun open source shell projects

moonmoji



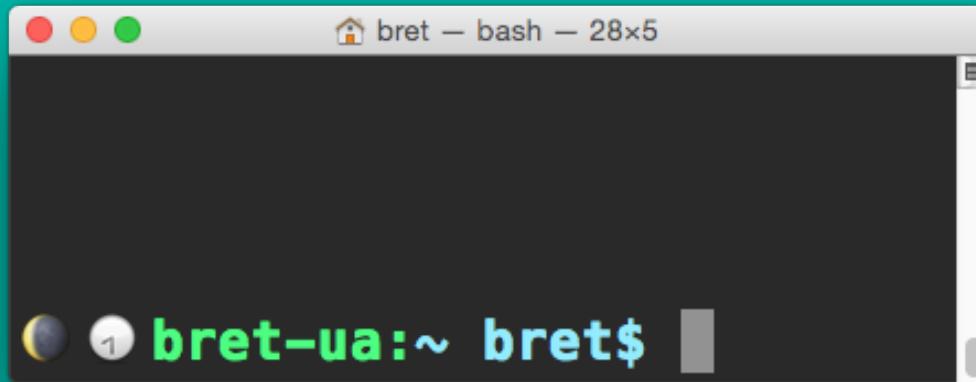
@lauradegroot

@bwinton

<https://github.com/rmillr/moonmoji>

# fun open source shell projects

## clockmoji



@bretolius

<https://github.com/Flet/clockmoji>

# moar shell widgets!

<https://github.com/alebcay/awesome-shell> (bash, zsh)

# resources

Digital Ocean Documentation

Eric Raymond's writings <http://www.catb.org/~esr/writings/taouu/html/ch02s01.html>

Jessica Dillon's Quickleft posts and ebook

# special thanks

- the organizers and volunteers of OS Bridge
- my coworkers at craigslist
- @rmurphey
- @bwinton, @muanchiou, @jackyalcine, @jennschiffer, @apblardo, @jllord, @ekrubnivek, @RandallAGordon, @bretolius, @makenai, @McDonnellDean, @rmill, @waferbaby, @notwaldorf, @rhodesjason, @maxogden, @joshtriplett, @potch, @lauradegroot, @adcadien, @daveg, @jergason, @saikobee, @dam, @richardiii, @gggritso, @jonmagic, @aredridel, @dbaker\_h, @inky, @jonursenbach, @jergason, @jedireza, @thealphanerd, @zeke, @JdotJdotF, @izs
- Alex Dergachev, screengif, <https://github.com/dergachev/screengif>
- to you!



cheers