

How The Command-Line Can Give You Superpowers

Eric Fung • @gnufmuffin



androidT0 • 2018-11-06 • #AndroidT0

Command line?

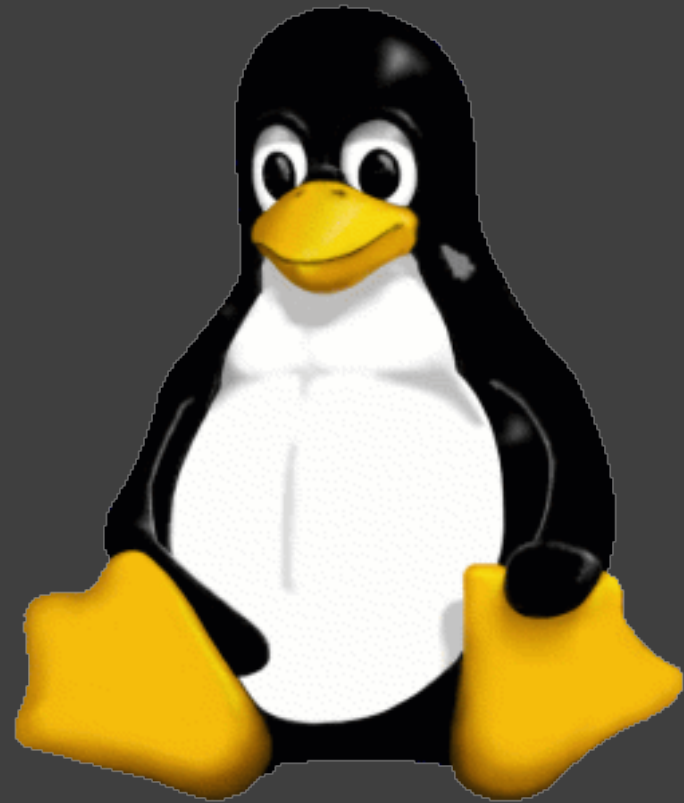
```
Last login: Sun Nov  4 11:46:00 on ttys003
11:46:27 [efung@jonagold:~]$ ls /usr/local/Cellar/bash-completion/1.3.3/etc/
bash_completion  bash_completion.d  profile.d
11:46:35 [efung@jonagold:~]$ cd !$
cd /usr/local/Cellar/bash-completion/1.3.3/etc/
11:46:43 [efung@jonagold:etc]$ head profile.d/bash_completion.sh
# Check for interactive bash and that we haven't already been sourced.
[ -z "$BASH_VERSION" -o -z "$PS1" -o -n "$BASH_COMPLETION" ] && return

# Check for recent enough version of bash.
bash=${BASH_VERSION%.*}; bmajor=${bash%.*}; bminor=${bash#*.}
if [ $bmajor -gt 3 ] || [ $bmajor -eq 3 -a $bminor -ge 2 ]; then
    if shopt -q progcomp && [ -r /usr/local/Cellar/bash-completion/1.3.3/etc/bas
h_completion ]; then
        # Source completion code.
        . /usr/local/Cellar/bash-completion/1.3.3/etc/bash_completion
    fi
fi
11:46:50 [efung@jonagold:etc]$ ls ~/bin
acap          fix-dropbox.sh      humpty.sh
atext         git-prompt.sh
dumpfarm      git-remove-submodule
11:46:57 [efung@jonagold:etc]$
```

```
C:\>dir /w |
```



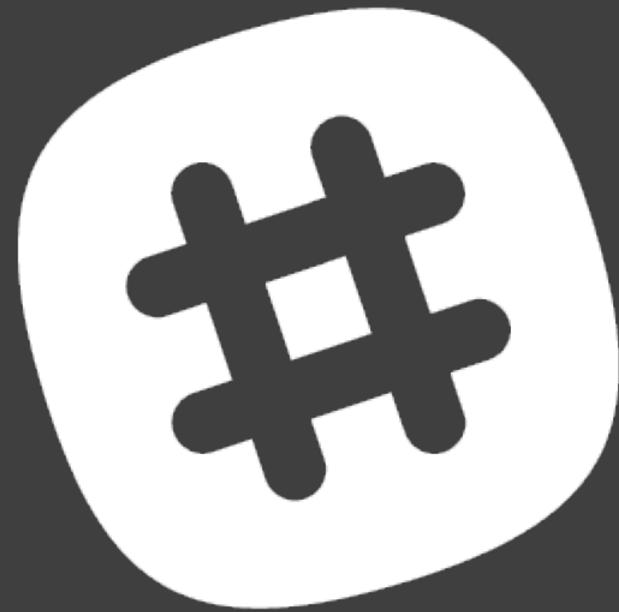
```
efung@mars:~$ ls -l
```



```
# pm list packages
```



/remind me Lunch w/Gemma at noon



Command-line interface: a way of interacting with a computer by giving it commands using text

Why would I use the command line?

- ▶ Gets you results more **quickly**
- ▶ Gives you more ways to **solve problems**
- ▶ Makes you more **versatile**

What superpowers?

- ▶ Learn to **type less**
- ▶ **Teleport** to your destination
- ▶ Search haystacks and **find** needles
- ▶ Rapidly **integrate** with remote APIs
- ▶ **Transmogrify** bits of data
- ▶ Control devices **telekinetically**

The basics

- Terminal
- Shell
- Package manager

a

b

c

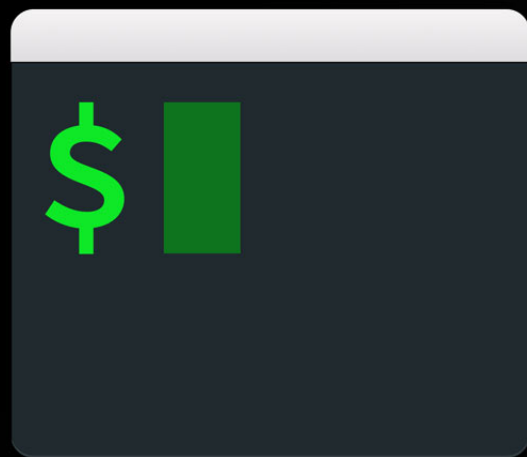
d

e

f

Terminal: iTerm2

- ▶ Extensive features and customization
- ▶ Favorites:
 - ▶ ⌘-click to open URLs and files
 - ▶ Highlights all search results
 - ▶ ⌘-; autocompletes from anything on screen



iTerm2

iTerm2 is a terminal emulator for macOS that does amazing things.



Shell: bash

- ▶ macOS default shell
- ▶ Widely used → lots of support
- ▶ May want to install newer version



Homebrew

The missing package manager for macOS

Package manager: Homebrew

- ▶ All kinds of software (e.g. ports of Linux tools)
- ▶ Over 4000+ packages
- ▶ Very easy to use

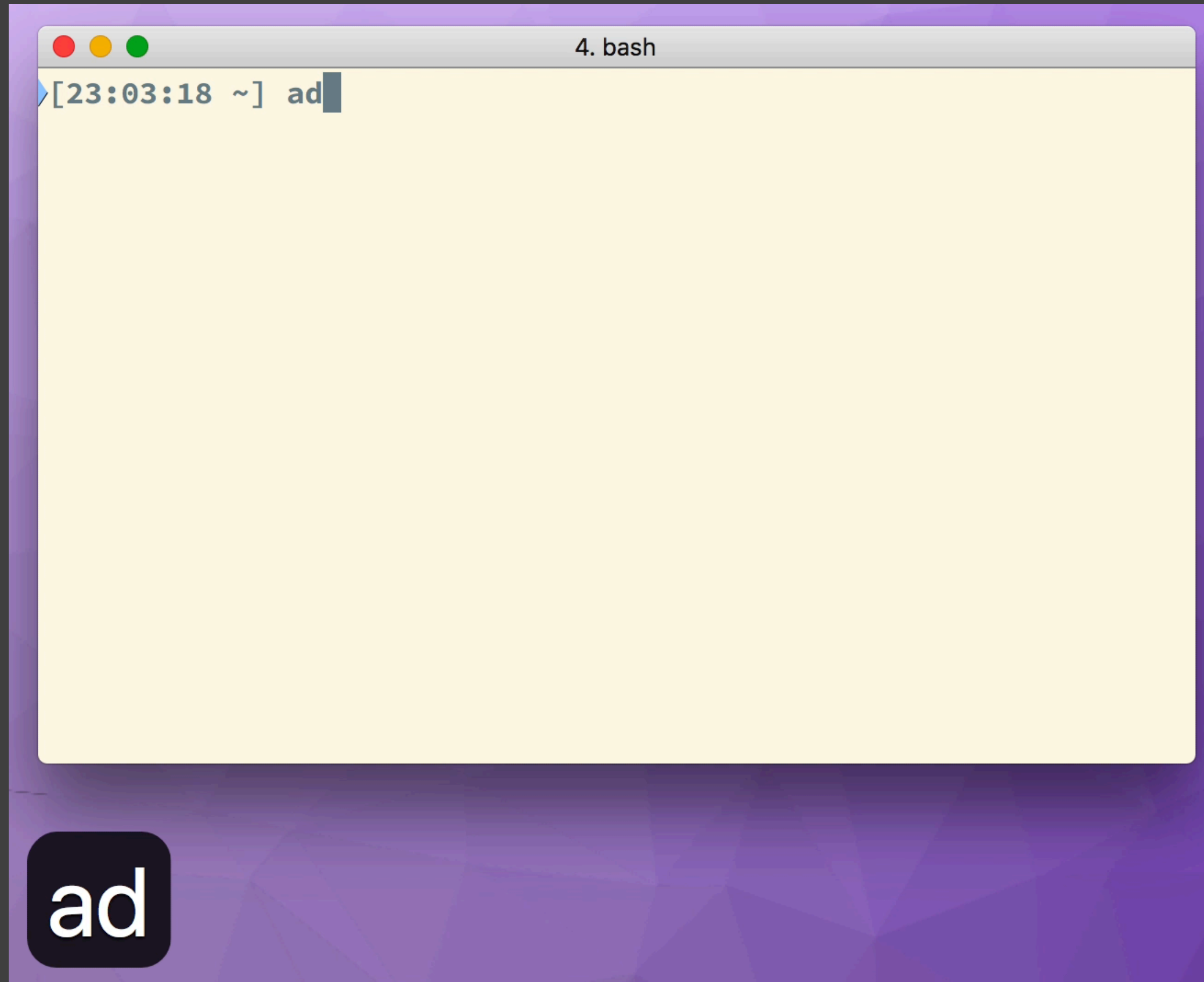
Typing less

Tab completion

- ▶ Built-in to bash
- ▶ Context sensitive
- ▶ For command names, option names, and option *values* too

adb completion

- ▶ `install`
 - ▶ completes with APKs in local directory
- ▶ `uninstall`
 - ▶ completes with installed application IDs
- ▶ `global -s` option
 - ▶ completes with serial numbers
- ▶ `shell pm`
 - ▶ completes with subcommands
- ▶ `pull`
 - ▶ completes from device filesystem



Demo:adb uninstall completion

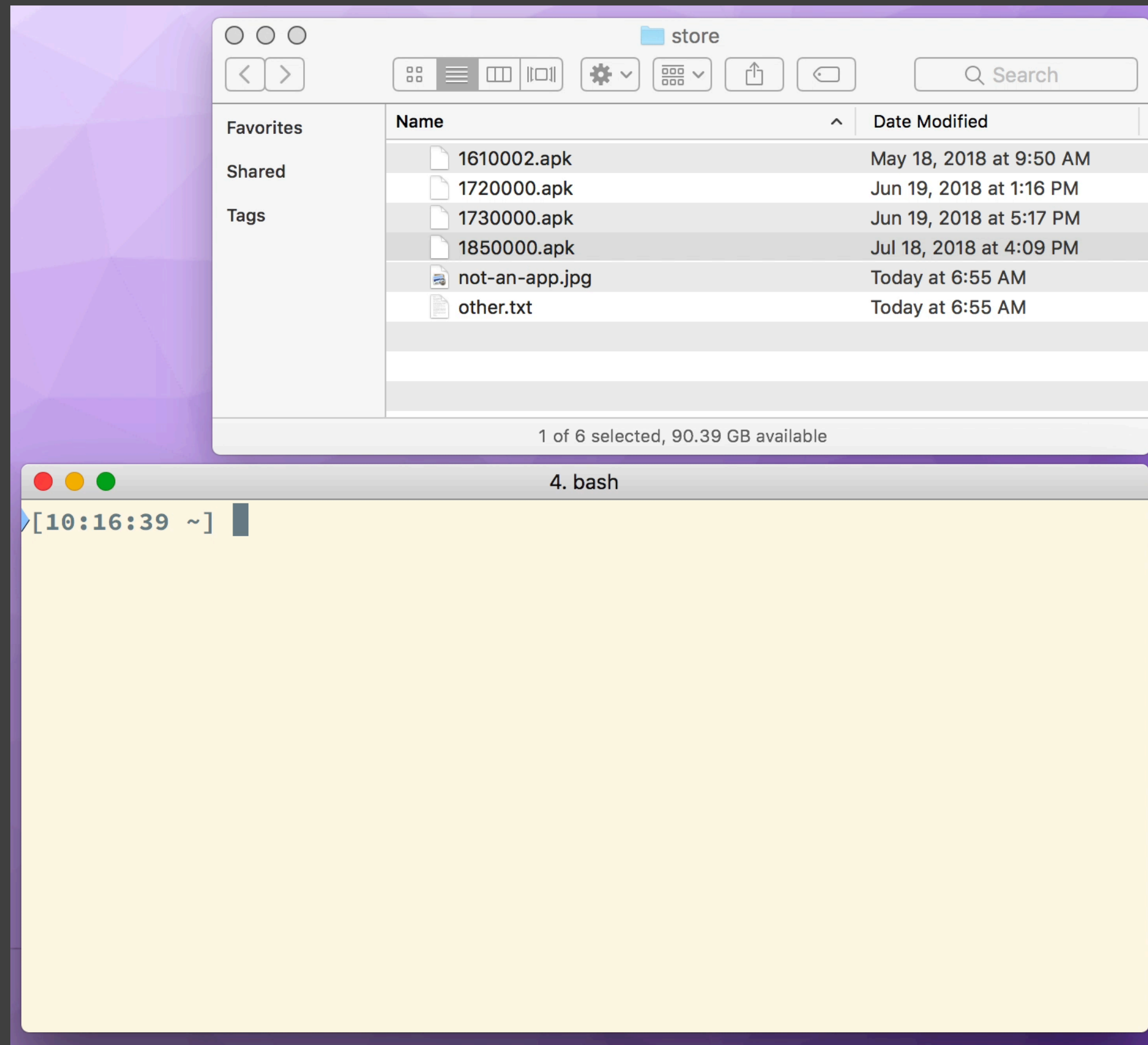
Working with Finder

From shell to Finder

- ▶ `open <thing>`
 - ▶ Equivalent to double-clicking on `<thing>`
- ▶ `open .`
 - ▶ Opens Finder with current directory

From Finder to shell

- ▶ Drag-and-drop file(s) onto Terminal
 - ▶ Pastes their filename(s), space-separated



Demo: Drag files to shell → paste filename

Navigation



cd 101

- ▶ `cd app/src/main/res/layout`
- ▶ `cd ../../java`
 - ▶ Go to a subdirectory
- ▶ `cd -`
 - ▶ Go to previous directory



4. bash

▶ [11:00:56 ~] j



Demo: cd with superpowers

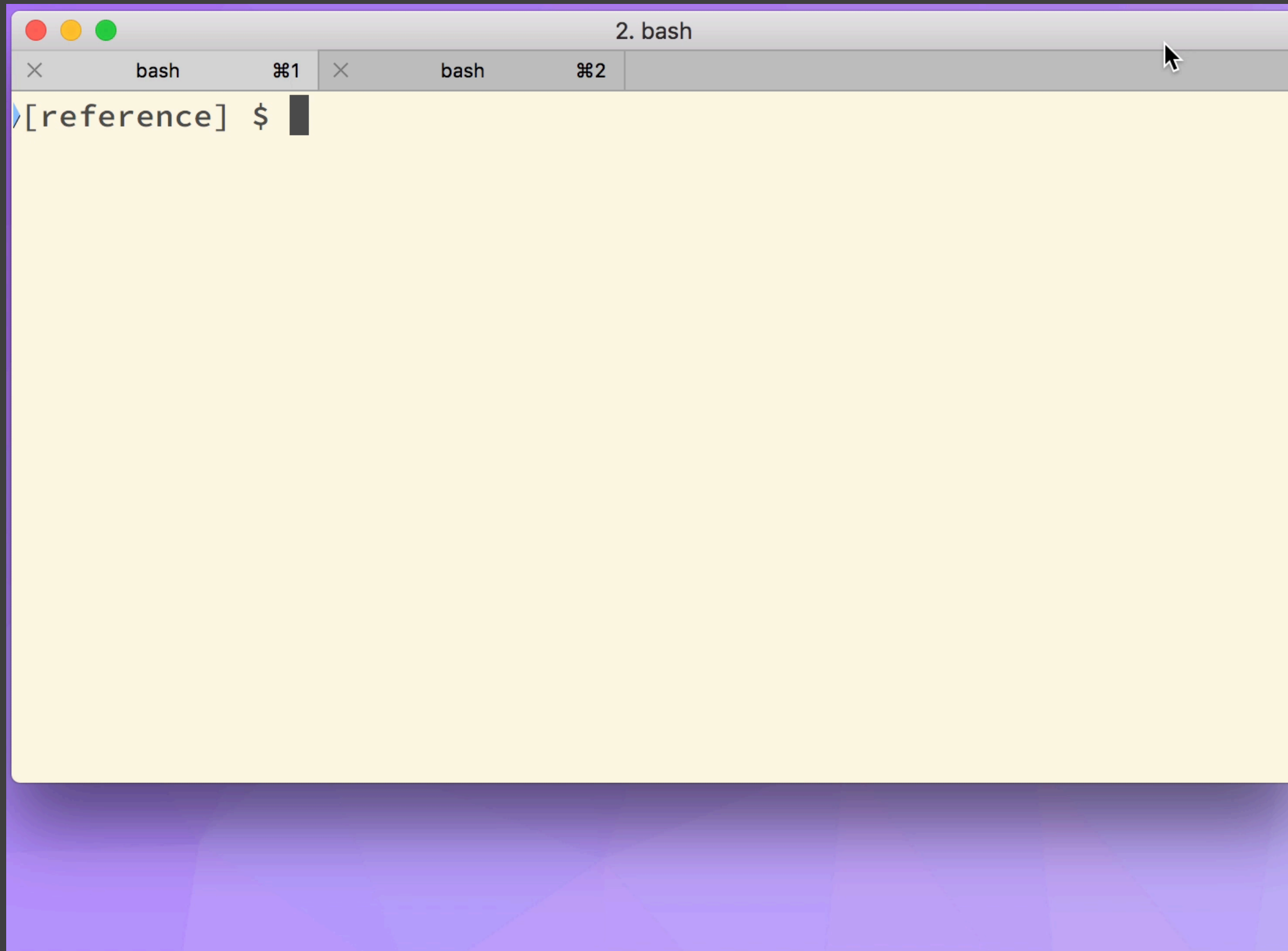
autojump

- ▶ Provide fast access to directories further away
- ▶ Selects directories based on usage
- ▶ Caveat: must be visited first

autojump

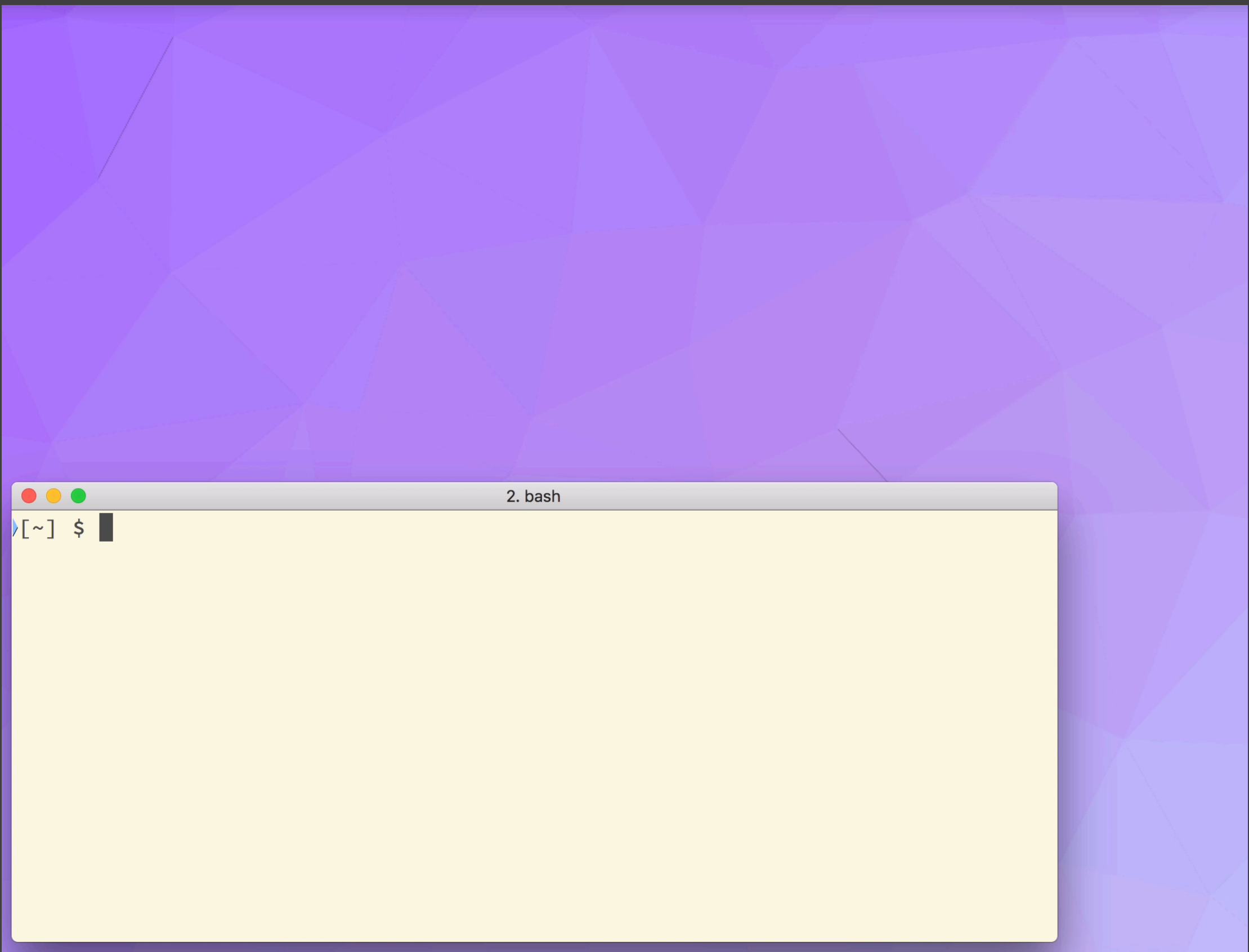
- ▶ `j name`
 - ▶ Go to most-used directory containing name
- ▶ `jc name`
 - ▶ Go to a subdirectory containing name
- ▶ `j name <Tab>`
 - ▶ Offers choices when there are multiple matches

Searching



ripgrep

- ▶ Supports `grep` options
- ▶ *Much* faster
- ▶ Knows about file types
- ▶ Respects revision control ignores
- ▶ Support for emoji 🐱



I/O



I/O 101

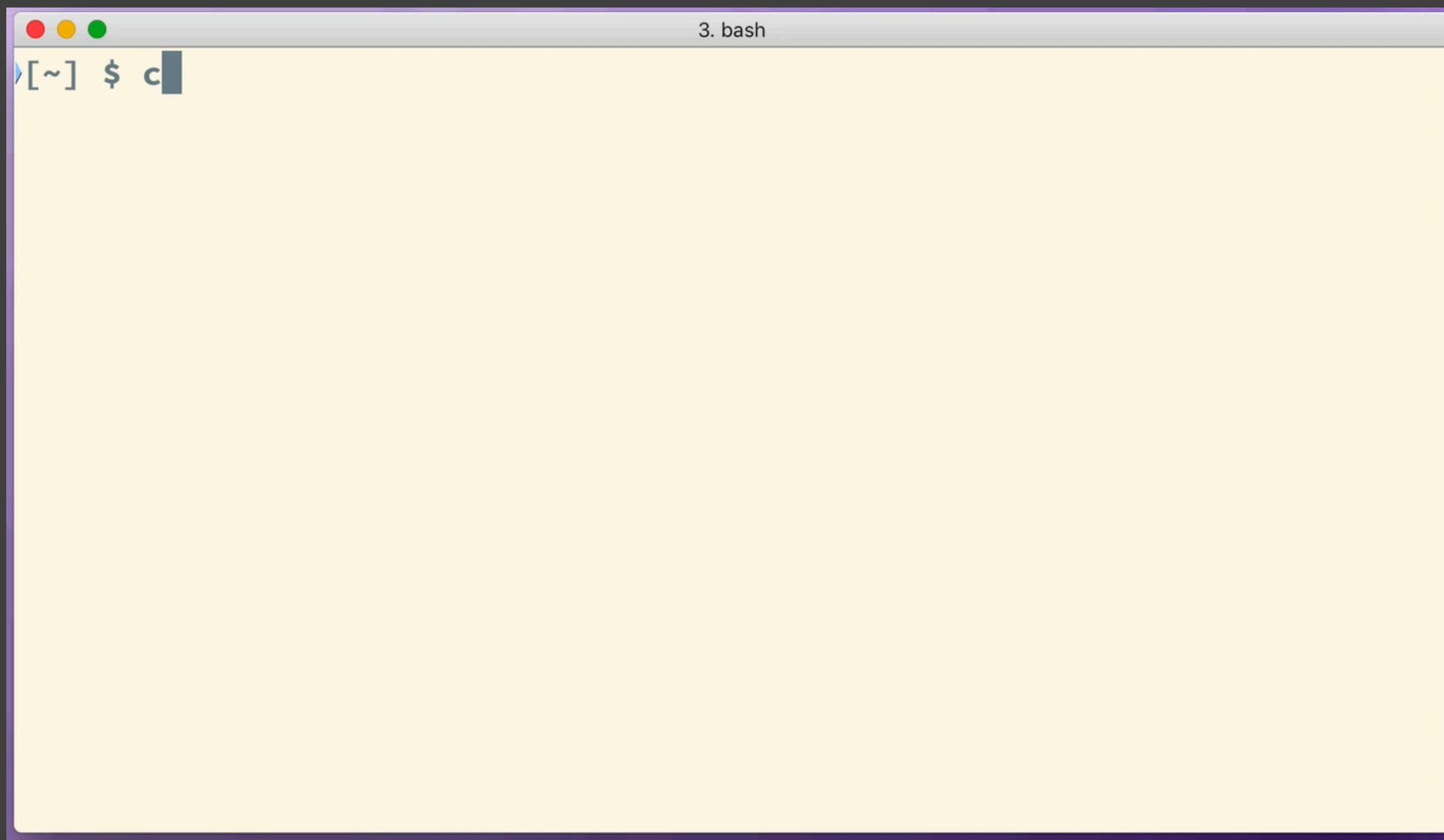
- ▶ Streams: stdin and stdout
- ▶ Redirection: > and <
- ▶ Pipelines: |

Working with Clipboard

- ▶ `pbcopy` (`⌘-C` for the shell)
 - ▶ Reads from `stdin`, copies to clipboard
- ▶ `pbpaste` (`⌘-V` for the shell)
 - ▶ Copies contents of clipboard, writes to `stdout`

The Web





curl



- ▶ Comprehensive client for working with Internet protocol transfers
 - ▶ *"the number of features will make your head spin!"*
- ▶ Supports HTTP, SMTP, IMAP, RTSP, and lots more
- ▶ Designed to be unattended

Debugging endpoints: --include

```
$ curl 'https://search.icons8.com/api/iconsets/v4/search' --include
HTTP/2 200
server: keycdn-engine
date: Thu, 09 Aug 2018 19:39:52 GMT
content-type: application/json; charset=utf-8
content-length: 96
cache-control: max-age=604800
expires: Thu, 16 Aug 2018 19:39:52 GMT
x-cache: HIT
x-shield: active
x-edge-location: usch
access-control-allow-origin: *
accept-ranges: bytes
```

Sending Firebase push notification

```
$ curl \
  --header "Authorization: key=${FIREBASE_LEGACY_SERVER_KEY}" \
  --header "Content-Type: application/json" \
  --data payload.json \
  https://fcm.googleapis.com/fcm/send
```

► More details in the [Firebase docs](#)

JSON



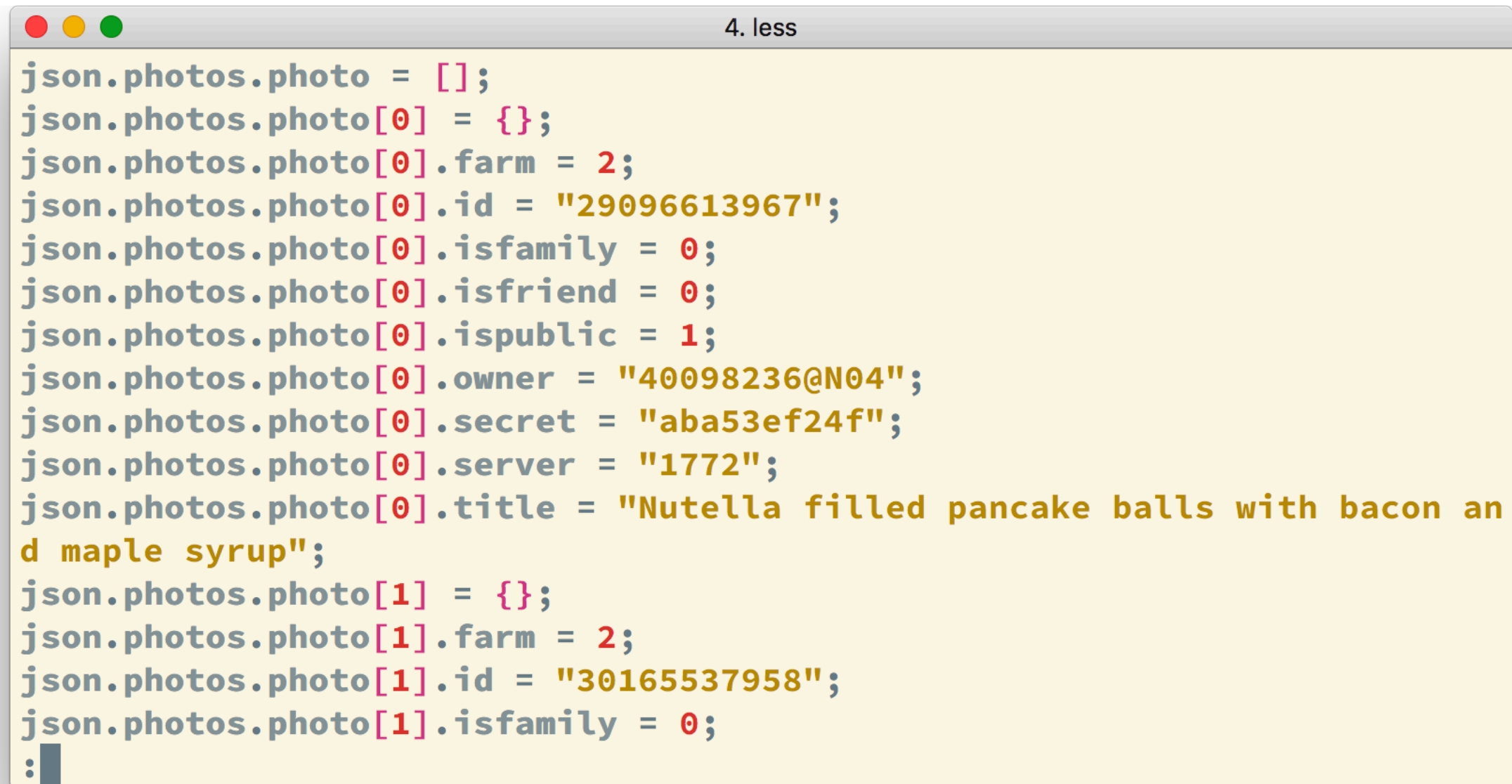
4. bash

```
[~] $ curl "https://api.flickr.com/services/rest/?method=flickr.photos.getRecent&api_key=${FLICKR_KEY}&per_page=8&format=json&nojsoncallback=1"
```

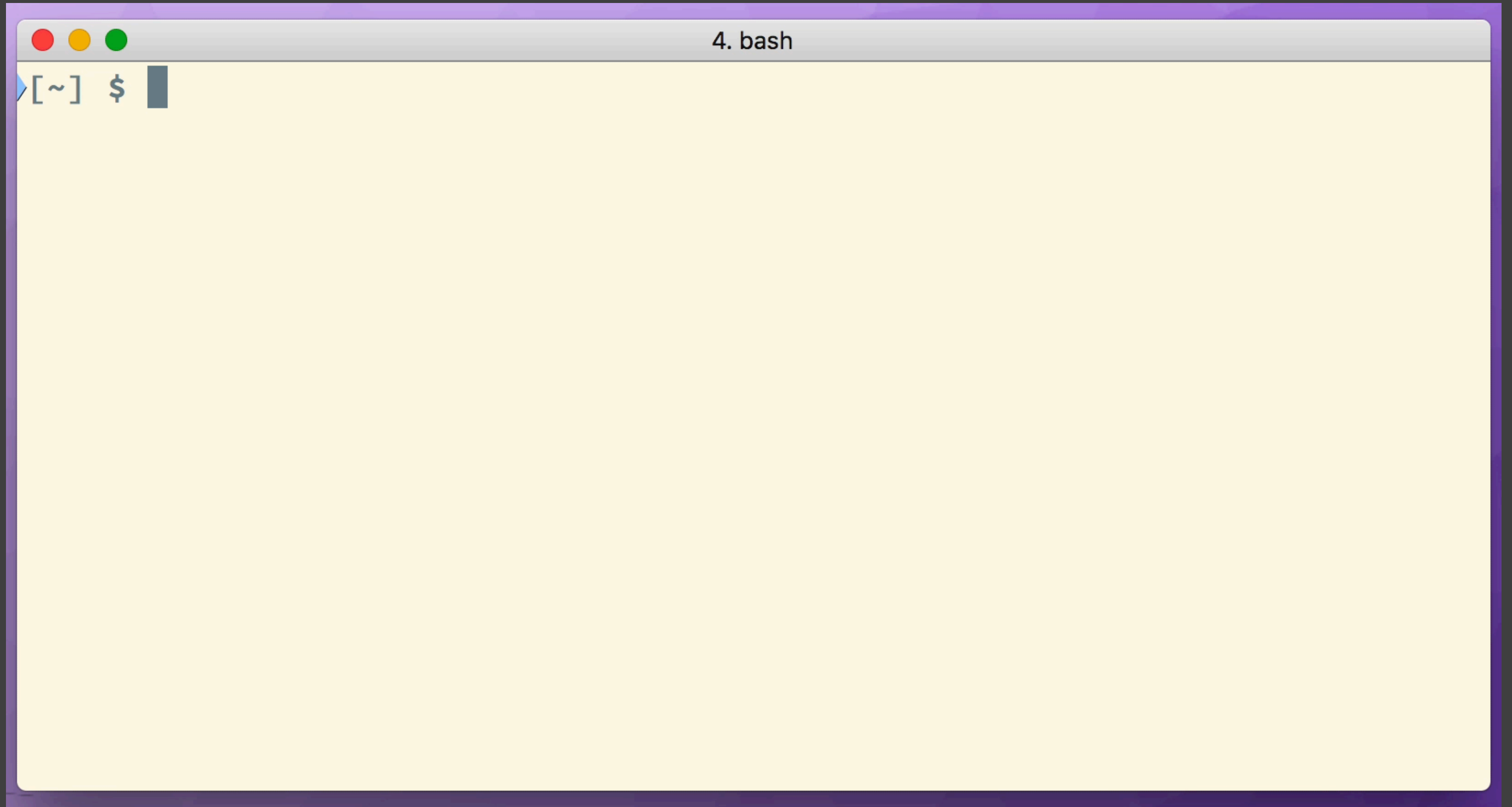
gron

- ▶ Turns JSON into line-oriented data
- ▶ Then, search with tools like `ag`
- ▶ Identify individual array elements

Example gron output



```
4. less
json.photos.photo = [];
json.photos.photo[0] = {};
json.photos.photo[0].farm = 2;
json.photos.photo[0].id = "29096613967";
json.photos.photo[0].isfamily = 0;
json.photos.photo[0].isfriend = 0;
json.photos.photo[0].ispublic = 1;
json.photos.photo[0].owner = "40098236@N04";
json.photos.photo[0].secret = "aba53ef24f";
json.photos.photo[0].server = "1772";
json.photos.photo[0].title = "Nutella filled pancake balls with bacon and maple syrup";
json.photos.photo[1] = {};
json.photos.photo[1].farm = 2;
json.photos.photo[1].id = "30165537958";
json.photos.photo[1].isfamily = 0;
:
```



jq

- ▶ Powerful, flexible processor
- ▶ Concise syntax
- ▶ Transform data you have, to data you *want*

Extract values from array of objects

```
$ jq -r '.photos.photo | .[] | .title' flickr.json
```

AJH-20180814-467-Edit.jpg

East Garston series: quintessential English countryside

IMG_20180723_215912_323

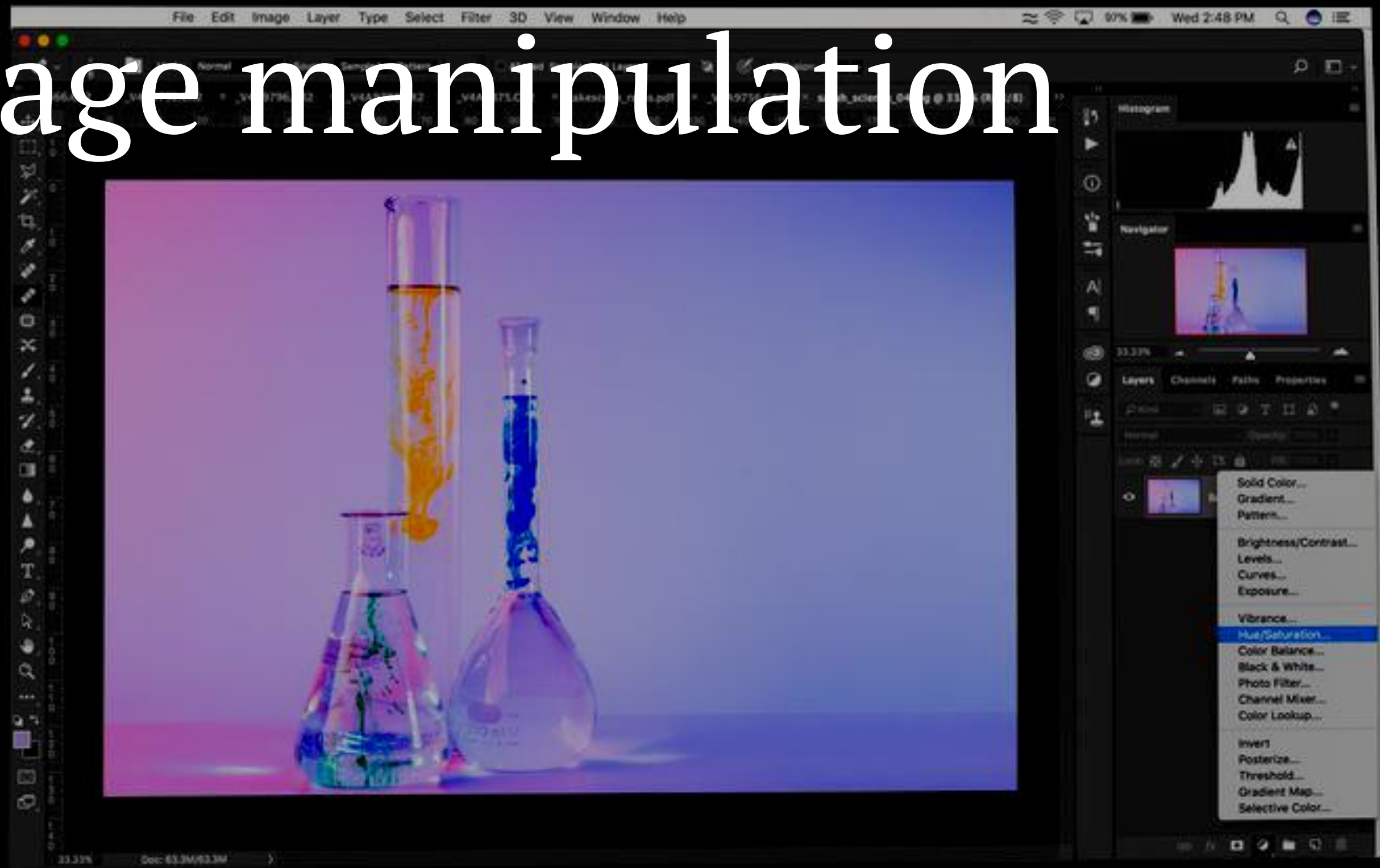
P1030365.jpg

Frankrijk 2018

Shirts on Sale (+6 03 6143 5225)Contact Us 03 6143 5225 <https://t.co/dBTffHWW99>

Tratamiento Alternativo Para El Colon Irritable

Image manipulation



Compositing images superpower

Tweaks to match latest designs #829

Closed

efung wants to merge 5 commits into android/807-pick-icon-template-screen from android/816-pick-template-tweak

Conversation3

Commits5

Checks0

Files changed22

+504-72

efung commented 7 days ago

What are you trying to accomplish?

Match latest design in Zeplin

2:31

Pick a template

Asdf

Asdf

Asdf

Asdf

Asdf

Asdf

Before

2:28

Pick a template

Asdf

Asdf

Asdf

Asdf

Asdf

Asdf

After

Reviewers

warefhaque

Assignees

No one—assign yourself

Labels

Android

Projects

None yet

Milestone

No milestone

Notifications

Unsubscribe

You're receiving notifications because you were mentioned.

3 participants

Lock conversation

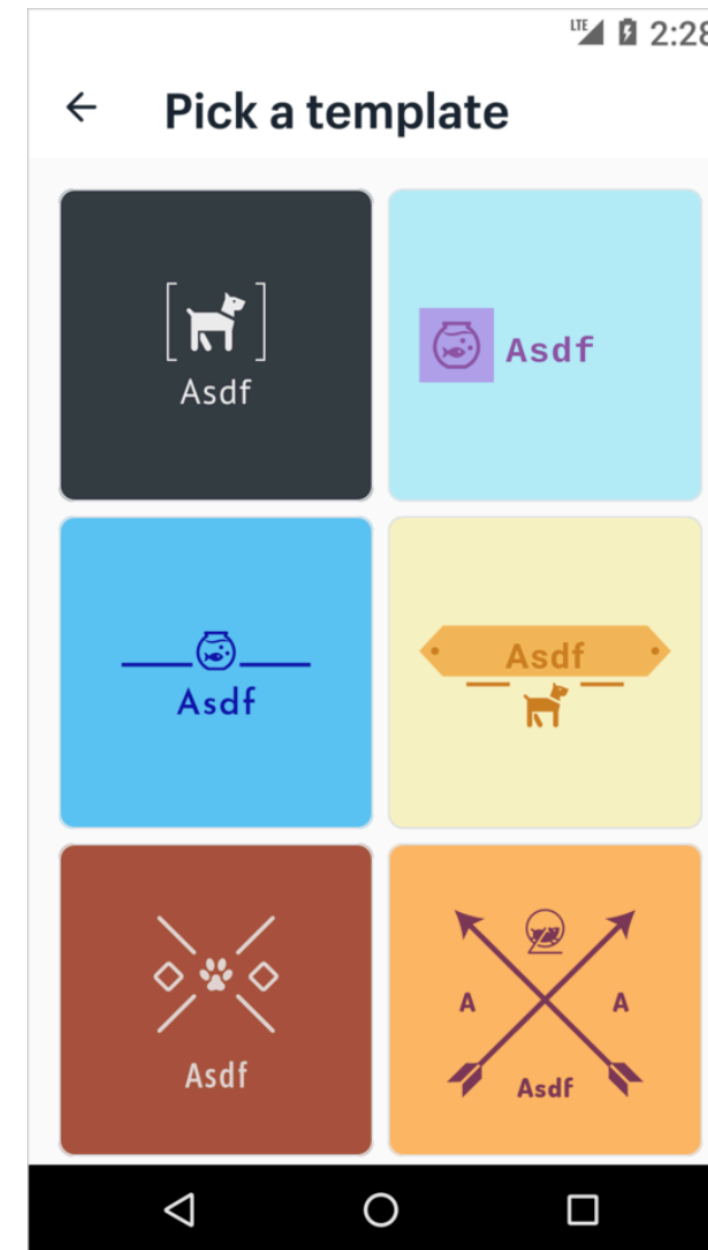
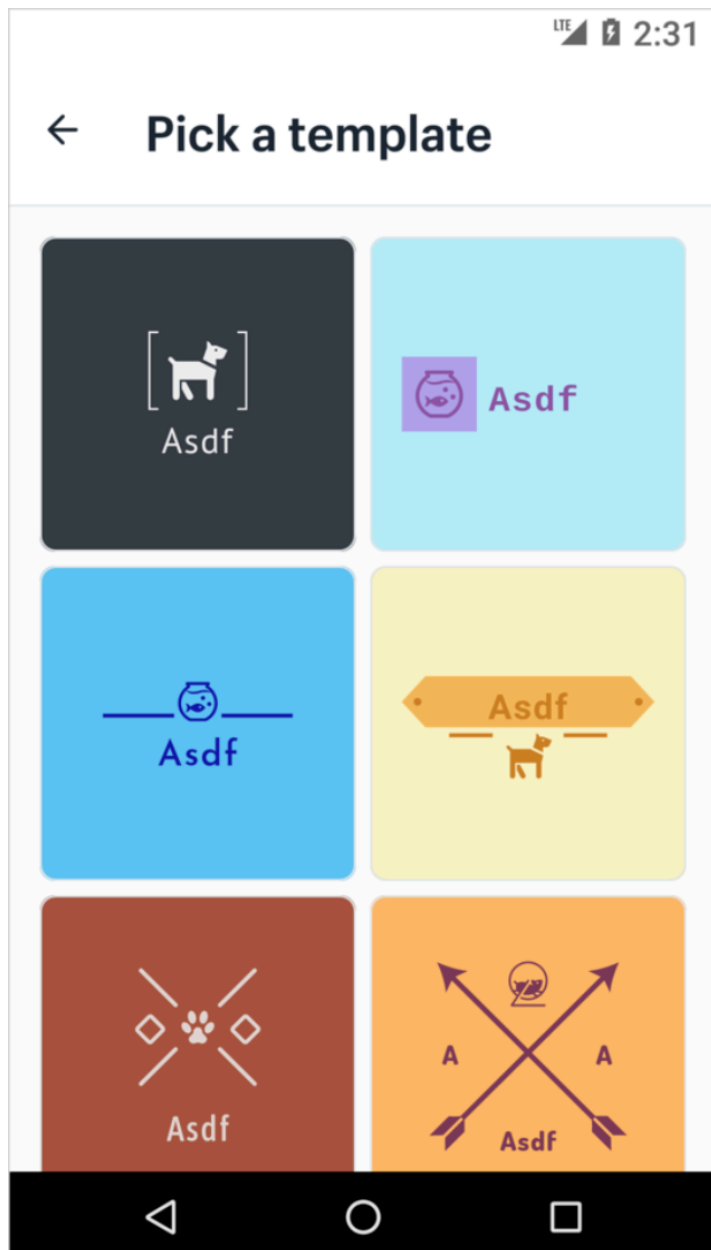
How The Command-Line Can Give You Superpowers • Eric Fung • AndroidTO 2018

ImageMagick

- ▶ Suite of command-line utilities
- ▶ Create, edit, compose, or convert bitmaps
- ▶ Read/write 200 image formats
- ▶ Handles really large images



Example: Composite images



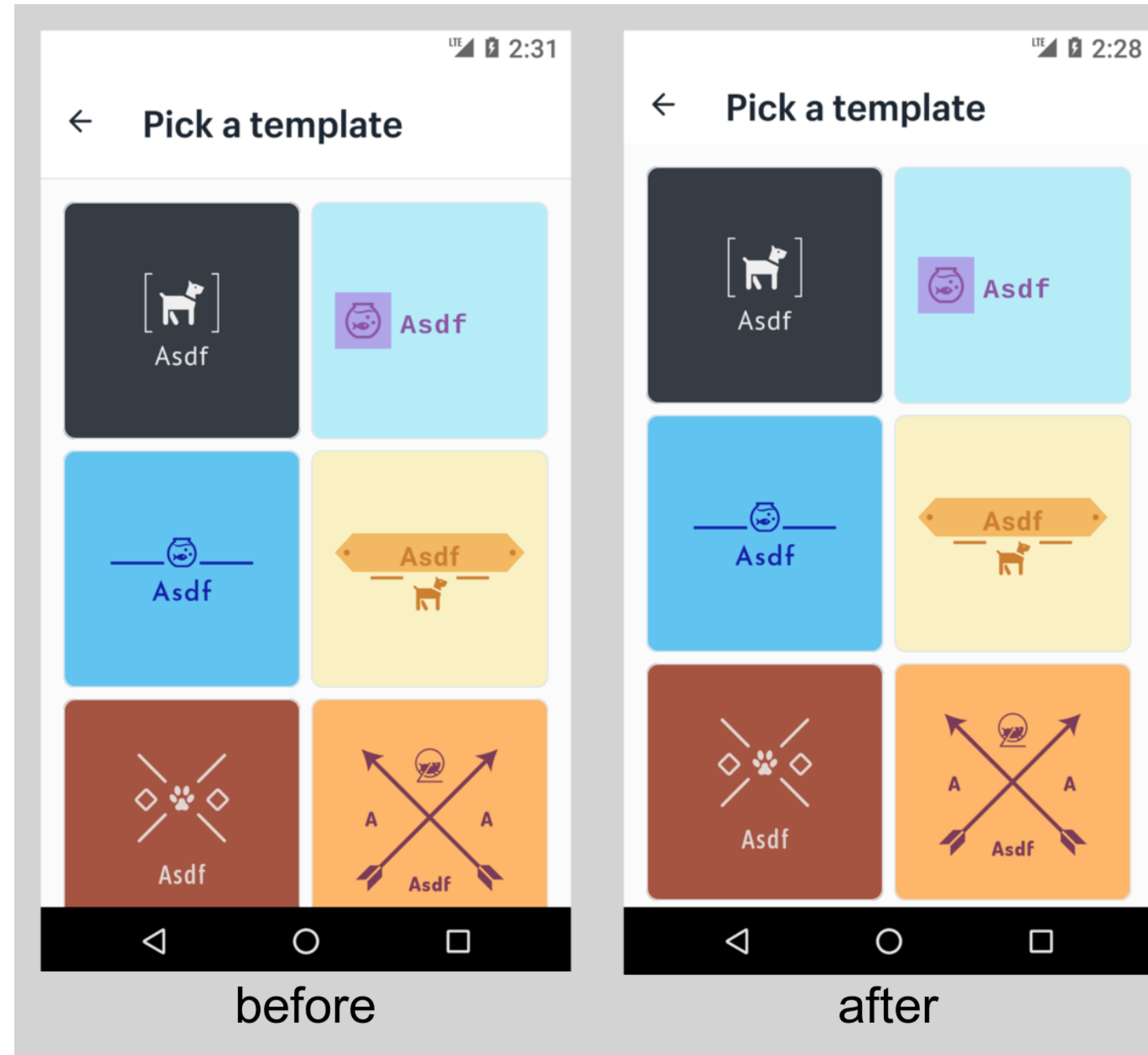
bash function using IM's montage

```
side_by_side() {  
    montage -background lightgray -pointsize 60 \  
        -label '%t' \  
        $* \  
        -tile x1 -geometry '+30+30' \  
        combined.png  
}
```

- ▶ `$*` is all the function's arguments
- ▶ `-label '%t'` is IM syntax defining label from each argument's base filename

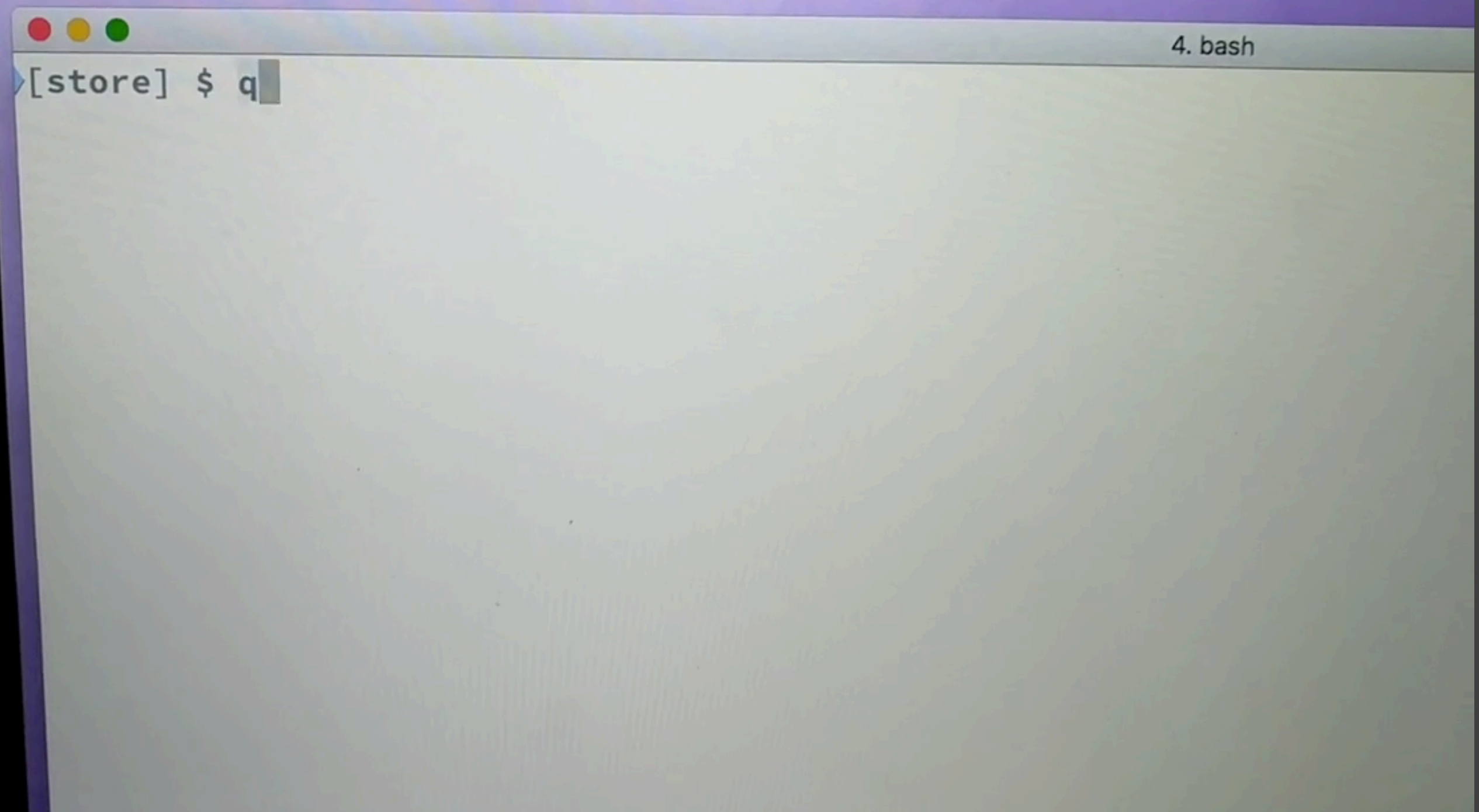
Usage

```
$ side_by_side before.png after.png
```





Deployment and testing



Demo: Install superpower

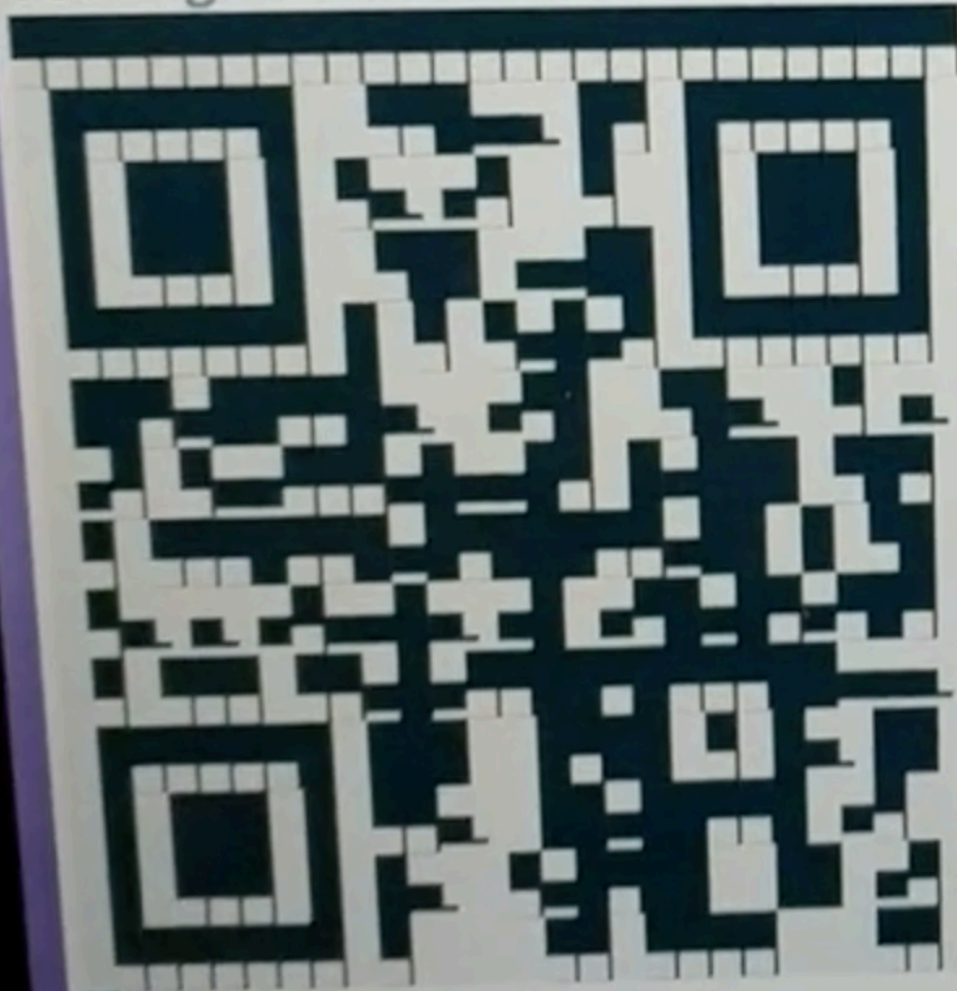
Installing APKs wirelessly

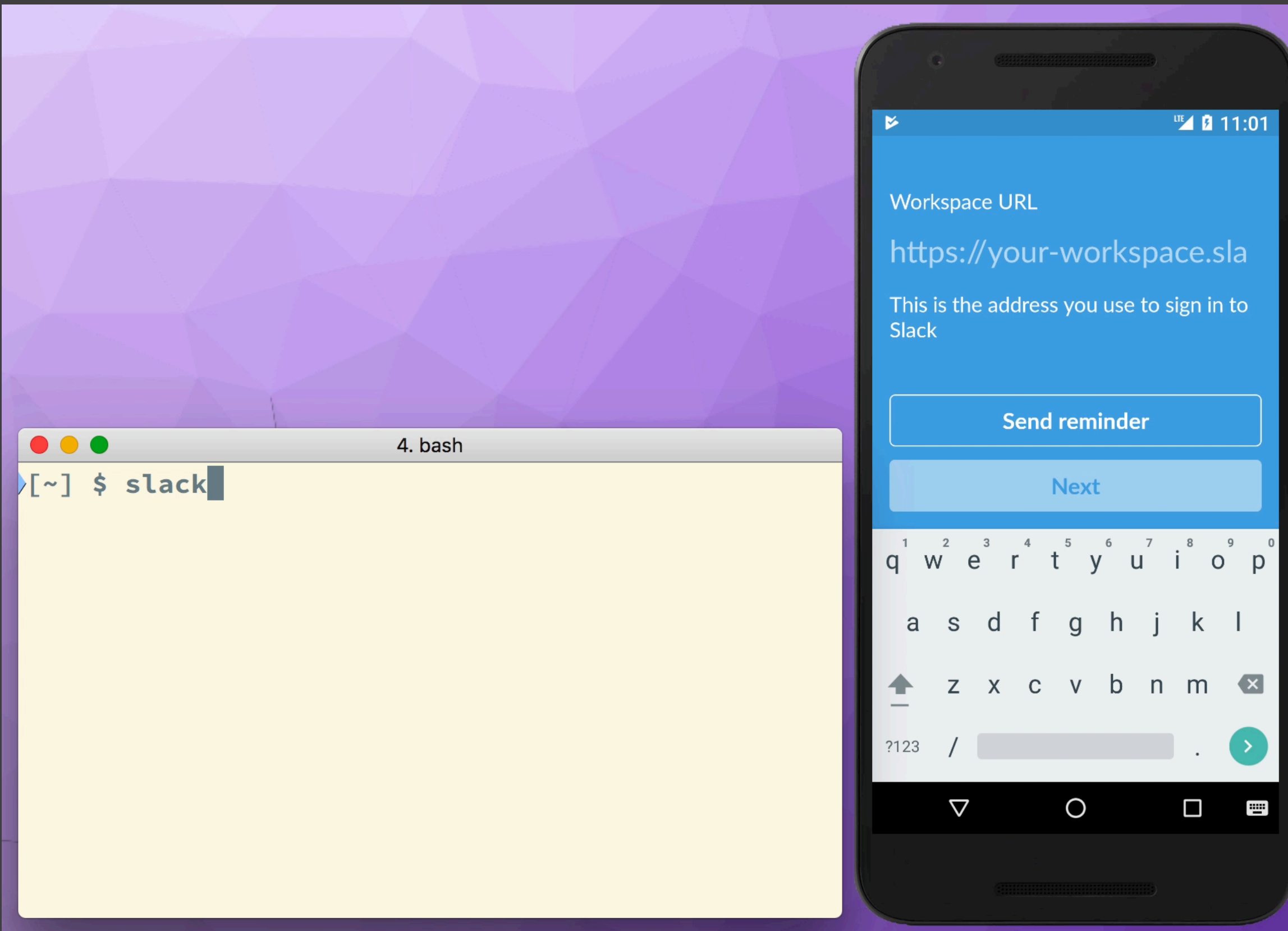
- ▶ Can be done over Wi-Fi (see Google's [adb user guide](#))
 - ▶ Process seems cumbersome
- ▶ Some devices block USB loading
- ▶ Don't always have a cable

qr-filetransfer

- ▶ Starts a local webserver hosting your file
- ▶ Displays QR code of URL to file

```
[store] $ qr-filetransfer 1850000.a  
Scan the following QR to start the  
Make sure that your smartphone is c  
Your generated address is http://19
```





Demo: Automation superpower

Automating device input

- ▶ `adb shell input text <text>`
 - ▶ Sends keycodes to device
- ▶ `adb shell input tap <x> <y>`
 - ▶ Sends touch events to device

Example: log in to Slack workspace

```
function slack_workspace_login() {  
    adb shell input text $1          # workspace  
    adb shell input tap 400 1000     # next  
    sleep 1  
    adb shell input text $2          # email  
    sleep 1  
    adb shell input tap 400 1000     # next  
    sleep 1  
    adb shell input text $3          # password  
}
```

► Naive, but good enough usually

Summary



What We've Learned

- ▶ Basic command-line setup
- ▶ How to make the command-line easier to use
- ▶ Many ~~tools~~ superpowers to help you get stuff done
 - ▶ Development
 - ▶ Design
 - ▶ Debugging

End

- ▶ **Contact** `eric.fung@shopify.com` • `@gnuvmuffin`
- ▶ **Work** `shopify.com/careers`
- ▶ **Blog** `code.gnuvmuffin.com`
- ▶ **Slides** `speakerdeck.com/efung`

Appendix: How to install (1/2)

Name	What it does	How to get it ¹
iTerm2	full-featured terminal	https://iterm2.com/
Homebrew	package manager	https://brew.sh/
bash completion	bash completions for macOS builtins	 bash-completion
adb completion	bash completions for adb	envsetup.sh + adb.bash
autojump (j)	navigate directories faster	 autojump
ripgrep (rg)	fast, intelligent search tool	 ripgrep
curl	transfer data with URLs	 curl
		everything curl book

¹  denotes Homebrew package name

Appendix: How to install (2/2)

Name	What it does	How to get it ¹
gron	make JSON line-oriented and searchable	 gron
jq	transform and query JSON	 jq
		jq manual
		jq cookbook
ImageMagick	utilities for image manipulation	 imagemagick
		usage examples
qr-filetransfer	transfer local files to device via QR codes	github.com/claودیodangelis/qr-filetransfer

¹  denotes Homebrew package name