

How The Command-Line Can Give You *Superpowers*

Eric Fung • @gnuvmuffin



Android Summit • 2018-08-16



Twitter

@gnufmuffin

#AndroidSummit2018

[https://speakerdeck.com/efung/
how-the-command-line-can-
give-you-superpowers-1](https://speakerdeck.com/efung/how-the-command-line-can-give-you-superpowers-1)

```
**** COMMODORE 64 BASIC V2 ****
64K RAM SYSTEM 38911 BASIC BYTES FREE
READY.
LOAD "$",8
SEARCHING FOR $
LOADING
READY.
LIST
0 "3D REVERSI.D64" 20
3 "LOADER 3D REV." PRG
17 "3D REVERSI" PRG
644 BLOCKS FREE.
READY.
```

```
[~] gcc --version
gcc (GCC) 4.0.2
Copyright (C) 2005 Free Software Foundation, Inc.
This is free software; see the source for copying conditions. There is NO
warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

[~] cat > hello.c << EOF
> #include <stdio.h>
>
> int main()
> {
>     printf("Hello world!\n");
>     return 0;
> }
> EOF
[~] gcc -Wall -o hello hello.c
[~] ./hello
Hello world!
[~]
```

Superpowers, eh?

- 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**

Why should I care?

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

Structure

- The basics
- Typing less
- Working with Finder
- Navigation
- Searching
- I/O
- Web
- JSON
- Image manipulation
- Deployment and testing
- Code generation

The basics

- Terminal
- Shell
- Package manager

a

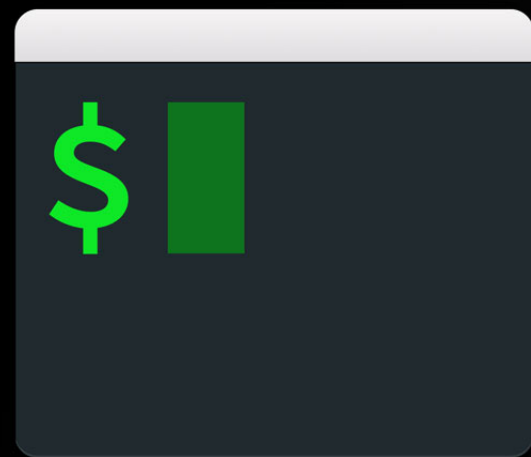
b

c

d

e

f



iTerm2

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

Terminal: iTerm2

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



Shell: bash

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

Package manager: Homebrew

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



Homebrew

The missing package manager for macOS

Typing less



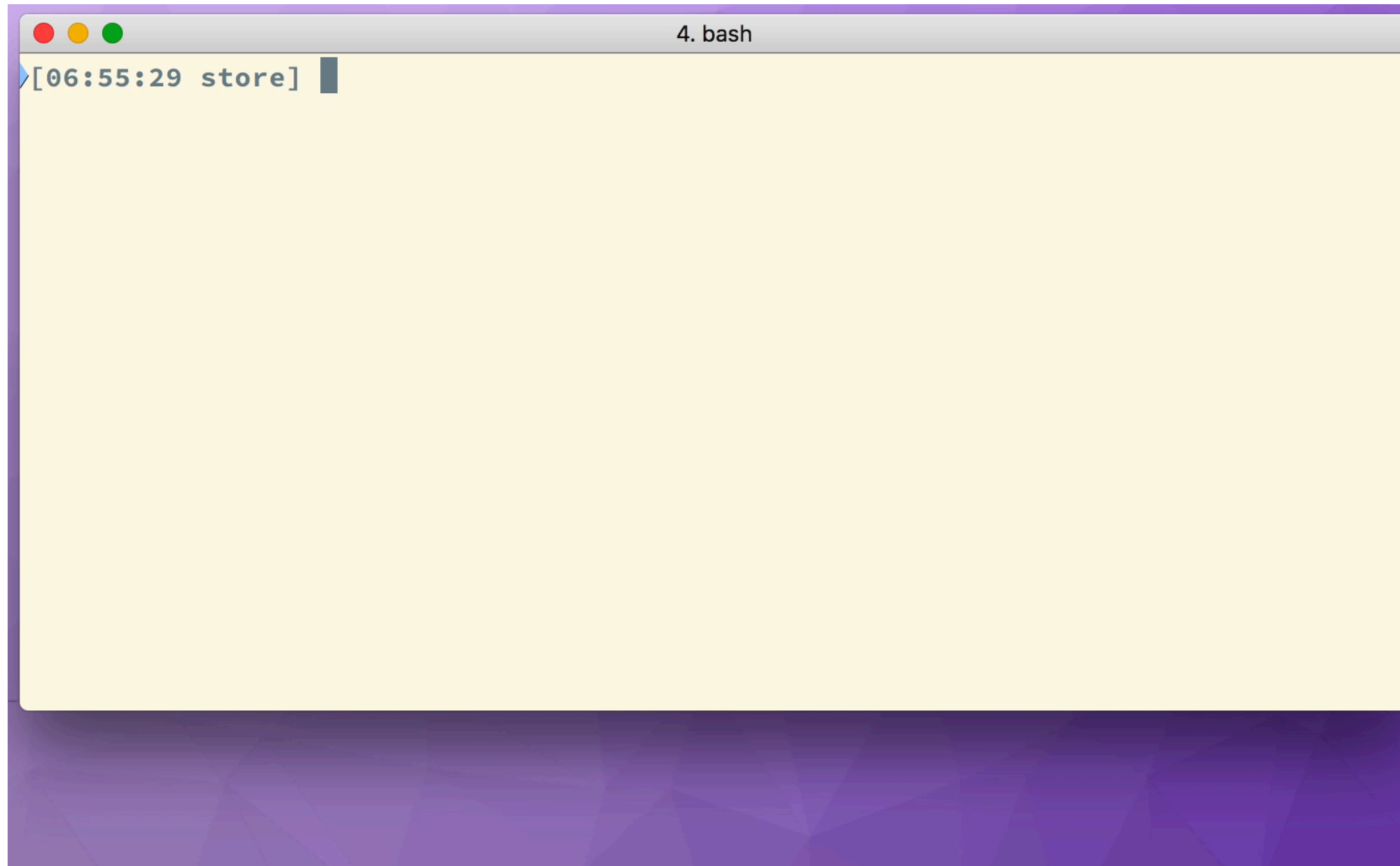
Completion

- Built-in to bash
- Context sensitive
- Completes command names and option names
- Completes option *values* too!

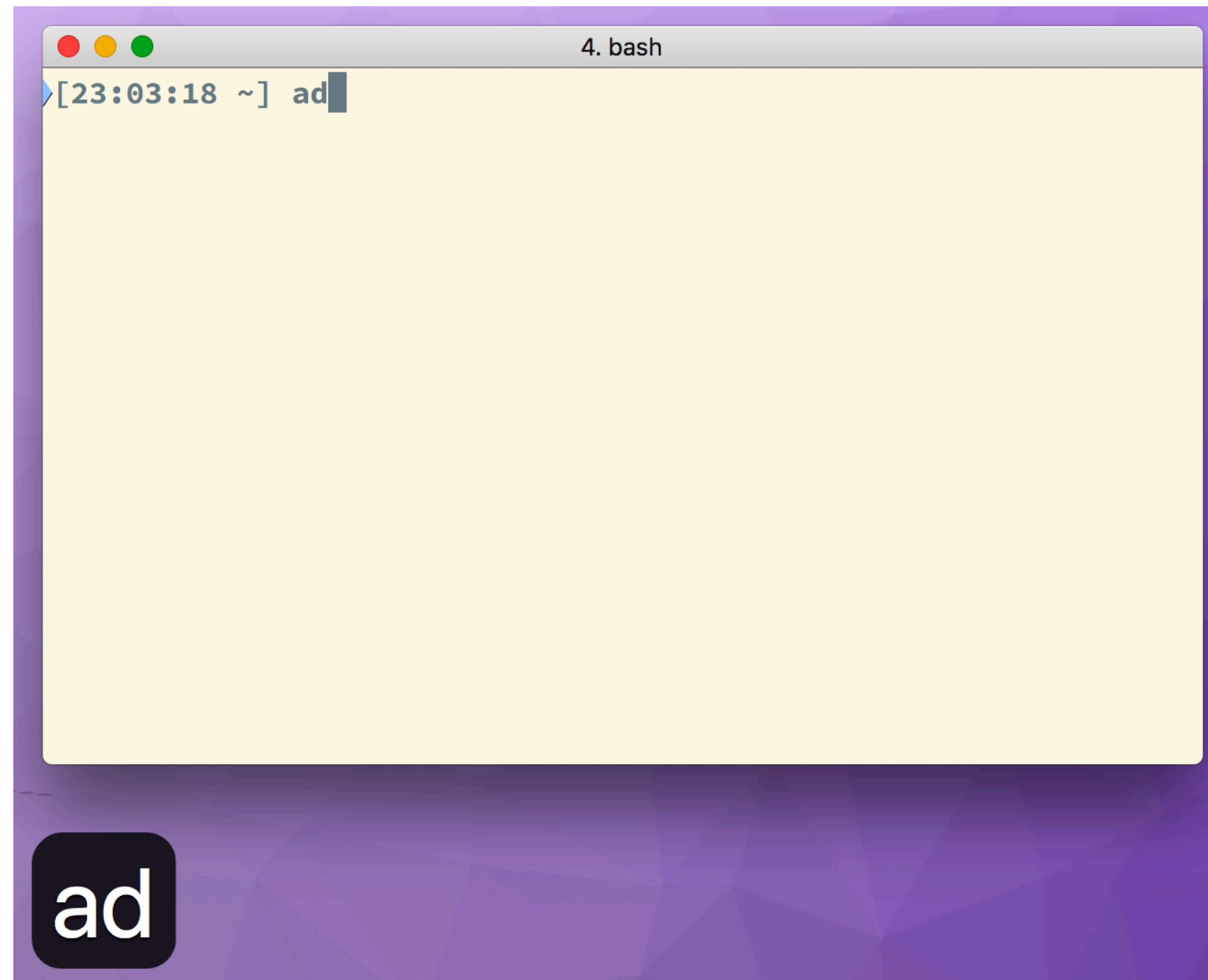
adb completion

- `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 install completion



Demo: adb uninstall completion

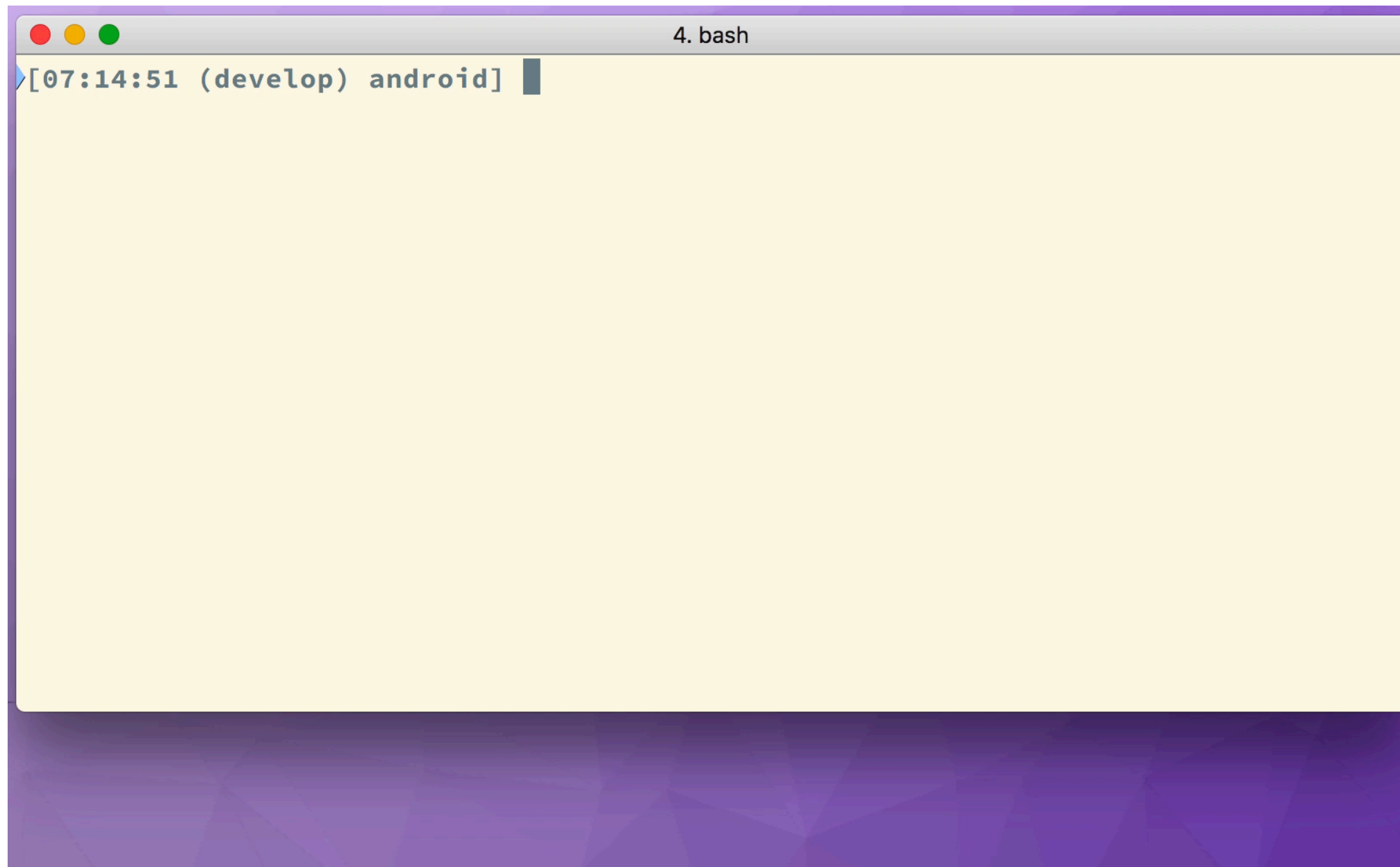




Gradle completion

- Knows properties, e.g. -
`Dorg.gradle.parallel`
- Fast, due to task cache
- Note: Implicit tasks not completed by default

Gradle completion



Working with Finder

From shell to Finder

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

`open -a <app> <file>`

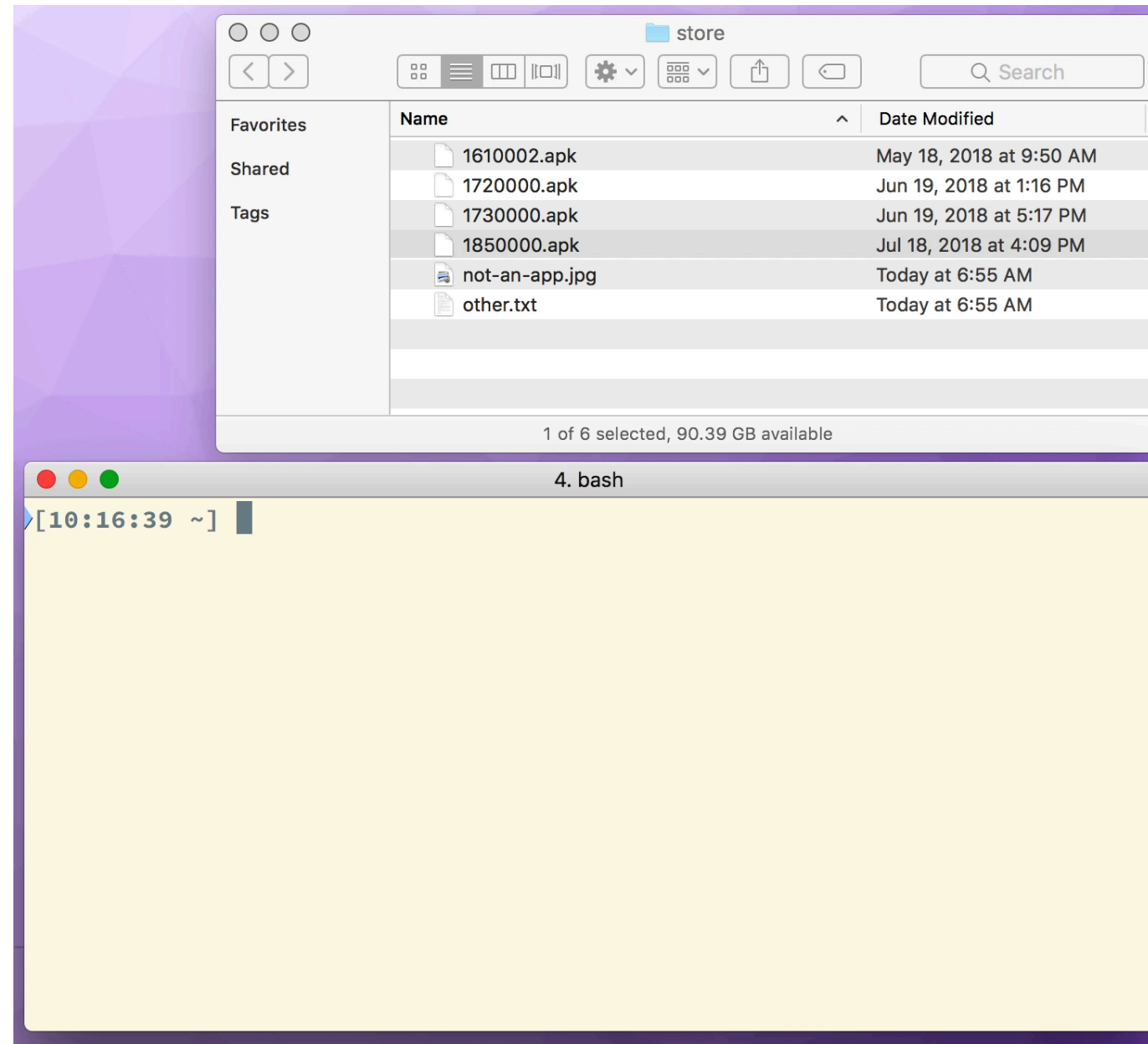
- Open `file` with a specific application
- Useful if file does not have default handler
- Tip: define a function or alias

```
asopen() {  
    open -a "/Applications/Android Studio.app" "$1"  
}
```

From Finder to shell

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

Drag files to shell → paste filename



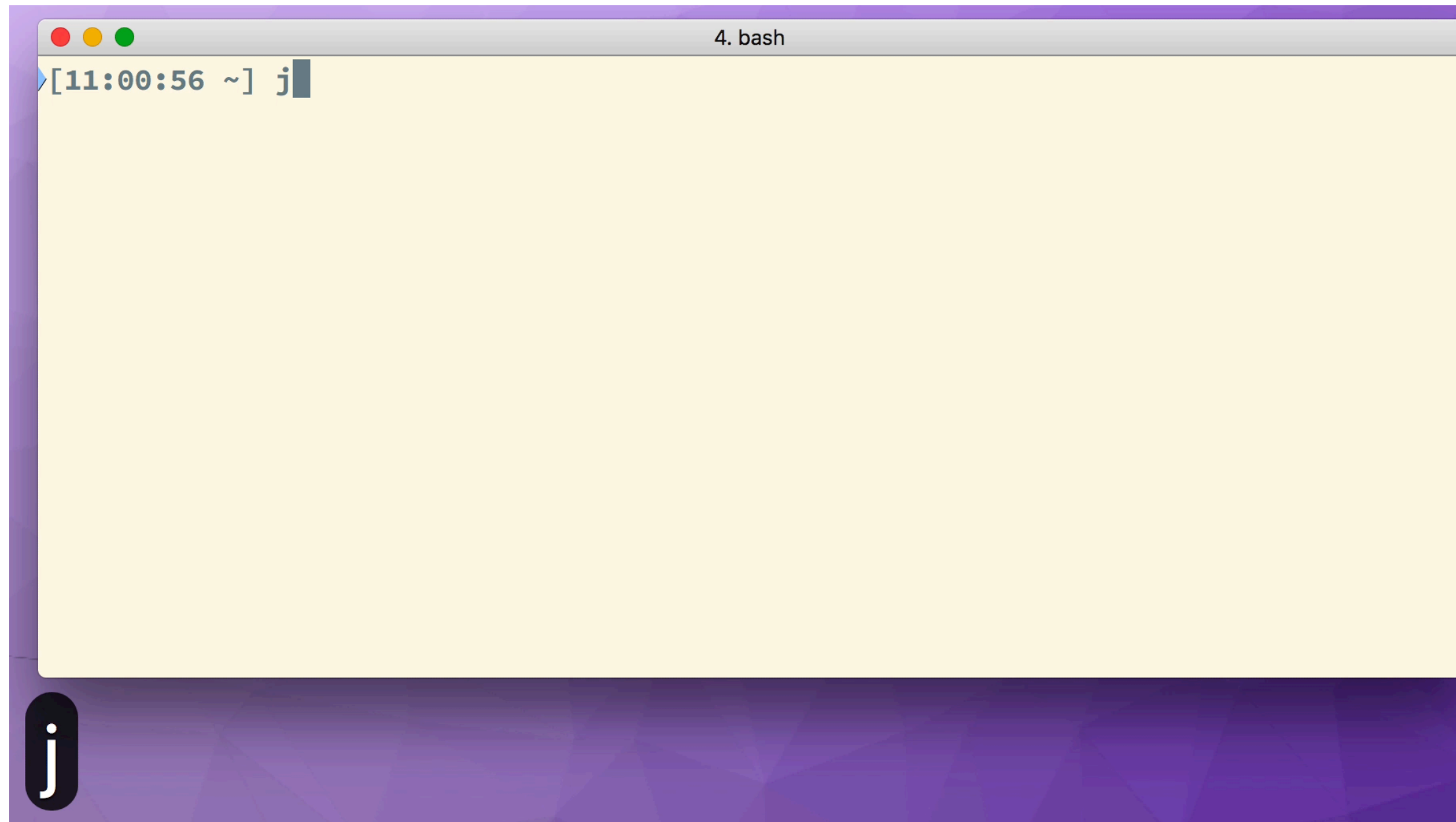
Navigation



cd 101

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

cd with superpowers



autojump

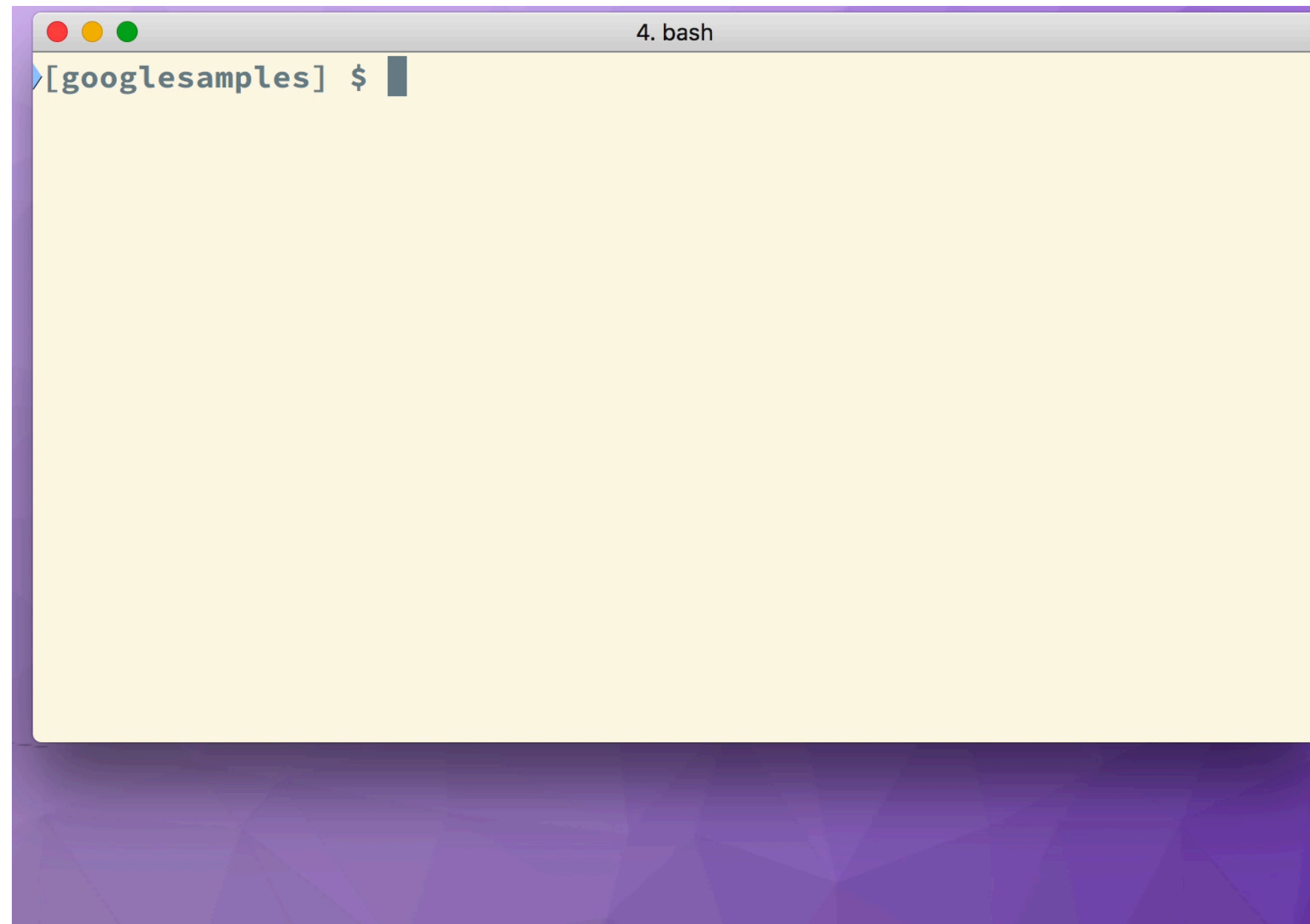
- Provide fast access to directories further away
- Selects directories based on usage
- Caveat: must be visited first

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

grep with superpowers

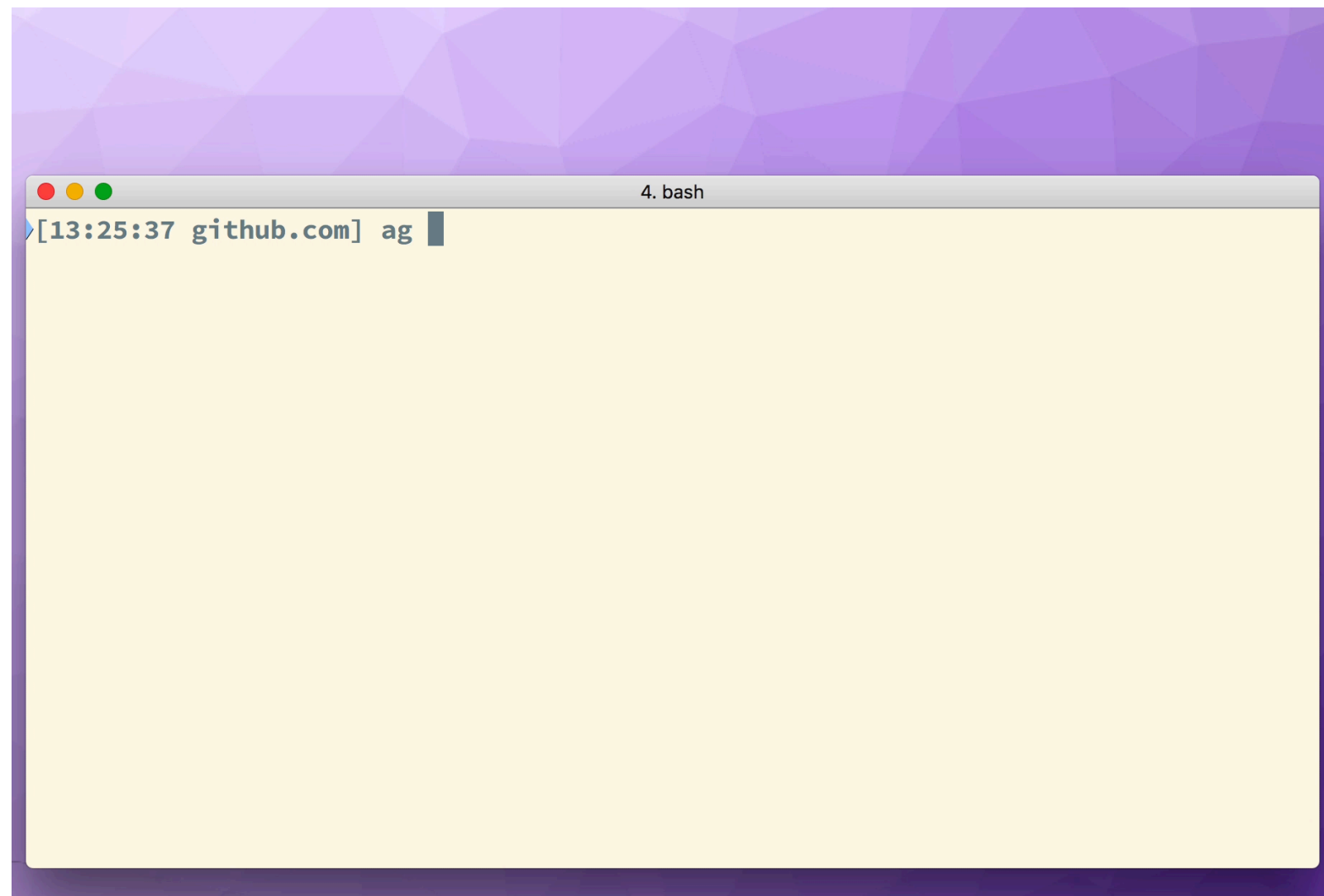


The Silver Searcher

- *Much* faster
- Supports grep options
- Knows about file types
- Respects revision control ignores
- Sensible defaults

Search Android documentation offline

`ag <term> $ANDROID_SDK/docs`



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



Web superpowers

A terminal window with a purple title bar and a yellow background. The title bar contains three colored window control buttons (red, yellow, green) on the left and the text "4. bash" on the right. The terminal shows a command prompt "[~] \$" followed by the command "curl --include https://img.icons8.com/color/256/puppy?token=\${KEY}" on the first line and "| ag cache" on the second line. A black cursor is positioned at the end of the second line.

```
4. bash
[~] $ curl --include https://img.icons8.com/color/256/puppy?token=${KEY}
| ag cache
```

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
```

Displaying fetched images

```
$ curl --header "Authorization: ${KEY}" \
    https://img.icons8.com/color/256/puppy | imgcat
```

— Pipes response through iTerm2's `imgcat` script

Debugging Firebase Cloud Messaging

- Typically, app server initiates push
- Console can only send notification messages
- Can simulate server from command-line
- More details in the [Firebase docs](#)

Sending a push notification

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

— Legacy server key avoids OAuth complexity

JSON



Visualize superpower

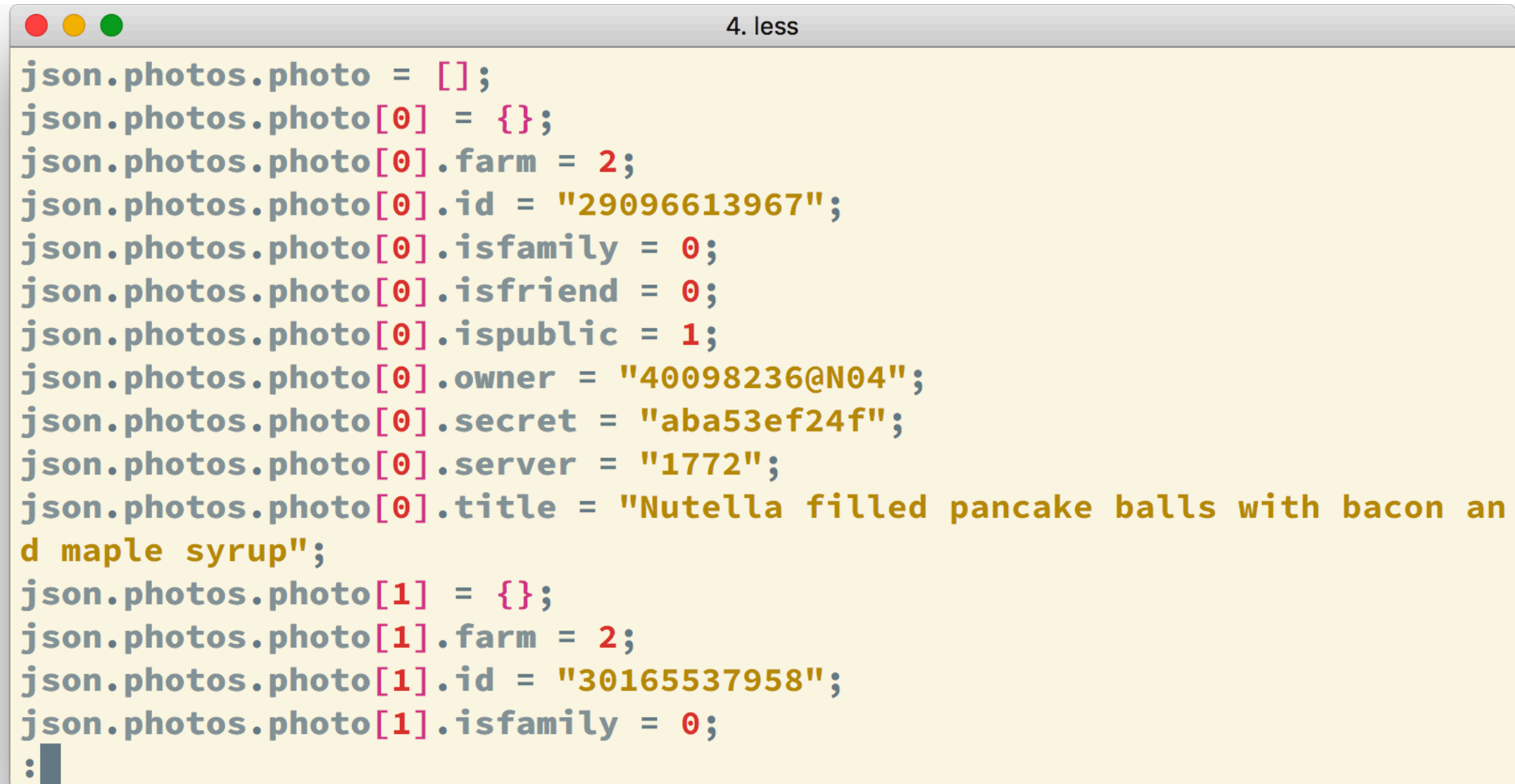
A terminal window with a purple title bar and three colored window control buttons (red, yellow, green) on the left. The title bar text is "4. bash". The terminal content shows a prompt "[~] \$" followed by a curl command: "curl 'https://api.flickr.com/services/rest/?method=flickr.photos.getRecent&api_key=\${FLICKR_KEY}&per_page=8&format=json&nojsoncallback=1'". The command is entered on a single line, and a cursor is visible at the end of the string.

```
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

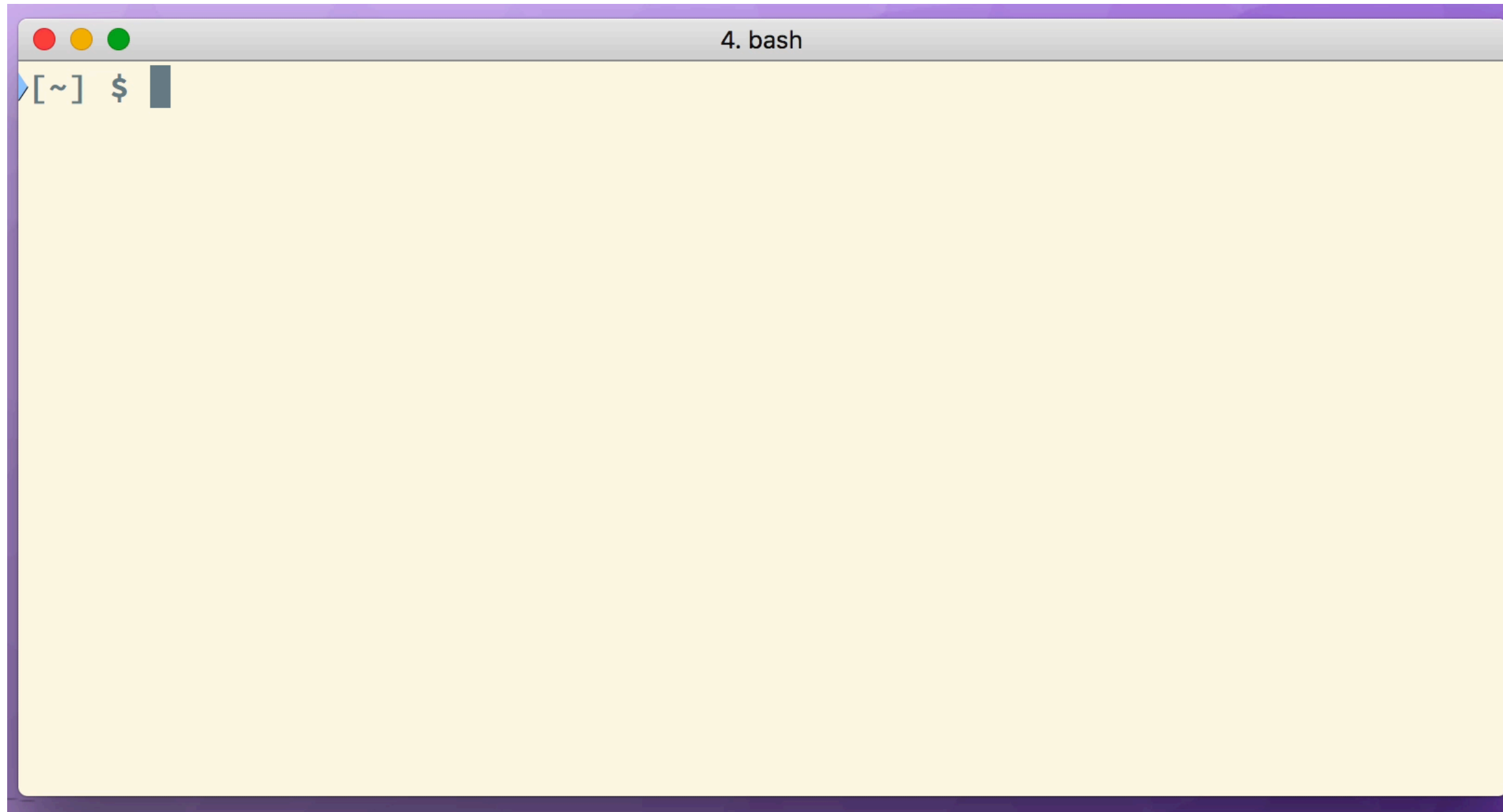
- Turns JSON into line-oriented data
- Then, search with tools like ag
- Identify individual array elements

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;
:
```

Slice and dice superpower



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>

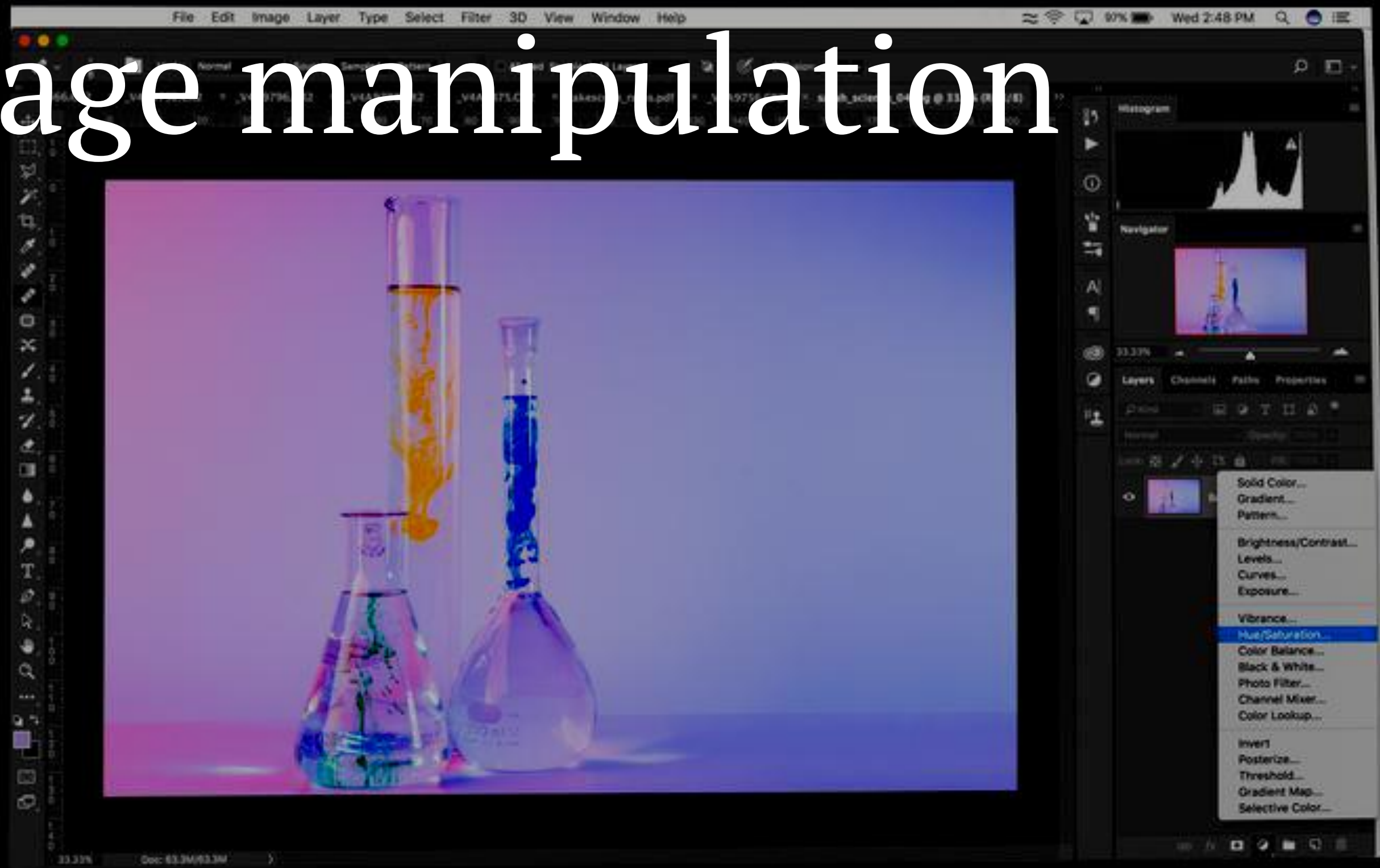
Tratamiento Alternativo Para El Colon Irritable

Filter elements by predicate

```
$ jq -r '[.photos.photo | .[] | select(.id | tonumber <= 43036557531)]' \
flickr.json
```

```
[
  {
    "id": "29099232337",
    "owner": "138111370@N03",
    "server": "1772",
    "farm": 2,
    "title": "AJH-20180814-467-Edit.jpg"
    ...
  },
  ...
]
```

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

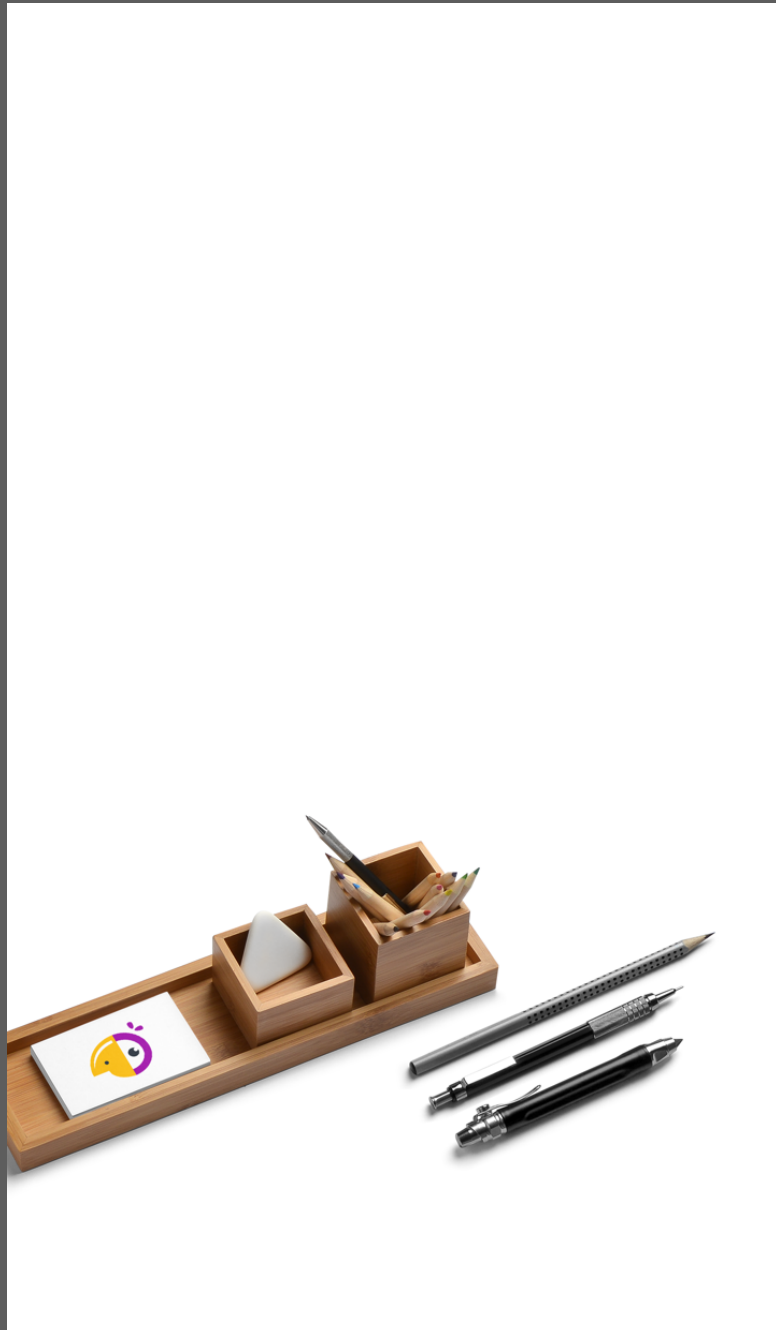
You're receiving notifications because you were mentioned.

3 participants

Lock conversation

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

Cropping many images superpower



Spot the difference superpower

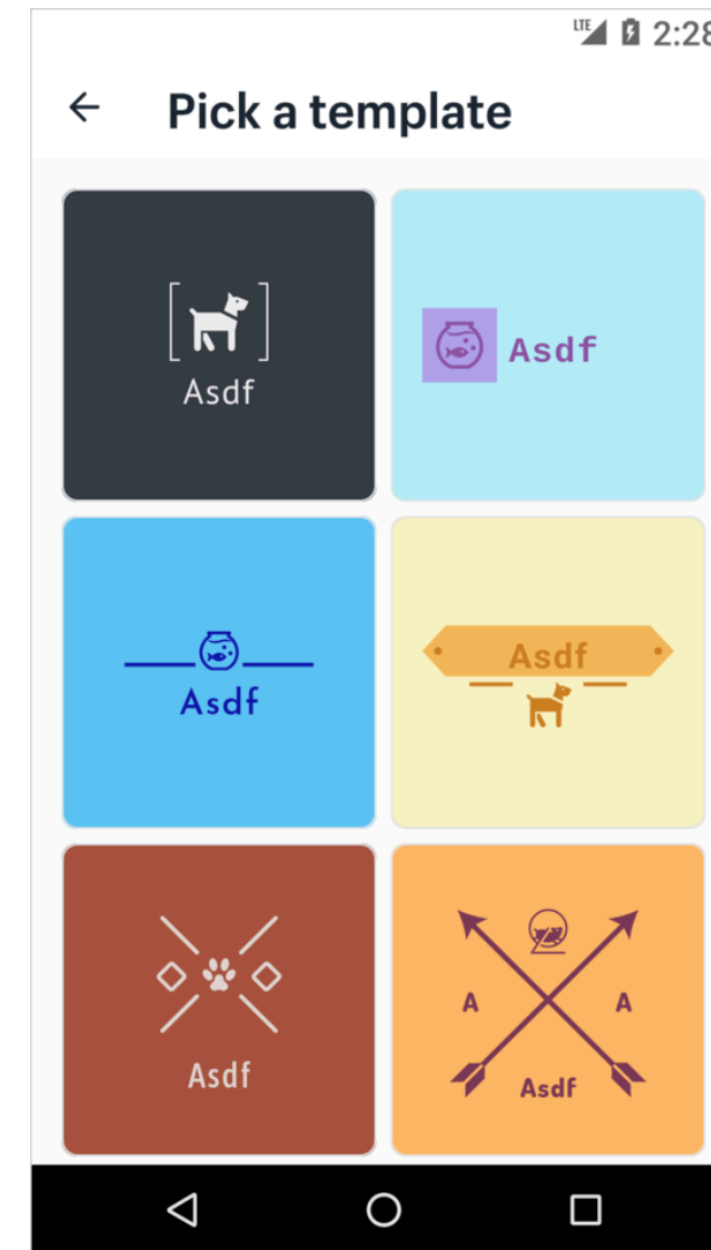
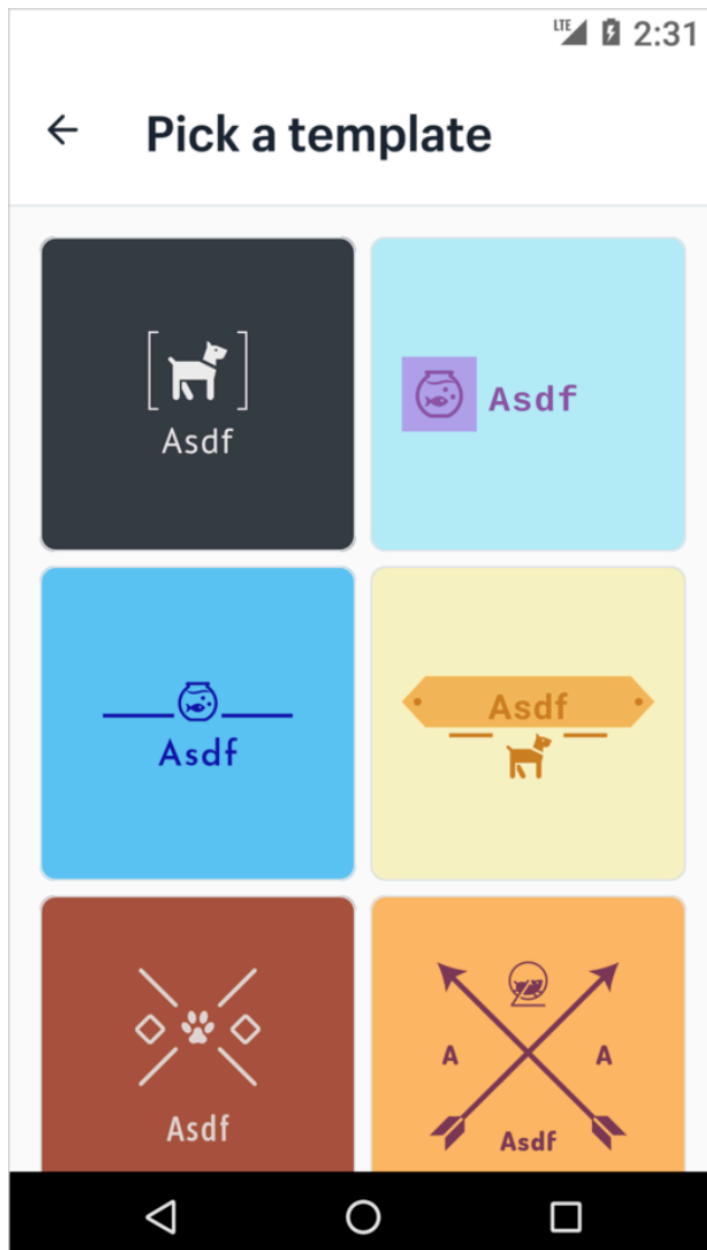


ImageMagick

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



Example: Composite images



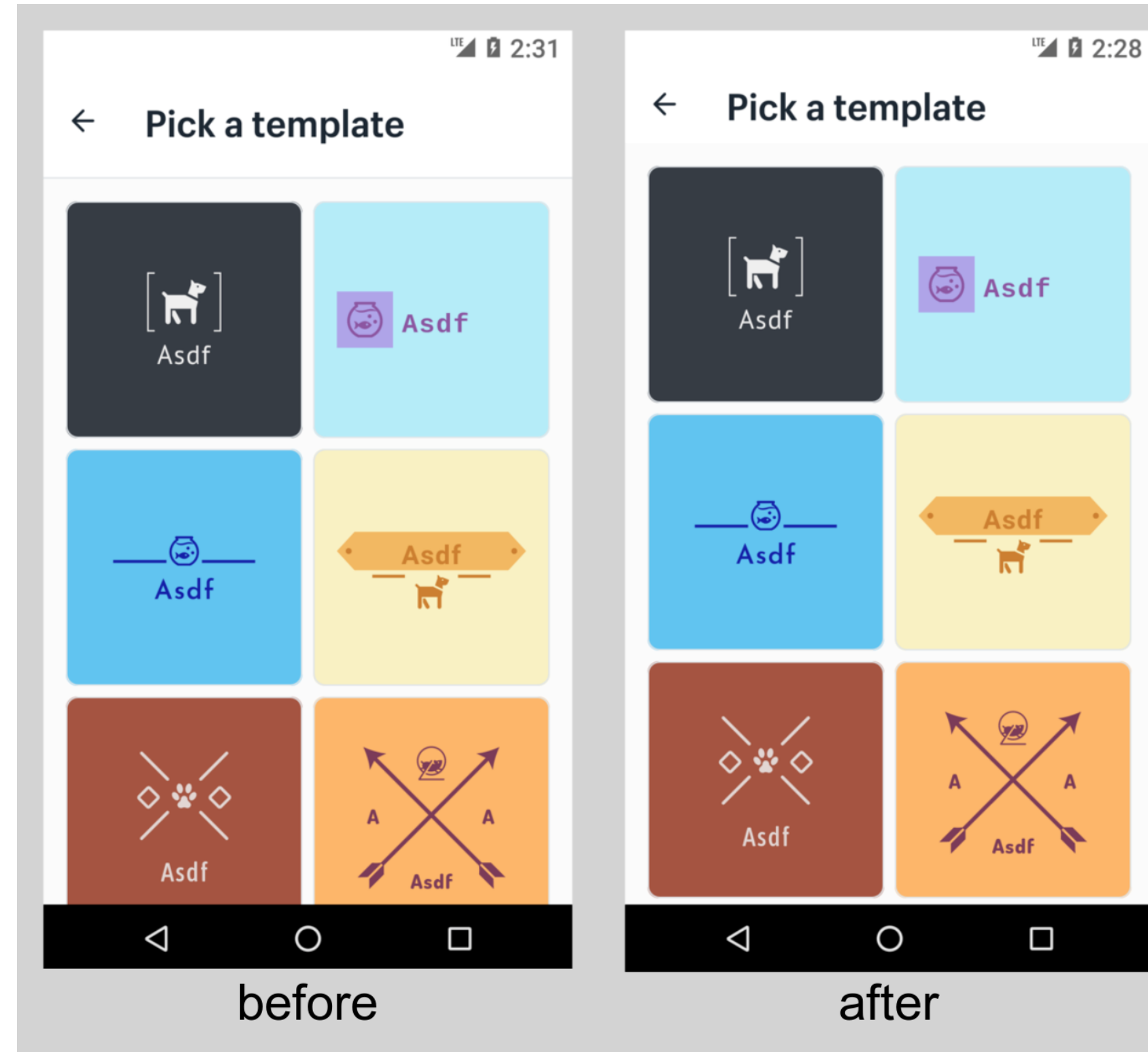
bash function using IM's montage

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

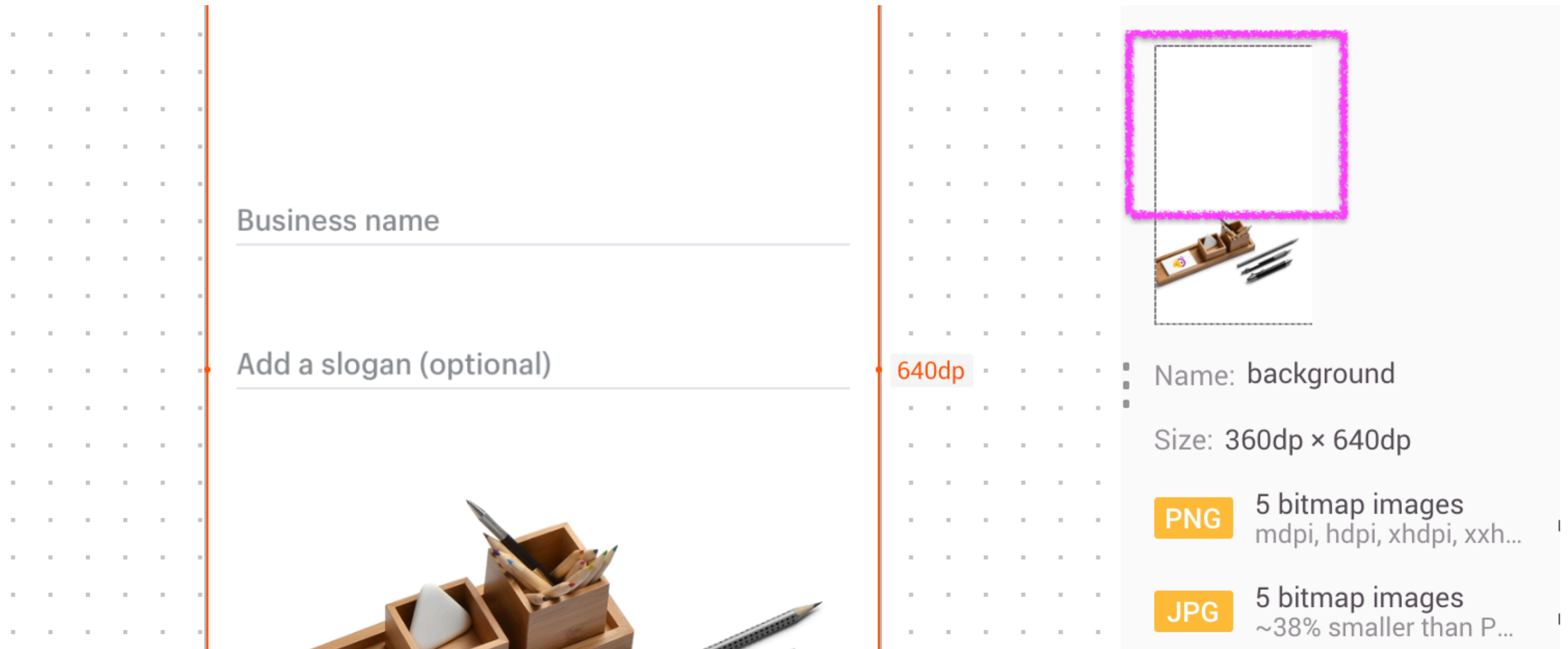
- `$*` is all the function's arguments
- `-label '%t'` is IM syntax defining label from each argument's base filename

Usage

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



Example: Crop multiple density bitmaps



bash loop using IM's convert

```
$ for r in m h xh xxh xxxh; do \
    convert -gravity SouthWest \
        -crop 100%x40%+0+0 \
        res/drawable-${r}dpi/background.jpg \
        res/drawable-${r}dpi/background_out.jpg; \
done
```

- `-gravity` defines origin
- Keep 40% of image in vertical direction

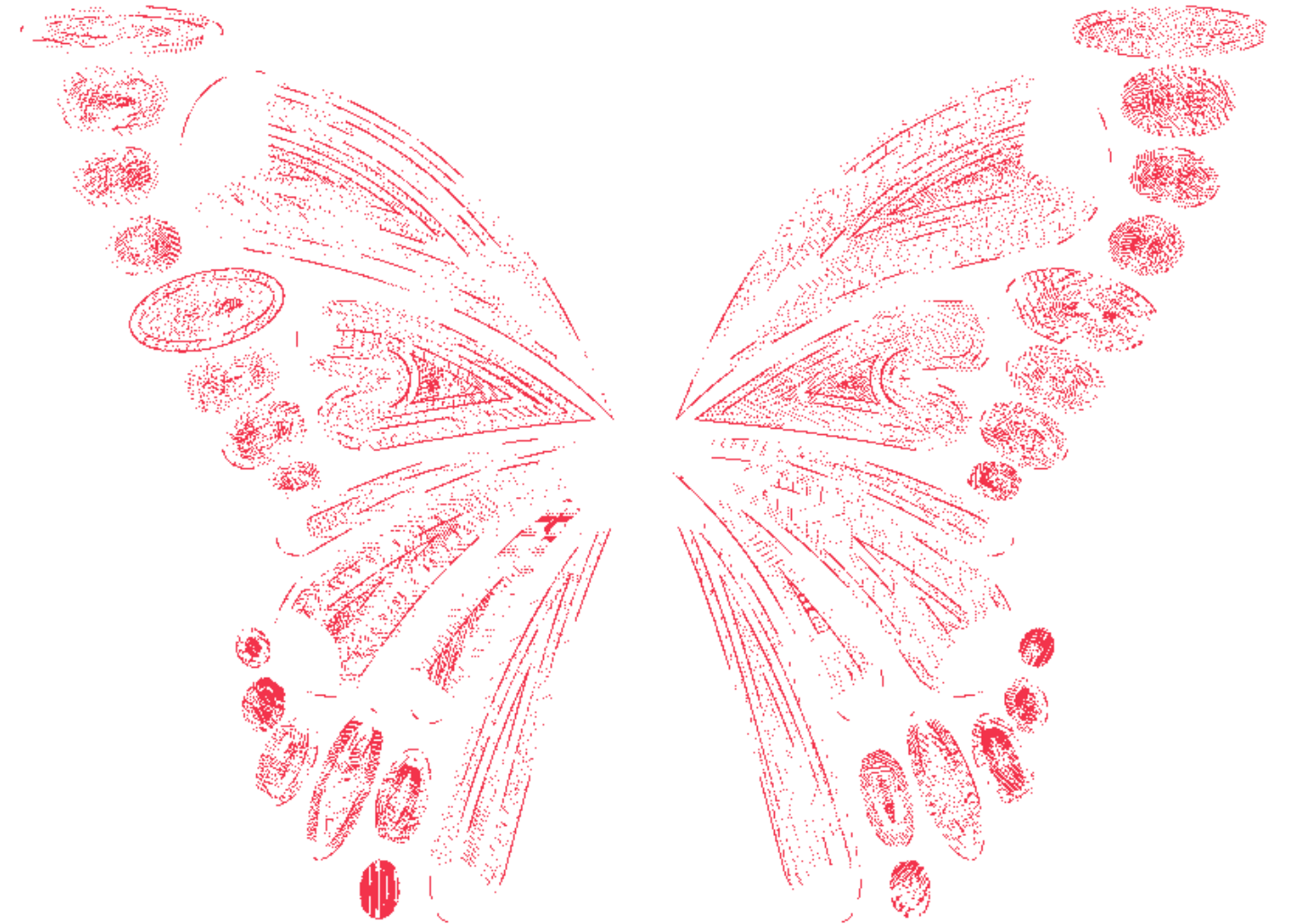
Example: compare images

- Shrink PNGs with pngcrush
- Optimize SVGs with svgo



IM's compare

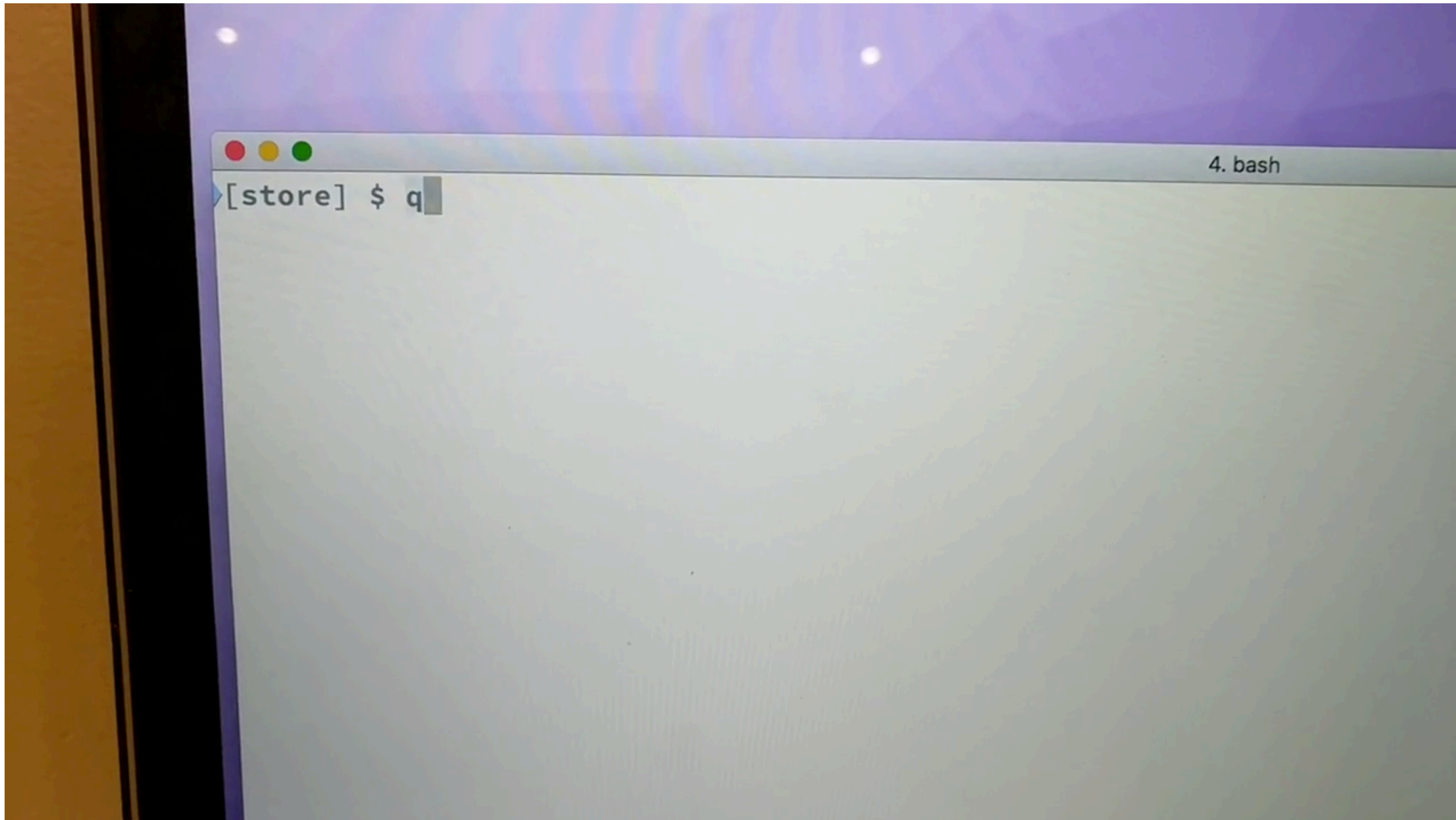
```
$ compare butterfly.png optimized.png \  
-compose Src difference.png
```





Deployment and testing

Install superpower



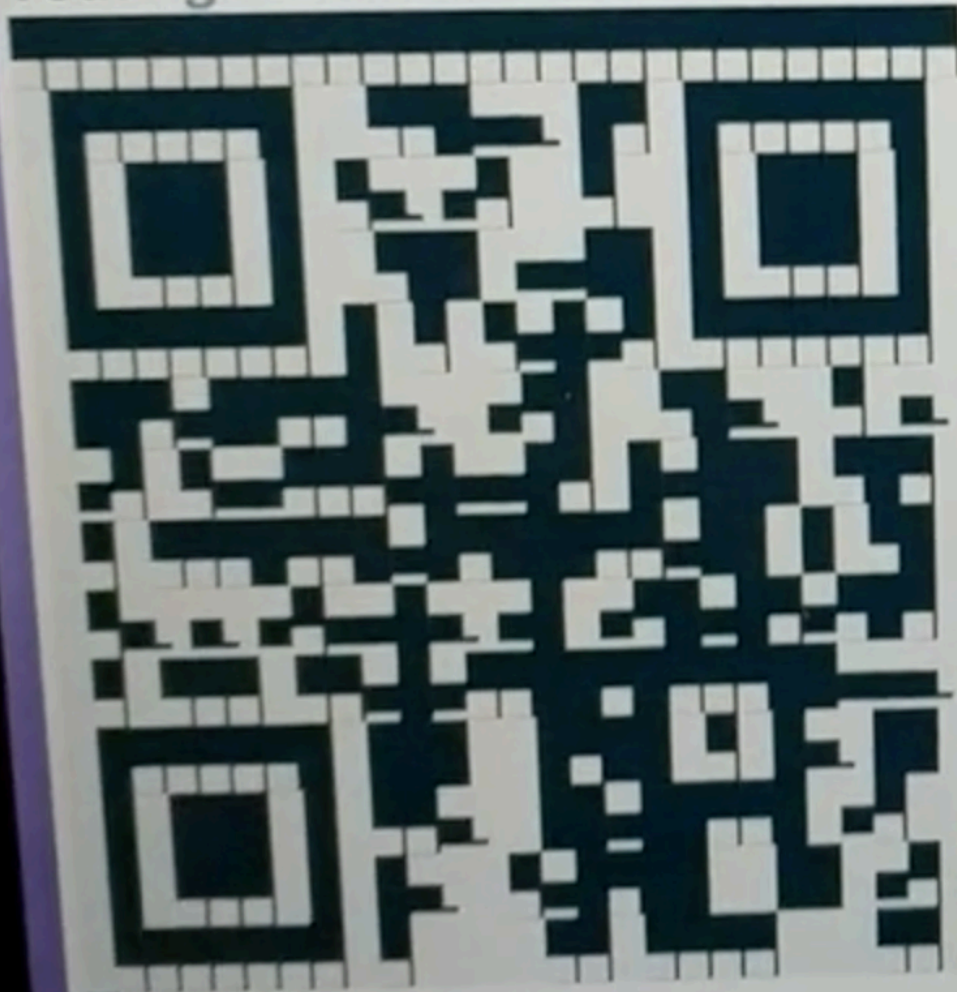
Installing APKs wirelessly

- APKs can be installed over WiFi (see Google's **adb over WiFi**)
- Process seems cumbersome
- Some devices block USB loading (unless you sign up)!

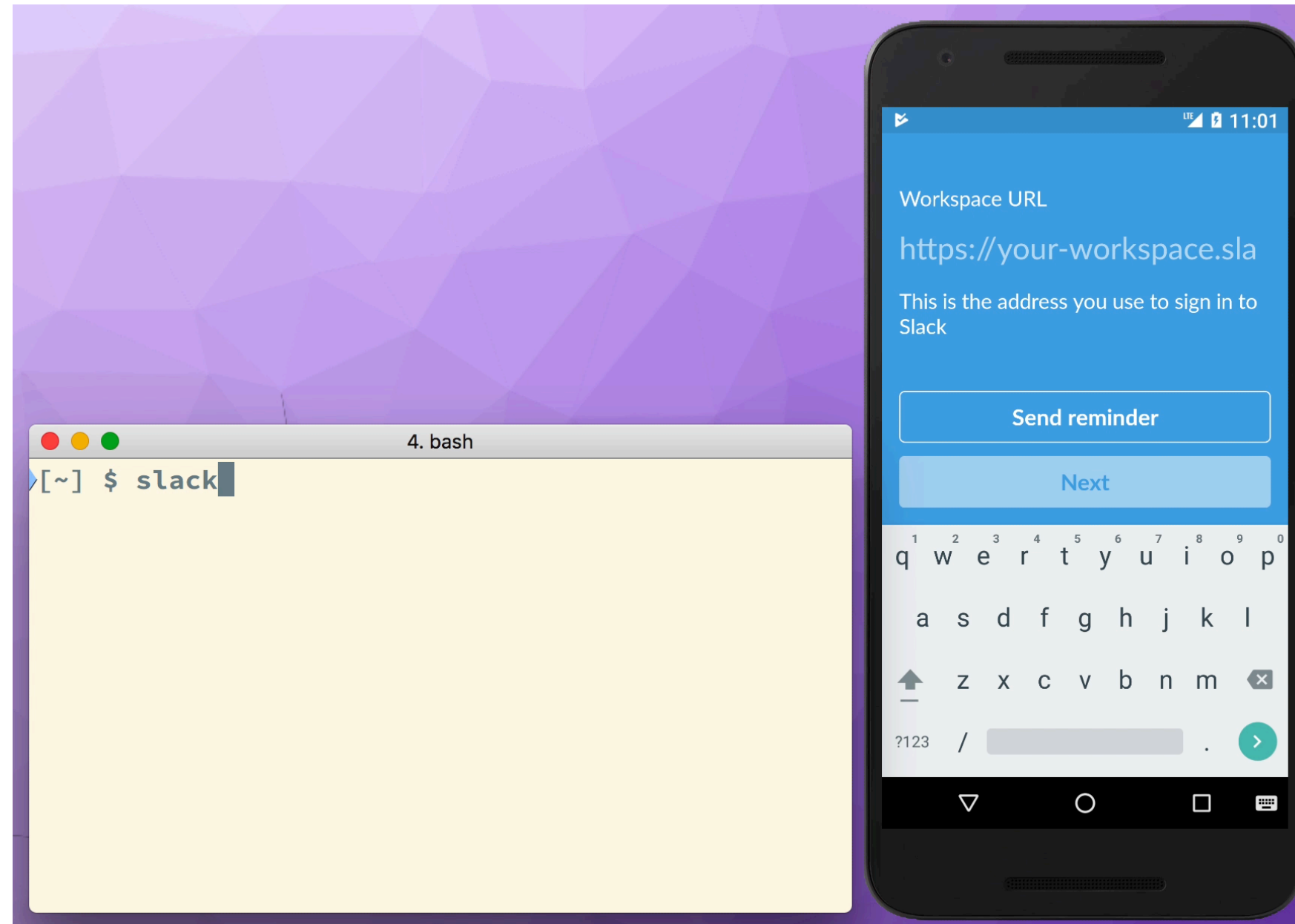
qr-filetransfer

- Starts a local webserver, with randomized URL hosting your file
- Generates QR code in terminal that points to that URL

```
[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
```



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

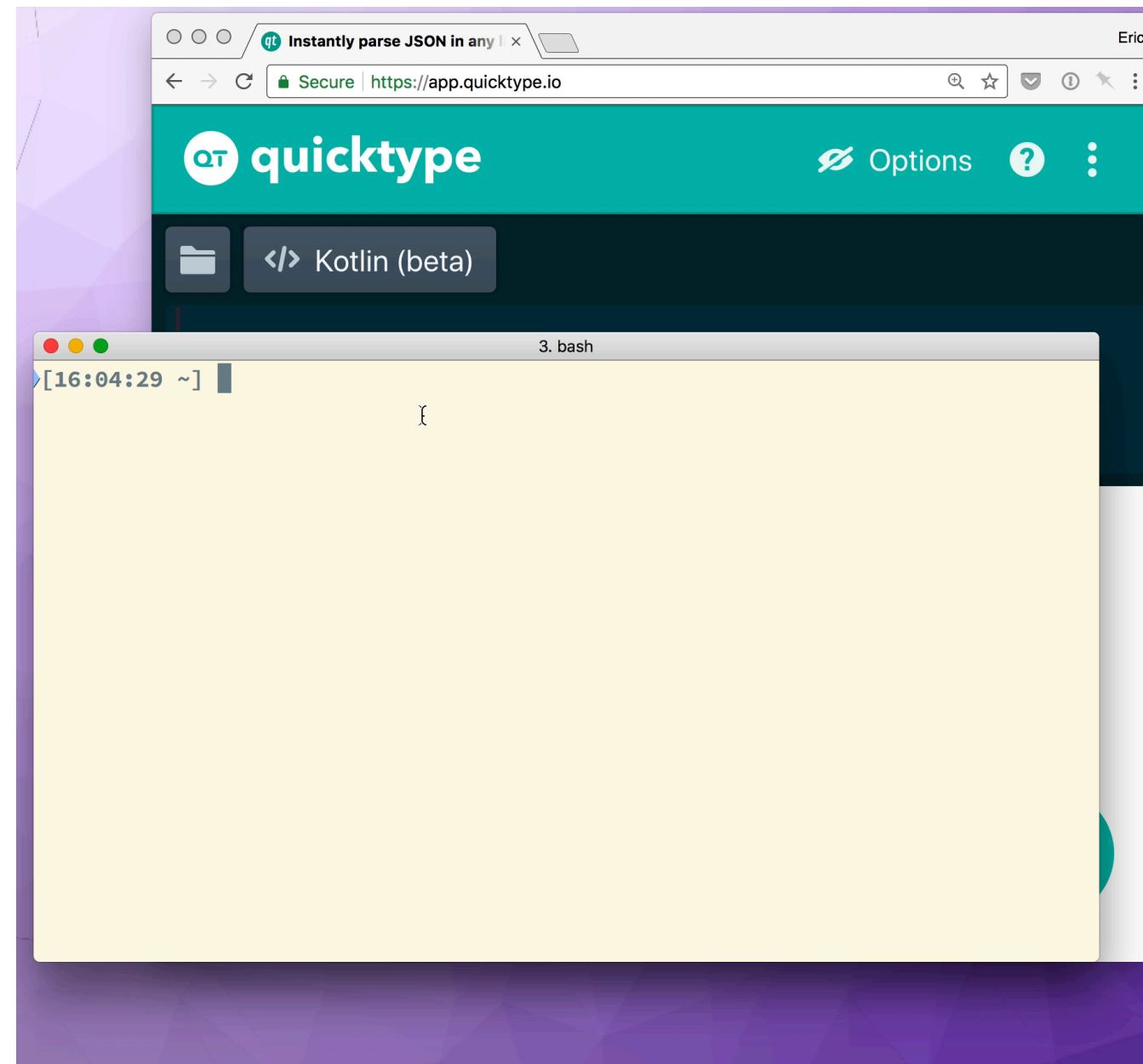
```
function slack_workspace_login() {  
    adb shell input text $1          # workspace  
    adb shell input tap 400 1000    # next  
    sleep 1  
    adb shell input text $2          # email  
    sleep 1  
    adb shell input tap 400 1000    # next  
    sleep 1  
    adb shell input text $3  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

— Naive, but good enough usually



Code generation

Model generation superpower



quicktype: automatic data models

- Web app and command-line tool to generate models from JSON
- Rapidly integrate with an unfamiliar API lacking an SDK
- Get sample JSON with curl, then paste into website

Scripting: write code to write code

- Use high-level language, e.g. Python, Ruby
- Turn data into code via templates
 - e.g. CSV to Kotlin enum values
 - e.g. CSV to bulk Google IAP SKUs

Summary



Summary

- Basic command-line setup
- How to make the command-line easier to use
- Many specialized tools 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's for	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
Gradle completion	bash completions for Gradle tasks	 gradle-completion
autojump	navigate directories faster	 autojump
The Silver Searcher (ag)	fast, intelligent code searcher	 the_silver_searcher
curl	transfer data with URLs	 curl
		everything curl book

¹  denotes Homebrew package name

Appendix: How to install (2/2)

Name	What it's for	How to get it ¹
gron	make JSON line-oriented and searchable	 gron
imgcat	display inline bitmaps in iTerm2	imgcat
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
quicktype	generate data classes from JSON	app.quicktype.io
		 quicktype

¹  denotes Homebrew package name