

Creating a Command Line Driven Dev Environment

- Practical / real world workflows
- Parse data on the command line
- Terminal, shell, scripting, prompt & history
- Using tmux and Vim
- Managing your dotfiles

Everything will work on **Linux, macOS and Windows**



References and Slides

<https://github.com/nickjj/nyhackr-cli-dev-env>

 <https://nickjanetakis.com>

 <https://twitter.com/nickjanetakis>

 <https://runninginproduction.com>

 Freelance Developer | Podcast Host | Course Creator

 Python | Ruby | Bash | Docker | Ansible | Terraform

Quickly switch between projects

tmux? Yes please

Vim? Of course, but maybe

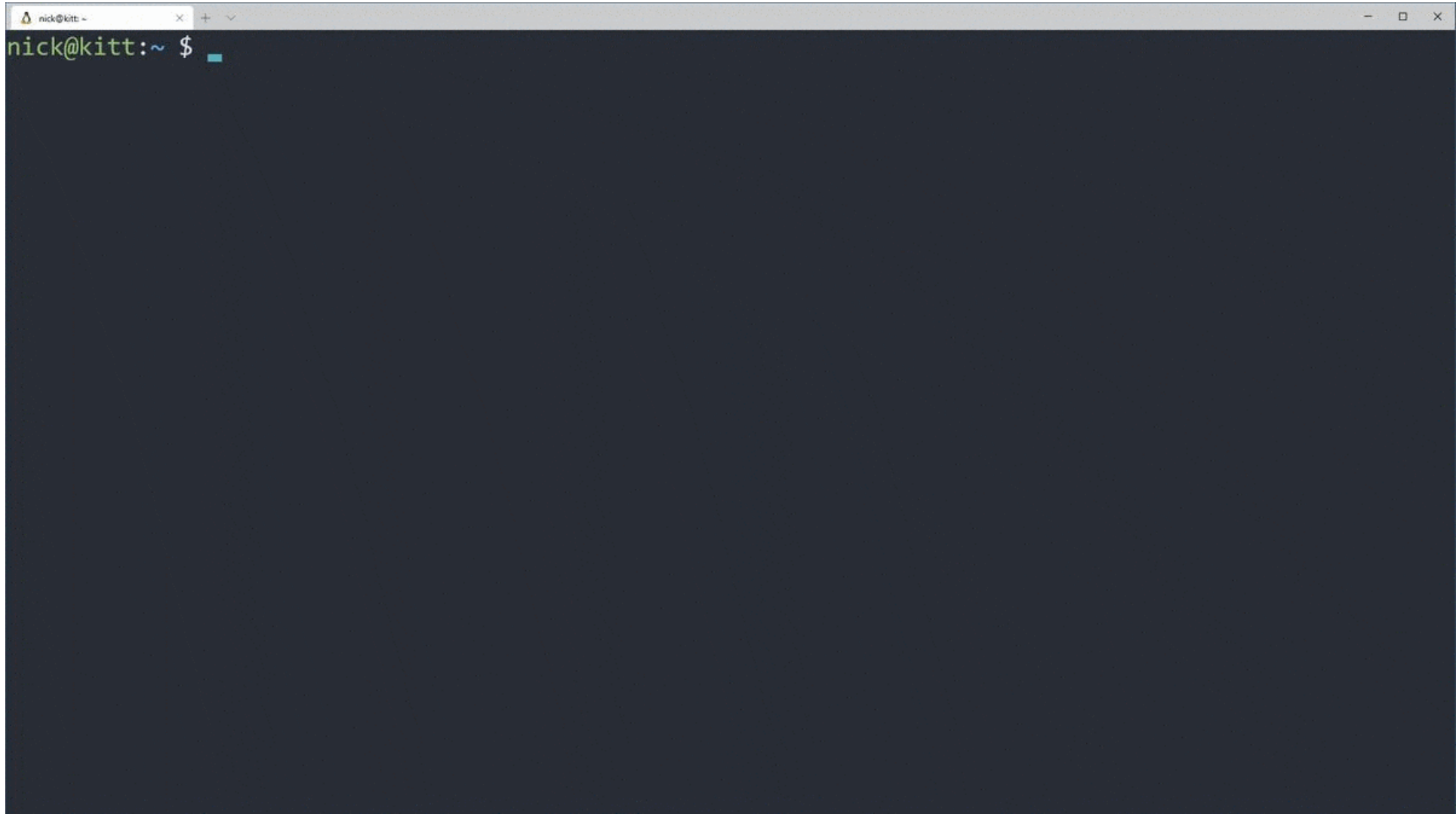
Practical / real world workflows

TIME FOR A ~~LIVE DEMO~~



WHAT COULD GO WRONG?

Attaching to an already running tmux session



Fuzzy searching for all tests in the project

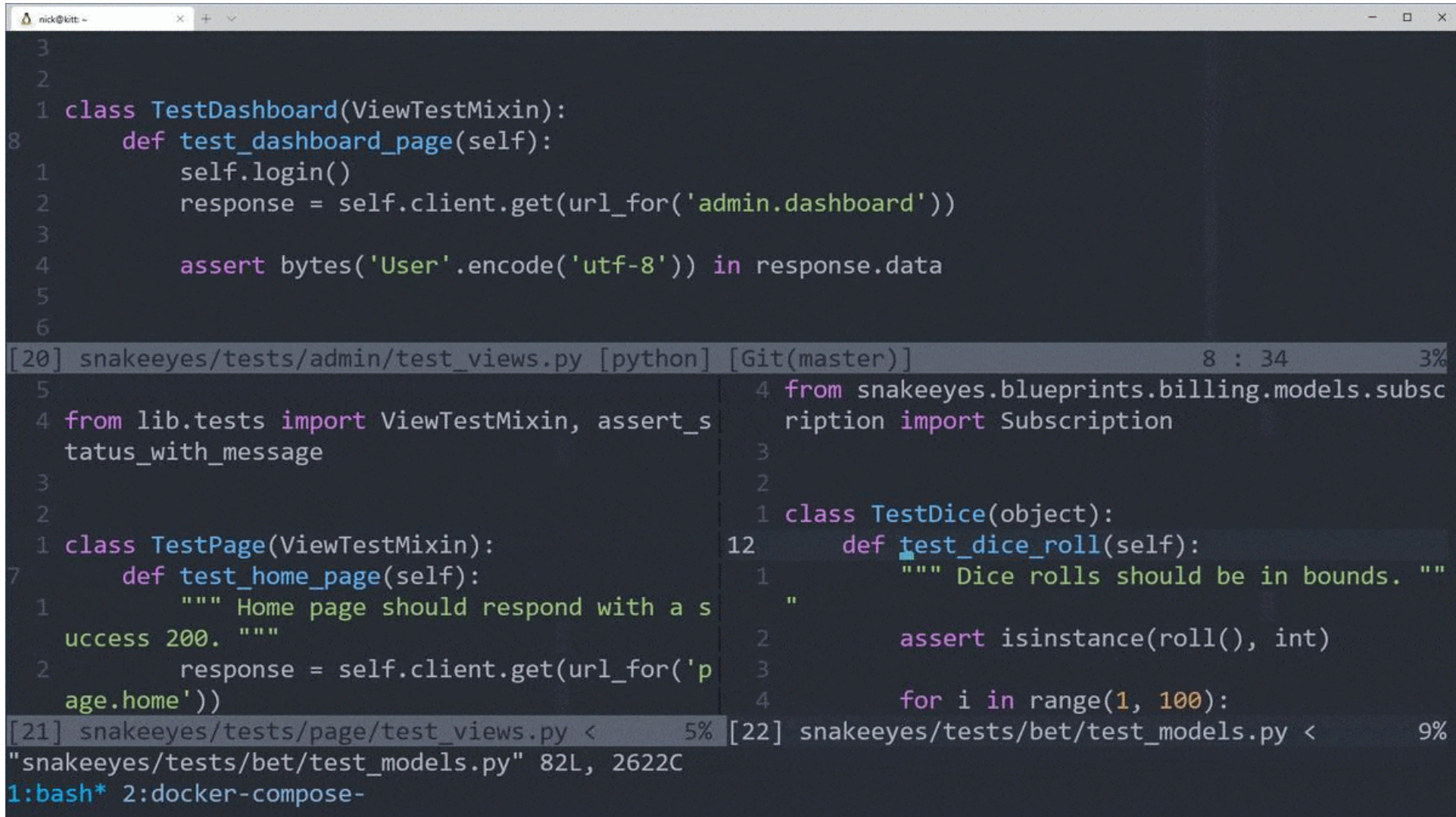
```
nick@kitt: ~  
5  
4     celery.Task = ContextTask  
3     return celery  
2  
1  
71 def create_app(settings_override=None):  
1     """  
2     Create a Flask application using the app factory pattern.  
3  
4     :param settings_override: Override settings  
5     :return: Flask app  
6     """  
7     app = Flask(__name__, static_folder='../public', static_url_path='')  
8  
9     app.config.from_object('config.settings')  
10  
11     if settings_override:  
12         app.config.update(settings_override)  
13  
14     stripe.api_key = app.config.get('STRIPE_SECRET_KEY')  
15     stripe.api_version = app.config.get('STRIPE_API_VERSION')  
16  
[4] snakeeyes/app.py [python] [Git(master)] 71 : 39 28%  
1: bash* 2: docker-compose-
```

Opening files in horizontal and vertical splits

```
nick@kitt: ~  
7 from flask import url_for  
6  
5 from lib.tests import ViewTestMixin, assert_status_with_message  
4 from snakeeyes.blueprints.user.models import User  
3  
2  
1 class TestDashboard(ViewTestMixin):  
8     def test_dashboard_page(self):  
1         self.login()  
2         response = self.client.get(url_for('admin.dashboard'))  
3  
4         assert bytes('User'.encode('utf-8')) in response.data  
5  
6  
7 class TestUsers(ViewTestMixin):  
8     def test_index_page(self):  
9         """ Index renders successfully. """  
10        self.login()  
11        response = self.client.get(url_for('admin.users'))  
12  
13        assert response.status_code == 200  
14  
[20] snakeeyes/tests/admin/test_views.py [python] [Git(master)]  
"snakeeyes/tests/admin/test_views.py" 136L, 4453C  
1: bash* 2: docker-compose-
```

1

Using Vim tabs to hold groups of buffers



```
nick@kitt: ~  
3  
2  
1 class TestDashboard(ViewTestMixin):  
8     def test_dashboard_page(self):  
1         self.login()  
2         response = self.client.get(url_for('admin.dashboard'))  
3  
4         assert bytes('User'.encode('utf-8')) in response.data  
5  
6  
[20] snakeeyes/tests/admin/test_views.py [python] [Git(master)] 8 : 34 3%  
5  
4 from lib.tests import ViewTestMixin, assert_s  
tatus_with_message  
3  
2  
1 class TestPage(ViewTestMixin):  
7     def test_home_page(self):  
1         """ Home page should respond with a s  
uccess 200. """  
2         response = self.client.get(url_for('p  
age.home'))  
[21] snakeeyes/tests/page/test_views.py < 5%  
"snakeeyes/tests/bet/test_models.py" 82L, 2622C  
1: bash* 2: docker-compose-
```

```
4 from snakeeyes.blueprints.billing.models.subsc  
ription import Subscription  
3  
2  
1 class TestDice(object):  
12     def test_dice_roll(self):  
1         """ Dice rolls should be in bounds. """  
"  
2         assert isinstance(roll(), int)  
3  
4         for i in range(1, 100):  
[22] snakeeyes/tests/bet/test_models.py < 9%
```

Using Vim to help make better git commit messages

```
nick@kitt: ~  
100101 Firefox/79.0' in 1929µs  
web_1      | 172.21.0.1 - - [27/Sep/2020:11:15:54 +0000] 'GET /_debug_toolbar/static/js/toolbar.js  
HTTP/1.1' 304 0 'http://localhost:8000/' 'Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:79.0) Gecko/2  
0100101 Firefox/79.0' in 1826µs  
web_1      | 172.21.0.1 - - [27/Sep/2020:11:15:54 +0000] 'GET /_debug_toolbar/static/js/jquery.tabl  
esorter.js HTTP/1.1' 304 0 'http://localhost:8000/' 'Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:79  
.0) Gecko/20100101 Firefox/79.0' in 3043µs  
web_1      | 172.21.0.1 - - [27/Sep/2020:11:15:54 +0000] 'GET /images/snake-eyes.jpg HTTP/1.1' 304  
0 'http://localhost:8000/' 'Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:79.0) Gecko/20100101 Firefo  
x/79.0' in 1666µs  
web_1      | 172.21.0.1 - - [27/Sep/2020:11:15:54 +0000] 'GET /_debug_toolbar/static/css/toolbar.cs  
s?0.038520099830944954 HTTP/1.1' 200 8117 'http://localhost:8000/' 'Mozilla/5.0 (Windows NT 10.0; Wi  
n64; x64; rv:79.0) Gecko/20100101 Firefox/79.0' in 1681µs  
web_1      | 172.21.0.1 - - [27/Sep/2020:11:15:54 +0000] 'GET /favicon.ico HTTP/1.1' 404 2717 '-' '  
Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:79.0) Gecko/20100101 Firefox/79.0' in 2824µs  
web_1      | 172.21.0.1 - - [27/Sep/2020:11:15:54 +0000] 'GET /_debug_toolbar/static/img/djdt_verti  
cal.png HTTP/1.1' 304 0 'http://localhost:8000/_debug_toolbar/static/css/toolbar.css?0.0385200998309  
44954' 'Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:79.0) Gecko/20100101 Firefox/79.0' in 1962µs  
redis_1    | 1:M 27 Sep 2020 12:10:14.088 * 1 changes in 3600 seconds. Saving...  
redis_1    | 1:M 27 Sep 2020 12:10:14.090 * Background saving started by pid 20  
redis_1    | 20:C 27 Sep 2020 12:10:14.101 * DB saved on disk  
redis_1    | 20:C 27 Sep 2020 12:10:14.102 * RDB: 6 MB of memory used by copy-on-write  
redis_1    | 1:M 27 Sep 2020 12:10:14.190 * Background saving terminated with success  
1: bash- 2: docker-compose*
```

Zooming in and out of tmux split panes

```
nick@kitt: ~  
n64; x64; rv:79.0) Gecko/20100101 Firefox/79.0' in 1681µs  
web_1 | 172.21.0.1 - - [27/Sep/2020:11:15:54 +0000] 'GET /favicon.ico HTTP/1.1' 404 2717 '-' '  
Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:79.0) Gecko/20100101 Firefox/79.0' in 2824µs  
web_1 | 172.21.0.1 - - [27/Sep/2020:11:15:54 +0000] 'GET /_debug_toolbar/static/img/djdt_verti  
cal.png HTTP/1.1' 304 0 'http://localhost:8000/_debug_toolbar/static/css/toolbar.css?0.0385200998309  
44954' 'Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:79.0) Gecko/20100101 Firefox/79.0' in 1962µs  
redis_1 | 1:M 27 Sep 2020 12:10:14.088 * 1 changes in 3600 seconds. Saving...  
redis_1 | 1:M 27 Sep 2020 12:10:14.090 * Background saving started by pid 20  
redis_1 | 20:C 27 Sep 2020 12:10:14.101 * DB saved on disk  
redis_1 | 20:C 27 Sep 2020 12:10:14.102 * RDB: 6 MB of memory used by copy-on-write  
redis_1 | 1:M 27 Sep 2020 12:10:14.190 * Background saving terminated with success  
  
nick@kitt:~/src/courses/bsawf (master)$  
  
1: bash- 2: docker-compose*
```

Switching tmux sessions and searching your history

```
nick@kitt: ~  
(0) + bsawf-0: 2 windows (group bsawf: bsawf-0) (attached)  
(1) + podcast-1: 2 windows (group podcast: podcast-1)  
  
bsawf-0 (sort: index)  
8 APP_DEPS=""  
9  
[23] Dockerfile [Dockerfile] [Git(master)]  
1 version: "3.4"  
1  
2 services: 1: bash  
3 postgres:  
4 env_file:  
5 - ".env"  
6 image: "postgres:12.3"  
  
Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:79.0) Gecko/20100101 Firefox/79.0  
web_1 | 172.21.0.1 - - [27/Sep/2020:11:15:00] "GET /cal.png HTTP/1.1" 304 0 "http://localhost:8000/_  
44954" 'Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:79.0) Gecko/20100101 Firefox/79.0'  
redis_1 | 0:14.088 * 1 c  
redis_1 | 2: docker-compose 0:14.090 * Bac  
redis_1 | 10:14.101 * DB  
redis_1 | 20:C 27 Sep 2020 12:10:14.102 * RD  
redis_1 | 1:M 27 Sep 2020 12:10:14.190 * Bac  
  
1: bash- 2: docker-compose*Z
```

Real time GitHub styled Markdown preview in Vim

The image shows a side-by-side comparison of a GitHub README preview and its raw Markdown source code in Vim.

Left Panel (GitHub Preview):

- Browser tab: [README]
- Address bar: 127.0.0.1:9031/page/4
- Page title: README
- Section header:

What is ansible-docker?
- Build status: build passing
- Text: It is an [Ansible](#) role to:
- List of features:
 - Install Docker (editions, channels and version pinning are all supported)
 - Install Docker Compose using PIP (version pinning is supported)
 - Install the `docker` PIP package so Ansible's `docker_*` modules work
 - Manage Docker registry login credentials
 - Configure 1 or more users to run Docker without needing root access
 - Configure the Docker daemon's options and environment variables
 - Configure a cron job to run Docker clean up commands
- Section header:

Why would you want to use this role?
- Text: If you're like me, you probably love Docker. This role provides everything you need to get going with a production ready Docker host.
- Text: By the way, if you don't know what Docker is, or are looking to become an expert with it then check out [Dive into Docker: The Complete Docker Course for Developers](#).

Right Panel (Vim Editor):

- Terminal window: nick@kitt ~
- Raw Markdown source code:

```
1 ## What is ansible-docker? [!Build Status](https://travis-ci.org/nickjj/ansible-docker.svg?branch=master)](https://travis-ci.org/nickjj/ansible-docker)
2 It is an [Ansible](http://www.ansible.com/home) role to:
3
4 - Install Docker (editions, channels and version pinning are all supported)
5 - Install Docker Compose using PIP (version pinning is supported)
6 - Install the `docker` PIP package so Ansible's `docker_` modules work
7 - Manage Docker registry login credentials
8 - Configure 1 or more users to run Docker without needing root access
9 - Configure the Docker daemon's options and environment variables
14] README.md <ster)] 1 : 4 Top
<DME.md" 410L, 12698C written c
1: bash*
```

OK, let's give tmux and Vim a break

```
nick@kitt: ~ $ drafts a cool blog post idea_
```

2

```
[1] src/sites/nickjanetakis/_drafts/2020-09-24-a-cool-blog-post-idea.md" 1L, 34C written
```

```
nick@kitt: ~  
1  #!/usr/bin/env bash  
2  # Create a dated Markdown file at a specific location.  
3  #  
4  # Usage:  
5  #   $ drafts some post idea (generate a new draft)  
6  #   $ drafts                (list all of your drafts)  
7  #  
8  # Produces:  
9  #   YYYY-MM-DD-some-post-idea.md in your $DRAFTS_DIRECTORY (this is set below).  
10  
11 set -e  
12  
13 readonly DRAFTS_DIRECTORY="${DRAFTS_DIRECTORY:-"${HOME}"/src/sites/nickjanetakis/_drafts}"  
14 readonly DRAFTS_EDITOR="${EDITOR}"  
15  
16 readonly DRAFTS_FILE="$(date +%Y-%m-%d)-${*}.md"  
17 readonly DRAFTS_PATH="${DRAFTS_DIRECTORY}/${DRAFTS_FILE// /-}"  
18  
19 if [[ $# -eq 0 ]]; then  
20     ls "${DRAFTS_DIRECTORY}"  
21     exit 0  
22 fi  
23  
24 if [ -e "${DRAFTS_PATH}" ]; then  
25     eval "${DRAFTS_EDITOR}" "${DRAFTS_PATH}"  
26 else  
27     touch "${DRAFTS_PATH}"  
28 fi  
[1] .local/bin/drafts [sh]
```

1 : 1 All

```
nick@kitt: ~ $  
  
nick@kitt: ~ $  
  
1: bash*
```

\$ weather nyc

nick@kitt: ~

Overcast
71..77 °F
↑ 0 mph
9 mi
0.0 in

Morning	Noon	Evening	Night
~/'"-. Patchy rain po... ,_(). 71 °F /(()) ↑ 7-9 mph 6 mi 0.0 in 29%	~/'"-. Light rain sho... ,_(). 75..78 °F /(()) ↑ 9-11 mph 6 mi 0.0 in 75%	~/'"-. Moderate or he... ,_(). 75..78 °F /(()) ↑ 14-23 mph 4 mi 0.1 in 76%	~/'"-. Light rain sho... ,_(). 73..77 °F /(()) ↑ 16-24 mph 4 mi 0.1 in 78%

A simple function

```
weather () {  
  curl https://wttr.in/"${1}"  
}
```

<https://wttr.in/:help>

Dozens of CLI tools

`find | sed | grep | cut | ...and friends`

```
$ cat title | cowsay -W 25
```

```
-----  
/ Obtaining, Scrubbing, \  
| and Exploring Data at |  
| the Command Line      |  
|                        |  
| Jeroen Janssens       |  
|                        |  
\  
\ @jeroenhjanssens      /  
-----
```



Yay for Unix commands
and writing your own scripts

Parsing out sales stats from a CSV file

gateway	transaction_at	gross_cents	fee_cents	net_cents	email	product
stripe	2020-09-29 01:10	5900	200	5700	hello@example.com	bsawf
stripe	2020-09-29 05:26	4900	172	4728	zero@cool.com	docker
stripe	2020-08-29 11:17	5900	200	5700	unix@pipes.com	bsawf

Vim has decent support for CSV files



```
nick@kitt: ~/src/talks/cli/examp x + v
1 gateway transaction_at gross_cents fee_cents net_cents email product
1 stripe 2020-09-29 01:10 5900 200 5700 hello@example.com bsawf
2 stripe 2020-09-29 05:26 4900 172 4728 zero@cool.com docker
3 stripe 2020-08-29 11:17 5900 200 5700 unix@pipes.com bsawf
~
~
~
[4] ~/src/talks/cli/examples/001-sales.csv [csv] 1 : 1 All
6
5 gateway transaction_at gross_cents fee_cents net_cents email product
4
3 stripe 2020-09-29 01:10 5900 200 5700 hello@example.com bsawf
2 stripe 2020-09-29 05:26 4900 172 4728 zero@cool.com docker
1 stripe 2020-08-29 11:17 5900 200 5700 unix@pipes.com bsawf
7
~
~
~
~
~
~
~
[5] [No Name] [+] 7 : 109-37 All
```

gateway	transaction_at	gross_cents	fee_cents	net_cents	email	product
stripe	2020-09-29 01:10	5900	200	5700	hello@example.com	bsawf
stripe	2020-09-29 05:26	4900	172	4728	zero@cool.com	docker
stripe	2020-08-29 11:17	5900	200	5700	unix@pipes.com	bsawf

```
$ sed 1d 001-sales.csv
```

```
stripe,2020-09-29 01:10,5900,200,5700,hello@example.com,bsawf  
stripe,2020-09-29 05:26,4900,172,4728,zero@cool.com,docker  
paypal,2020-08-29 11:17,5900,200,5700,unix@pipes.com,bsawf
```

WHY MUST THERE

BE SO MANY CHOICES?

memegenerator.net

```
$ tail -n +2 001-sales.csv  
$ sed 1d 001-sales.csv
```

```
stripe,2020-09-29 01:10,5900,200,5700,hello@example.com,bsawf  
stripe,2020-09-29 05:26,4900,172,4728,zero@cool.com,docker  
paypal,2020-08-29 11:17,5900,200,5700,unix@pipes.com,bsawf
```

Readability > Performance (sometimes)

Tail is still useful

```
$ sed 1d 001-sales.csv | cut -d "," -f 5
```

```
5700  
4728  
5700
```

```
sed 1d 001-sales.csv
```

```
stripe,2020-09-29 01:10,5900,200,5700,hello@example.com,bsawf  
stripe,2020-09-29 05:26,4900,172,4728,zero@cool.com,docker  
paypal,2020-08-29 11:17,5900,200,5700,unix@pipes.com,bsawf
```

There's lots of tools to choose from

```
awk | date | diff | grep | head | tail | sed | sort | uniq | ...
```

```
$ sed 1d 001-sales.csv | cut -d "," -f 5
```

```
5700  
4728  
5700
```

```
sed 1d 001-sales.csv
```

```
stripe,2020-09-29 01:10,5900,200,5700,hello@example.com,bsawf  
stripe,2020-09-29 05:26,4900,172,4728,zero@cool.com,docker  
paypal,2020-08-29 11:17,5900,200,5700,unix@pipes.com,bsawf
```

```
$ sed 1d 001-sales.csv | cut -d "," -f 4
```

```
200  
172  
200
```

```
sed 1d 001-sales.csv
```

```
stripe,2020-09-29 01:10,5900,200,5700,hello@example.com,bsawf  
stripe,2020-09-29 05:26,4900,172,4728,zero@cool.com,docker  
paypal,2020-08-29 11:17,5900,200,5700,unix@pipes.com,bsawf
```

```
$ sed 1d 001-sales.csv | cut -d "," -f 5 | paste -sd+
```

```
5700+4728+5700
```

```
sed 1d 001-sales.csv
```

```
stripe,2020-09-29 01:10,5900,200,5700,hello@example.com,bsawf  
stripe,2020-09-29 05:26,4900,172,4728,zero@cool.com,docker  
paypal,2020-08-29 11:17,5900,200,5700,unix@pipes.com,bsawf
```

```
$ sed 1d 001-sales.csv | cut -d "," -f 5 | paste -sd+ | bc
```

```
16128
```

```
sed 1d 001-sales.csv
```

```
stripe,2020-09-29 01:10,5900,200,5700,hello@example.com,bsawf  
stripe,2020-09-29 05:26,4900,172,4728,zero@cool.com,docker  
paypal,2020-08-29 11:17,5900,200,5700,unix@pipes.com,bsawf
```

```
$ grep ",bsawf$" 001-sales.csv | cut -d "," -f 5 | paste -sd+ | b
```

```
11400
```

```
sed 1d 001-sales.csv
```

```
stripe,2020-09-29 01:10,5900,200,5700,hello@example.com,bsawf  
stripe,2020-09-29 05:26,4900,172,4728,zero@cool.com,docker  
paypal,2020-08-29 11:17,5900,200,5700,unix@pipes.com,bsawf
```

```
# Both of these produce the same output.  
$ grep ",2020-09-" 001-sales.csv | wc -l  
$ grep ",2020-09-" 001-sales.csv -c
```

```
2
```

```
sed 1d 001-sales.csv
```

```
stripe,2020-09-29 01:10,5900,200,5700,hello@example.com,bsawf  
stripe,2020-09-29 05:26,4900,172,4728,zero@cool.com,docker  
paypal,2020-08-29 11:17,5900,200,5700,unix@pipes.com,bsawf
```

Command crash course complete

Picking a terminal and shell

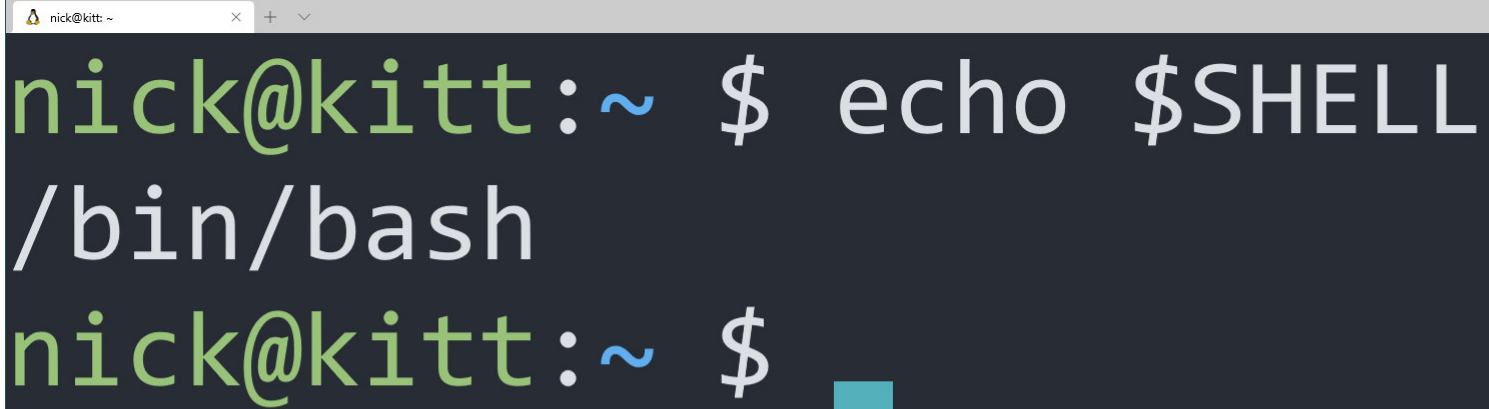
tmux super charges your terminal

- Speed / low input latency
- Customizable hotkeys
- Zoom hotkeys
- Clickable URLs
- Unicode / Emoji support

- **Windows**
 - Microsoft Terminal
 - wsltty
- **Linux**
 - xterm
- **macOS**
 - iTerm2
- **Cross platform**
 - Alacritty

Speed and quality of life improvements

```
nick@kitt: ~ $ echo $0
-bash
nick@kitt: ~ $ █
```



A terminal window with a dark background and a light gray title bar. The title bar contains the text "nick@kitt: ~" and standard window control icons. The terminal shows a user prompt "nick@kitt:~" in green, followed by a command "\$ echo \$SHELL" in white. The output "/bin/bash" is displayed on the next line. The prompt "nick@kitt:~" appears again, followed by a white dollar sign and a blue cursor block.

```
nick@kitt:~ $ echo $SHELL
/bin/bash
nick@kitt:~ $
```

Bourne Shell (**sh**) 1979

Bourne Again Shell (**bash**) 1989

```
nick@kitt: ~  
$ echo $0  
-bash  
nick@kitt: ~  
$
```

```
nick@kitt: ~ $ sh
$ echo $0
sh
$ echo $SHELL
/bin/bash
$
```

Try to use **sh** when writing shell scripts

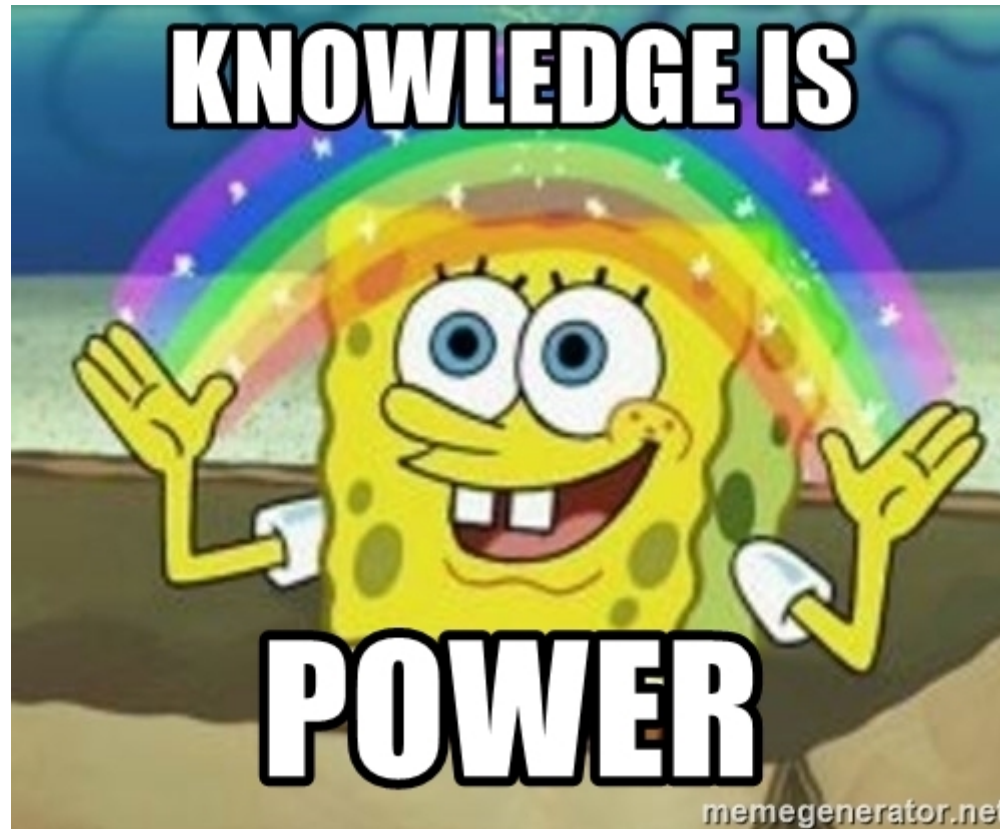
Z Shell, AKA **zsh**

<https://github.com/zsh-users/zsh>

macOS Catalina (2019) defaults to zsh
but Bash is still available to use

Friendly Interactive Shell, AKA **fish**

<https://fishshell.com/>





DO NOT SHAVE

A HAIRY YAK IS A HAPPY YAK

bash, **zsh** and **fish** share a decent
amount of configuration

```
$ ls -la ~
```

```
-rw-r--r--  1 nick nick    38 Aug 17 11:39 .bashrc  
-rw-r--r--  1 nick nick    39 May 27 21:35 .profile
```

```
$ cat ~/.profile
```

```
#!/usr/bin/env bash  
# This file runs once at login.  
  
# ... (22 lines including comments)
```

```
$ cat ~/.profile
```

```
# ...
```

```
# Add all local binary paths to the system path.
```

```
export PATH="${PATH}:${HOME}/.local/bin"
```

```
$ cat ~/.profile
```

```
# ...
```

```
# Default programs to run.
```

```
export EDITOR="vim"
```

```
$ cat ~/.profile
```

```
# ...
```

```
# Add colors to the less and man commands.
```

```
export LESS=-R
```

```
export LESS_TERMCAP_mb="$ (printf '%b' '\001[1;31m') "; a="${a%_}"
```

```
export LESS_TERMCAP_md="$ (printf '%b' '\001[1;36m') "; a="${a%_}"
```

```
export LESS_TERMCAP_me="$ (printf '%b' '\001[0m') "; a="${a%_}"
```

```
export LESS_TERMCAP_so="$ (printf '%b' '\001[01;44;33m') "; a="${a%_}"
```

```
export LESS_TERMCAP_se="$ (printf '%b' '\001[0m') "; a="${a%_}"
```

```
export LESS_TERMCAP_us="$ (printf '%b' '\001[1;32m') "; a="${a%_}"
```

```
export LESS_TERMCAP_ue="$ (printf '%b' '\001[0m') "; a="${a%_}"
```

\$ man grep

```
nick@kitt: ~  
GREP(1) User Commands GREP(1)  
  
NAME  
    grep, egrep, fgrep, rgrep - print lines that match patterns  
  
SYNOPSIS  
    grep [OPTION...] PATTERNS [FILE...]  
    grep [OPTION...] -e PATTERNS ... [FILE...]  
    grep [OPTION...] -f PATTERN_FILE ... [FILE...]  
  
DESCRIPTION  
    grep searches for PATTERNS in each FILE. PATTERNS is one or more patterns separated by  
    newline characters, and grep prints each line that matches a pattern. Typically PATTERNS  
    should be quoted when grep is used in a shell command.  
  
    A FILE of "-" stands for standard input. If no FILE is given, recursive searches examine  
    the working directory, and nonrecursive searches read standard input.  
  
    In addition, the variant programs egrep, fgrep and rgrep are the same as grep -E, grep -F,  
    and grep -r, respectively. These variants are deprecated, but are provided for backward  
    compatibility.  
  
OPTIONS  
    Generic Program Information  
    Manual page grep(1) line 1 (press h for help or q to quit)
```

```
$ cat ~/.profile
```

```
# ...
```

```
# If bash is the login shell, then source ~/.bashrc if it exists.  
echo "${0}" | grep "bash$" >/dev/null \  
    && [ -f "${HOME}/.bashrc" ] && source "${HOME}/.bashrc"
```

```
$ cat ~/.bashrc
```

```
#!/usr/bin/env bash  
# This file runs every time you open a new terminal window.  
# ... (73 lines including comments)
```

```
$ cat ~/.bashrc
```

```
# ...

# HISTFILESIZE controls the history lines saved on disk and
# HISTSIZE controls lines stored in memory.
export HISTFILESIZE=50000
export HISTSIZE=50000

# Add a timestamp to each command.
export HISTTIMEFORMAT="%Y/%m/%d %H:%M:%S:  "

# Duplicates and lines starting with a space get ignored.
export HISTCONTROL=ignoreboth

# Append to the history file, don't overwrite it.
shopt -s histappend
```

```
$ ls -la ~
```

```
-rw----- 1 nick nick    555 Sep 29 16:10 .bash_history  
-rw----- 1 nick nick 58967 Sep 29 16:10 .bash_history.BAK
```

```
$ history
```

```
1 2020/09/29 06:26:08: cd ~/src/ansible
3 2020/09/29 08:35:53: cd ~
4 2020/09/29 08:35:54: ls -la
5 2020/09/29 08:36:01: vim .bash_history
6 2020/09/29 08:36:59: exit
```

```
$ cat ~/.bashrc
```

```
# ...
```

```
# Ensure $LINES and $COLUMNS always get updated.
```

```
shopt -s checkwinsize
```

```
# Enable bash completion.
```

```
[ -f /etc/bash_completion ] && source /etc/bash_completion
```

```
# Improve output of less for binary files.
```

```
[ -x /usr/bin/lesspipe ] && eval "$(SHELL=/bin/sh lesspipe)"
```

```
$ ls -la /etc/skel/
```

```
-rw-r--r--  1 root root  220 Feb 25  2020 .bash_logout  
-rw-r--r--  1 root root 3771 Feb 25  2020 .bashrc  
-rw-r--r--  1 root root  807 Feb 25  2020 .profile
```

```
$ cat ~/.bashrc
```

```
# ...
```

```
# Load aliases if they exist.
```

```
[ -f "${HOME}/.aliases" ] && source "${HOME}/.aliases"
```

```
[ -f "${HOME}/.aliases.local" ] && source "${HOME}/.aliases.local"
```

```
$ ls -la ~
```

```
-rw-r--r--  1 nick nick 2362 Sep 23 10:17 .aliases  
-rw-r--r--  1 nick nick  682 Mar 08  2020 .aliases.local
```

```
$ cat ~/.aliases
```

```
if [ -x /usr/bin/dircolors ]; then
    test -r ~/.dircolors && eval "$(dircolors -b ~/.dircolors
        || eval "$(dircolors -b) "

    alias ls="ls --color=auto"

    alias grep="grep --color=auto"
    alias fgrep="fgrep --color=auto"
    alias egrep="egrep --color=auto"
fi

alias ll="ls -alF"
alias la="ls -A"
alias l="ls -CF"
```

```
$ cat ~/.aliases
```

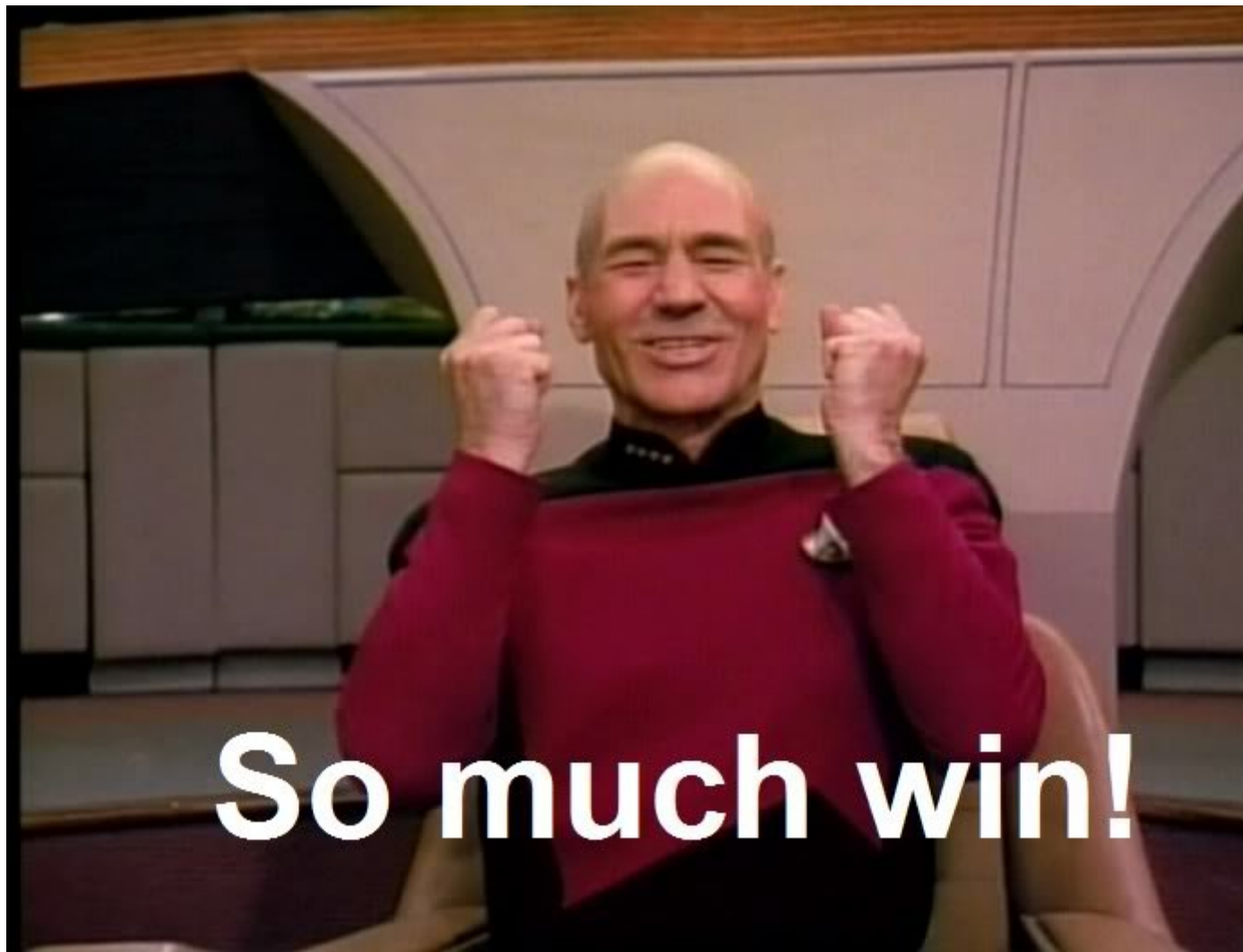
```
# ...
```

```
weather () {  
    curl https://wttr.in/"${1}"  
}
```

```
$ cat ~/.aliases
```

```
# ...
```

```
alias toggle-dark-mode="toggle-dark-mode && source ${HOME}/.bashrc"
```



So much win!

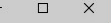
Let's talk about prompts

nick@kitt: /tmp

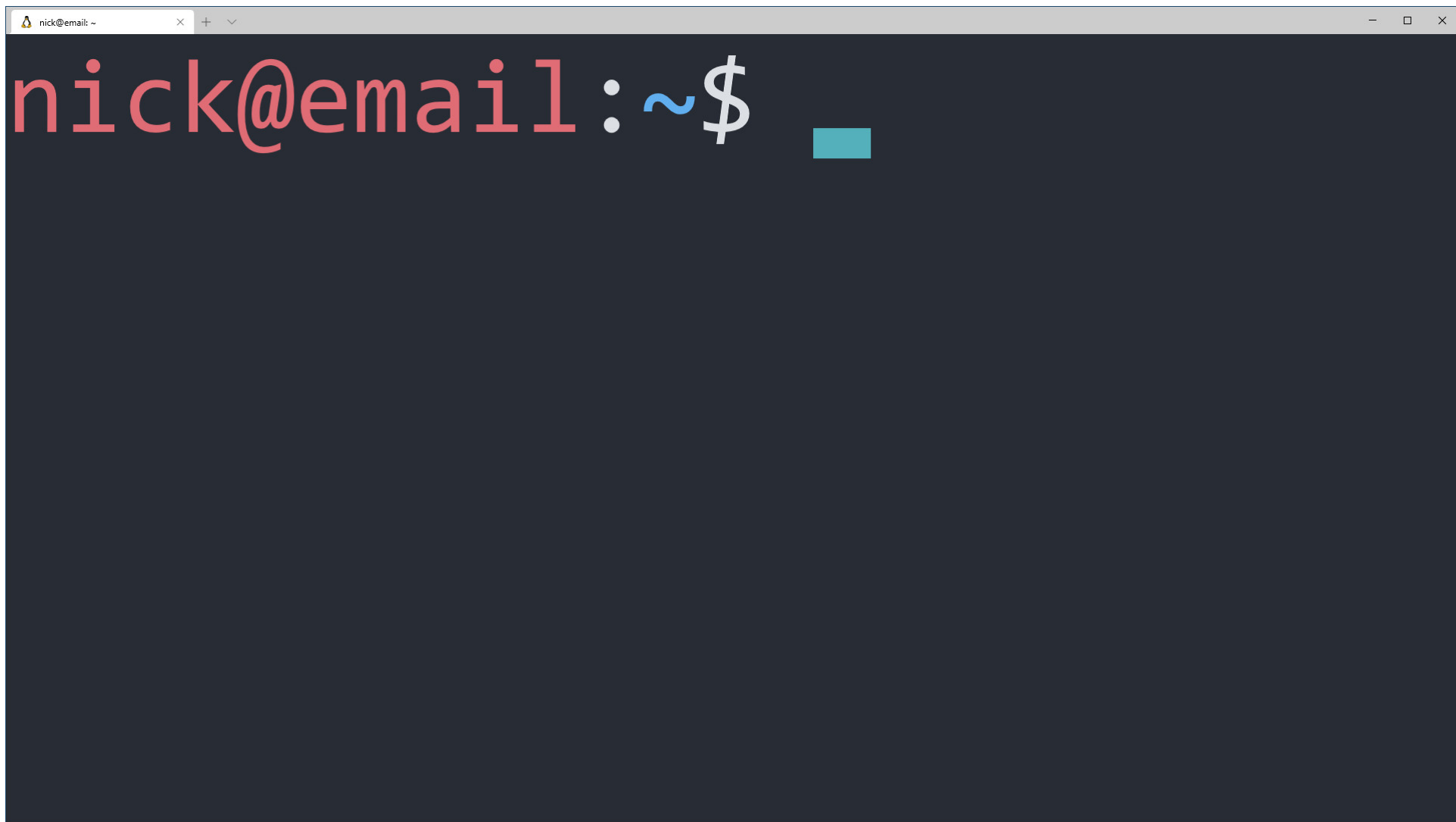


nick@kitt: /tmp \$ █

nick@kitt: ~/src/courses/bsawf



```
nick@kitt:~/src/courses/bsawf (master)$
```



1: Terminal ▾



```
~/Downloads > git clone https://github.com/starship/starship
Cloning into 'starship'...
remote: Enumerating objects: 37, done.
remote: Counting objects: 100% (37/37), done.
remote: Compressing objects: 100% (30/30), done.
remote: Total 2331 (delta 14), reused 14 (delta 6), pack-reused 2294
Receiving objects: 100% (2331/2331), 9.53 MiB | 11.35 MiB/s, done.
Resolving deltas: 100% (1570/1570), done.
~/Downloads took 3s > cd starship/
starship on  master is  v0.16.0 via  v1.36.0 > 
```

```
nick@kitt: /tmp $
```

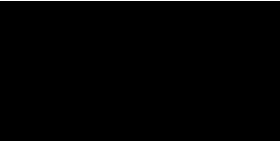
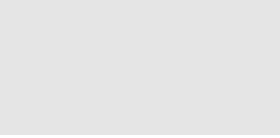
```
1 # Set a non-distracting prompt.  
2 PS1='\^[[01;32m\]\u@\h\^[[00m\]:\^[[01;34m\]\w\^[[00m\] \^[[01;33m\]$(parse_git_branch)\^[[00m\]\$ '
```

```
nick@kitt: /tmp $
```

```
1 # Set a non-distracting prompt.  
2 PS1='\[\^[[01;32m\]\u@\h\[\^[[00m\]:\[\^[[01;34m\]\w\[\^[[00m\] \[\^[[01;33m\]$(parse_git_branch)\[\^[[00m\]\$ '
```

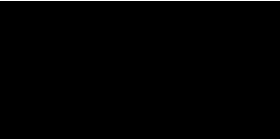
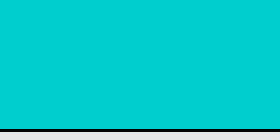
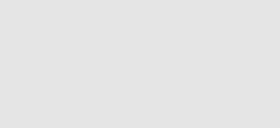
```
nick@kitt: /tmp $
```

```
1 # Set a non-distracting prompt.  
2 PS1='\^[[01;32m\]\u@\h\^[[00m\]:\^[[01;34m\]\w\^[[00m\] \^[[01;33m\]$(parse_git_branch)\^[[00m\]\$ '
```

Name	FG Code	BG Code	Color
Black	30	40	
Red	31	41	
Green	32	42	
Yellow	33	43	
Blue	34	44	
Magenta	35	45	
Cyan	36	46	
White	37	47	

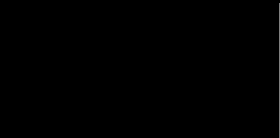
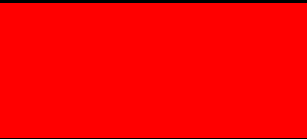
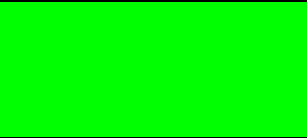
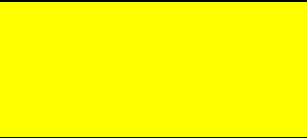
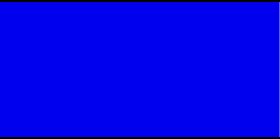
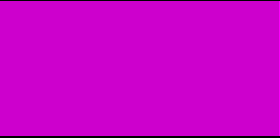
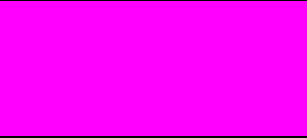
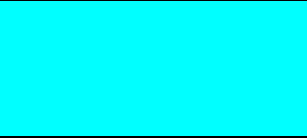
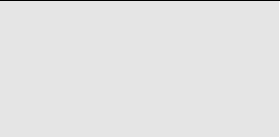
```
nick@kitt: /tmp $
```

```
1 # Set a non-distracting prompt.  
2 PS1='\[\^[[01;32m\]\u@\h\[\^[[00m\]:\[\^[[01;34m\]\w\[\^[[00m\] \[\^[[01;33m\]$(parse_git_branch)\[\^[[00m\]\$ '
```

Name	FG Code	BG Code	Color
Black	30	40	
Red	31	41	
Green	32	42	
Yellow	33	43	
Blue	34	44	
Magenta	35	45	
Cyan	36	46	
White	37	47	

```
nick@kitt: /tmp $
```

```
1 # Set a non-distracting prompt.  
2 PS1='\[\[\[01;32m\]\u@\h\[\[\[00m\]:\[\[\[01;34m\]\w\[\[\[00m\] \[\[\[01;33m\]$(parse_git_branch)\[\[\[00m\]\$ '
```

Name	FG Code	BG Code	Color	Bright
Black	30	40		
Red	31	41		
Green	32	42		
Yellow	33	43		
Blue	34	44		
Magenta	35	45		
Cyan	36	46		
White	37	47		

```
nick@kitt: /tmp $
```

```
1 # Set a non-distracting prompt.  
2 PS1='\[\^[[01;32m\]\u@\h\[\^[[00m\]:\[\^[[01;34m\]\w\[\^[[00m\] \[\^[[01;33m\]$(parse_git_branch)\[\^[[00m\]\$
```

```
nick@kitt: ~/src/courses/bsawf (master)$
```

```
1 # Determine git branch.
2 parse_git_branch() {
3     git branch 2> /dev/null | sed -e '/^[^*]/d' -e 's/* \(.*\)/(\1)/'
4 }
5
6 # Set a non-distracting prompt.
7 PS1='\[\001;32m\]\u@\h\[\001;34m\]:\[\001;34m\]\w\[\001;34m\] \[\001;33m\]$(parse_git_branch)\[\001;34m\]\$ '
```

nick@kitt: ~/src/courses/bsawf (master)\$

```
1 # Determine git branch.
2 parse_git_branch() {
3     git branch 2> /dev/null | sed -e '/^[^*]/d' -e 's/* \(.*\)/(\1)/'
4 }
5
6 # Set a non-distracting prompt.
7 PS1='\[\^[[01;32m\]\u@\h\[\^[[00m\]:\[\^[[01;34m\]\w\[\^[[00m\] \[\^[[01;33m\]$(parse_git_branch)\[\^[[00m\]\$ '
```

```
$ cat ~/.bashrc
```

```
# ...

# If it's an xterm compatible terminal, set title to user@host: d
case "${TERM}" in
xterm*|rxvt*)
    PS1="\[\e]0;\u@\h: \w\a\]${PS1}"
    ;;
*)
    ;;
esac
```

```
$ cat ~/.bashrc
```

```
# ...
```

```
# Enable asdf to manage various programming runtime versions.
```

```
# Requires: https://asdf-vm.com/#/
```

```
source "${HOME}"/.asdf/asdf.sh
```

```
$ cat ~/.bashrc
```

```
# ...  
  
# Enable a better reverse search experience.  
#   Requires: https://github.com/junegunn/fzf (to use fzf in general  
#   Requires: https://github.com/BurntSushi/ripgrep (for using rg be  
export FZF_DEFAULT_COMMAND="rg --files --hidden --follow --glob '!.g  
export FZF_DEFAULT_OPTS="--color=dark"  
[ -f "${HOME}/.fzf.bash" ] && source "${HOME}/.fzf.bash"
```

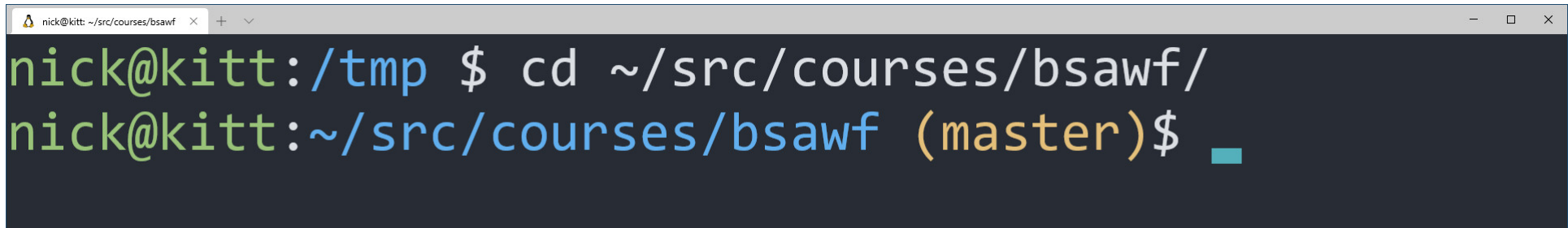
A side quest on searching your history

CTRL + r (without FZF)

A terminal window with a dark background. The title bar shows 'nick@kitt: /tmp' and standard window controls. The command '(reverse-i-search)`bs`: cd ~/src/courses/bsawf/' is entered, with 'bsawf/' highlighted in light blue.

```
(reverse-i-search)`bs`: cd ~/src/courses/bsawf/
```

CTRL + r (without FZF)

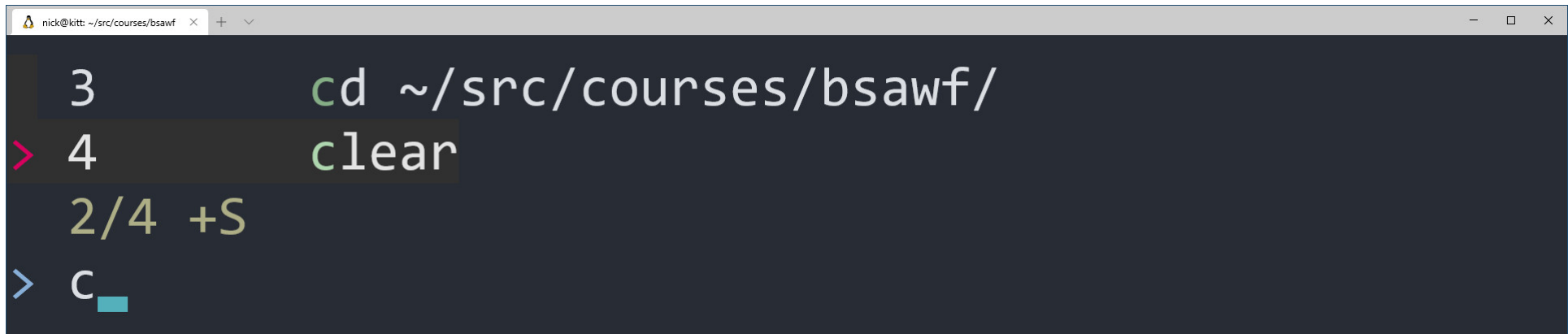
A terminal window with a dark background and light-colored text. The window title bar shows 'nick@kitt: ~/src/courses/bsawf' and standard window controls. The terminal content shows a user prompt 'nick@kitt:~/tmp' followed by a command 'cd ~/src/courses/bsawf/'. The next line shows the prompt 'nick@kitt:~/src/courses/bsawf' followed by '(master)\$' and a cursor. The text is color-coded: 'nick@kitt:' is green, paths are blue, and '(master)' is orange.

```
nick@kitt:~/tmp $ cd ~/src/courses/bsawf/  
nick@kitt:~/src/courses/bsawf (master)$
```

CTRL + r (with FZF)

```
nick@kitt: ~/src/courses/bsawf
1      ls -la
2      mv .bash_history .bash_history.BAK
3      cd ~/src/courses/bsawf/
> 4    clear
    4/4 +S
> 
```

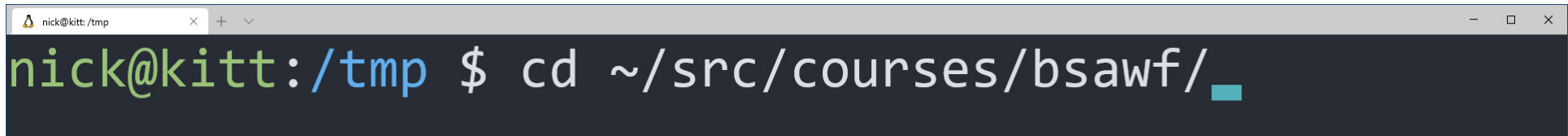
CTRL + r (with FZF)



```
nick@kitt: ~/src/courses/bsawf
3      cd ~/src/courses/bsawf/
> 4    clear
      2/4 +S
> C_
```

A terminal window with a dark background. The title bar shows 'nick@kitt: ~/src/courses/bsawf'. The terminal content shows a command history list with line numbers 3 and 4. Line 3 is 'cd ~/src/courses/bsawf/' and line 4 is 'clear'. A search filter '2/4 +S' is applied, highlighting line 4. The prompt '>' is followed by 'C_' and a cursor.

CTRL + r (with FZF)

A terminal window with a dark background. The title bar shows 'nick@kitt: /tmp' and standard window controls. The prompt is 'nick@kitt: /tmp \$'. The command 'cd ~/src/courses/bsawf/' is being entered, followed by a blue cursor block.

```
nick@kitt: /tmp $ cd ~/src/courses/bsawf/
```

```
$ cat ~/.bashrc
```

```
# ...  
  
# Enable a better reverse search experience.  
#   Requires: https://github.com/junegunn/fzf (to use fzf in general  
#   Requires: https://github.com/BurntSushi/ripgrep (for using rg be  
export FZF_DEFAULT_COMMAND="rg --files --hidden --follow --glob '!.g  
export FZF_DEFAULT_OPTS="--color=dark"  
[ -f "${HOME}/.fzf.bash" ] && source "${HOME}/.fzf.bash"
```

ripgrep

<https://github.com/BurntSushi/ripgrep>

Let's install everything a bit later

Once we cover dotfiles this process will be streamlined

```
# WSL 2 specific settings.
if grep -q "microsoft" /proc/version &>/dev/null; then
    export DISPLAY="$(/sbin/ip route | awk '/default/ { print $3 }'
fi

# WSL 1 specific settings.
if grep -qE "(Microsoft|WSL)" /proc/version &>/dev/null; then
    if [ "$(umask)" = "0000" ]; then
        umask 0022
    fi

    export DISPLAY=:0
fi
```



Shebang

```
#!/usr/bin/env bash
```

```
#!/usr/bin/env sh
```

```
#!/usr/bin/env python3
```

```
#!/usr/bin/env Rscript
```

```
#!/bin/bash
```

```
#!/usr/bin/env bash
```

POSIX compliance

Learning by example





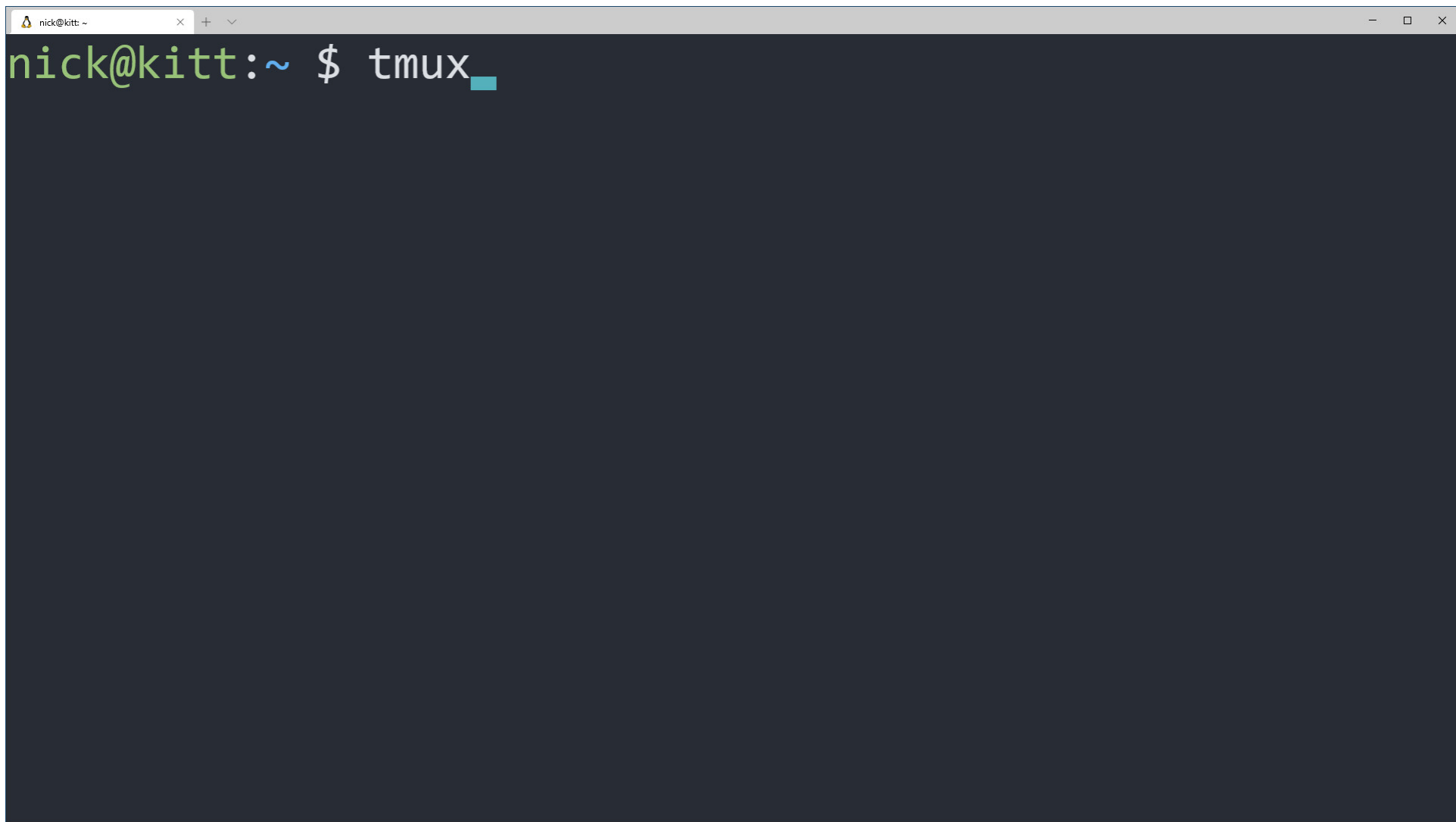
tmux

Installing tmux

Prefix / Leader key

CTRL + b is the default prefix key

I remapped mine to ` (back tick)

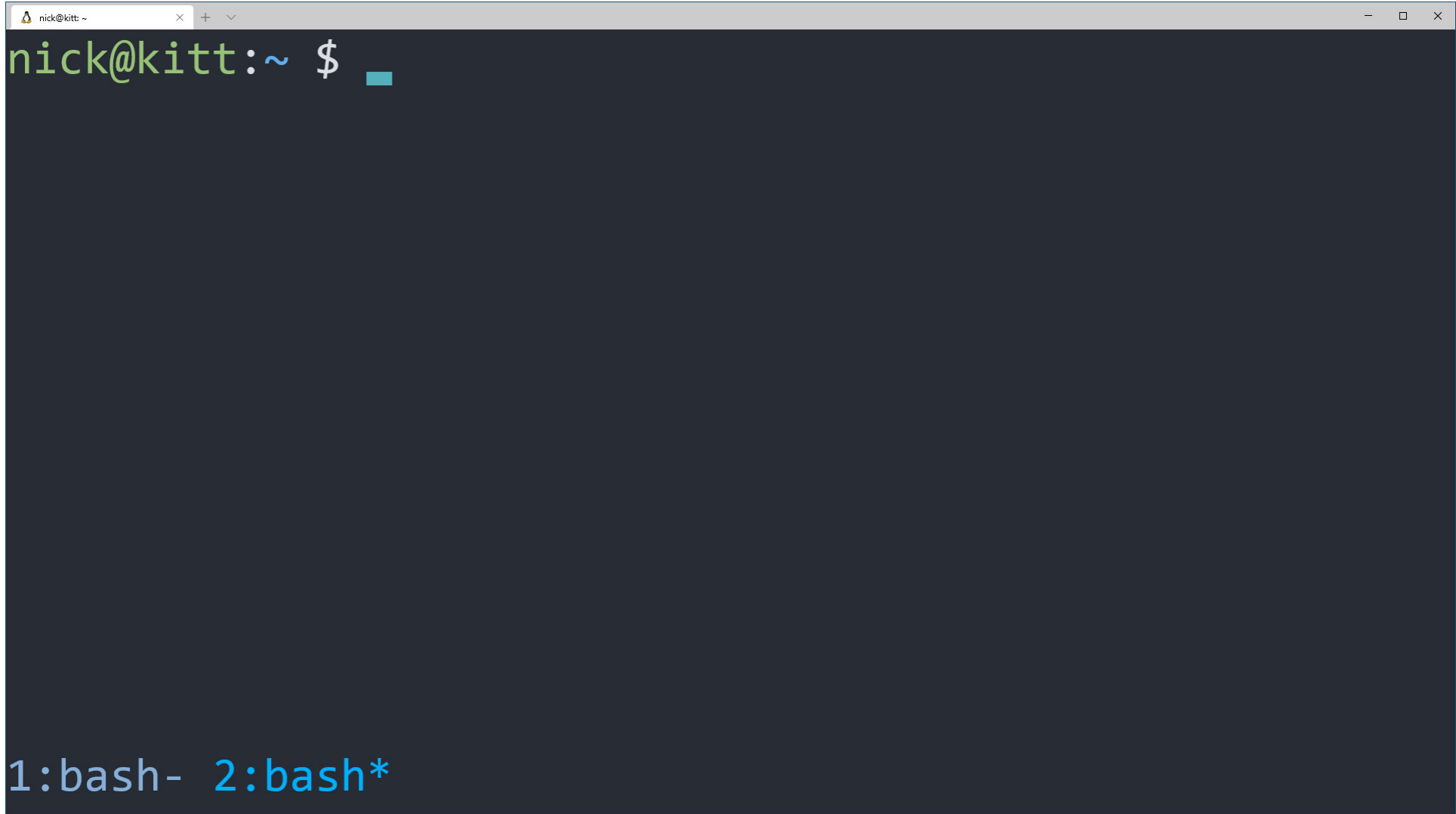


A terminal window with a dark blue background. The title bar at the top shows a small icon, the text "nick@kitt: ~", and window control buttons (close, maximize, and a dropdown arrow). The terminal content shows the prompt "nick@kitt:~" in green, followed by a white dollar sign "\$" and the word "tmux" in white. A small cyan cursor block is positioned at the end of the word "tmux".

```
nick@kitt:~ $ tmux
```

```
nick@kitt: ~  
nick@kitt:~ $  
  
1: bash*
```

prefix + c

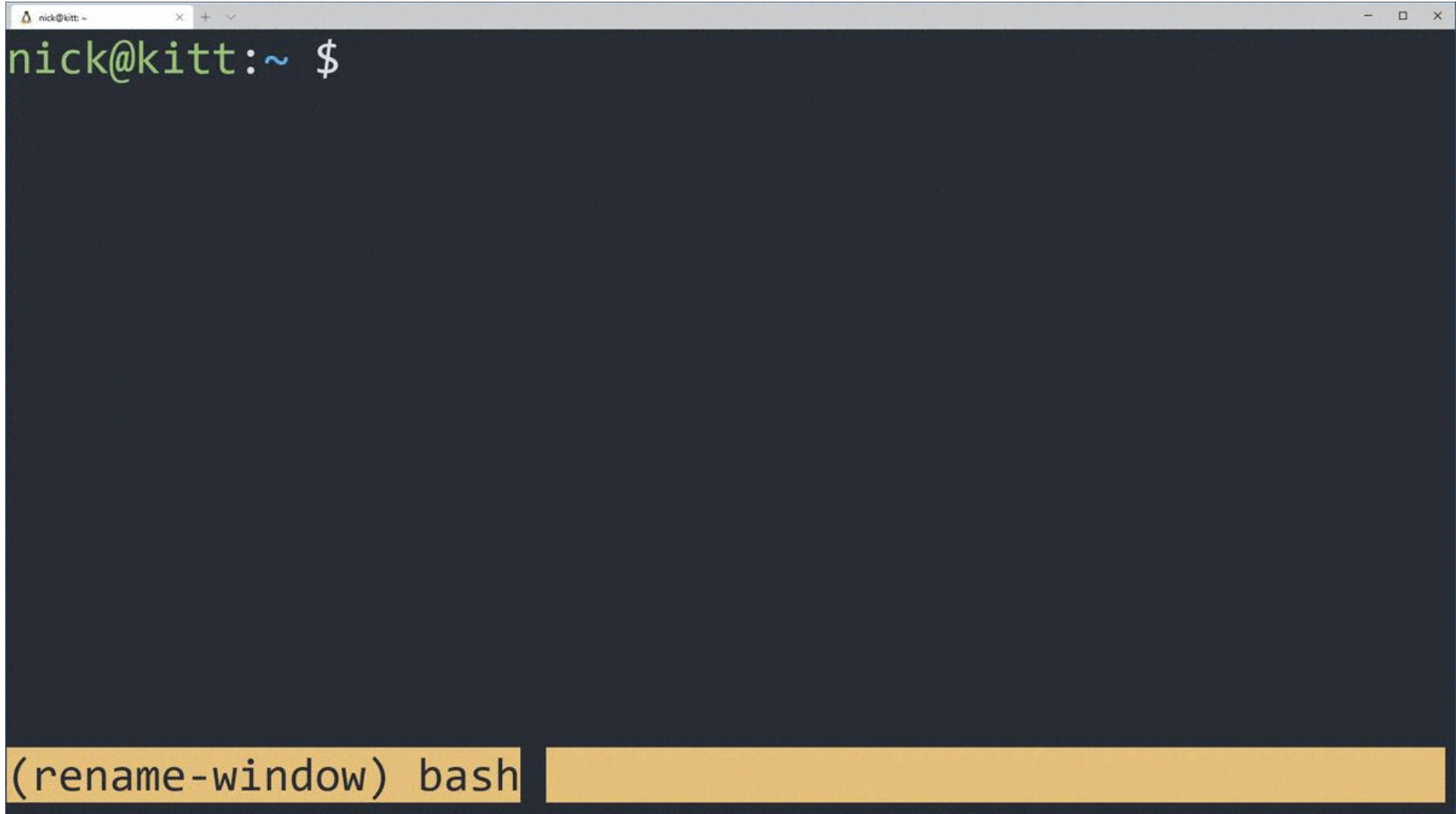


A terminal window with a dark background. The title bar shows 'nick@kitt: ~' and standard window controls. The prompt 'nick@kitt:~ \$' is in green and blue. A red cursor is at the end of the prompt. At the bottom, '1: bash-' and '2: bash*' are shown in blue.

```
nick@kitt:~ $
```

1: bash- 2: bash*

prefix + ,

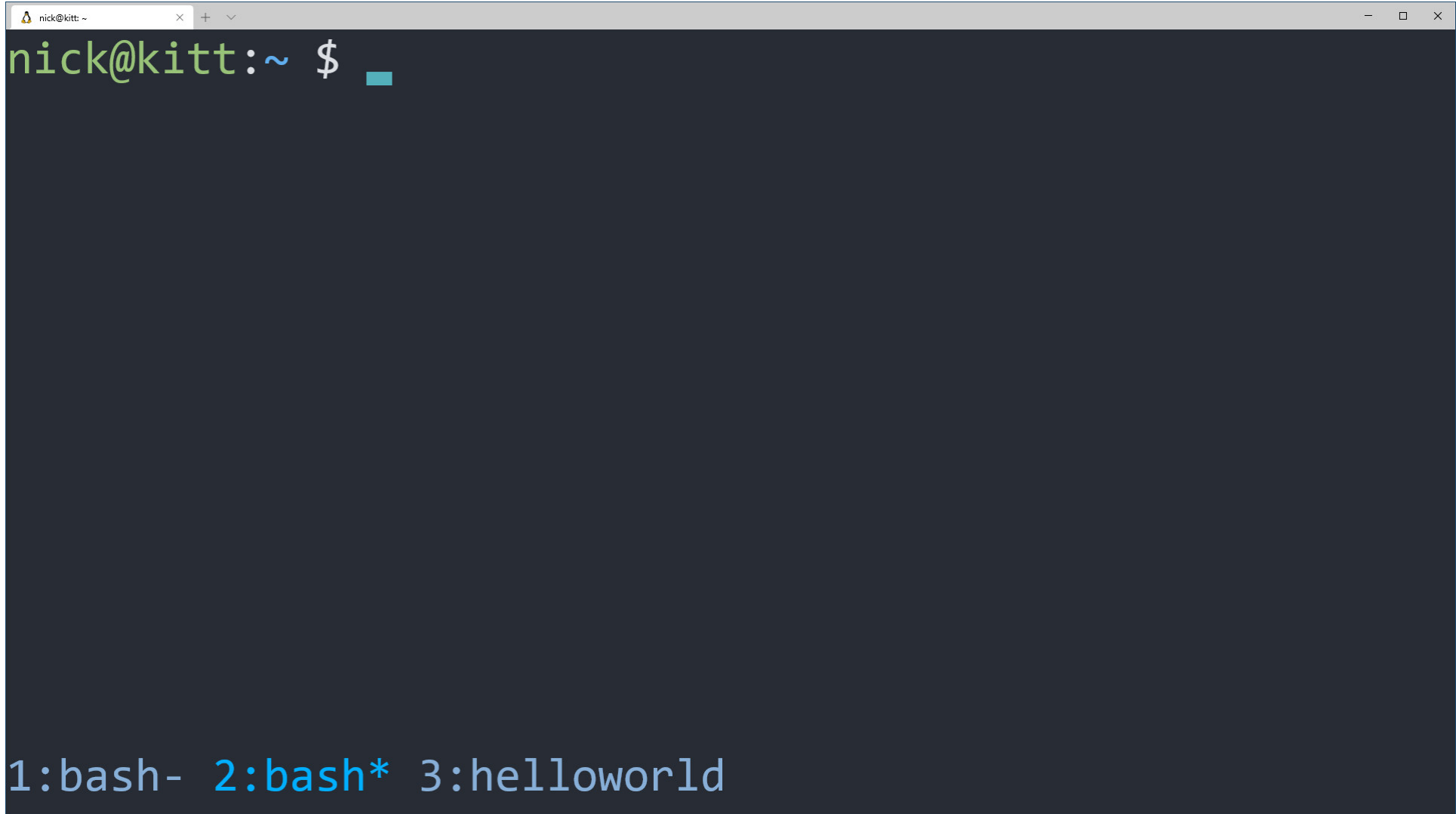


A terminal window with a dark blue background. The title bar at the top shows 'nick@kitt ~' and standard window controls. The prompt 'nick@kitt:~ \$' is displayed in green and blue. At the bottom, a yellow bar contains the text '(rename-window) bash'.

```
nick@kitt:~ $
```

(rename-window) bash

prefix + 2

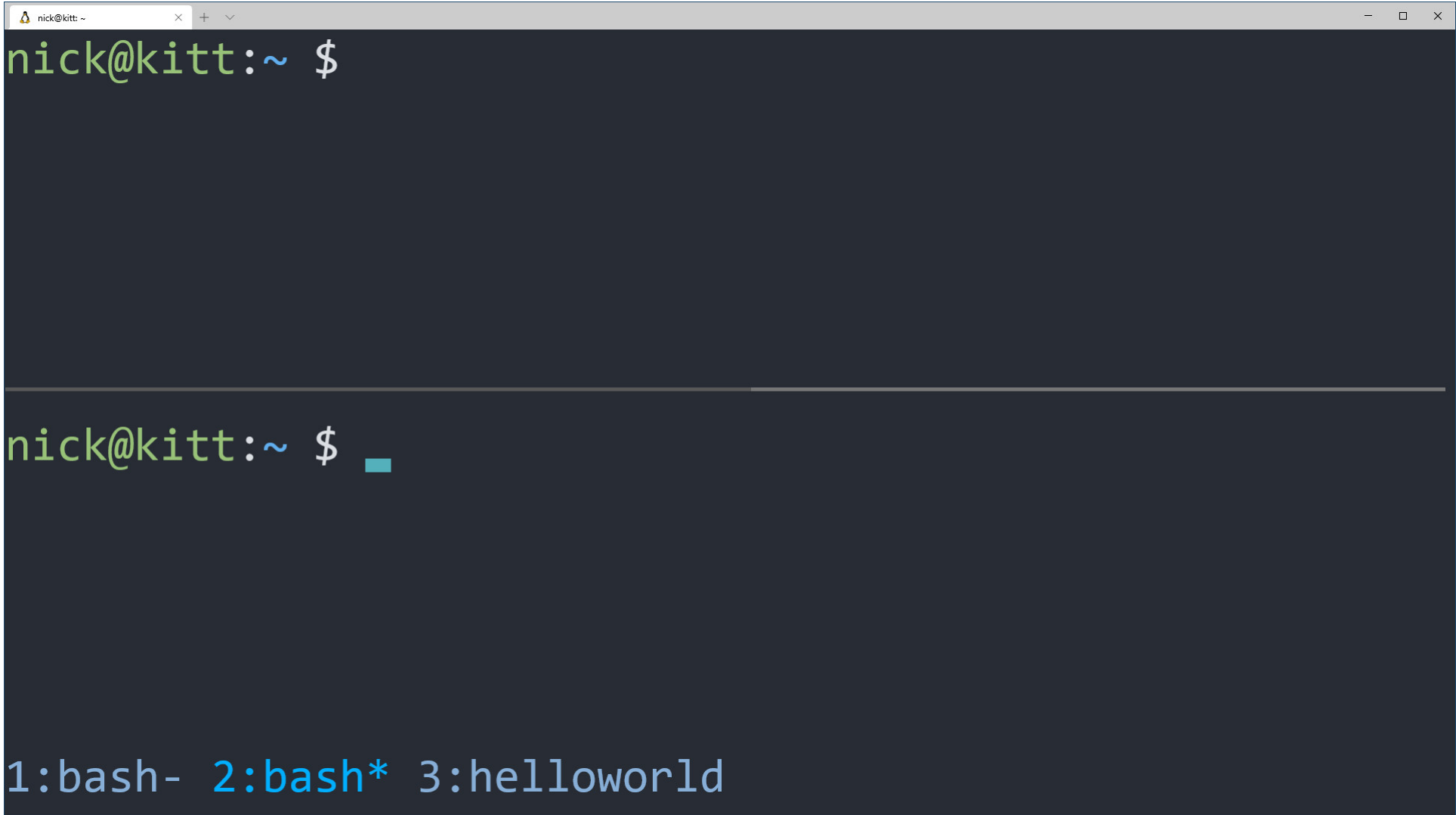


A terminal window with a dark background. The title bar shows 'nick@kitt: ~' and standard window controls. The prompt is 'nick@kitt:~ \$' with a blue cursor. At the bottom, environment variables are listed: '1: bash- 2: bash* 3: helloworld'.

```
nick@kitt:~ $  
  
1: bash- 2: bash* 3: helloworld
```

prefix + ?

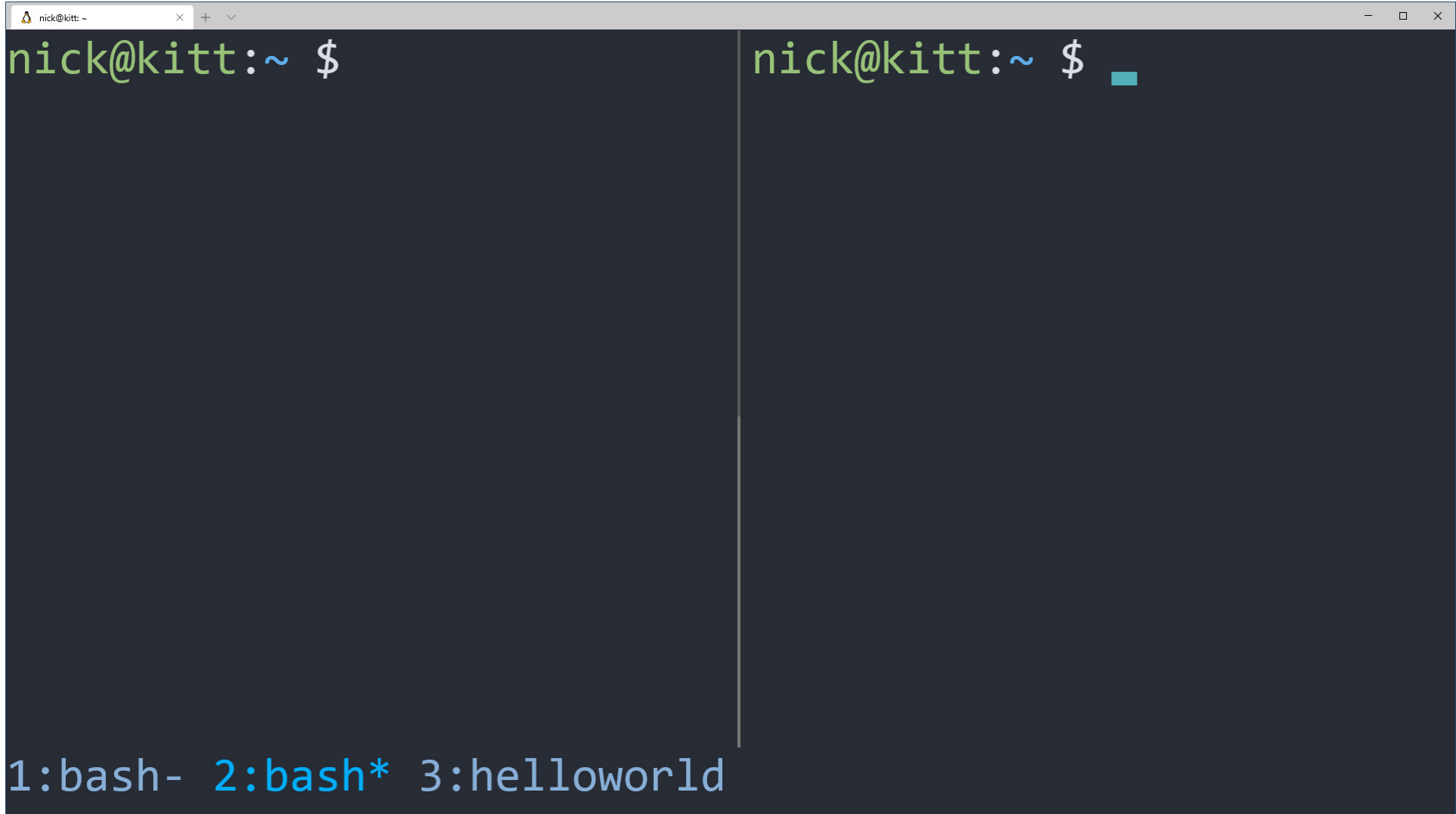
prefix + b (ALT + up / down to switch)



```
nick@kitt: ~  
nick@kitt:~ $  
  
nick@kitt:~ $  
  
1: bash- 2: bash* 3: helloworld
```

The image shows a terminal window with a dark background. The window title bar at the top indicates the user is 'nick' on a host named 'kitt', with a tilde symbol representing the home directory. The prompt 'nick@kitt:~ \$' is shown in green. Below the prompt, there is a blank line, followed by another prompt 'nick@kitt:~ \$' with a blue cursor. At the bottom of the window, a command history list is displayed: '1: bash- 2: bash* 3: helloworld'.

prefix + v (ALT + left / down to switch)



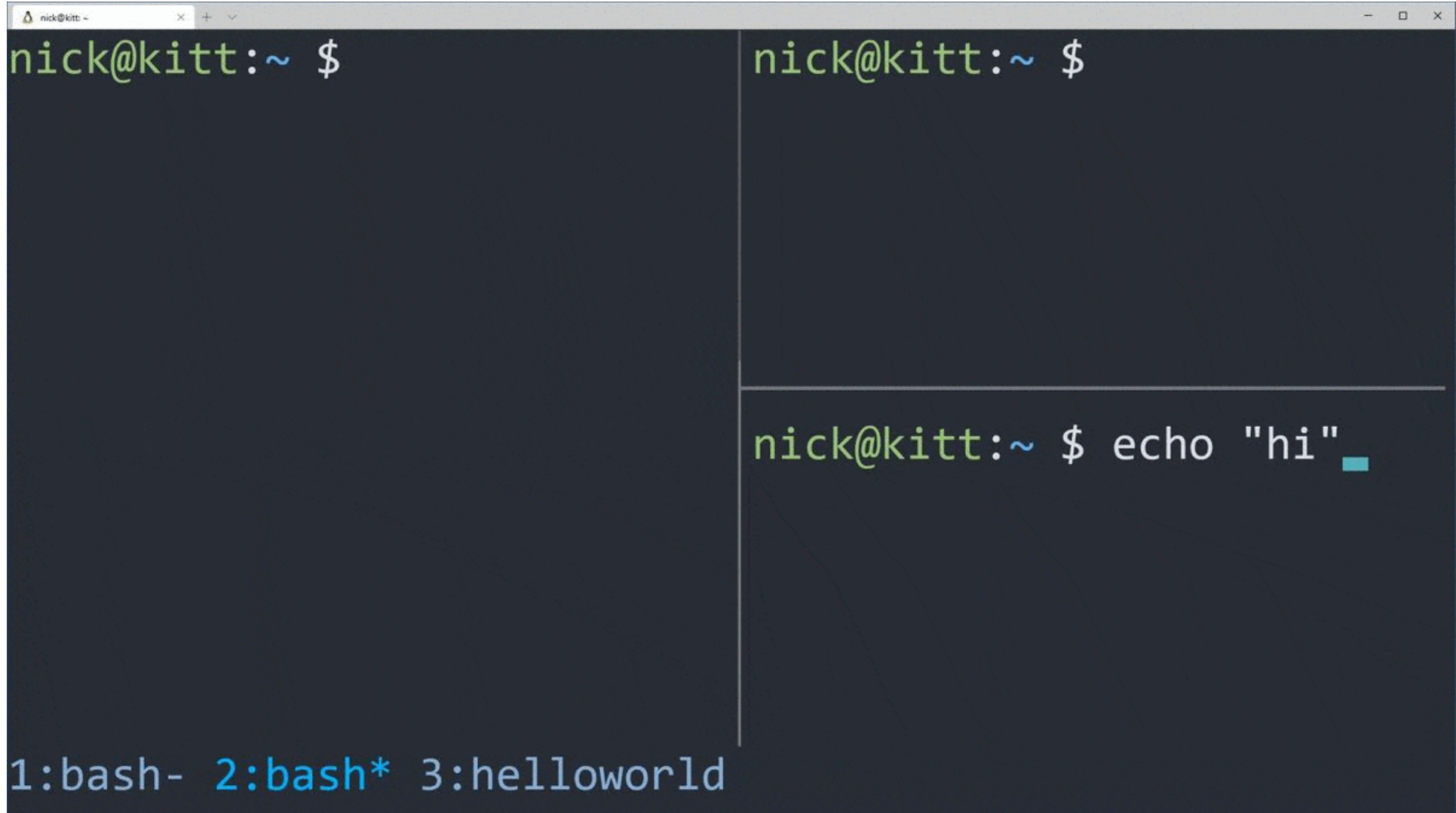
The image shows two terminal windows side-by-side. Both windows have a title bar with a flame icon, the text 'nick@kitt: ~', and window control buttons. The left window displays the prompt 'nick@kitt:~ \$' and a command history at the bottom: '1: bash- 2: bash* 3: helloworld'. The right window displays the prompt 'nick@kitt:~ \$' followed by a blue cursor block.

```
nick@kitt:~ $
```

```
1: bash- 2: bash* 3: helloworld
```

```
nick@kitt:~ $
```

prefix + z



The image shows a terminal window with a dark background. The window title is "nick@kitt: ~". The prompt is "nick@kitt:~ \$". The command "echo 'hi'" is being entered, followed by a cursor. The output of the command is "1: bash- 2: bash* 3: helloworld".

```
nick@kitt:~ $  
nick@kitt:~ $ echo "hi"  
1: bash- 2: bash* 3: helloworld
```

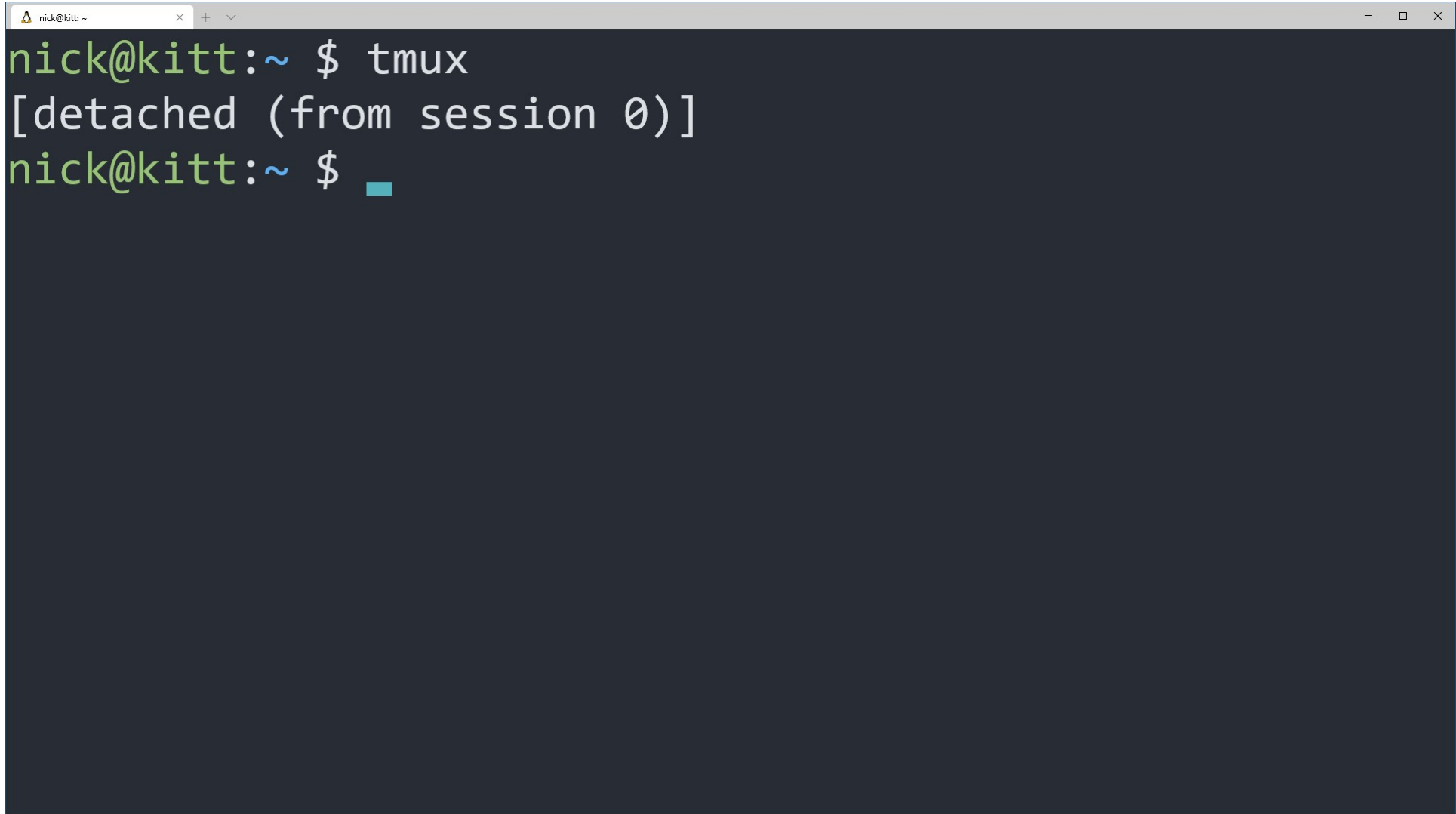
BUT WAIT



THERE'S MORE

memegenerator.net

prefix + d

A terminal window with a dark background and light-colored text. The window title bar shows 'nick@kitt: ~' and standard window controls. The terminal content shows a user typing 'tmux' at the prompt 'nick@kitt:~ \$', followed by a message '[detached (from session 0)]' and a new prompt 'nick@kitt:~ \$' with a blue cursor.

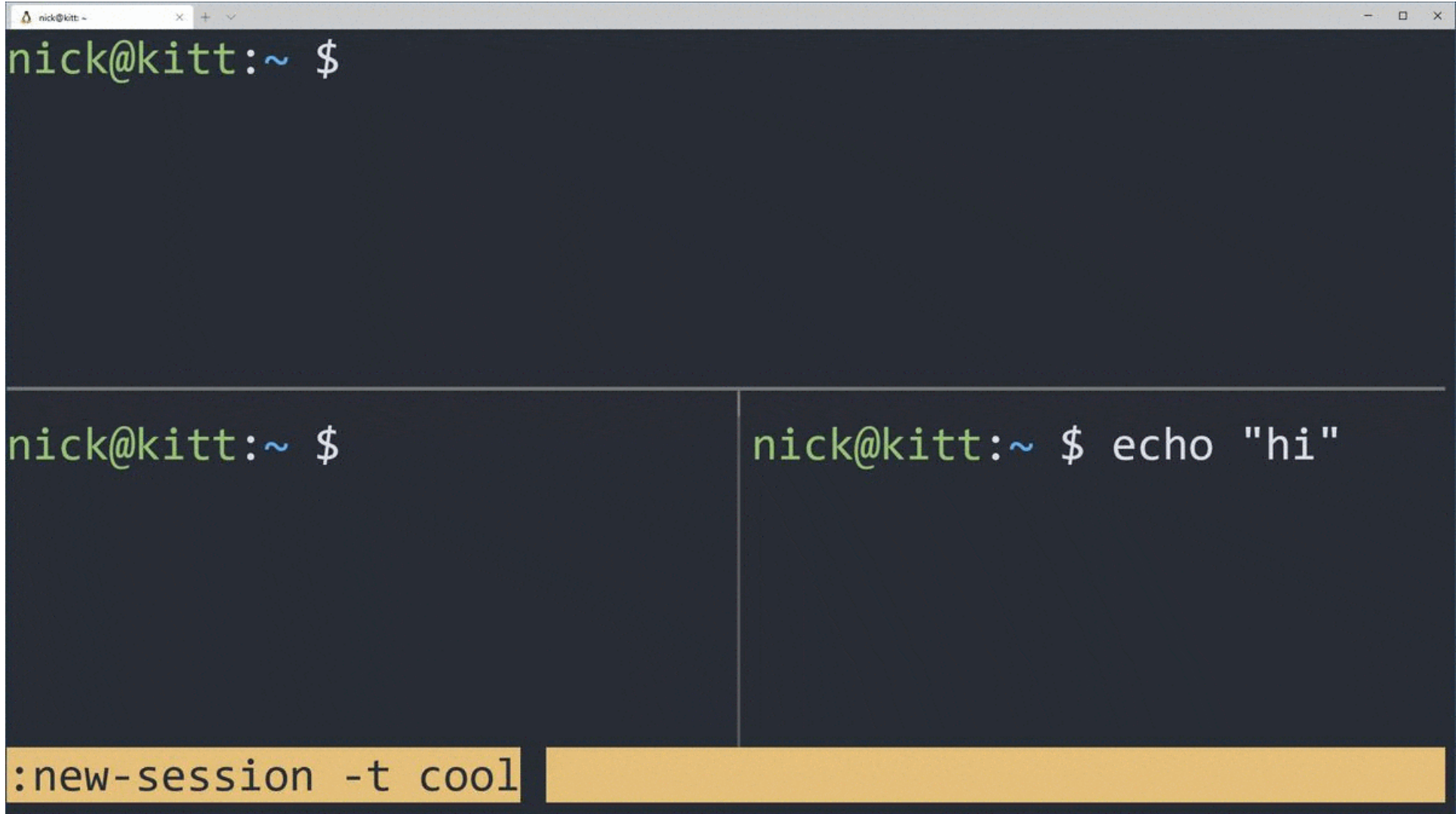
```
nick@kitt:~ $ tmux
[detached (from session 0)]
nick@kitt:~ $
```

```
nick@kitt: ~  
nick@kitt:~ $ tmux  
[detached (from session 0)]  
nick@kitt:~ $ tmux ls  
0: 3 windows (created Sat Sep 26 17:21:37 2020)  
nick@kitt:~ $
```

```
nick@kitt: ~  
0: 3 windows (created Sat Sep 26 17:21:37 2020)  
nick@kitt:~$ tmux ls  
nick@kitt:~$ tmux attach -t 0
```

```
nick@kitt: ~ $  
  
nick@kitt:~ $  
nick@kitt:~ $ echo "hi"  
1: bash 2: bash* 3: helloworld-
```

prefix + :new-session -t cool



A terminal window titled "nick@kitt ~" is shown. The window is split into two panes. The left pane shows the prompt "nick@kitt:~ \$" and the right pane shows the command "nick@kitt:~ \$ echo "hi"". At the bottom of the window, a yellow bar contains the text ":new-session -t cool".

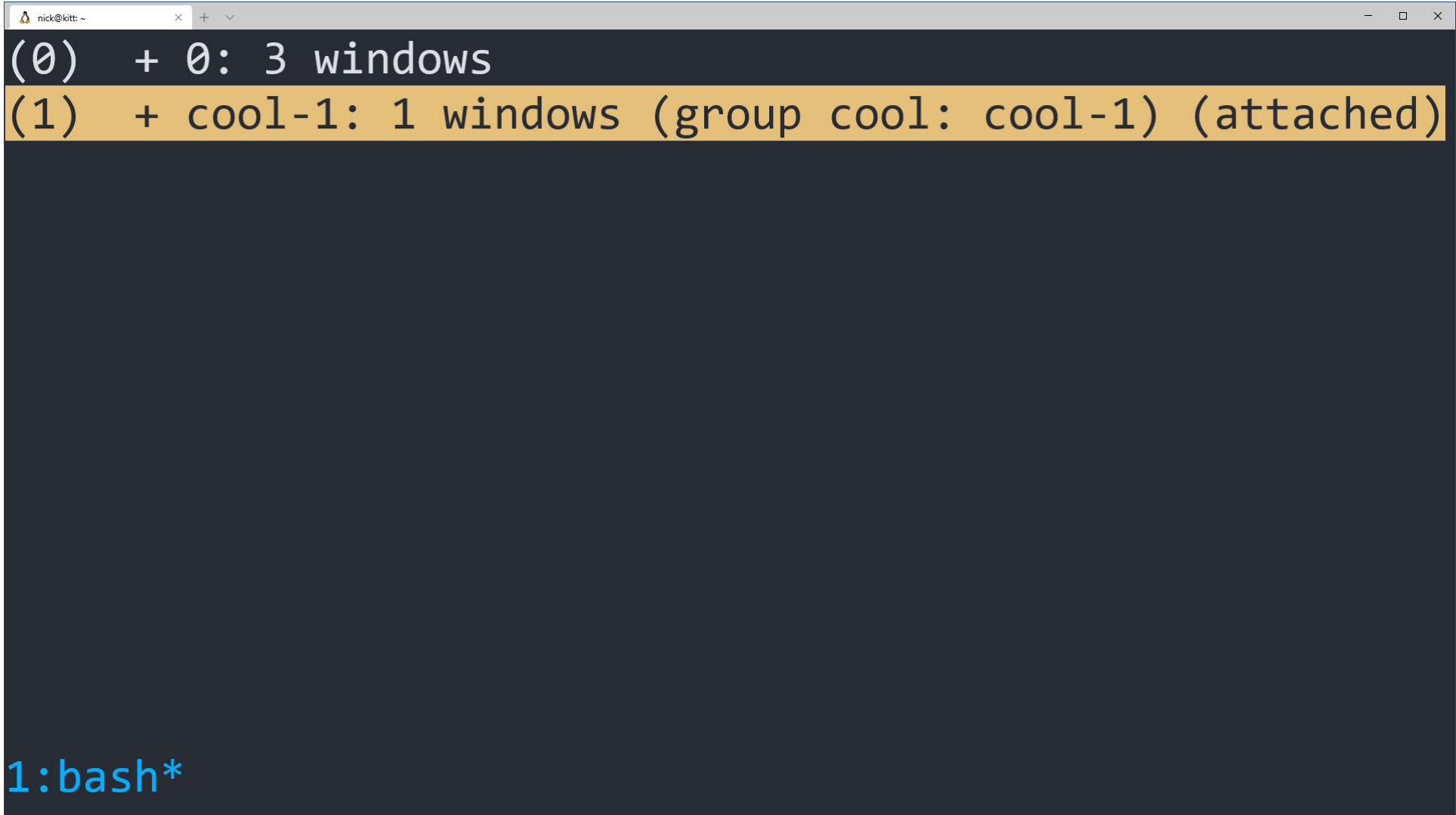
```
nick@kitt:~ $
```

```
nick@kitt:~ $ echo "hi"
```

```
:new-session -t cool
```

```
nick@kitt: ~  
$ tmux new-session -t hmm  
sessions should be nested with care, unset $TMUX to force  
nick@kitt:~ $  
  
1: bash*
```

prefix + s



```
nick@kitt: ~  
(0) + 0: 3 windows  
(1) + cool-1: 1 windows (group cool: cool-1) (attached)  
  
1: bash*
```

A terminal window with a dark background and a light gray title bar. The title bar contains the text "nick@kitt: ~" and standard window control buttons. The terminal displays two lines of text: "(0) + 0: 3 windows" and "(1) + cool-1: 1 windows (group cool: cool-1) (attached)". The second line is highlighted with a yellow background. At the bottom left, there is a blue prompt "1: bash*" followed by an asterisk.

(0) + 0: 3 windows

(1) + cool-1: 1 windows (group cool: cool-1) (attached)

0 (sort: index)

kitt:~ \$

kitt:~ \$

kitt:~ \$

1: bash

2: bash

3: helloworld

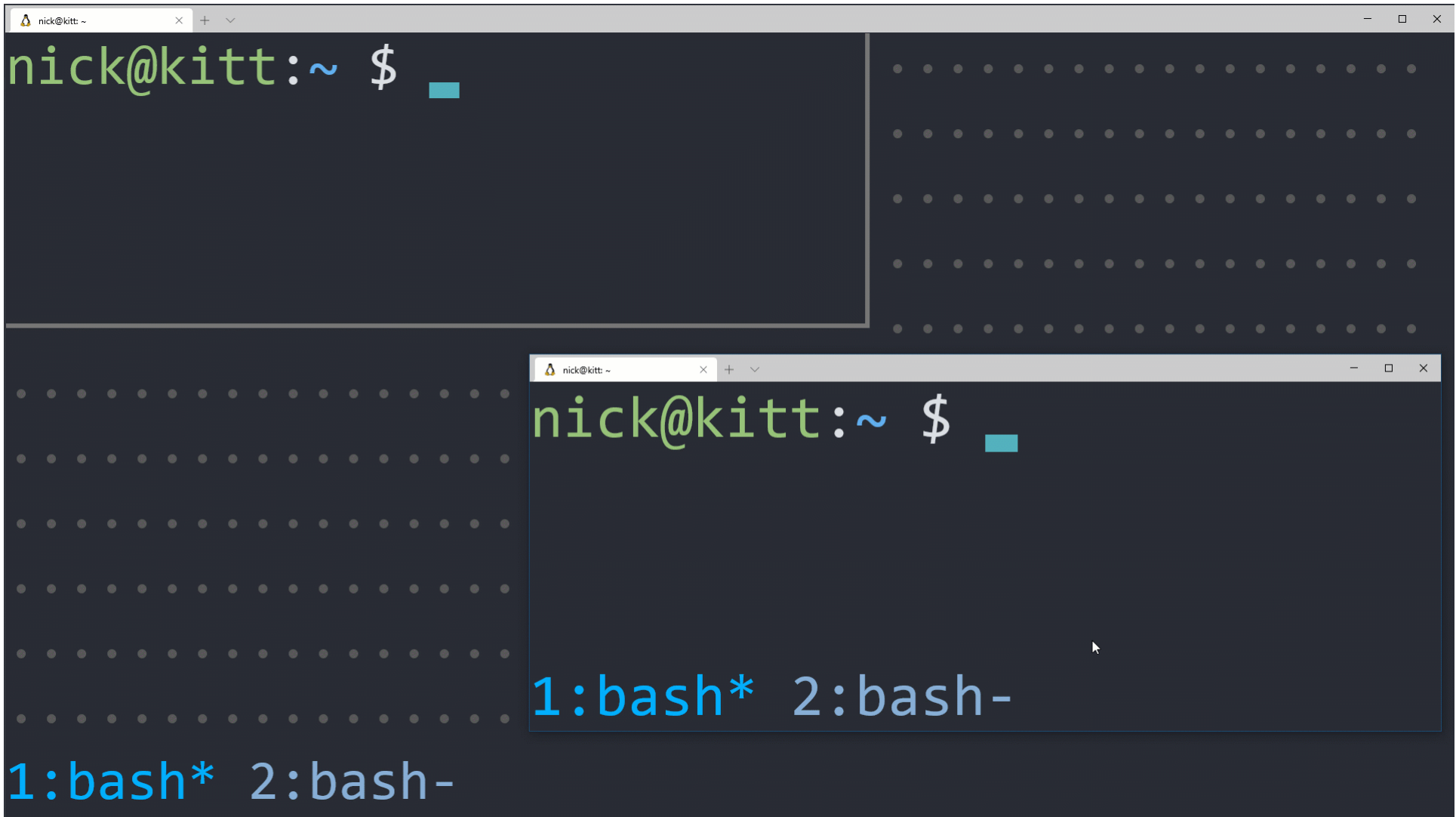
1: bash*

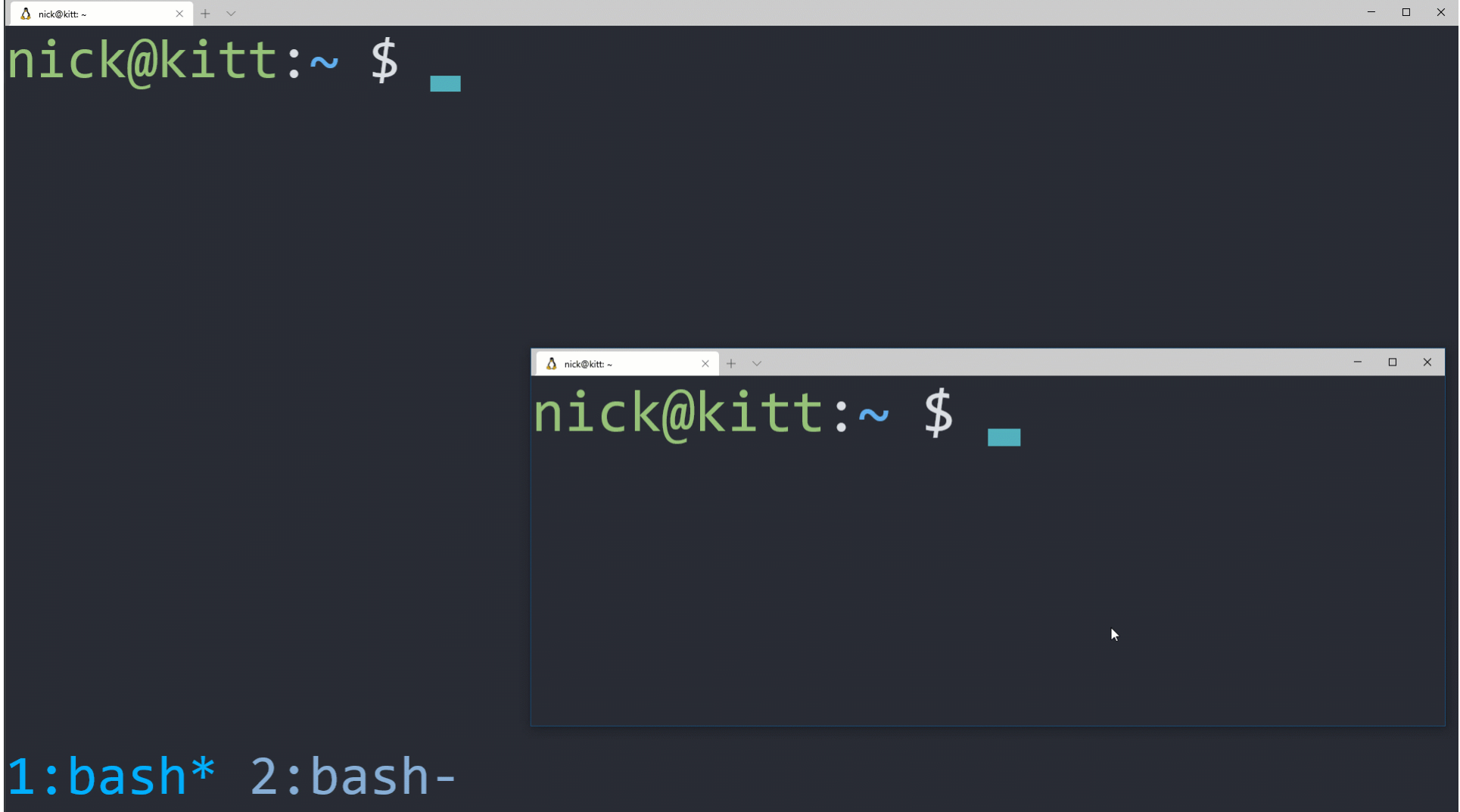
Mouse wheel up + / + tmux + n

```
nick@kitt: ~  
drwx----- 2 nick nick 4096 Aug 14 17:50 .ssh  
-rw-r--r-- 1 nick nick 0 May 27 19:41 .sudo_as_admin_successful  
drwxr-xr-x 2 nick nick 4096 Sep 5 08:50 .terraform.d  
drwxr-xr-x 3 nick nick 4096 Aug 16 11:53 .tmux  
lrwxrwxrwx 1 nick nick 41 Aug 17 11:38 .tmux.conf -> /home/nick/src/github/dotfiles/.tmux.conf  
-rw-r--r-- 1 nick nick 26 May 28 09:04 .tool-versions  
drwxr-xr-x 8 nick nick 4096 Aug 3 08:50 .vim  
-rw----- 1 nick nick 45952 Sep 26 17:17 .viminfo  
lrwxrwxrwx 1 nick nick 37 Aug 17 11:39 .vimrc -> /home/nick/src/github/dotfiles/.vimrc  
-rw-r--r-- 1 nick nick 178 Aug 21 11:03 .wget-hsts  
drwxr-xr-x 3 nick nick 4096 May 27 21:31 .yarn  
-rw-r--r-- 1 nick nick 116 Jun 5 06:16 .yarnrc  
drwxr-xr-x 2 nick nick 4096 Sep 3 13:40 notes  
drwxr-xr-x 12 nick nick 4096 Sep 7 08:51 src  
nick@kitt:~ $  
1: bash*
```

Select and copy text

```
nick@kitt: ~  
drwx----- 2 nick nick 4096 Aug 14 17:50 .ssh [0/106]  
-rw-r--r-- 1 nick nick 0 May 27 19:41 .sudo_as_admin_successful  
drwxr-xr-x 2 nick nick 4096 Sep 5 08:50 .terraform.d  
drwxr-xr-x 3 nick nick 4096 Aug 16 11:53 .tmux  
lrwxrwxrwx 1 nick nick 41 Aug 17 11:38 .tmux.conf -> /home/nick/src/github/dotfiles/.tmux.conf  
-rw-r--r-- 1 nick nick 26 May 28 09:04 .tool-versions  
drwxr-xr-x 8 nick nick 4096 Aug 3 08:50 .vim  
-rw----- 1 nick nick 45952 Sep 26 17:17 .viminfo  
lrwxrwxrwx 1 nick nick 37 Aug 17 11:39 .vimrc -> /home/nick/src/github/dotfiles/.vimrc  
-rw-r--r-- 1 nick nick 178 Aug 21 11:03 .wget-hsts  
drwxr-xr-x 3 nick nick 4096 May 27 21:31 .yarn  
-rw-r--r-- 1 nick nick 116 Jun 5 06:16 .yarnrc  
drwxr-xr-x 2 nick nick 4096 Sep 3 13:40 notes  
drwxr-xr-x 12 nick nick 4096 Sep 7 08:51 src  
nick@kitt:~ $  
1: bash*
```





```
nick@kitt: ~  
nick@kitt:~ $ tmux attach  
[detached (from session cool-1)]  
nick@kitt:~ $ tmux ls  
0: 3 windows (created Sat Sep 26 17:21:37 2020)  
cool-1: 1 windows (created Sat Sep 26 18:03:11 2020) (group cool)  
nick@kitt:~ $ tmux kill-server  
nick@kitt:~ $ tmux ls  
no server running on /tmp/tmux-1000/default  
nick@kitt:~ $
```

What happens when you reboot?

Let's take a look at configuring tmux

```
$ cat ~/.tmux.conf
```

```
# Set a new prefix / leader key.  
set -g prefix `  
bind ` send-prefix
```

```
$ cat ~/.tmux.conf
```

```
# ...
```

```
# Allow opening multiple sessions independently.
```

```
setw -g aggressive-resize on
```

```
$ cat ~/.tmux.conf
```

```
# ...
```

```
set -g status-left ''
```

```
set -g status-left-length 0
```

```
set -g status-right ''
```

```
set -g status-right-length 0
```

```
# Display a clock on the bottom right of the status bar.
```

```
#set -g status-right '%a %Y-%m-%d %H:%M'
```

```
#set -g status-right-length 20
```

```
$ cat ~/.tmux.conf
```

```
# ...
```

```
# Ensure window index numbers get reordered on delete.
```

```
set-option -g renumber-windows on
```

```
# Start windows and panes indexes at 1, not 0.
```

```
set -g base-index 1
```

```
setw -g pane-base-index 1
```

```
$ cat ~/.tmux.conf
```

```
# ...
```

```
# Reload the tmux config.
```

```
bind-key r source-file ~/.tmux.conf
```

```
$ cat ~/.tmux.conf
```

```
# ...
```

```
# List of plugins.
```

```
set -g @plugin 'tmux-plugins/tpm'
```

```
set -g @plugin 'tmux-plugins/tmux-yank'
```

```
set -g @plugin 'tmux-plugins/tmux-resurrect'
```



Plug is one of Vim's plugin managers

Fuzzy searching files, text and more

```
Plug '~/fzf'  
Plug 'junegunn/fzf.vim'
```

Project tree visualization and manipulating files

```
Plug 'lambdalisue/fern.vim'
```

```
Plug 'lambdalisue/fern-mapping-mark-children.vim'
```

Showing git changes in the gutter

```
Plug 'mhinz/vim-signify'
```

Real-time browser based Markdown preview

```
Plug 'iamcco/markdown-preview.nvim', { 'do': 'cd app & yarn insta
```

Syntax highlighting and tabular views for CSV files

```
Plug 'chrisbra/csv.vim'
```

Snippets and auto-complete

```
Plug 'SirVer/ultisnips' | Plug 'honza/vim-snippets'  
Plug 'vim-scripts/AutoComplPop'
```

Zooming in and out of splits

```
Plug 'dhruvasagar/vim-zoom'
```

Highlighting, searching and replacing

```
" Highlight characters to jump to when using horizontal movement
Plug 'unblevable/quick-scope'

" Modify * to also work with visual selections.
Plug 'nelstrom/vim-visual-star-search'

" Automatically clear search highlights after you move your cursor
Plug 'haya14busa/is.vim'

" Handle multi-file find and replace.
Plug 'mhinz/vim-grepper'
```

High quality themes with dark and light modes

```
" Atom One Dark / Light theme.  
Plug 'rkr/vim-one'  
  
" Gruvbox theme.  
Plug 'gruvbox-community/gruvbox'
```

**SO I SEE YOU HAVE AROUND 200
PLUGINS**

**HOW MANY HAVE YOU ACTUALLY
USED?**

makeameme.org

Dotfiles

3 places where they might exist

1. .sometool file in \$HOME

```
$ ls -la ~
```

```
-rw-r--r--  1 nick nick  2347 Sep 26 08:11 .bashrc  
-rw-r--r--  1 nick nick  3396 Sep 26 08:11 .tmux.conf
```

3 places where they might exist

1. `..sometool` file in `$HOME`
2. `$HOME/.sometool/` directory

```
$ ls -la ~
```

```
drwxr-xr-x 11 nick nick 4096 May 27 21:32 .fzf/  
drwxr-xr-x  3 nick nick 4096 Aug  8 13:50 .vim/
```

3 places where they might exist

1. `.sometool` file in `$HOME`
2. `$HOME/.sometool/` directory
3. `$HOME/.config/sometool/` directory

```
$ ls -la ~/.config
```

```
drwx----- 2 nick nick 4096 May 27 21:38 htop/  
drwxr-xr-x  2 nick nick 4096 Jul  9 10:36 latest-releases/
```

```
nick@kitt:~ $ ls -la ~
total 260
drwxr-xr-x 26 nick nick 4096 Sep 26 09:36 .
drwxr-xr-x  3 root root 4096 May 27 19:27 ..
lrwxrwxrwx  1 nick nick   39 May 28 08:15 .aliases -> /home/nick/src/github/dotfiles/.aliases
lrwxrwxrwx  1 nick nick   38 Aug 17 11:39 .bashrc -> /home/nick/src/github/dotfiles/.bashrc
```

```
nick@kitt: ~/src/github/dotfiles (master)$ ls -la
total 108
drwxrwxrwx  7 nick nick 4096 Sep 26 08:11 .
drwxr-xr-x 63 nick nick 4096 Sep  9 13:43 ..
-rw-r--r--  1 nick nick 3375 Mar 17 2019 .Xresources
-rw-r--r--  1 nick nick 2362 Sep 23 10:17 .aliases
-rw-r--r--  1 nick nick 2347 Sep 26 08:11 .bashrc
-rw-r--r--  1 nick nick  19 Dec 18 2018 .gemrc
drwxr-xr-x  8 nick nick 4096 Sep 26 09:24 .git
-rw-r--r--  1 nick nick  100 May 29 08:58 .gitconfig
-rw-r--r--  1 nick nick  62 May 29 09:04 .gitconfig.user
-rw-r--r--  1 nick nick  214 Aug 17 11:12 .gitignore
drwxr-xr-x  3 nick nick 4096 Jul 30 14:40 .local
-rw-r--r--  1 nick nick  844 Sep 25 16:35 .profile
-rw-r--r--  1 nick nick 3396 Sep 26 08:11 .tmux.conf
```



 [TheLocehiliosan / yadm](#)

Yet Another Dotfiles Manager

 yadm.io/

 GPL-3.0 License

 1.9k stars  107 forks

Documentation

- [View screenshots](#) of the current set up along with previous set ups
- [Quickly get set up with these dotfiles](#)
 - [Debian](#) (native or WSL)
 - [Ubuntu LTS](#) (native or WSL)
 - [MacOS](#)
 - [Extra WSL 1 and WSL 2 steps](#)
- [FAQ](#)
 - [How to personalize these dotfiles?](#)
 - [How to fix Vim taking a long time to open when inside of WSL?](#)
- [About the author](#)

```
sudo apt-get update && sudo apt-get install -y \  
    curl \  
    git \  
    gpg \  
    htop \  
    jq \  
    pass \  
    pwgen \  
    python3-pip \  
    ripgrep \  
    rsync \  
    shellcheck \  
    tmux \  
    unzip \  
    vim-gtk
```

```
git clone https://github.com/nickjj/dotfiles ~/dotfiles

# The last symlink is only necessary if you're using WSL.
mkdir -p ~/.local/bin && mkdir -p ~/.vim/spell \
  && ln -s ~/dotfiles/.aliases ~/.aliases \
  && ln -s ~/dotfiles/.bashrc ~/.bashrc \
  && ln -s ~/dotfiles/.gemrc ~/.gemrc \
  && ln -s ~/dotfiles/.gitconfig ~/.gitconfig \
  && ln -s ~/dotfiles/.profile ~/.profile \
  && ln -s ~/dotfiles/.tmux.conf ~/.tmux.conf \
  && ln -s ~/dotfiles/.vimrc ~/.vimrc \
  && ln -s ~/dotfiles/.vim/spell/en.utf-8.add ~/.vim/spell/en.utf-8.add \
  && ln -s ~/dotfiles/.local/bin/toggle-dark-mode ~/.local/bin/toggle-dark-m
  && sudo ln -s ~/dotfiles/etc/wsl.conf /etc/wsl.conf
```

```
# Install Plug (Vim plugin manager).
curl -fLo ~/.vim/autoload/plug.vim --create-dirs \
    https://raw.githubusercontent.com/junegunn/vim-plug/master/plug.vim

# Install TPM (Tmux plugin manager).
git clone https://github.com/tmux-plugins/tpm ~/.tmux/plugins/tpm

# Install FZF (fuzzy finder on the terminal and used by a Vim plugin).
git clone --depth 1 https://github.com/junegunn/fzf.git ~/.fzf && ~/.fzf/install

# Install ASDF (version manager which I use for non-Dockerized apps).
git clone https://github.com/asdf-vm/asdf.git ~/.asdf --branch v0.7.8
```

```
# Open Vim and install the configured plugins. You would type in  
# :PlugInstall command from within Vim and then hit Enter.  
vim .  
:PlugInstall  
  
# Start a tmux session and install the configured plugins.  
# You would type in ` followed by I (capital i) to issue the command  
tmux  
`I
```

```
# Sanity check to see if you can run some of the tools we install
ruby --version
node --version
ansible --version
aws --version
terraform --version

# Check to make sure git is configured with your custom settings.
git config --list

# If you're using Docker Desktop with WSL, these should be access
docker info
docker-compose --version
```

dotfiles github



All



News



Videos



Images

About 643,000 results (0.40 seconds)



<i>Knowns</i>	<i>Known Knowns</i> <i>Things we are aware of and understand.</i>	<i>Known Unknowns</i> <i>Things we are aware of but don't understand.</i>
	<i>Unknown Knowns</i> <i>Things we understand but are not aware of.</i>	<i>Unknown Unknowns</i> <i>Things we are neither aware of nor understand.</i>
	<i>Knowns</i>	<i>Unknowns</i>

 <https://nickjanetakis.com>

 <https://twitter.com/nickjanetakis>

 <https://runninginproduction.com>

 Freelance Developer | Podcast Host | Course Creator

 Python | Ruby | Bash | Docker | Ansible | Terraform