

Z shell

\$ oh my zsh



 **fzf**

command-line fuzzy finder



NERD FONTS

Putting together the perfect terminal experience

The relatively short version

 **tmux**

Who am I?



- Aspiring Speakers Group member #1
- I have a cute accent
- I spend a lot of time using the terminal
- Principal Developer at Comply Advantage
- Disclaimer: Nothing that I say or do has anything to do with my employer:

Comply Advantage

But we are hiring
<https://complyadvantage.com/careers/>

Motivation

- I'm a IntelliJ IDEA kind of person
- Keyboard shortcuts are ♥
- Thanks to the whole DevOps movement I'm expending more and more time using the terminal
- Old school SysAds surely can do a lot in the terminal
- Can I become more productive with the terminal?



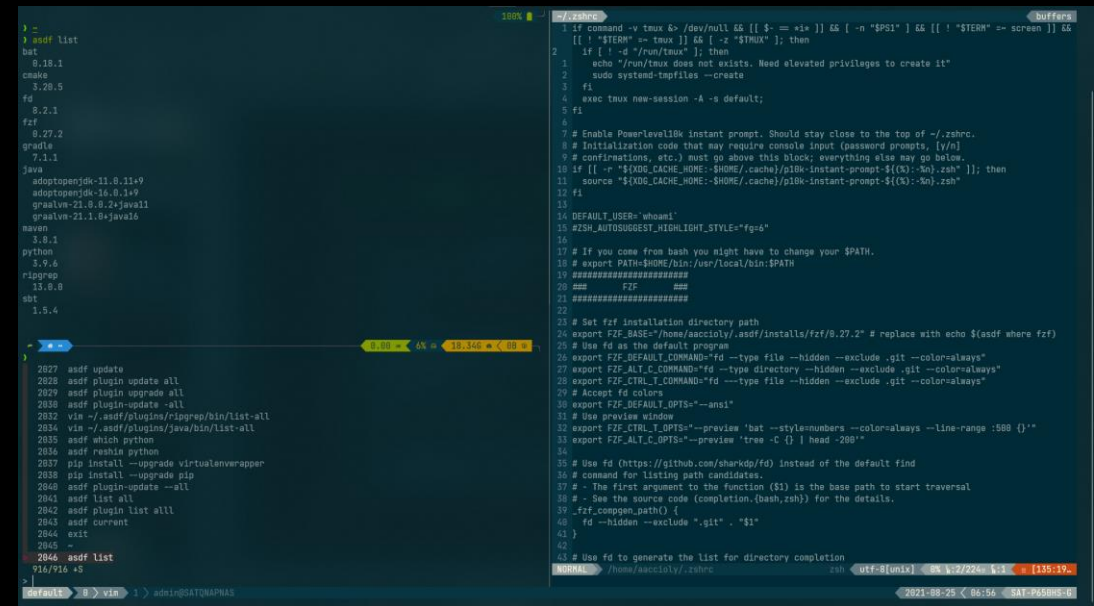
<https://www.youtube.com/watch?v=41CC-F6KRP8>

So, what's the “perfect” terminal experience?

It's up to you. My wishlist:

- Terminal doesn't get in my way
- Fast
- Low energy consumption
- Syntax highlighting and auto-completion
- I can quickly find files, commands, etc
- Term can handle context switching
- It remembers what I was doing 3 days ago so that I don't have to do it

And it looks like this:



The screenshot shows a terminal window with two panes. The left pane displays the output of the `asdf list` command, showing a list of installed plugins and their versions. The right pane displays the `~/.asdfrc` file, which contains configuration for the asdf tool manager.

```
1 2
3 asdf list
4 bat
5 8.18.1
6 cmake
7 3.28.5
8 fd
9 8.2.1
10 fzf
11 8.27.2
12 gradle
13 7.1.1
14 java
15 adoptopenjdk-11.0.11+9
16 adoptopenjdk-16.0.1+9
17 graalvm-21.0.0-rc1-jdk11
18 graalvm-21.0.0-rc1-jdk16
19 mvn
20 3.8.1
21 python
22 3.9.6
23 ripgrep
24 13.0.0
25 sbt
26 1.5.4
27
28 2827 asdf update
29 2828 asdf plugin update all
30 2829 asdf plugin upgrade all
31 2830 asdf plugin update --all
32 2831 vim ~/.asdf/plugins/ripgrep/bin/list-all
33 2832 vim ~/.asdf/plugins/ripgrep/bin/list-all
34 2833 vim ~/.asdf/plugins/java/bin/list-all
35 2834 asdf which python
36 2835 asdf which python
37 2836 pip install --upgrade virtualenvwrapper
38 2837 pip install --upgrade pip
39 2838 asdf plugin update --all
40 2839 asdf list all
41 2840 asdf plugin list all
42 2841 asdf current
43 2842 exit
44 2843
45 2844 asdf list
46 916/916 as
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
```

```
1 if command -v tmux && [ $(tmux has -p -t $1) ]; then
2   [[ ! "$TERM" =~ tmux ]] && [ -z "$TMUX" ]; then
3     if [ ! -d ~/.run/tmux ]; then
4       echo "/run/tmux does not exist. Need elevated privileges to create it"
5       sudo systemd-tmpfiles --create
6       fi
7     exec tmux new-session -A -s default;
8   fi
9
10 # Enable Powerlevel10k instant prompt. Should stay close to the top of ~/.zshrc.
11 # Initialization code that may require console input (password prompts, y/n)
12 # confirmations, etc.) must go above this block; everything else may go below.
13 if [[ -n "${XDG_CACHE_HOME:-$HOME/.cache}" ]]; then
14   source "${XDG_CACHE_HOME:-$HOME/.cache}/p10k-instant-prompt-${(KX)-$sh}.zsh"
15 fi
16
17 # If you come from bash you might have to change your $PATH.
18 # export PATH=$HOME/bin:/usr/local/bin:$PATH
19 # =====
20 # FZF
21 # =====
22
23 # Set fzf installation directory path
24 export FZF_BASE="/home/ascciofy/asdf/install/fzf/0.27.2" # replace with echo $(asdf where fzf)
25 # Use fd as the default program
26 export FZF_DEFAULT_COMMAND="fd --type file --hidden --exclude .git --color=always"
27 export FZF_ALT_C_COMMAND="fd --type directory --hidden --exclude .git --color=always"
28 export FZF_CTRL_T_COMMAND="fd --type file --hidden --exclude .git --color=always"
29 # Accept fd colors
30 export FZF_DEFAULT_OPTS="--ansi"
31 # Use preview window
32 export FZF_CTRL_T_OPTS="--preview 'bat --style=numbers --color=always --line-range :500 {}'"
33 export FZF_ALT_C_OPTS="--preview 'tree -C {} | head -200'"
34
35 # Use fd (https://github.com/sharkdp/fd) instead of the default find
36 # command for listing path candidates.
37 # - The first argument to the function ($1) is the base path to start traversal
38 # - See the source code (completion.bash.zsh) for the details.
39 _fzf_compgen_path() {
40   fd --hidden --exclude ".git" . "$1"
41 }
42
43 # Use fd to generate the list for directory completion
44 _fzf_compgen_dir() {
45   fd --hidden --exclude ".git" . "$1"
46 }
47
48 # =====
49 # FZF
50 # =====
51
52 # Set fzf installation directory path
53 export FZF_BASE="/home/ascciofy/asdf/install/fzf/0.27.2" # replace with echo $(asdf where fzf)
54 # Use fd as the default program
55 export FZF_DEFAULT_COMMAND="fd --type file --hidden --exclude .git --color=always"
56 export FZF_ALT_C_COMMAND="fd --type directory --hidden --exclude .git --color=always"
57 export FZF_CTRL_T_COMMAND="fd --type file --hidden --exclude .git --color=always"
58 # Accept fd colors
59 export FZF_DEFAULT_OPTS="--ansi"
60 # Use preview window
61 export FZF_CTRL_T_OPTS="--preview 'bat --style=numbers --color=always --line-range :500 {}'"
62 export FZF_ALT_C_OPTS="--preview 'tree -C {} | head -200'"
63
64 # Use fd (https://github.com/sharkdp/fd) instead of the default find
65 # command for listing path candidates.
66 # - The first argument to the function ($1) is the base path to start traversal
67 # - See the source code (completion.bash.zsh) for the details.
68 _fzf_compgen_path() {
69   fd --hidden --exclude ".git" . "$1"
70 }
71
72 # Use fd to generate the list for directory completion
73 _fzf_compgen_dir() {
74   fd --hidden --exclude ".git" . "$1"
75 }
76
77 # =====
78 # FZF
79 # =====
80
81 # Set fzf installation directory path
82 export FZF_BASE="/home/ascciofy/asdf/install/fzf/0.27.2" # replace with echo $(asdf where fzf)
83 # Use fd as the default program
84 export FZF_DEFAULT_COMMAND="fd --type file --hidden --exclude .git --color=always"
85 export FZF_ALT_C_COMMAND="fd --type directory --hidden --exclude .git --color=always"
86 export FZF_CTRL_T_COMMAND="fd --type file --hidden --exclude .git --color=always"
87 # Accept fd colors
88 export FZF_DEFAULT_OPTS="--ansi"
89 # Use preview window
90 export FZF_CTRL_T_OPTS="--preview 'bat --style=numbers --color=always --line-range :500 {}'"
91 export FZF_ALT_C_OPTS="--preview 'tree -C {} | head -200'"
92
93 # Use fd (https://github.com/sharkdp/fd) instead of the default find
94 # command for listing path candidates.
95 # - The first argument to the function ($1) is the base path to start traversal
96 # - See the source code (completion.bash.zsh) for the details.
97 _fzf_compgen_path() {
98   fd --hidden --exclude ".git" . "$1"
99 }
100
101 # Use fd to generate the list for directory completion
102 _fzf_compgen_dir() {
103   fd --hidden --exclude ".git" . "$1"
104 }
```

Let's get something out of the way

- I will be using Ubuntu Today
- This is not a talk for Linux Gurus
- The OS itself isn't as important as you may think
- I'll be using a terminal emulator called kitty:
<https://sw.kovidgoyal.net/kitty/>
- Your choice of terminal emulator is important
- Pick one that:
 - Is GPU Accelerated
 - Has a Quake / drop-down mode
 - Has good font support (including ligatures)



+



+



+



Eye Candy:

Fonts

- You want a Monospaced font
- At least Powerline Glyphs
- Optional: Font ligatures
- Pick one from <https://www.nerdfonts.com/>:
 - JetBrains Mono
 - Cascadia Code
 - Fira Code
 - Meslo
 - Source Code Pro / Hasklig
 - Ubuntu Mono

Colours

- Pick a colour scheme that doesn't make your eyes bleed
- I don't have a lot of imagination, so I keep coming back to my own patched version of [Solarized](#) Dark
- A few other suggestions:
 - Base16 schemes: <https://github.com/chriskempson/base16>
 - Dracula
 - Gruvbox
 - Monokai
 - PaperColor



Shell



- [Bash](#) is great, [Zsh](#) is even better
- Install a recent version and make it default:

```
chsh -s $(which zsh)
```

- Notable features:
 - Smart tab completion
 - History filter with up arrow
 - case / typo correction
 - ** expansion
 - Glob Qualifiers: `ls *(m-7)`

Framework / Plugin Managers

- Extract even more from zsh with a collection of plugins and themes.
[Use Oh My Zsh](#)
- Install Oh My Zsh and start enjoying extra functionality out of the box:
 - Easier navigation:
 - .., ...,
 - No need to type cd
 - Partial path matching
 - New commands / alias by default such as:
 - d / cd -N
 - rd
 - take
- Install custom plugin sand edit your .zshrc file to enable / disable plugins
 - [zsh-autosuggestions](#)
 - [zsh-syntax-highlighting](#)
 - [z](#)
 - [web-search](#)
- If you are using too many plugins and your shell is taking a while to start, have a look at a plugin manager able to load plugins asynchronously (such as [Zinit](#))

A powerful / lightweight / flexible theme

- [Powerlevel10K](#) is my favourite theme
- Notable features:
 - Make it look the way you want
 - Fast (and can also make zsh plugins load faster)
 - Transient Prompt
 - Enable and Disable “Segments”



Terminal Multiplexer

- [Tmux](#) for the win!



- Create, use and manage multiple sessions, windows and panes
- Nest Tmux inside Tmux
- Tmux + SSH = Don't lose your work
- Customize and install plugins ([TPM](#) strongly recommended)



Fuzzy Find all the things



- Find what you need with [fzf](#)
- A standalone command that integrates with what you have
- ** expansion
- Integrates by default with commands such as kubectlx, kubens, etc
- Shortcuts:
 - CTRL+T to search files
 - CTRL+R to search history
 - ALT+C cd into directory

Honourable mentions:

- [asdf](#): nvm + pyenv+ rbenv + SDKMAN + Install and manage versions for all your cli tools. Go watch my talk ⇒
- [bat](#): “better” cat with syntax highlighting, Git Integration, automatic paging, etc
- [fd](#): “better”, faster, saner alternative to find written in Rust
- [ripgrep](#): “better”, faster”, saner alternative to grep written in Rust



<https://www.youtube.com/watch?v=6NNbqDTSFD0&t=646s>

Oh, by the Way

- I avoid text editors as much as I can but...
- When I need an editor I use [neovim](#):



- ... and [vim-plug](#)

- Getting all of this to work can take some time
- It's a hobby like any other
- Make sure to keep your files in version control
- Take some time to ~~steal ideas from~~ get inspired by other people's dotfile repositories.





That's all Folks!

Questions?



<https://www.linkedin.com/in/aaccioly>



<https://stackoverflow.com/users/664577>



<https://a.acciolty.7rtc.com>



[a.acciolty at protonmail.com](mailto:a.acciolty@protonmail.com)



Putting together the perfect terminal experience

© 2022 by [Anthony Acciolty](#) is licensed under

[CC BY-SA 4.0](#)