

## • TERMINAL PRODUCTIVITY TOOLKIT

# 5 Shell Tools That Remember So You Don't

Stop retyping the same commands over and over. These autocomplete tools save developers hours every week — and take under 5 minutes to set up.

⚡ Ghost-text autocomplete

🔍 Fuzzy history search

📄 navi & pet snippet tools

🧠 Smart command recall

🍺 Install with Homebrew

```
git commit -m "fix auth bug" --no-verify
```

# Why Your Terminal Should Finish Your Sentences

Every developer has typed `git checkout feature/user-auth-redesign-v2` from scratch — then watched their hands slow down as they try to remember the exact branch name. Or re-run the same long Docker command for the tenth time this week.

The tools in this guide plug directly into your shell (zsh, bash) and **remember what you type so you don't have to**. Ghost-text suggestions, fuzzy search, snippet libraries — all free, all open-source, and all installed in minutes.

**~5**

min total install time

**\$0**

cost — all free &amp; open-source

**100%**

tmux &amp; cmux compatible

## WHAT'S INSIDE

- 1 zsh-autosuggestions** Ghost-text completion as you type, based on your history
- 2 fzf** Interactive fuzzy search across history, files and more
- 3 atuin** Smarter history with timestamps, exit codes and sync
- 4 navi** Snippet tool — search your cheatsheets by description, run the command
- 5 pet** Snippet tool — save any command with a tag, search and run it later

# zsh-autosuggestions

Ghost text that finishes your commands as you type

## WHAT IT DOES

As you type any command, zsh-autosuggestions shows a **faded grey suggestion** to the right of your cursor — pulled from your command history. Press the right arrow key to accept the full suggestion, or keep typing to ignore it. It's like having autocomplete in every browser field, but for your terminal.

## HOW IT WORKS

- 1 Start typing any command — e.g. `git push origin`
- 2 The tool looks through your history and shows the most likely completion in faint grey text
- 3 Press **→ (right arrow)** to accept the full suggestion instantly
- 4 Or press **Ctrl+F** to accept just the next word in the suggestion

## INSTALLATION

```
# Step 1: Install via Homebrew
brew install zsh-autosuggestions

# Step 2: Add to ~/.zshrc
echo 'source $(brew --prefix)/share/zsh-autosuggestions/zsh-autosuggestions.zsh' >>
~/.zshrc

# Step 3: Reload your shell
source ~/.zshrc
```

## KEY SHORTCUTS

**→** Accept the full suggestion

**Ctrl + F** Accept next word only

**Ctrl + E** Move to end of line

**Esc** Ignore the suggestion

# fzf

Fuzzy finder — search anything interactively in your terminal

## WHAT IT DOES

fzf (fuzzy finder) turns your command history into a **searchable, filterable list**. Press **Ctrl+R** and type any part of a command — even out of order — and fzf narrows it down instantly. It also works for file paths, git branches, process IDs, and much more. Think of it as "search-as-you-type" for your entire terminal.

## HOW IT WORKS

- 1 Press **Ctrl+R** — a searchable history panel opens over your terminal
- 2 Type any fragment — e.g. `docker run` — and it fuzzy-matches all matching commands
- 3 Use arrow keys to scroll through matches, then press **Enter** to run
- 4 Press **Ctrl+T** to fuzzy-search files and insert the path into your command

## INSTALLATION

```
# Step 1: Install via Homebrew
brew install fzf

# Step 2: Install shell integrations (say y to both prompts)
$(brew --prefix)/opt/fzf/install

# Step 3: Reload your shell
source ~/.zshrc
```

## KEY SHORTCUTS

**Ctrl + R** Fuzzy search command history

**Ctrl + T** Fuzzy search files & insert path

**Alt + C** Fuzzy cd into a directory

**Enter** Run the selected command

# atuin

Your shell history — smarter, searchable, and syncable across machines

## WHAT IT DOES

Atuin replaces your default **Ctrl+R** history search with a rich TUI (terminal UI) that shows **timestamps, exit codes, working directory, and how often you've used each command**. You can see not just what you ran, but when it ran, whether it failed, and where you were when you ran it. It also optionally syncs history across all your machines via an encrypted server.

## HOW IT WORKS

- 1 Press **Ctrl+R** — atuin opens a full-panel interactive history browser
- 2 Each result shows the command, the folder it ran in, the date, and a ✓ or ✗ exit status
- 3 Filter by current directory or session using the toolbar — great for project-specific history
- 4 Optional: create a free atuin account to sync history encrypted across all your Macs/servers

## INSTALLATION

```
# Step 1: Install via Homebrew
brew install atuin

# Step 2: Add to ~/.zshrc
echo 'eval "$(atuin init zsh)"' >> ~/.zshrc

# Step 3: Reload your shell
source ~/.zshrc
```

## KEY SHORTCUTS

**Ctrl + R** Open the smart history browser

**Ctrl + D** Filter to current directory only

**Ctrl + S** Filter to current session only

**Enter** Run the selected command

# navi

Snippet tool — search cheatsheets by description, get the exact command

## WHAT IT DOES

Navi is a **cheatsheet browser for your terminal**. Instead of googling "how do I list all running Docker containers", you type a description in navi and it returns the exact command — ready to run. You can use built-in community cheatsheets or write your own. It's the tool that eliminates the "I know what I want to do, but not the exact flags" problem.

## HOW IT WORKS

- 1 Run `navi` to open the interactive panel
- 2 Type a description like *"list docker containers"* — navi filters cheatsheets in real time
- 3 Select the result, fill in any variable prompts (like container name), press Enter
- 4 Add your own `.cheat` files to build a personal command library

## INSTALLATION

```
# Step 1: Install via Homebrew
brew install navi

# Step 2: Optional — add shell widget (Ctrl+G shortcut)
echo 'eval "$(navi widget zsh)"' >> ~/.zshrc

# Step 3: Reload and launch
source ~/.zshrc && navi
```

## KEY SHORTCUTS

**navi** Open interactive cheatsheet browser

**Ctrl + G** Open navi inline (with widget)

**Tab** Fill in a variable field

**Enter** Run the selected command

# pet

Snippet tool — save any command, tag it, find it in seconds

## WHAT IT DOES

Pet is a **personal command snippet library**. Save any command you use repeatedly — with a short description and optional tags. Then search it instantly by keyword. It's simpler than navi (no cheatsheet format required) and perfect for saving your own project-specific commands, one-liners, and scripts you always forget the exact syntax for.

## HOW IT WORKS

- 1 Run a command, then use `pet new` to save it with a description
- 2 Later, run `pet search` to fuzzy-search your saved snippets
- 3 Select a result to copy it to clipboard or run it directly
- 4 Snippets are stored in a plain TOML file — easy to back up or share with a team

## INSTALLATION

```
# Step 1: Install dependencies
brew install peco

# Step 2: Install pet
brew install knqyf263/pet/pet

# Step 3: Save your first snippet
pet new # follow the prompts to add a command + description
```

## KEY COMMANDS

`pet new` Save a new snippet

`pet search` Fuzzy-search your snippets

`pet list` Show all saved snippets

`pet edit` Edit snippets in your text editor

# Install All 5 in One Go

Copy these commands — everything you need from start to finish

## **zsh-autosuggestions** — ghost text completion

```
brew install zsh-autosuggestions
echo 'source $(brew --prefix)/share/zsh-autosuggestions/zsh-autosuggestions.zsh' >>
~/.zshrc
```

## **fzf** — fuzzy history search

```
brew install fzf
$(brew --prefix)/opt/fzf/install # answer y to both prompts
```

## **atuin** — smart history with context

```
brew install atuin
echo 'eval "$(atuin init zsh)"' >> ~/.zshrc
```

## **navi** — cheatsheet browser

```
brew install navi
echo 'eval "$(navi widget zsh)"' >> ~/.zshrc
```

## **pet** — snippet manager

```
brew install peco knqyf263/pet/pet
```

FINAL STEP — ALWAYS RUN THIS AFTER EDITING ~/.ZSHRC

```
source ~/.zshrc
```



# Want More Guides Like **This One?**

AI Doctor breaks down developer tools, AI workflows, and tech shortcuts into plain language — no jargon, no fluff. Follow along for free resources, tutorials, and tools that actually save you time.



**Follow AI Doctor**  
for more free guides & tools



**Free resources every week**  
AI tools, terminal tricks, dev shortcuts

---

## AI Doctor

TECH MADE SIMPLE · 2026