fzf Fuzzy finder

START HERE

`<cmd> | fzf`
Pick a line from any list

`Alt+C`
cd into chosen directory

`fzf`
Browse files here

`fzf -m`
Select several with Tab

`Ctrl+T`
Paste chosen file path

`fzf --preview '<cmd> {}'`
Preview the focused line

`Ctrl+R`
Fuzzy search history

`vim **<Tab>`
Fuzzy path completion

CONCEPTS

fzf reads lines on stdin, shows a filterable list and prints the chosen line(s) on stdout. It is a filter, so anything that makes lines can feed it.

Source stdin, find, or walker
↓ lines
fzf type to filter, move, mark
↓ selection
stdout chosen line(s)
↓ consumed by
Command vim, cd, kill, git ...
How fzf fits in a pipeline

Run eval "$(fzf --bash)" in ~/.bashrc (or source <(fzf --zsh)) to get Ctrl+T, Ctrl+R, Alt+C and ** completion.

RUNNING FZF

`<cmd> | fzf`
Choose one line from the input

`fzf`
Browse files under the current dir

`fzf -m`
Multi-select (Tab marks, Shift+Tab unmarks)

`fzf -q '<query>'`
Start with a query typed in

`fzf -1 -0`
Auto-pick a single match, exit on none

`fzf --height 40%`
Inline list instead of full screen

`fzf --layout reverse`
Prompt at the top, list grows down

`fzf --border`
Draw a border around the finder

`fzf --ansi`
Keep colours from the input

`fzf --print0`
Separate output by NUL (pair with xargs -0)

`fzf --disabled`
Turn off filtering; list is driven by reload

`fzf --tmux 80%`
Open in a tmux popup
Needs tmux 3.3+; silently ignored elsewhere

KEYS INSIDE THE FINDER

`Ctrl+J / Ctrl+N / Down`
Move down

`Ctrl+K / Ctrl+P / Up`
Move up

`Enter`
Accept the focused item

KEYS INSIDE THE FINDER (cont.)

`Esc / Ctrl+C / Ctrl+G`
Quit without selecting

`Tab / Shift+Tab`
Mark or unmark item (with -m)

`Ctrl+A / Ctrl+E`
Cursor to start or end of query

`Ctrl+W / Ctrl+U`
Delete word or whole query

`PgUp / PgDn`
Page through the list

`Shift+Up / Shift+Down`
Scroll the preview

SEARCH SYNTAX

Tokens (space = AND)

Token	Matches
`sbtrkt`	fuzzy
`'wild`	exact substring
`'wild'`	exact, word boundaries
`^music`	prefix
`.mp3$`	suffix
`!fire`	does not contain
`!^music`	does not start with
`a | b`	a OR b

`fzf -e`
Exact matching by default (' makes a term fuzzy)

`fzf -i / +i`
Force case-insensitive or case-sensitive

`fzf --scheme=path`
Rank for file paths

Terms are ANDed, so 'core go$ | rb$' narrows quickly.

SHELL INTEGRATION

`eval "$(fzf --bash)"`
Enable bindings and completion in bash

`source <(fzf --zsh)`
Enable in zsh

`fzf --fish | source`
Enable in fish

`Ctrl+T`
Paste selected files or dirs at the cursor

`Ctrl+R`
Fuzzy search command history

`Alt+C`
cd into a selected directory

`Ctrl+R (again)`
Toggle relevance and chronological sort

`<cmd> **<Tab>`
Fuzzy completion of paths, e.g. vim **<Tab>

`kill -9 **<Tab>`
Pick a process to kill

`ssh **<Tab>`
Pick a host from config and hosts files

PREVIEW AND LAYOUT

`fzf --preview 'cat {}'`
Show a preview of the focused line

`fzf --preview-window right,50%`
Position and size the preview

`--preview-window up,40%,border-bottom`
Put the preview above the list

`--preview-window hidden`
Start hidden (toggle with a bind)

`--preview-window '~3'`
Pin the first 3 preview lines as a header

`fzf --header '<text>'`
Add a fixed header line

`fzf --header-lines 1`
Treat input's first line as header

`fzf --prompt '<text> '`
Change the prompt string

{} in a preview or bind is the focused line in single quotes; {+} is every marked line, {q} is the query.

FIELDS AND OUTPUT

`--delimiter ':'`
Split each line into fields

`--nth 2`
Search only field 2

`--with-nth 2..`
Show only fields from 2 onwards

`--accept-nth 1`
Print only field 1 of the selection

`{1} {2} {-1}`
Field placeholders in preview or bind commands

`--walker file,dir,follow,hidden`
Choose what the built-in walker lists

`--walker-skip .git,node_modules`
Skip directories while walking

BINDINGS AND ACTIONS

`--bind 'ctrl-/:toggle-preview'`
Bind a key to a preview toggle

`--bind 'enter:become(vim {})'`
Replace fzf with a command on Enter

`--bind 'ctrl-y:execute-silent(<cmd>)+abort'`
Run a command silently, then quit

BINDINGS AND ACTIONS (cont.)

`--bind 'ctrl-o:execute(<cmd> {})'`
Run a command, then return to fzf

`--bind 'ctrl-r:reload(<cmd>)'`
Replace the list with new output

`--bind 'change:reload(<cmd> {q})'`
Re-run a command on each keystroke

`--bind 'start:<action>'`
Event bindings: start, change, focus, result

CONFIGURATION

`FZF_DEFAULT_OPTS`
Options applied to every run

`FZF_DEFAULT_OPTS_FILE`
Read default options from a file (~/.fzfrc)

`FZF_DEFAULT_COMMAND`
Where fzf gets candidates with no stdin

`FZF_CTRL_T_COMMAND`
Override the Ctrl+T source

`FZF_CTRL_T_OPTS`
Extra options for Ctrl+T (e.g. preview)

`FZF_CTRL_R_OPTS`
Extra options for Ctrl+R

Do not put --preview in FZF_DEFAULT_OPTS: it also applies to history, ps and other non-file inputs. Use FZF_CTRL_T_OPTS instead.

RECIPES

Open a file with a preview

```
vim $(fzf --preview 'head -n 50 {}')
# or, safer with spaces
fzf --preview 'head -n 50 {}' --bind 'enter:become(vim {})'
```

Checkout a git branch

```
# local branches (names with spaces or * break a plain pipe)
git switch "$(git branch --format='%(refname:short)' | fzf)"
# include remotes
git switch "$(git branch --all --format='%(refname:short)' | fzf)"
```

Kill processes by picking

```
ps -ef | fzf -m --header-lines=1 | awk '{print $2}' | xargs kill
```

Live ripgrep launcher

```
rg --line-number --color=always '' |
  fzf --ansi --delimiter : \
      --bind 'change:reload(rg --line-number --color=always {q} || true)' \
      --disabled --bind 'enter:become(vim {1} +{2})'
```