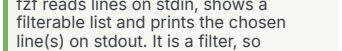
fzf Fuzzy finder

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.



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

KEYS INSIDE THE FINDER

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

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

`Enter`
Accept the focused item

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

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

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

SHELL INTEGRATION

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

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

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

`Ctrl+R`
Fuzzy search command history

`Alt+C`
cd into a selected directory

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

PREVIEW AND LAYOUT

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

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

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

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

CONFIGURATION

`FZF_DEFAULT_OPTS`
Options applied to every run