

**NAME**

fzf - a command-line fuzzy finder

**SYNOPSIS**

fzf [options]

**DESCRIPTION**

fzf is a general-purpose command-line fuzzy finder.

**OPTIONS****Search mode****-x, --extended**

Extended-search mode. Since 0.10.9, this is enabled by default. You can disable it with **+x** or **--no-extended**.

**-e, --exact**

Enable exact-match

**-i** Case-insensitive match (default: smart-case match)**+i** Case-sensitive match**--literal**

Do not normalize latin script letters for matching.

**--scheme=SCHEME**

Choose scoring scheme tailored for different types of input.

**default** Generic scoring scheme designed to work well with any type of input

**path** Scoring scheme well suited for file paths

**history** Scoring scheme well suited for command history or any input where chronological ordering is important

Sets **--tiebreak=index** as well.

**--algo=TYPE**

Fuzzy matching algorithm (default: v2)

- v2** Optimal scoring algorithm (quality)
- v1** Faster but not guaranteed to find the optimal result (performance)

**-n, --nth=N[,...]**

Comma-separated list of field index expressions for limiting search scope. See **FIELD INDEX EXPRESSION** for the details.

**--with-nth=N[,...]**

Transform the presentation of each line using field index expressions

**-d, --delimiter=STR**

Field delimiter regex for **--nth** and **--with-nth** (default: AWK-style)

**--disabled**

Do not perform search. With this option, fzf becomes a simple selector interface rather than a "fuzzy finder". You can later enable the search using **enable-search** or **toggle-search** action.

## Search result

**+s, --no-sort**

Do not sort the result

**--track**

Make fzf track the current selection when the result list is updated. This can be useful when browsing logs using fzf with sorting disabled. It is not recommended to use this option with **--tac** as the resulting behavior can be confusing. Also, consider using **track** action instead of this option.

e.g.

```
git log --oneline --graph --color=always | nl |  
fzf --ansi --track --no-sort --layout=reverse-list
```

**--tac**

Reverse the order of the input

e.g.

```
history | fzf --tac --no-sort
```

**--tiebreak=CRI[,...]**

Comma-separated list of sort criteria to apply when the scores are tied.

**length** Prefers line with shorter length  
**chunk** Prefers line with shorter matched chunk (delimited by whitespaces)  
**begin** Prefers line with matched substring closer to the beginning  
**end** Prefers line with matched substring closer to the end  
**index** Prefers line that appeared earlier in the input stream

- Each criterion should appear only once in the list
- **index** is only allowed at the end of the list
- **index** is implicitly appended to the list when not specified
- Default is **length** (or equivalently **length,index**)
- If **end** is found in the list, fzf will scan each line backwards

## Interface

### **-m, --multi**

Enable multi-select with tab/shift-tab. It optionally takes an integer argument which denotes the maximum number of items that can be selected.

### **+m, --no-multi**

Disable multi-select

### **--no-mouse**

Disable mouse

### **--bind=KEYBINDS**

Comma-separated list of custom key bindings. See **KEY/EVENT BINDINGS** for the details.

### **--cycle**

Enable cyclic scroll

### **--keep-right**

Keep the right end of the line visible when it's too long. Effective only when the query string is empty.

### **--scroll-off=LINES**

Number of screen lines to keep above or below when scrolling to the top or to the bottom (default: 0).

### **--no-hscroll**

Disable horizontal scroll

**--hscroll-off=COLS**

Number of screen columns to keep to the right of the highlighted substring (default: 10). Setting it to a large value will cause the text to be positioned on the center of the screen.

**--filepath-word**

Make word-wise movements and actions respect path separators. The following actions are affected:

**backward-kill-word**

**backward-word**

**forward-word**

**kill-word**

**--jump-labels=CHARS**

Label characters for **jump** and **jump-accept**

**Layout****--height=[~]HEIGHT[%]**

Display fzf window below the cursor with the given height instead of using the full screen. When prefixed with ~, fzf will automatically determine the height in the range according to the input size. Note that adaptive height is not compatible with top/bottom margin and padding given in percent size.

**--min-height=HEIGHT**

Minimum height when **--height** is given in percent (default: 10). Ignored when **--height** is not specified.

**--layout=LAYOUT**

Choose the layout (default: default)

**default**     Display from the bottom of the screen

**reverse**     Display from the top of the screen

**reverse-list** Display from the top of the screen, prompt at the bottom

**--reverse**

A synonym for **--layout=reverse**

**--border[=BORDER\_OPT]**

Draw border around the finder

**rounded**    Border with rounded corners (default)  
**sharp**     Border with sharp corners  
**bold**      Border with bold lines  
**double**    Border with double lines  
**block**     Border using block elements; suitable when using different background colors  
**thinblock** Border using legacy computing symbols; may not be displayed on some terminals  
**horizontal** Horizontal lines above and below the finder  
**vertical**   Vertical lines on each side of the finder  
**top** (up)  
**bottom** (down)  
**left**  
**right**  
**none**

If you use a terminal emulator where each box-drawing character takes 2 columns, try setting **RUNEWIDTH\_EASTASIAN** environment variable to **0** or **1**. If the border is still not properly rendered, set **--no-unicode**.

#### **--border-label[=*LABEL*]**

Label to print on the horizontal border line. Should be used with one of the following **--border** options.

- \* **rounded**
- \* **sharp**
- \* **bold**
- \* **double**
- \* **horizontal**
- \* **top** (up)
- \* **bottom** (down)

e.g.

# ANSI color codes are supported

# (with <https://github.com/busyloop/lolcat>)

label=\$(curl -s http://metaphorpsum.com/sentences/1 | lolcat -f)

# Border label at the center

fzf --height=10 --border --border-label="<?> \$label <?>" --color=label:italic:black

**# Left-aligned (positive integer)**

```
fzf --height=10 --border --border-label="<?> $label <?>" --border-label-pos=3
--color=label:italic:black
```

**# Right-aligned (negative integer) on the bottom line (:bottom)**

```
fzf --height=10 --border --border-label="<?> $label <?>" --border-label-pos=-3:bottom
--color=label:italic:black
```

**--border-label-pos[=*N[:top|bottom]*]**

Position of the border label on the border line. Specify a positive integer as the column position from the left. Specify a negative integer to right-align the label. Label is printed on the top border line by default, add **:bottom** to put it on the border line on the bottom. The default value **0 (or center)** will put the label at the center of the border line.

**--no-unicode**

Use ASCII characters instead of Unicode drawing characters to draw borders, the spinner and the horizontal separator.

**--margin=*MARGIN***

Comma-separated expression for margins around the finder.

**TRBL** Same margin for top, right, bottom, and left

**TB,RL** Vertical, horizontal margin

**T,RL,B** Top, horizontal, bottom margin

**T,R,B,L** Top, right, bottom, left margin

Each part can be given in absolute number or in percentage relative to the terminal size with % suffix.

e.g.

```
fzf --margin 10%
```

```
fzf --margin 1,5%
```

**--padding=*PADDING***

Comma-separated expression for padding inside the border. Padding is distinguishable from margin only when **--border** option is used.

e.g.

```
fzf --margin 5% --padding 5% --border --preview 'cat {}' \
    --color bg:#222222,preview-bg:#333333
```

**TRBL** Same padding for top, right, bottom, and left

**TB,RL** Vertical, horizontal padding

**T,RL,B** Top, horizontal, bottom padding

**T,R,B,L** Top, right, bottom, left padding

### **--info=STYLE**

Determines the display style of finder info (match counters).

**default** Display on the next line to the prompt

**right** Display on the right end of the next line to the prompt

**inline** Display on the same line with the default separator ' < '

**inline:SEPARATOR** Display on the same line with a non-default separator

**inline-right** Display on the right end of the same line

**hidden** Do not display finder info

### **--no-info**

A synonym for **--info=hidden**

### **--separator=STR**

The given string will be repeated to form the horizontal separator on the info line (default: '<?>' or '-' depending on **--no-unicode**).

ANSI color codes are supported.

### **--no-separator**

Do not display horizontal separator on the info line. A synonym for **--separator=""**

### **--scrollbar=CHAR1[CHAR2]**

Use the given character to render scrollbar. (default: '|' or ':' depending on **--no-unicode**). The optional **CHAR2** is used to render scrollbar of the preview window.

**--no-scrollbar**

Do not display scrollbar. A synonym for **--scrollbar=""**

**--prompt=STR**

Input prompt (default: '> ')

**--pointer=STR**

Pointer to the current line (default: '>')

**--marker=STR**

Multi-select marker (default: '>')

**--header=STR**

The given string will be printed as the sticky header. The lines are displayed in the given order from top to bottom regardless of **--layout** option, and are not affected by **--with-nth**. ANSI color codes are processed even when **--ansi** is not set.

**--header-lines=N**

The first N lines of the input are treated as the sticky header. When **--with-nth** is set, the lines are transformed just like the other lines that follow.

**--header-first**

Print header before the prompt line

**--ellipsis=STR**

Ellipsis to show when line is truncated (default: '..')

**Display****--ansi**

Enable processing of ANSI color codes

**--tabstop=SPACES**

Number of spaces for a tab character (default: 8)

**--color=[BASE\_SCHEME][,COLOR\_NAME[:ANSI\_COLOR][:ANSI\_ATTRIBUTES]]...**

Color configuration. The name of the base color scheme is followed by custom color mappings.

**BASE SCHEME:**

(default: dark on 256-color terminal, otherwise 16)



**dark** Color scheme for dark 256-color terminal  
**light** Color scheme for light 256-color terminal  
**16** Color scheme for 16-color terminal  
**bw** No colors (equivalent to **--no-color**)

## COLOR NAMES:

**fg** Text  
**preview-fg** Preview window text  
**bg** Background  
**preview-bg** Preview window background  
**hl** Highlighted substrings  
**fg+** Text (current line)  
**bg+** Background (current line)  
**gutter** Gutter on the left  
**hl+** Highlighted substrings (current line)  
**query** Query string  
**disabled** Query string when search is disabled (**--disabled**)  
**info** Info line (match counters)  
**border** Border around the window (**--border** and **--preview**)  
**scrollbar** Scrollbar  
**preview-border** Border around the preview window (**--preview**)  
**preview-scrollbar** Scrollbar  
**separator** Horizontal separator on info line  
**label** Border label (**--border-label** and **--preview-label**)  
**preview-label** Border label of the preview window (**--preview-label**)  
**prompt** Prompt  
**pointer** Pointer to the current line  
**marker** Multi-select marker  
**spinner** Streaming input indicator  
**header** Header

## ANSI COLORS:

**-1** Default terminal foreground/background color  
(or the original color of the text)  
**0 ~ 15** 16 base colors  
**black**  
**red**  
**green**  
**yellow**  
**blue**

**magenta**  
**cyan**  
**white**  
**bright-black** (gray | grey)  
**bright-red**  
**bright-green**  
**bright-yellow**  
**bright-blue**  
**bright-magenta**  
**bright-cyan**  
**bright-white**  
**16 ~ 255** ANSI 256 colors  
**#rrggbb** 24-bit colors

**ANSI ATTRIBUTES: (Only applies to foreground colors)**

**regular** Clears previously set attributes; should precede the other ones  
**bold**  
**underline**  
**reverse**  
**dim**  
**italic**  
**strikethrough**

**EXAMPLES:**

**# Seoul256 theme with 8-bit colors**

**# (<https://github.com/junegunn/seoul256.vim>)**

**fzf --color='bg:237,bg+:236,info:143,border:240,spinner:108' \**  
**--color='hl:65,fg:252,header:65,fg+:252' \**  
**--color='pointer:161,marker:168,prompt:110,hl+:108'**

**# Seoul256 theme with 24-bit colors**

**fzf --color='bg:#4B4B4B,bg+:#3F3F3F,info:#BDBB72,border:#6B6B6B,spinner:#98BC99' \**  
**--color='hl:#719872,fg:#D9D9D9,header:#719872,fg+:#D9D9D9' \**  
**--color='pointer:#E12672,marker:#E17899,prompt:#98BEDE,hl+:#98BC99'**

**--no-bold**

Do not use bold text

**--black**

Use black background

## History

**--history**=*HISTORY\_FILE*

Load search history from the specified file and update the file on completion. When enabled, **CTRL-N** and **CTRL-P** are automatically remapped to **next-history** and **prev-history**.

**--history-size**=*N*

Maximum number of entries in the history file (default: 1000). The file is automatically truncated when the number of the lines exceeds the value.

## Preview

**--preview**=*COMMAND*

Execute the given command for the current line and display the result on the preview window. {} in the command is the placeholder that is replaced to the single-quoted string of the current line. To transform the replacement string, specify field index expressions between the braces (See **FIELD INDEX EXPRESSION** for the details).

e.g.

```
fzf --preview='head -$LINES {}'
```

```
ls -l | fzf --preview="echo user={3} when={-4..-2}; cat {-1}" --header-lines=1
```

fzf exports **\$FZF\_PREVIEW\_LINES** and **\$FZF\_PREVIEW\_COLUMNS** so that they represent the exact size of the preview window. (It also overrides **\$LINES** and **\$COLUMNS** with the same values but they can be reset by the default shell, so prefer to refer to the ones with **FZF\_PREVIEW\_** prefix.)

A placeholder expression starting with + flag will be replaced to the space-separated list of the selected lines (or the current line if no selection was made) individually quoted.

e.g.

```
fzf --multi --preview='head -10 {+}'
```

```
git log --oneline | fzf --multi --preview 'git show {+1}'
```

When using a field index expression, leading and trailing whitespace is stripped from the replacement string. To preserve the whitespace, use the **s** flag.

Also, **{q}** is replaced to the current query string, and **{n}** is replaced to zero-based ordinal index of the line. Use **{+n}** if you want all index numbers when multiple lines are selected.

A placeholder expression with **f** flag is replaced to the path of a temporary file that holds the evaluated list. This is useful when you multi-select a large number of items and the length of the evaluated string may exceed **ARG\_MAX**.

e.g.

```
# Press CTRL-A to select 100K items and see the sum of all the numbers.
# This won't work properly without 'f' flag due to ARG_MAX limit.
seq 100000 | fzf --multi --bind ctrl-a:select-all \
    --preview "awk '{sum+=\$1} END {print sum}' {+f}"
```

Note that you can escape a placeholder pattern by prepending a backslash.

Preview window will be updated even when there is no match for the current query if any of the placeholder expressions evaluates to a non-empty string or **{q}** is in the command template.

Since 0.24.0, fzf can render partial preview content before the preview command completes. ANSI escape sequence for clearing the display (**CSI 2 J**) is supported, so you can use it to implement preview window that is constantly updating.

e.g.

```
fzf --preview 'for i in $(seq 100000); do
(( i % 200 == 0 )) && printf "\033[2J"
echo "$i"
sleep 0.01
done'
```

Since 0.43.0, fzf has experimental support for Kitty graphics protocol, so if you use Kitty, you can make fzf display an image in the preview window.

e.g.

```
fzf --preview='
if file --mime-type {} | grep -qF "image/"; then
    kitty icat --clear --transfer-mode=memory --stdin=no
--place=${FZF_PREVIEW_COLUMNS}x${FZF_PREVIEW_LINES}@0x0 {} | sed \${d
else
    bat --color=always {}
fi
,
```

**--preview-label[=*LABEL*]**

Label to print on the horizontal border line of the preview window. Should be used with one of the following **--preview-window** options.

- \* **border-rounded** (default on non-Windows platforms)
- \* **border-sharp** (default on Windows)
- \* **border-bold**
- \* **border-double**
- \* **border-block**
- \* **border-thinblock**
- \* **border-horizontal**
- \* **border-top**
- \* **border-bottom**

**--preview-label-pos[=*N[:top|bottom]*]**

Position of the border label on the border line of the preview window. Specify a positive integer as the column position from the left. Specify a negative integer to right-align the label. Label is printed on the top border line by default, add **:bottom** to put it on the border line on the bottom. The default value 0 (or **center**) will put the label at the center of the border line.

**--preview-window=[*POSITION*][,*SIZE*[%]][,*border-BORDER\_OPT*][,*[no]wrap*][,*[no]follow*][,*[no]cycle*][,*[no]hidd*]**

**POSITION:** (default: **right**)

**up**  
**down**  
**left**  
**right**

Determines the layout of the preview window.

\* If the argument contains **:hidden**, the preview window will be hidden by default until **toggle-preview** action is triggered.

\* If size is given as 0, preview window will not be visible, but fzf will still execute the command in the background.

\* Long lines are truncated by default. Line wrap can be enabled with **wrap** flag.

\* Preview window will automatically scroll to the bottom when **follow** flag is set, similarly to how **tail -f** works.

e.g.

```
fzf --preview-window follow --preview 'for i in $(seq 100000); do
  echo "$i"
  sleep 0.01
  (( i % 300 == 0 )) && printf "\033[2J"
done'
```

\* Cyclic scrolling is enabled with **cycle** flag.

\* To change the style of the border of the preview window, specify one of the options for **--border** with **border-** prefix. e.g. **border-rounded** (border with rounded edges, default), **border-sharp** (border with sharp edges), **border-left**, **border-none**, etc.

\* **[+SCROLL[OFFSETS][/DENOM]]** determines the initial scroll offset of the preview window.

- **SCROLL** can be either a numeric integer or a single-field index expression that refers to a numeric integer.

- The optional **OFFSETS** part is for adjusting the base offset. It should be given as a series of signed integers (**-INTEGER** or **+INTEGER**).

- The final **/DENOM** part is for specifying a fraction of the preview window height.

\* **~HEADER\_LINES** keeps the top N lines as the fixed header so that they are always visible.

\* **default** resets all options previously set to the default.

e.g.

```
# Non-default scroll window positions and sizes
fzf --preview="head {}" --preview-window=up,30%
fzf --preview="file {}" --preview-window=down,1

# Initial scroll offset is set to the line number of each line of
# git grep output *minus* 5 lines (-5)
git grep --line-number '' |
fzf --delimiter : --preview 'nl {1}' --preview-window '+{2}-5'
```

```

# Preview with bat, matching line in the middle of the window below
# the fixed header of the top 3 lines
#
# ~3   Top 3 lines as the fixed header
# +{2} Base scroll offset extracted from the second field
# +3   Extra offset to compensate for the 3-line header
# /2   Put in the middle of the preview area
#
git grep --line-number '' |
fzf --delimiter : \
    --preview 'bat --style=full --color=always --highlight-line {2} {1}' \
    --preview-window '~3,+{2}+3/2'

# Display top 3 lines as the fixed header
fzf --preview 'bat --style=full --color=always {}' --preview-window '~3'

```

\* You can specify an alternative set of options that are used only when the size of the preview window is below a certain threshold. Note that only one alternative layout is allowed.

e.g.

```
fzf --preview 'cat {}' --preview-window 'right,border-left,<30(up,30%,border-bottom)'
```

## Scripting

**-q, --query=STR**

Start the finder with the given query

**-1, --select-1**

If there is only one match for the initial query (**--query**), do not start interactive finder and automatically select the only match

**-0, --exit-0**

If there is no match for the initial query (**--query**), do not start interactive finder and exit immediately

**-f, --filter=STR**

Filter mode. Do not start interactive finder. When used with **--no-sort**, fzf becomes a fuzzy-version of grep.

**--print-query**

Print query as the first line

**--expect=KEY[,...]**

Comma-separated list of keys that can be used to complete fzf in addition to the default enter key. When this option is set, fzf will print the name of the key pressed as the first line of its output (or as the second line if **--print-query** is also used). The line will be empty if fzf is completed with the default enter key. If **--expect** option is specified multiple times, fzf will expect the union of the keys. **--no-expect** will clear the list.

e.g.

```
fzf --expect=ctrl-v,ctrl-t,alt-s --expect=f1,f2,~,@
```

**--read0**

Read input delimited by ASCII NUL characters instead of newline characters

**--print0**

Print output delimited by ASCII NUL characters instead of newline characters

**--no-clear**

Do not clear finder interface on exit. If fzf was started in full screen mode, it will not switch back to the original screen, so you'll have to manually run **tput rmcup** to return. This option can be used to avoid flickering of the screen when your application needs to start fzf multiple times in order. (Note that in most cases, it is preferable to use **reload** action instead.)

e.g.

```
foo=$(seq 100 | fzf --no-clear) || (  
  # Need to manually switch back to the main screen when cancelled  
  tput rmcup  
  exit 1  
) && seq "$foo" 100 | fzf
```

**--sync**

Synchronous search for multi-staged filtering. If specified, fzf will launch ncurses finder only after the input stream is complete.

e.g. **fzf --multi | fzf --sync**

**--listen[=HTTP\_PORT]**



Start HTTP server on the given port. It allows external processes to send actions to perform via POST method. If the port number is omitted or given as 0, fzf will choose the port automatically and export it as **FZF\_PORT** environment variable to the child processes started via **execute** and **execute-silent** actions. If **FZF\_API\_KEY** environment variable is set, the server would require sending an API key with the same value in the **x-api-key** HTTP header.

e.g.

```
# Start HTTP server on port 6266
```

```
fzf --listen 6266
```

```
# Get program state in JSON format (experimental)
```

```
curl localhost:6266
```

```
# Send action to the server
```

```
curl -XPOST localhost:6266 -d 'reload(seq 100)+change-prompt(hundred> )'
```

```
# Start HTTP server on port 6266 and send an authenticated action
```

```
export FZF_API_KEY="$(head -c 32 /dev/urandom | base64)"
```

```
curl -XPOST localhost:6266 -H "x-api-key: $FZF_API_KEY" -d 'change-query(yo)'
```

```
# Choose port automatically and export it as $FZF_PORT to the child process
```

```
fzf --listen --bind 'start:execute-silent:echo $FZF_PORT > /tmp/fzf-port'
```

## **--version**

Display version information and exit

Note that most options have the opposite versions with **--no-** prefix.

## **ENVIRONMENT VARIABLES**

### **FZF\_DEFAULT\_COMMAND**

Default command to use when input is tty. On \*nix systems, fzf runs the command with **\$SHELL -c** if **SHELL** is set, otherwise with **sh -c**, so in this case make sure that the command is POSIX-compliant.

### **FZF\_DEFAULT\_OPTS**

Default options. e.g. **export FZF\_DEFAULT\_OPTS="--extended --cycle"**

**FZF\_API\_KEY**

Can be used to require an API key when using **--listen** option. If not set, no authentication will be required by the server. You can set this value if you need to protect against DNS rebinding and privilege escalation attacks.

**EXIT STATUS**

- 0** Normal exit
- 1** No match
- 2** Error
- 130** Interrupted with **CTRL-C** or **ESC**

**FIELD INDEX EXPRESSION**

A field index expression can be a non-zero integer or a range expression ([BEGIN]..[END]). **--nth** and **--with-nth** take a comma-separated list of field index expressions.

**Examples**

- 1** The 1st field
- 2** The 2nd field
- 1** The last field
- 2** The 2nd to last field
- 3..5** From the 3rd field to the 5th field
- 2..** From the 2nd field to the last field
- ..3** From the 1st field to the 3rd to the last field
- ..** All the fields

**EXTENDED SEARCH MODE**

Unless specified otherwise, fzf will start in "extended-search mode". In this mode, you can specify multiple patterns delimited by spaces, such as: **'wild ^music .mp3\$ sbtrkt !rmx**

You can prepend a backslash to a space (\ ) to match a literal space character.

**Exact-match (quoted)**

A term that is prefixed by a single-quote character (') is interpreted as an "exact-match" (or "non-fuzzy") term. fzf will search for the exact occurrences of the string.

### Anchored-match

A term can be prefixed by `^`, or suffixed by `$` to become an anchored-match term. Then fzf will search for the lines that start with or end with the given string. An anchored-match term is also an exact-match term.

### Negation

If a term is prefixed by `!`, fzf will exclude the lines that satisfy the term from the result. In this case, fzf performs exact match by default.

### Exact-match by default

If you don't prefer fuzzy matching and do not wish to "quote" (prefixing with `'`) every word, start fzf with `-e` or `--exact` option. Note that when `--exact` is set, `'`-prefix "unquotes" the term.

### OR operator

A single bar character term acts as an OR operator. For example, the following query matches entries that start with **core** and end with either **go**, **rb**, or **py**.

e.g. `^core go$ | rb$ | py$`

## KEY/EVENT BINDINGS

`--bind` option allows you to bind **a key** or **an event** to one or more **actions**. You can use it to customize key bindings or implement dynamic behaviors.

`--bind` takes a comma-separated list of binding expressions. Each binding expression is **KEY:ACTION** or **EVENT:ACTION**.

e.g.

`fzf --bind=ctrl-j:accept,ctrl-k:kill-line`

### AVAILABLE KEYS: (SYNONYMS)

*ctrl-[a-z]*

*ctrl-space*

*ctrl-delete*

*ctrl-\*

*ctrl-]*

*ctrl-^* (*ctrl-6*)  
*ctrl-/* (*ctrl-\_*)  
*ctrl-alt-[a-z]*  
*alt-[\*]* (Any case-sensitive single character is allowed)  
*f[1-12]*  
*enter* (*return ctrl-m*)  
*space*  
*bspace* (*bs*)  
*alt-up*  
*alt-down*  
*alt-left*  
*alt-right*  
*alt-enter*  
*alt-space*  
*alt-bspace* (*alt-bs*)  
*tab*  
*btabs* (*shift-tab*)  
*esc*  
*del*  
*up*  
*down*  
*left*  
*right*  
*home*  
*end*  
*insert*  
*pgup* (*page-up*)  
*pgdn* (*page-down*)  
*shift-up*  
*shift-down*  
*shift-left*  
*shift-right*  
*shift-delete*  
*alt-shift-up*  
*alt-shift-down*  
*alt-shift-left*  
*alt-shift-right*  
*left-click*  
*right-click*  
*double-click*

*scroll-up*  
*scroll-down*  
*preview-scroll-up*  
*preview-scroll-down*  
*shift-left-click*  
*shift-right-click*  
*shift-scroll-up*  
*shift-scroll-down*  
 or any single character

## AVAILABLE EVENTS:

### *start*

Triggered only once when fzf finder starts. Since fzf consumes the input stream asynchronously, the input list is not available unless you use *--sync*.

e.g.

```
# Move cursor to the last item and select all items
seq 1000 | fzf --multi --sync --bind start:last+select-all
```

### *load*

Triggered when the input stream is complete and the initial processing of the list is complete.

e.g.

```
# Change the prompt to "loaded" when the input stream is complete
(seq 10; sleep 1; seq 11 20) | fzf --prompt 'Loading> ' --bind 'load:change-prompt:Loaded> '
```

### *change*

Triggered whenever the query string is changed

e.g.

```
# Move cursor to the first entry whenever the query is changed
fzf --bind change:first
```

### *focus*

Triggered when the focus changes due to a vertical cursor movement or a search result update.

e.g.

```
fzf --bind 'focus:transform-preview-label:echo [ {} ]' --preview 'cat {}'
```

```
# Any action bound to the event runs synchronously and thus can make the interface sluggish
# e.g. lolcat isn't one of the fastest programs, and every cursor movement in
#   fzf will be noticeably affected by its execution time
```

```
fzf --bind 'focus:transform-preview-label:echo [ {} ] | lolcat -f' --preview 'cat {}'
```

**# Beware not to introduce an infinite loop**

```
seq 10 | fzf --bind 'focus:up' --cycle
```

*one*

Triggered when there's only one match. **one:accept** binding is comparable to **--select-1** option, but the difference is that **--select-1** is only effective before the interactive finder starts but **one** event is triggered by the interactive finder.

e.g.

**# Automatically select the only match**

```
seq 10 | fzf --bind one:accept
```

*zero*

Triggered when there's no match. **zero:abort** binding is comparable to **--exit-0** option, but the difference is that **--exit-0** is only effective before the interactive finder starts but **zero** event is triggered by the interactive finder.

e.g.

**# Reload the candidate list when there's no match**

```
echo $RANDOM | fzf --bind 'zero:reload(echo $RANDOM)+clear-query' --height 3
```

*backward-eof*

Triggered when the query string is already empty and you try to delete it backward.

e.g.

```
fzf --bind backward-eof:abort
```

## AVAILABLE ACTIONS:

A key or an event can be bound to one or more of the following actions.

ACTION:	DEFAULT BINDINGS (NOTES):
<b>abort</b>	<i>ctrl-c ctrl-g ctrl-q esc</i>
<b>accept</b>	<i>enter double-click</i>
<b>accept-non-empty</b>	(same as <b>accept</b> except that it prevents fzf from exiting without selection)
<b>backward-char</b>	<i>ctrl-b left</i>
<b>backward-delete-char</b>	<i>ctrl-h bspace</i>
<b>backward-delete-char/eof</b>	(same as <b>backward-delete-char</b> except aborts fzf if query is empty)
<b>backward-kill-word</b>	<i>alt-bs</i>
<b>backward-word</b>	<i>alt-b shift-left</i>

**become(...)** (replace fzf process with the specified command; see below for the details)  
**beginning-of-line** *ctrl-a home*  
**cancel** (clear query string if not empty, abort fzf otherwise)  
**change-border-label(...)** (change **--border-label** to the given string)  
**change-header(...)** (change header to the given string; doesn't affect **--header-lines**)  
**change-preview(...)** (change **--preview** option)  
**change-preview-label(...)** (change **--preview-label** to the given string)  
**change-preview-window(...)** (change **--preview-window** option; rotate through the multiple option sets separated by '|')  
**change-prompt(...)** (change prompt to the given string)  
**change-query(...)** (change query string to the given string)  
**clear-screen** *ctrl-l*  
**clear-selection** (clear multi-selection)  
**close** (close preview window if open, abort fzf otherwise)  
**clear-query** (clear query string)  
**delete-char** *del*  
**delete-char/eof** *ctrl-d* (same as **delete-char** except aborts fzf if query is empty)  
**deselect**  
**deselect-all** (deselect all matches)  
**disable-search** (disable search functionality)  
**down** *ctrl-j ctrl-n down*  
**enable-search** (enable search functionality)  
**end-of-line** *ctrl-e end*  
**execute(...)** (see below for the details)  
**execute-silent(...)** (see below for the details)  
**first** (move to the first match; same as **pos(1)**)  
**forward-char** *ctrl-f right*  
**forward-word** *alt-f shift-right*  
**ignore**  
**jump** (EasyMotion-like 2-keystroke movement)  
**jump-accept** (jump and accept)  
**kill-line**  
**kill-word** *alt-d*  
**last** (move to the last match; same as **pos(-1)**)  
**next-history** (*ctrl-n* on **--history**)  
**next-selected** (move to the next selected item)  
**page-down** *pgdn*  
**page-up** *pgup*  
**half-page-down**  
**half-page-up**

**hide-preview**  
**offset-down** (similar to CTRL-E of Vim)  
**offset-up** (similar to CTRL-Y of Vim)  
**pos(...)** (move cursor to the numeric position; negative number to count from the end)  
**prev-history** (*ctrl-p* on **--history**)  
**prev-selected** (move to the previous selected item)  
**preview(...)** (see below for the details)  
**preview-down** *shift-down*  
**preview-up** *shift-up*  
**preview-page-down**  
**preview-page-up**  
**preview-half-page-down**  
**preview-half-page-up**  
**preview-bottom**  
**preview-top**  
**print-query** (print query and exit)  
**put** (put the character to the prompt)  
**put(...)** (put the given string to the prompt)  
**refresh-preview**  
**rebind(...)** (rebind bindings after **unbind**)  
**reload(...)** (see below for the details)  
**reload-sync(...)** (see below for the details)  
**replace-query** (replace query string with the current selection)  
**select**  
**select-all** (select all matches)  
**show-preview**  
**toggle** (*right-click*)  
**toggle-all** (toggle all matches)  
**toggle+down** *ctrl-i* (*tab*)  
**toggle-header**  
**toggle-in** (**--layout=reverse\* ? toggle+up : toggle+down**)  
**toggle-out** (**--layout=reverse\* ? toggle+down : toggle+up**)  
**toggle-preview**  
**toggle-preview-wrap**  
**toggle-search** (toggle search functionality)  
**toggle-sort**  
**toggle-track**  
**toggle+up** *btabs* (*shift-tab*)  
**track** (track the current item; automatically disabled if focus changes)  
**transform-border-label(...)** (transform border label using an external command)



**transform-header(...)** (transform header using an external command)  
**transform-preview-label(...)** (transform preview label using an external command)  
**transform-prompt(...)** (transform prompt string using an external command)  
**transform-query(...)** (transform query string using an external command)  
**unbind(...)** (unbind bindings)  
**unix-line-discard** *ctrl-u*  
**unix-word-rubout** *ctrl-w*  
**up** *ctrl-k ctrl-p up*  
**yank** *ctrl-y*

## ACTION COMPOSITION

Multiple actions can be chained using + separator.

e.g.

```

fzf --multi --bind 'ctrl-a:select-all+accept'
fzf --multi --bind 'ctrl-a:select-all' --bind 'ctrl-a:+accept'

```

## ACTION ARGUMENT

An action denoted with (...) suffix takes an argument.

e.g.

```

fzf --bind 'ctrl-a:change-prompt(NewPrompt> )'
fzf --bind 'ctrl-v:preview(cat {})' --preview-window hidden

```

If the argument contains parentheses, fzf may fail to parse the expression. In that case, you can use any of the following alternative notations to avoid parse errors.

```

action-name[...]
action-name{...}
action-name<...>
action-name~...~
action-name!...!
action-name@...@
action-name#...#
action-name$...$
action-name%...%
action-name^...^
action-name&...&

```

**action-name\*...\***

**action-name;...;**

**action-name/.../**

**action-name|...|**

**action-name:...**

The last one is the special form that frees you from parse errors as it does not expect the closing character. The catch is that it should be the last one in the comma-separated list of key-action pairs.

## COMMAND EXECUTION

With **execute(...)** action, you can execute arbitrary commands without leaving fzf. For example, you can turn fzf into a simple file browser by binding **enter** key to **less** command like follows.

```
fzf --bind "enter:execute(less {})"
```

You can use the same placeholder expressions as in **--preview**.

fzf switches to the alternate screen when executing a command. However, if the command is expected to complete quickly, and you are not interested in its output, you might want to use **execute-silent** instead, which silently executes the command without the switching. Note that fzf will not be responsive until the command is complete. For asynchronous execution, start your command as a background process (i.e. appending **&**).

On \*nix systems, fzf runs the command with **\$SHELL -c** if **SHELL** is set, otherwise with **sh -c**, so in this case make sure that the command is POSIX-compliant.

**become(...)** action is similar to **execute(...)**, but it replaces the current fzf process with the specified command using **execve(2)** system call.

```
fzf --bind "enter:become(vim {})"
```

**become(...)** is not supported on Windows.

## RELOAD INPUT

**reload(...)** action is used to dynamically update the input list without restarting fzf. It takes the same command template with placeholder expressions as **execute(...)**.

See <https://github.com/junegunn/fzf/issues/1750> for more info.

e.g.

```
# Update the list of processes by pressing CTRL-R
ps -ef | fzf --bind 'ctrl-r:reload(ps -ef)' --header 'Press CTRL-R to reload' \
    --header-lines=1 --layout=reverse
```

# Integration with ripgrep

```
RG_PREFIX="rg --column --line-number --no-heading --color=always --smart-case "
INITIAL_QUERY="foobar"
FZF_DEFAULT_COMMAND="$RG_PREFIX '$INITIAL_QUERY'" \
    fzf --bind "change:reload:$RG_PREFIX {q} || true" \
    --ansi --disabled --query "$INITIAL_QUERY"
```

**reload-sync(...)** is a synchronous version of **reload** that replaces the list only when the command is complete. This is useful when the command takes a while to produce the initial output and you don't want fzf to run against an empty list while the command is running.

e.g.

```
# You can still filter and select entries from the initial list for 3 seconds
seq 100 | fzf --bind 'load:reload-sync(sleep 3; seq 1000)+unbind(load)'
```

## PREVIEW BINDING

With **preview(...)** action, you can specify multiple different preview commands in addition to the default preview command given by **--preview** option.

e.g.

```
# Default preview command with an extra preview binding
fzf --preview 'file {}' --bind '?:preview:cat {}'
```

```
# A preview binding with no default preview command
# (Preview window is initially empty)
fzf --bind '?:preview:cat {}'
```

```
# Preview window hidden by default, it appears when you first hit '?'
fzf --bind '?:preview:cat {}' --preview-window hidden
```

## CHANGE PREVIEW WINDOW ATTRIBUTES

**change-preview-window** action can be used to change the properties of the preview window. Unlike

the **--preview-window** option, you can specify multiple sets of options separated by '|' characters.

e.g.

```
# Rotate through the options using CTRL-/
fzf --preview 'cat {}' --bind 'ctrl-/:change-preview-window(right,70%|down,40%,border-
horizontal|hidden|right)'
```

# The default properties given by '--preview-window' are inherited, so an empty string in the list is interpreted as the default

```
fzf --preview 'cat {}' --preview-window 'right,40%,border-left' --bind 'ctrl-/:change-preview-
window(70%|down,border-top|hidden)'
```

# This is equivalent to toggle-preview action

```
fzf --preview 'cat {}' --bind 'ctrl-/:change-preview-window(hidden)'
```

## AUTHOR

Junegunn Choi (*junegunn.c@gmail.com*)

## SEE ALSO

### Project homepage:

*<https://github.com/junegunn/fzf>*

### Extra Vim plugin:

*<https://github.com/junegunn/fzf.vim>*

## LICENSE

MIT