

Navigation (open)

<cr>	Open the file or fugitive object under the cursor.
o	Open the file or fugitive object under the cursor in a new split.
gO	Open the file or fugitive object under the cursor in a new vertical split.
O	Open the file or fugitive object under the cursor in a new tab.
p	Open the file or fugitive object under the cursor in a preview window.
~	Open the current file in the [count]th first ancestor.
P	Open the current file in the [count]th parent.
C	Open the commit containing the current file.
gi	Open .git/info/exclude in a split. Use a count to open .gitignore.

Navigation (jump to file)

gu	Jump to file [count] in the "Untracked" or "Unstaged" section.
gU	Jump to file [count] in the "Unstaged" section.
gs	Jump to file [count] in the "Staged" section.
gp	Jump to file [count] in the "Unpushed" section.
gP	Jump to file [count] in the "Unpulled" section.
gr	Jump to file [count] in the "Rebasing" section.

Staging/unstaging

s	Stage (add) the file or hunk under the cursor.
u	Unstage (reset) the file or hunk under the cursor.
-	Stage or unstage the file or hunk under the cursor.
U	Unstage everything.
X	Discard the change under the cursor
=	Toggle an inline diff of the file under the cursor.
>	Insert an inline diff of the file under the cursor.
<	Remove the inline diff of the file under the cursor.
gI	Open .git/info/exclude in a split and add the file under the cursor.
I	Invoke :Git add --patch or reset --patch on the file under the cursor.
P	On untracked files, this instead calls :Git add --intent-to-add.

Checkout/branch

coo	Check out the commit under the cursor.
cb< Spa c e>	Populate command line with ":Git branch ".
co< Spa c e>	Populate command line with ":Git checkout ".
cb?	Show this help.
co?	Same as "cb?"



By isaacnewton

cheatography.com/isaacnewton/

Published 3rd August, 2021.

Last updated 3rd August, 2021.

Page 1 of 3.

Sponsored by **ApolloPad.com**

Everyone has a novel in them. Finish Yours!

<https://apollopad.com>

Stash

<code>czz</code>	Push stash. Pass a [count] of 1 to add <code>--include -untracked</code> or 2 to add <code>--all</code> .
<code>czw</code>	Push stash of worktree. Like <code>czz</code> with <code>--keep-index</code> .
<code>czA</code>	Apply topmost stash, or <code>stash@{count}</code> .
<code>cza</code>	Apply topmost stash, or <code>stash@{count}</code> , preserving the index.
<code>czP</code>	Pop topmost stash, or <code>stash@{count}</code> .
<code>czp</code>	Pop topmost stash, or <code>stash@{count}</code> , preserving the index.
<code>cz< space c</code> <code>e></code>	Populate command line with <code>":Git stash "</code> .
<code>cz?</code>	Show this help.

Rebase

<code>ri</code>	Perform an interactive rebase.
<code>u</code>	Same as <code>"ri"</code>
<code>rf</code>	Perform an autosquash rebase without editing the todo list.
<code>ru</code>	Perform an interactive rebase against <code>@{upstream}</code> .
<code>rp</code>	Perform an interactive rebase against <code>@{push}</code> .
<code>rr</code>	Continue the current rebase.
<code>rs</code>	Skip the current commit and continue the current rebase.
<code>ra</code>	Abort the current rebase.
<code>re</code>	Edit the current rebase todo list.
<code>rw</code>	Perform an interactive rebase with the commit under the cursor set to <code>reword</code> .
<code>rm</code>	Perform an interactive rebase with the commit under the cursor set to <code>edit</code> .

Rebase (cont)

<code>rd</code>	Perform an interactive rebase with the commit under the cursor set to <code>drop</code> .
<code>r<space pac -</code> <code>e></code>	Populate command line with <code>":Git rebase "</code> .
<code>r?</code>	Show this help.

Navigation (jump)

<code>(</code>	Jump to the previous file, hunk, or revision.
<code>)</code>	Jump to the next file, hunk, or revision.
<code>[c</code>	Jump to previous hunk, expanding inline diffs automatically.
<code>]c</code>	Jump to next hunk, expanding inline diffs automatically.
<code>[/</code>	Jump to previous file, collapsing inline diffs automatically.
<code>[m</code>	Same as <code>"[/"</code>
<code>] /</code>	Jump to next file, collapsing inline diffs automatically.
<code>]m</code>	Same as <code>"] /"</code>
<code>i</code>	Jump to the next file or hunk, expanding inline diffs automatically.
<code>[[</code>	Jump [count] sections backward.
<code>]]</code>	Jump [count] sections forward.
<code>[]</code>	Jump [count] section ends backward.
<code>] [</code>	Jump [count] section ends forward.
<code>*</code>	On the first column of a + or - diff line, search for the corresponding - or + line.
<code>#</code>	Same as <code>"**"</code> , but search backward.



By isaacnewton

cheatography.com/isaacnewton/

Published 3rd August, 2021.

Last updated 3rd August, 2021.

Page 2 of 3.

Sponsored by **ApolloPad.com**

Everyone has a novel in them. Finish Yours!

<https://apollopadd.com>

Diff

- dp Invoke `:Git diff` on the file under the cursor.
- dd Perform a `:Gdiff split` on the file under the cursor.
- dv Perform a `:Gvdif fsplit` on the file under the cursor.
- ds Perform a `:Ghdif fsplit` on the file under the cursor.
- dh Perform a `:Ghdif fsplit` on the file under the cursor.
- dq Close all but one diff buffer, and `:diff! off!` the last one.
- d? Show this help.

Commit

- cc Create a commit.
- ca Amend the last commit and edit the message.
- ce Amend the last commit without editing the message.
- cw Reword the last commit.
- cvc Create a commit with `-v`.
- cva Amend the last commit with `-v`.
- crc Revert the commit under the cursor.
- crn Revert the commit under the cursor in the index and work tree, but do not actually commit the changes.
- c? Show this help.

Commit (command-line)

- `c<s pac e` Populate command line with `":Git commit "`.
- `>`
- `cr< spa c` Populate command line with `":Git revert "`.
- `e>`
- `cm< spa c` Populate command line with `":Git merge "`.
- `e>`

Commit (create a...)

- cf Create a **fixup!** commit for the commit under the cursor.
- cF Create a **fixup!** commit for the commit under the cursor and immediately rebase it.
- cs Create a **squash!** commit for the commit under the cursor.
- cS Create a **squash!** commit for the commit under the cursor and immediately rebase it.
- cA Create a **squash!** commit for the commit under the cursor and edit the message.

Miscellaneous

- gq Close the status buffer.
- `.` Start a `:` command line with the file under the cursor prepopulated.
- g? Show help for `fugitive-maps`

Global

- `<c- r>< c-` On the command line, recall the path to the current `fugitive-object`.
- `g>`
- `["x] y<c -` Yank the path to the current `fugitive-object`.
- `g>`



By isaacnewton

cheatography.com/isaacnewton/

Published 3rd August, 2021.

Last updated 3rd August, 2021.

Page 3 of 3.

Sponsored by [ApolloPad.com](https://apollopad.com)

Everyone has a novel in them. Finish

Yours!

<https://apollopad.com>