

### Navigation (open)

<c	Open the file or <code>fugitive-object</code> under the cursor.
r>	
o	Open the file or <code>fugitive-object</code> under the cursor in a new split.
gO	Open the file or <code>fugitive-object</code> under the cursor in a new vertical split.
O	Open the file or <code>fugitive-object</code> under the cursor in a new tab.
p	Open the file or <code>fugitive-object</code> 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 <code>.git/info/exclude</code> in a split. Use a count to open <code>.gitignore</code> .

### 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 <code>.git/info/exclude</code> in a split and add the file under the cursor.
I	Invoke <code>:Git add --patch</code> or <code>reset --patch</code> on the file under the cursor.
P	On untracked files, this instead calls <code>:Git add --intent-to-add</code> .

### Checkout/branch

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



By **isaacnewton**

Published 3rd August, 2021.

Last updated 3rd August, 2021.

Page 1 of 3.

Sponsored by **Readable.com**

Measure your website readability!

<https://readable.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&lt;space&gt;ace&gt;</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&lt;-</code>	Populate command line with <code>:"Git rebase "</code> .
<code>spa- ce&gt;</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>	
<code>*</code>	On the first column of a <code>+ or -</code> diff line, search for the corresponding <code>- or +</code> line.
<code>#</code>	Same as <code>"**"</code> , but search backward.



By **isaacnewton**

Published 3rd August, 2021.  
Last updated 3rd August, 2021.  
Page 2 of 3.

Sponsored by **Readable.com**  
Measure your website readability!  
<https://readable.com>

### Diff

<code>dp</code>	Invoke <code>:Git diff</code> on the file under the cursor.
<code>dd</code>	Perform a <code>:Gdiffsplit</code> on the file under the cursor.
<code>dv</code>	Perform a <code>:Gvdiffsplit</code> on the file under the cursor.
<code>ds</code>	Perform a <code>:Ghdiffsplit</code> on the file under the cursor.
<code>dh</code>	Perform a <code>:Ghdiffsplit</code> on the file under the cursor.
<code>dq</code>	Close all but one diff buffer, and <code>:diffoff!</code> the last one.
<code>d?</code>	Show this help.

### Commit

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

### Commit (command-line)

<code>c&lt;space&gt;</code>	Populate command line with <code>":Git commit "</code> .
<code>cr&lt;space&gt;</code>	Populate command line with <code>":Git revert "</code> .
<code>cm&lt;space&gt;</code>	Populate command line with <code>":Git merge "</code> .

### Commit (create a...)

<code>cf</code>	Create a <code>fixup!</code> commit for the commit under the cursor.
<code>cF</code>	Create a <code>fixup!</code> commit for the commit under the cursor and immediately rebase it.
<code>cs</code>	Create a <code>squash!</code> commit for the commit under the cursor.
<code>cS</code>	Create a <code>squash!</code> commit for the commit under the cursor and immediately rebase it.
<code>cA</code>	Create a <code>squash!</code> commit for the commit under the cursor and edit the message.

### Miscellaneous

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

### Global

<code>&lt;c-r&gt;-</code>	On the command line, recall the path to the current <code>fugitive-object</code>
<code>&lt;c-g&gt;</code>	<code>fugitive-object</code>
<code>["x]y&lt;-</code>	Yank the path to the current <code>fugitive-object</code> .
<code>c-g&gt;</code>	



By **isaacnewton**

Published 3rd August, 2021.  
Last updated 3rd August, 2021.  
Page 3 of 3.

Sponsored by **Readable.com**  
Measure your website readability!  
<https://readable.com>