
git-nav-upstream(1)

Manual Page

NAME

`git-nav-upstream` - Checkout the upstream branch of the currently checked out branch.

SYNOPSIS

`git nav-upstream`

DESCRIPTION

git-nav-upstream takes the current branch, finds it's upstream branch, and checks that out.

EXAMPLE

```
$ git map-branches
origin/master
  cool_feature
    subfeature *
  fixit
  frozen_branch
$ git nav-upstream
Switched to branch
'cool_feature'
Your branch is ahead of
'origin/master' by 4 commits.
  (use "git push" to publish
your local commits)
$ git map-branches
origin/master
  cool_feature *
    subfeature
  fixit
  frozen_branch
$ git nav-upstream
Note: switching to
'origin/master'.

You are in 'detached HEAD'
state. You can look around,
make experimental <1>
changes and commit them, and
```

you can discard any commits you make in this state without impacting any branches by switching back to a branch.

If you want to create a new branch to retain commits you create, you may do so (now or later) by using `-c` with the switch command.

Example:

```
git switch -c <new-branch-name>
```

Or undo this operation with:

```
git switch -
```

Turn off this advice by setting config variable `advice.detachedHead` to false

HEAD is now at beec6f4 Make ReflectorImpl use mailboxes

```
$ git map-branches
```

```
origin/master
```

```
cool_feature
```

```
subfeature
```

```
fixit  
  frozen_branch
```

1. Note that being in a *detached HEAD* state is OK. You'll just want to create a [git-new-branch\(1\)](#) from here.

SUGGESTED ALIASES

Some common short-hand aliases. Feel free to add these to your `~/.gitconfig` file.

```
[alias]  
  git up = nav-upstream
```

SEE ALSO

[git-map-branches\(1\)](#), [git-nav-downstream\(1\)](#)