git-map-branches(1) Manual Page

NAME

git-map-branches - Helper script to display all local git branches with ‘upstream’ hierarchy in colorized terminal format.

SYNOPSIS

> *git map-branches*

DESCRIPTION

Git map-branches displays all local branches such that:

- Current branch is cyan.
    - The branch which will be modified with git-commit is denoted with an asterisk (`*`) after the name.
- Local branches are green.
- Remote branches are red (usually, the root of all other branches).
- *branch-heads* branches are blue.
- `{NO UPSTREAM}` is a special placeholder in magenta.

    - Branches which have this as their parent are usually misconfigured, and should be assigned a parent by checking out the branch and running git branch --set-upstream-to=<correct parent branch>.

> **Note** If multiple branches are on the same commit, they will all be cyan.

OPTIONS

--no-color
: Turn off colors.

-v
: Be more verbose. Pass once to show tracking info, twice for hash and review URL, thrice for review status.

EXAMPLE

Given the hypothetical demo repo in git-map(1)'s EXAMPLE section, and assuming that the `frozen_changes` branch was currently checked out, running *git map-branches* would result in an output like:

```
$ git map-branches
{NO_UPSTREAM}
  no_upstream
origin/master
  cool_feature
    subfeature
  fixit
    frozen_branch *
$ git map-branches -v
{NO_UPSTREAM}
```

```
  no_upstream
origin/master
  cool_feature         [ ahead 4
]
    subfeature         [ ahead 2 | behind
1 ]
  fixit                [ ahead 2
]
    frozen_branch *    [ ahead 3
]
```

SUGGESTED ALIASES

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

```
[alias]
  git bmap = map-branches
```

SEE ALSO

git-map(1)