

Git

Dynamic access to current branch name with the `current_branch` function

```
git pull origin $(current_branch)
grb publish $(current_branch) origin
```

Alias	Command
<i>g</i>	git
<i>ga</i>	git add
<i>gau</i>	git add -u
<i>gaa</i>	git add --all
<i>gapa</i>	git add --patch
<i>gb</i>	git branch
<i>gba</i>	git branch -a
<i>gbd</i>	git branch -d
<i>gbl</i>	git blame -b -w
<i>gbnm</i>	git branch --no-merged
<i>gbr</i>	git branch --remote
<i>gbs</i>	git bisect
<i>gbsb</i>	git bisect bad
<i>gbsg</i>	git bisect good
<i>gbsr</i>	git bisect reset
<i>gbss</i>	git bisect start
<i>gc</i>	git commit -v
<i>gc!</i>	git commit -v --amend
<i>gca</i>	git commit -v -a
<i>gca!</i>	git commit -v -a --amend
<i>gcan!</i>	git commit -v -a --no-edit --amend
<i>gcans!</i>	git commit -v -a -s --no-edit --amend
<i>gcam</i>	git commit -a -m
<i>gcsn</i>	git commit -s -m
<i>gcb</i>	git checkout -b
<i>gcf</i>	git config --list
<i>gcl</i>	git clone --recursive
<i>gclean</i>	git clean -id
<i>gpristine</i>	git reset --hard && git clean -dfx
<i>gcm</i>	git checkout master
<i>gcd</i>	git checkout develop
<i>gcmsg</i>	git commit -m
<i>gco</i>	git checkout
<i>gcount</i>	git shortlog -sn

Alias	Command
<i>gcp</i>	git cherry-pick
<i>gcpa</i>	git cherry-pick --abort
<i>gcpc</i>	git cherry-pick --continue
<i>gcs</i>	git commit -S
<i>gd</i>	git diff
<i>gdca</i>	git diff --cached
<i>gdct</i>	git describe --tags `git rev-list --tags --max-count=1`
<i>gds</i>	git diff --staged
<i>gdt</i>	git diff-tree --no-commit-id --name-only -r
<i>gdw</i>	git diff --word-diff
<i>gf</i>	git fetch
<i>gfa</i>	git fetch --all --prune
<i>gfo</i>	git fetch origin
<i>gg</i>	git gui citool
<i>gga</i>	git gui citool --amend
<i>ggpnp</i>	git pull origin \$(current_branch) && git push origin \$(current_branch)
<i>ggpull</i>	git pull origin \$(current_branch)
<i>ggl</i>	git pull origin \$(current_branch)
<i>ggpur</i>	git pull --rebase origin \$(current_branch)
<i>glum</i>	git pull upstream master
<i>ggpush</i>	git push origin \$(current_branch)
<i>ggp</i>	git push origin \$(current_branch)
<i>ggfl</i>	git push --force-with-lease origin <your_argument>/\$(current_branch)
<i>ggsup</i>	git branch --set-upstream-to=origin/\$(current_branch)
<i>gpsup</i>	git push --set-upstream origin \$(current_branch)
<i>gignore</i>	git update-index --assume-unchanged
<i>gignored</i>	git ls-files -v grep "^ .lower: "
<i>git-svn-dcommit-push</i>	git svn dcommit && git push github master:svntrunk
<i>gk</i>	gitk --all --branches
<i>gl</i>	git pull
<i>glg</i>	git log --stat --max-count = 10
<i>glgg</i>	git log --graph --max-count = 10
<i>glgga</i>	git log --graph --decorate --all
<i>glo</i>	git log --oneline --decorate --color
<i>glog</i>	git log --oneline --decorate --color --graph
<i>glp</i>	_git_log_prettily (git log --pretty=\$1)
<i>gm</i>	git merge
<i>gma</i>	git merge --abort
<i>gmt</i>	git mergetool --no-prompt

Alias	Command
<i>gp</i>	git push
<i>gpoat</i>	git push origin --all && git push origin --tags
<i>gr</i>	git remote
<i>grb</i>	git rebase
<i>grba</i>	git rebase --abort
<i>grbc</i>	git rebase --continue
<i>grbd</i>	git rebase develop
<i>grbm</i>	git rebase master
<i>grbs</i>	git rebase --skip
<i>grbi</i>	git rebase -i
<i>grh</i>	git reset HEAD
<i>grhh</i>	git reset HEAD --hard
<i>grmv</i>	git remote rename
<i>grrm</i>	git remote remove
<i>grs</i>	git restore
<i>grset</i>	git remote set-url
<i>grt</i>	cd \$(git rev-parse --show-toplevel echo ".")
<i>grup</i>	git remote update
<i>grv</i>	git remote -v
<i>gsd</i>	git svn dcommit
<i>gsps</i>	git show --pretty = short --show-signature
<i>gsr</i>	git svn rebase
<i>gss</i>	git status -s
<i>gst</i>	git status
<i>gsta</i>	git stash save
<i>gstaa</i>	git stash apply
<i>gstd</i>	git stash drop
<i>gstl</i>	git stash list
<i>gstp</i>	git stash pop
<i>gsts</i>	git stash show --text
<i>gsu</i>	git submodule update
<i>gts</i>	git tag -s
<i>gunignore</i>	git update-index --no-assume-unchanged
<i>gunwip</i>	git log -n 1 grep -q -c "--wip--" && git reset HEAD~1
<i>gup</i>	git pull --rebase
<i>gvt</i>	git verify-tag
<i>gwhch</i>	git whatchanged -p --abbrev-commit --pretty = medium
<i>gwip</i>	git add -A; git ls-files --deleted -z xargs -r0 git rm; git commit -m "--wip--"

You also find these commands in Dash as a Cheat-sheet.