```
# Print repo summary
bash-3.2# g
[No repos to report on]
=======================


# This is the example folder/repo tree we're working with. my-
repo, test and some-repo are Git repos.
bash-3.2# tree /tmp/example/ -L 2
/tmp/example/
├── repos-1
│   ├── my-repo
│   └── test
└── repos-2
    └── some-repo

5 directories, 0 files


# Add a folder and a repo as 'bookmarks'
bash-3.2# g -a /tmp/example/repos-1 /tmp/example/repos-2/some-
repo/


# Print repo summary. Green = working copy changes, red =
behind/ahead of upstream
bash-3.2# g
1) /tmp/example/repos-1 2) /tmp/example/repos-2/some-repo
=========================================================
/tmp/example/repos-1/my-repo (1) *1.5.4 another-branch
/tmp/example/repos-1/test (2) *master
/tmp/example/repos-2/some-repo (3) *test-branch


# Only show repos in /tmp/example/repos-2
bash-3.2# g /tmp/example/repos-2
1) /tmp/example/repos-1 2) /tmp/example/repos-2/some-repo
=========================================================
/tmp/example/repos-2/some-repo (1) *test-branch


# Only show repos in /tmp/example/repos-2/some-repo -
references number 2) on the top line of `g`'s previous output
bash-3.2# g 2
1) /tmp/example/repos-1 2) /tmp/example/repos-2/some-repo
=========================================================
/tmp/example/repos-2/some-repo (1) *test-branch


# The 'test' repo has upstream changes
bash-3.2# g
1) /tmp/example/repos-1 2) /tmp/example/repos-2/some-repo
=========================================================
/tmp/example/repos-1/my-repo (1) *1.5.4 another-branch
/tmp/example/repos-1/test (2) *master
/tmp/example/repos-2/some-repo (3) *test-branch


# Quick `git pull` on /tmp/example/repos-1/test (references
(1) in the list in `g`'s previous output)
bash-3.2# gp 2
Updating d18feb6..51557cf
Fast-forward
 README.md | 1 +
 1 file changed, 1 insertion(+)


# `g` only shows repos with working copy changes, branches not
named 'master', or branches that have upstream changes. We
`git pull`ed the 'test' repo, so now it no longer shows:
bash-3.2# g
1) /tmp/example/repos-1 2) /tmp/example/repos-2/some-repo
=========================================================
/tmp/example/repos-1/my-repo (1) *1.5.4 another-branch
/tmp/example/repos-2/some-repo (2) *test-branch


# `ga` behaves exactly the same as `g` except it will include
repos regardless of their state
bash-3.2# ga
1) /tmp/example/repos-1 2) /tmp/example/repos-2/some-repo
=========================================================
/tmp/example/repos-1/my-repo (1) *1.5.4 another-branch
/tmp/example/repos-1/test (2) *master
/tmp/example/repos-2/some-repo (3) *test-branch


# Deleting 'bookmarks'
bash-3.2# g -d /tmp/example/repos-1 /tmp/example/repos-2/some-
repo/


# Empty again now
bash-3.2# ga
[No repos to report on]
=======================


# Adding them back
bash-3.2# g -a /tmp/example/repos-1 /tmp/example/repos-2/some-
repo/


# And they're back
bash-3.2# g
1) /tmp/example/repos-1 2) /tmp/example/repos-2/some-repo
=========================================================
/tmp/example/repos-1/my-repo (1) *1.5.4 another-branch
/tmp/example/repos-2/some-repo (2) *test-branch


# Currently in /tmp
bash-3.2# pwd
/tmp


# cd into /tmp/example/repos-2/some-repo (references (2) in
the list in `g`/`ga`'s previous output)
bash-3.2# gc 2


# Now in /tmp/example/repos-2/some-repo
bash-3.2# pwd
/tmp/example/repos-2/some-repo


# Open . in SourceTree
bash-3.2# gs .


# Open /tmp/example/repos-2/some-repo in SourceTree
bash-3.2# gs 2


# Start the daemon that fetches each of your 'bookmarks',
fetching a single one every 30 seconds. No stop function
(yet), just kill it manually if you want to stop it. You don't
have to restart it if you change your bookmarks.
bash-3.2# gfbd
```