

```
## git log --name-status|--name-only
```

```
# show file names and statuses
```

```
$ git log -1 --name-status
```

```
commit 714a14ef23acb056e3ad0b03c2fd515e3c308603 (HEAD -> dev, github/dev)
```

```
...
```

```
M README.md
```

```
A rm.Extensions/BitSet.cs
```

```
M rm.Extensions/Properties/AssemblyInfo.cs
```

```
M rm.Extensions/rm.Extensions.csproj
```

```
A rm.ExtensionsTest/BitSetTest.cs
```

```
M rm.ExtensionsTest/rm.ExtensionsTest.csproj
```

```
# show file names
```

```
$ git difftool head~ --name-only
```

```
## grep, log grep, ls-files
```

```
# greps in working directory only
```

```
$ git grep "woohoo"
```

```
$ git grep -i "woohoo"          # ignore case
```

```
$ git grep "woohoo" -- *.gtt    # in .gtt files only
```

```
# greps in entire repo (slow)
```

```
$ git log -G "woohoo"
```

```
# greps in log messages only
```

```
$ git log --grep "woohoo"
```

```
# greps in tip of all branches (not just current)
```

```
$ git branch -a \
```

```
    | cut -c3- | xargs git grep "woohoo"
```

```
## git add -i|-p
```

```
$ git status  
(changes)
```

```
$ git add -i
```

```
# *** Commands ***
```

```
# 1: status      2: update      3: revert      4: add untracked
```

```
# 5: patch      6: diff        7: quit        8: help
```

```
# What now>
```

```
(some work to add/remove/patch to index)
```

```
$ git commit
```

```
# --patch for reset, checkout also
```

```
$ git add -p
```

```
$ git reset -p
```

```
$ git checkout -p
```

```
## rev-list
```

```
# get total commit count
```

```
$ git rev-list --all --count
```

```
# get total commit count for master
```

```
$ git rev-list master --count
```

```
#   ---o---o---o master
```

```
#
```

```
  \
```

```
#      o---o topic
```

```
# get left-right commit count
```

```
$ git rev-list topic...master --left-right --count
```

```
2    1
```

```
# get left-right commit
```

```
$ git rev-list topic...master --left-right
```

```
# Where did I branched from?
```

```
# alias: git bfrom
```

```
$ git bfrom master@{u} next@{u}
```

```
## template dir and git hooks
```

```
# template for when you init a repo  
# also works for existing repo  
# .gitconfig
```

```
[init]
```

```
    templateDir = ~/.git-templates
```

```
# copy contents of template dir to .git/ dir  
$ git init
```

```
# to apply git hooks to all repos
```

```
[core]
```

```
    hooksPath = ~/.git-hooks
```

```
# to skip the commit hooks
```

```
$ git commit -n
```