

## ## piping

```
# There are more things in git and piping, Horatio,  
# than are dreamt of in your philosophy.
```

```
#           - echo "WS" \  
#           | sed 's/WS/R/' \  
#           | awk '{print $0"M}'  
#
```

```
# text tools that support piping
```

```
sed streaming editor
```

```
awk text processing tool
```

```
# files with different statuses
```

```
$ git status -s | grep "^UU"
```

```
$ git status -s | grep "^ M"
```

```
# o is shortcut for my text editor, scite
```

```
# open conflicted files
```

```
$ git status -s | grep "^UU" | cut -c3- | xargs o
```

```
# open files with BitSet in name
```

```
$ git ls-files *BitSet* | xargs o
```

```
# alias hidden: show hidden files
```

```
$ git ls-files -v | grep '^h' | cut -c3-
```

```
$ git hidden
```

```
# hide and unhide are aliases
```

```
$ git hide
```

```
$ git unhide
```

```
# hide modified files and then unhide them
```

```
$ git status -s | grep "^ M" | xargs git hide
```

```
$ git hidden | xargs git unhide
```

```
# open files with space in them
```

```
$ git ls-files "<spc>"
```

```
    | sed 's/^/"/' | sed 's/$/"/'
```

```
    | xargs o
```

```
$ git ls-files *<spc>*
```

```
    | awk '{print "\""$0"\""}'
```

```
    | xargs o
```

```
$ git ls-files -z *<spc>*
```

```
    | xargs -0 o
```

```
# 'wrap' wraps all paths in quotes to handle spaces
```

```
$ git ls-files "<spc>" | wrap | xargs o
```

```
# 'wrap' is piping-friendly
```

```
$ o ~/bin/wrap
```

```
# open files with "next"
```

```
$ git grep "next" | cut -d':' -f1 | wrap | xargs o
```

```
# delete merged branches except few
```

```
$ git branch --merged  
    | egrep -v "(\*|master|pu|maint|next)"  
    | egrep "cr/"  
    | xargs git branch -d
```

```
# delete oldest 3 cr/* branches except current
```

```
$ git branch --sort=committerdate  
    | egrep -v "\*"  
    | egrep "cr/"  
    | head -3  
    | xargs git b -d
```

```
# top 2 items
```

```
$ ... | head -2
```

```
# last 2 items
```

```
$ ... | tail -2
```

```
# 2nd line item
```

```
$ ... | sed -n '2 p'
```

## ## advanced aliases

```
$ git coe <partial branch>
```

### # multiple branches

```
$ git coe e
fatal: multiple branches matched
  master
  dev
```

### # fuzzy checkout to branch

```
$ git coe z
Switched to branch 'temp/lazy'
```

### # custom commands

```
$ git l
l = git log --pretty=two
```

### # pretty formats

```
two
one
```

### # aliases running as shell scripts run from /

```
l = !"git lg2"
# this will still run from /
$ cd to/some/dir && git l .
```

```
## reflog
```

```
# <commit> (refs) HEAD@{<i>} summary of operation  
#      where <commit> is after the operation
```

```
# ref log
```

```
$ git reflog -3
```

```
# 87dce5f (HEAD -> dev, master) HEAD@{0}: reset: moving to master  
# 0733641 HEAD@{1}: rebase -i (finish): returning to refs/heads/dev  
# 0733641 HEAD@{2}: rebase -i (pick): Adding BitSet implementation.
```

```
$ git reset --hard HEAD@{1}
```

## bisect

```
$ git bisect start
# head is bad
$ git bisect bad
# some commit in past is good
$ git bisect good <commit>
(bisecting)
$ git bisect good|bad
(bisecting)
(08b9bd4 is the first bad commit)
# exit bisect state
$ git bisect reset
```

```
## shortlog
```

```
#
```

```
# The owls stats are not what they seem.
```

```
#
```

```
# shortlog with summary, sort, email
```

```
$ git shortlog -sne
```

```
# .mailmap
```

```
- consolidate name/emails
```

```
# What to do?
```

```
- place at root OR
```

```
  outside of repo with mailmap.file path in config
```

```
- log.mailmap true
```