

File Management

```
Bulk rename files in a folder:
for i in FILE*; do mv $i
${i/FILE1/FILE2}; done

Add a prefix to all files:
for f in*;do mv "$f""PRE_$f";done

Delete everything except matching file/folder
ls | grep -v '{file_or_folder}' |
xargs rm -rf
```

Misc

```
IP address
ifconfig | grep "inet " | grep -v
127.0.0.1 | cut -d\ -f2

List Daemons and orphaned processes
ps -ef | awk '$3 == 1'
```

Call Function From Shell

```
$ cat test.sh
testA() {
    echo "TEST A $1";
}
# call arguments verbatim:
$@
$ bash test.sh
$ bash test.sh testA
TEST A
```

Search

```
restrict search by file type
grep --include=.js 'some string'
[path]/

find by filename
find [path] -name [filename]

restrict shell to basic letters (ASCII-7)
export LC_CTYPE=C && export LANG=C

find and replace string in multiple files
grep -rl 'wl:' app | xargs sed -i
".js" -e 's/wl:/working-list:/g'

find occurrences of 'word-' and sort them
grep -roh 'word[a-z|0-9]*' ./ |
sort | uniq

recursively remove file
find . -name "*.swo" -type f -
delete
```

Git

```
tag a version
git tag -a v1.4 -m 'my version 1.4'

remove untracked files and directories
git clean -f -d

delete remote branch
git branch -d {the_local_branch}

git push origin :
{the_remote_branch}

save password (OSX only)
git config --global
credential.helper osxkeychain
erase password - use Keychain Access.app or
...
git credential-osxkeychain erase
host=github.com
protocol=https

revert an individual file in working directory
git checkout HEAD <file>

in staging area
git reset HEAD <file>

reset everything
git fetch origin
git reset --hard origin/master

undo last commit in working branch
git reset --soft HEAD~1

log deletes
git log --diff-filter=D --summary

search for <string> in commit messages
git log -g --grep=<string>

checkout a tagged version
git checkout tags/0.0.1

log all branches
git log --source --all
```

Heroku

```
Adding to existing git project
heroku create
heroku apps:rename [app-name]
git push heroku master
heroku open

DNS
a-f.dns.zerigo.net
http://stackoverflow.com/questions/17586017/when-do-i-find-nameservers-of-my-heroku-app/29320779#29320779
```

SVN

```
add all unrevised files
svn add --force --depth infinity
{path}

if you want to see what the last commit did:
svn diff -r PREV:COMMITTED {path}

last change revision
svn info {path}

Edit ignore props for current dir.
svn propedit svn:ignore .

Patching:
svn diff > ~/fix_ugly_bug.diff
patch -p0 -i ~/fix_ugly_bug.diff
revert working copy
svn revert --recursive .
remove uncommitted files
svn st | grep '^?' | awk '{print
$2}' | xargs rm -rf

create diffs for 224453 and 224462 (e.g. by
svn diff -r 224452:224453 > diff1.patch).
check out 224446 (svn up -r224446)
apply the diffs (e.g. patch -p0 -i diff1.patch)
create a diff of that against 224445 (svn diff -r
224445 > diff2.patch)
```

