Git Tricks

Automatic deployment

In some cases (like managing websites in git) it's usefull to have remote working directory (production checkout) updated on each push. So you just do "git push" and you can see changes imediately on the production website. Also note that you might want to have two remote repositories (one for development and another for production). On "client" side you will later add two remotes, one called "origin" for development and another you can name like "production" or whatever else… sou you can use "git push" for everyday development versioning and "git push production" to deploy.

All you have to do on production repository is add post-receive hook to .git directory. Note it has to be chmod +x.

repository.git/hooks/post-receive

```sh
#!/bin/sh
git --work-tree=/var/www/www.example.org checkout -f
```

On older git versions you may try this one:

repository.git/hooks/post-receive

```sh
#!/bin/sh
GIT_WORK_TREE=/var/www/www.example.org git checkout -f
```

Rebasing history

Fix encoding

This is quite nice way to cleanup code (eg. change character encoding, remove windows newlines, reindent, etc…) through whole history of commits. Well… you can do this only for latest revision by single commit, but this can be usefull when you work with history a lot…

```
git filter-branch -f --tree-filter 'find | while read file; do dos2unix -b --d2u "$file"; done' master
```

Search and destroy files

This can delete file(s) from all revisions

```
git filter-branch -f --tree-filter 'rm -f dir/unwanted_file.txt' master
```

Search and destroy strings in files

You can use this one eg. if you want to delete a password that have been commited long time ago…

```
git filter-branch -f --tree-filter '/tmp/clean.sh' master
```

/tmp/clean.sh

```
clean() {
    [ -f "$1" ] && sed -i 's/SEARCH/REPLACE/g' "$1"
}

clean nodes.php
clean wwwroot/nodes.php
true
```

Remove trailing blanks

```
git filter-branch -f --tree-filter '/tmp/clean.sh' master
```

/tmp/clean.sh

```
clean() {
    [ -f "$1" ] && sed -i 's/[ \t]*$//' "$1"
}
find | while read file; do clean "$file"; done
true
```

Re-indent

```
vim -esc "normal gg=G" -c "wq" -e "file.php"
```