Cool Git Tricks

(That I learn when things go badly.)

@cczona

Carina C. Zona

@cczona
www.cczona.com

I write a detailed commit message.

(About some other changeset.)

```
$ git commit

# Changes to be committed:
#...
#modified:   foo.txt
```

```
$ git commit -v

# Changes to be committed:
#...
# modified:   foo.txt
#...
-old line
+new line
```

I create long branch names.

(Then have to type all that.)

git/contrib/completion/
git-completion.bash

```
cp git-completion.bash ~/.git-completion.sh

echo source ~/.git-completion.sh >> ~/.bashrc
```

```
$ git checkout <tab><tab>

0.1-mvp-release          marketing-content-dump
2.0-public-release       master
authentication-basic     optimize-for-webkit
button-to-pay-me         redesign-for-fall-2012
live-customer-chat       ui-performance-enhance
```

```
$ git checkout b<tab>
Switched to branch 'button-to-pay-me'
```

I use ‘git status’

(A lot.)

@cczona

git/contrib/completion/
git-prompt.sh

```
cp git-prompt.sh ~/.git-prompt.sh

echo source ~/.git-completion.sh >> ~/.bashrc
```

```
#master|MERGE $
#master|REBASE-i $
#master + $ # tracked
#master $ $ # stash
#master> $ # upstream
```

```
3 pending on foundation#master
~/repo/foundation/css $ git add .
3 pending on foundation#master
~/repo/foundation/css $ git commit
...
0 pending on foundation#master
~/repo/foundation/css $ touch new.txt
1 pending on foundation#master
~/repo/foundation/css $
```

I pathologically avoid
`git add .`

(Now.)

@cczona

```
$ git grep -e stupid \
      --and -e boss
index.html: The boss is a stupidhead
```

```
$ git grep -e promotion \
    --or -e bargain \
    --not -e (coupon|discount)
```

```
# working area tracked
$ git grep


# working area tracked & untracked
$ git grep --untracked
```

```
# working area all (ignore .gitignore)
$ git grep --untracked --no-exclude-standard


# staged
$ git grep --cached
```

```
# arbitrary commit
$ git grep “2007” HEAD^

# range
$ git grep  “2007” c0c1e80..141a16a

# every commit
$ git grep “2007” $(git rev-list --all)
```

I don’t necessarily detect a problem immediately.

(Time machine needed.)

```
$ git bisect start
$ git bisect bad HEAD
$ git bisect good v2.0
Bisecting: 44 revisions
left to test after this
(roughly 6 steps)
...
b047b0 is first bad commit
```

I sometimes need to pretend a commit *never happened.*

(Like, even rebase ain't gonna cut it.)

@cczona

```
$ git checkout HEAD~10

...

$ ls

about.html

contact.html

uber-secret-never-commit-this.txt
```

(leaving out many crucial details...)

```
filter-branch # remove from local
commits...

reflog expire # make artifacts
eligible for garbage collection

gc # do the garbage collection

push --force  # rewrite remote
history
```

Git can be a pain.

It also has cool tricks to take pain away.

(Whew.)

Questions?

Carina C. Zona
www.cczona.com
@cczona

Learn More

- http://git-scm.com/book
- http://blog.bitfluent.com/post/27983389/git-utilities-you-cant-live-without
- https://help.github.com/articles/remove-sensitive-data

@cczona