

Practical Git tips and tricks

 [tipishev/git_workshop](https://github.com/tipishev/git_workshop)



Motivation

- refresh memory
- rekindle interest in Git
- simplify workflow

Motivation

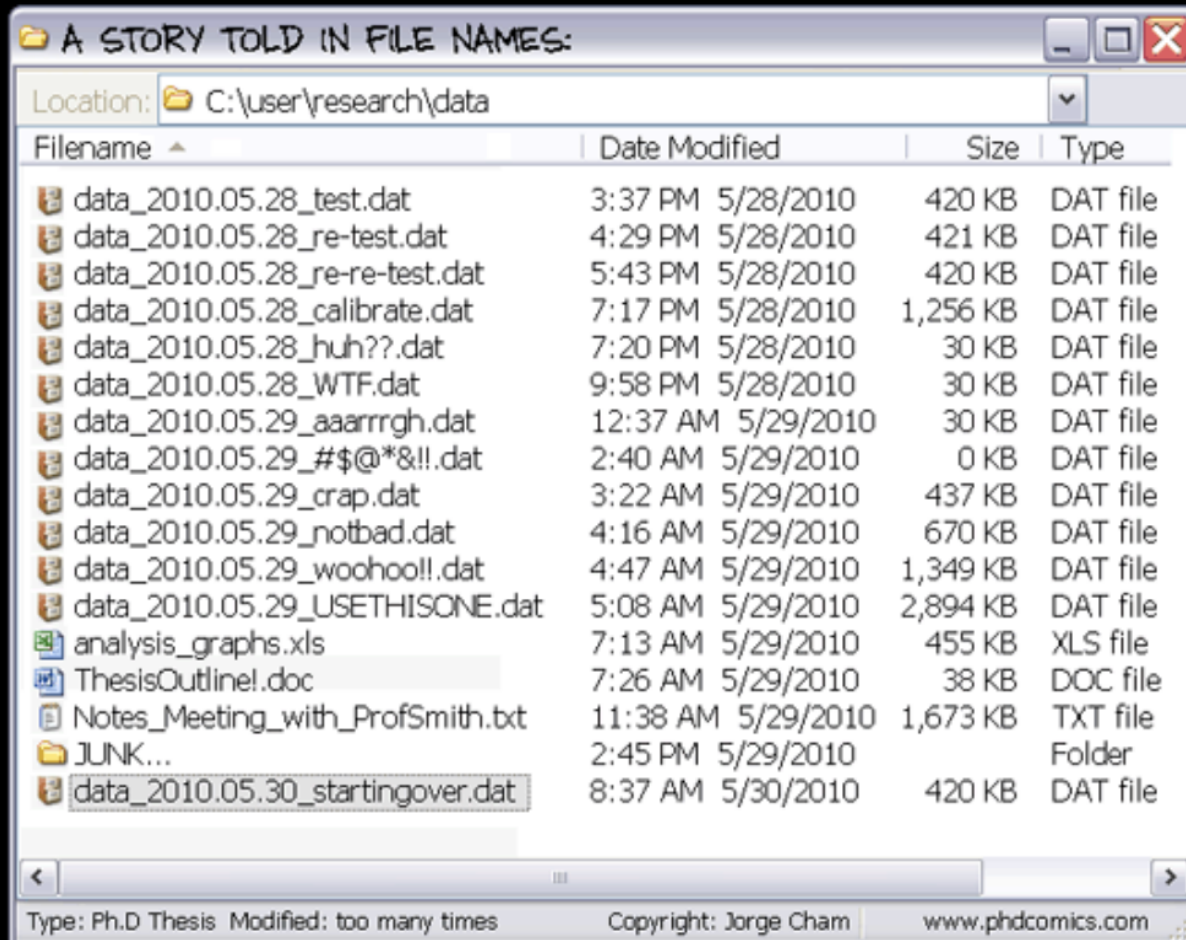
- refresh memory
- rekindle interest in Git
- simplify workflow

* myself included

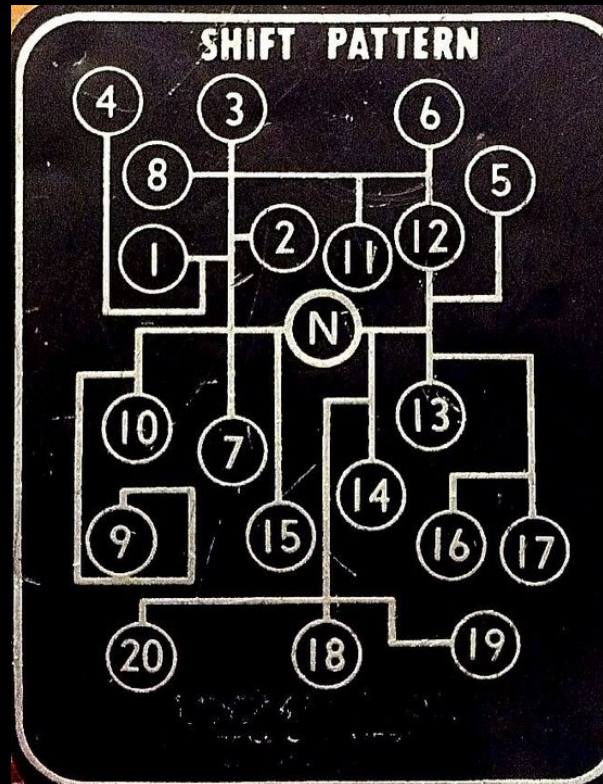
Who I am and what I do

- Python/Django backender
- Sportamore
 - 20 devs in 3 teams
 - Github + Teamcity
 - feature → dev → master

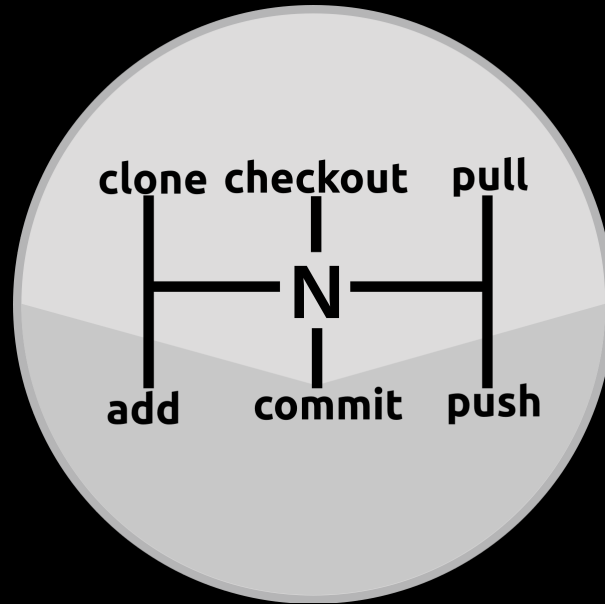
Naïve Version Control



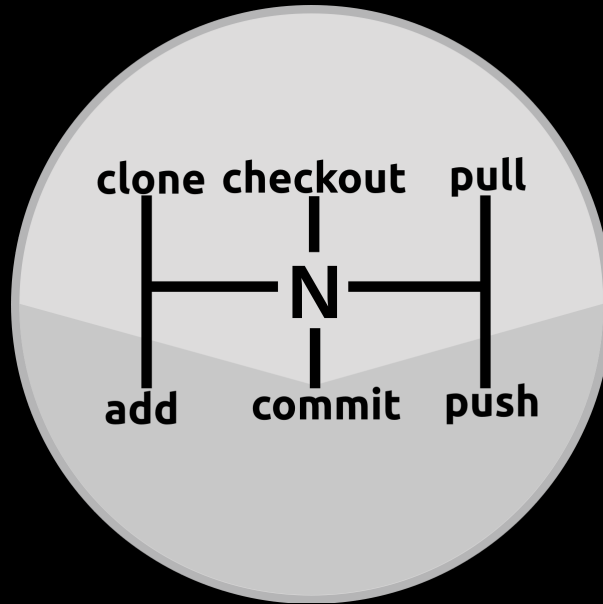
Do we need this complexity?



Maybe Not...



Maybe Not...



git **help** everyday

Le.git (git-legit.org)

Git Workflow for Humans



Star

5,322

Feature branch workflows are dead simple.

```
$ git switch <branch>
# Switches to branch. Stashes and restores unstaged changes.

$ git sync
# Synchronizes current branch. Auto-merge/rebase, un/stash.

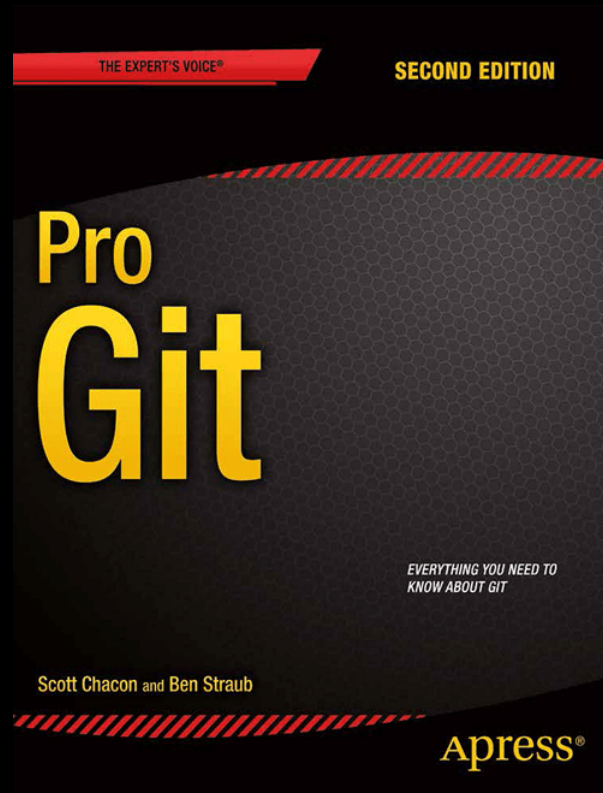
$ git publish <branch>
# Publishes branch to remote server.

$ git unpublish <branch>
# Removes branch from remote server.

$ git branches
# Nice & pretty list of branches + publication status.

$ git undo [--hard]
# Removes the last commit from history.
```

Pro-git



Configuration

- `--system` for all users
- `--global` just for you
- `--local` for the project

Configuration

- `--system` for all users
- `--global` just for you
- `--local` for the project
- ~~`--emperor` FOR THE EMPRAH!~~

Examples

- Aliases

Aside: Sexy Bash Prompt

```
todd at Euclid in ~/non-git
$ cd /tmp/git/synced
todd at Euclid in /tmp/git/synced on master
$ cd /tmp/git/dirty-synced
todd at Euclid in /tmp/git/dirty-synced on master*
$ cd /tmp/git/unpushed
todd at Euclid in /tmp/git/unpushed on master△
$ cd /tmp/git/dirty-unpushed
todd at Euclid in /tmp/git/dirty-unpushed on master▲
$ cd /tmp/git/unpulled
todd at Euclid in /tmp/git/unpulled on master▽
$ cd /tmp/git/dirty-unpulled
todd at Euclid in /tmp/git/dirty-unpulled on master▼
$ cd /tmp/git/unpushed-unpulled
todd at Euclid in /tmp/git/unpushed-unpulled on master○
$ cd /tmp/git/dirty-unpushed-unpulled
todd at Euclid in /tmp/git/dirty-unpushed-unpulled on master●
$
```

Examples

- Aliases
- Autocorrect

Examples

- Aliases
- Autocorrect
- Colors

Other Configuration Tips

- Always rebase on pull

```
git config --global pull.rebase true
```

- See all configs values*

```
git config --list
```

Other Configuration Tips

- Always rebase on pull

```
git config --global pull.rebase true
```

- See all configs values*

```
git config --list
```

* you may be surprised

Git Hooks

- Enforce rules
- Trigger actions

Git Hooks

- Enforce rules
 - Trigger actions
-

- Remote - enforced
- Local - optional*

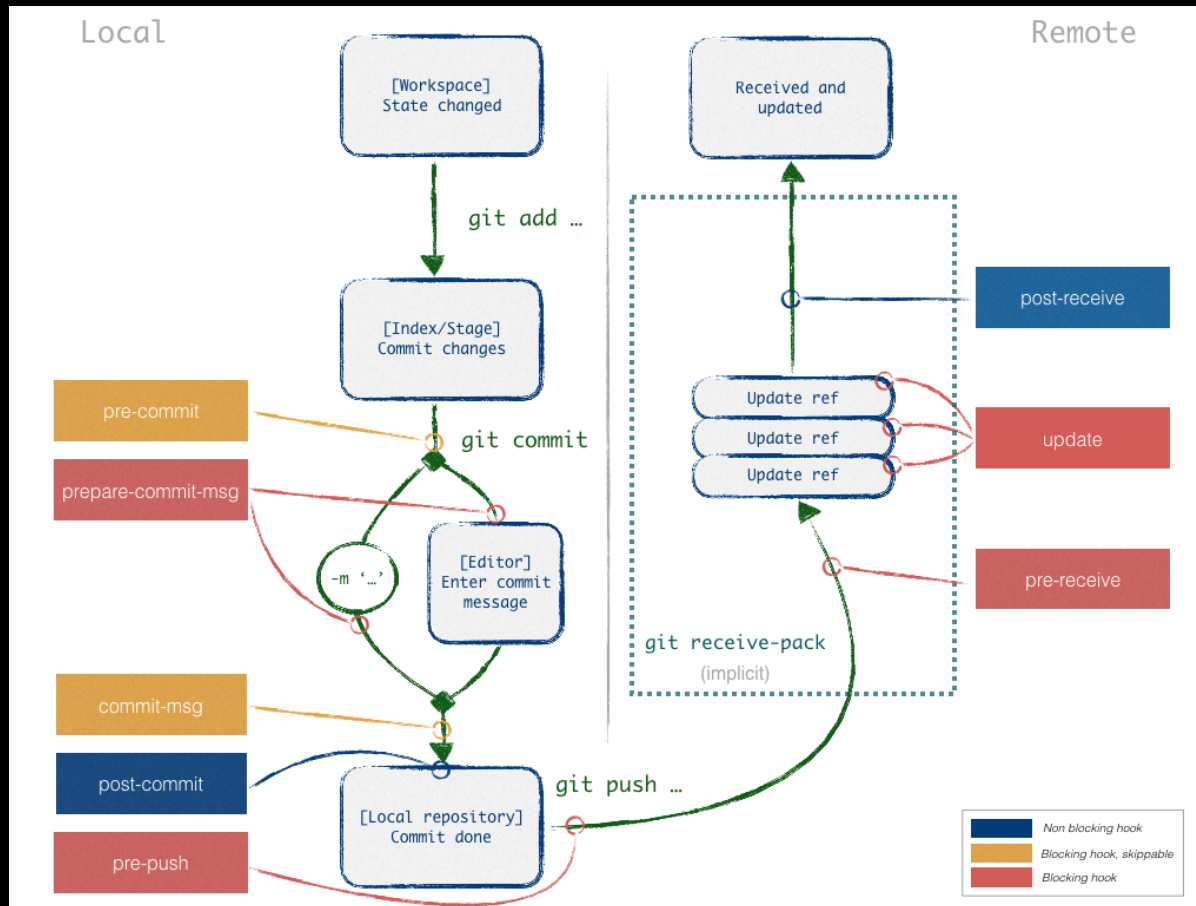
Git Hooks

- Enforce rules
 - Trigger actions
-

- Remote - enforced
- Local - optional*

* Can be bypassed with `--no-verify`

Git Hooks



Archeology

git-grep

- faster than `grep`
- useful flags
 - `--count`
 - `--line-number`
 - `--show-function`

Blame

```
git blame
```

- the starting point
- likely to blame refactorer
- ignore-flags
 - `-w` white space
 - `-M` moving within file
 - `-C` moving between files

Pickaxe

```
git log -S {string}
```

The ultimate code-digging tool
across the whole history

Realism vs Perfection

- aka Sausage Making

Realism vs Perfection

- aka Sausage Making
- Realism
 - better for historians
- Perfection
 - easier to read

Tools

- separate commits (-p)
- separate branches
- interactive rebasing

Tools

- separate commits (-p)
- separate branches
- interactive rebasing

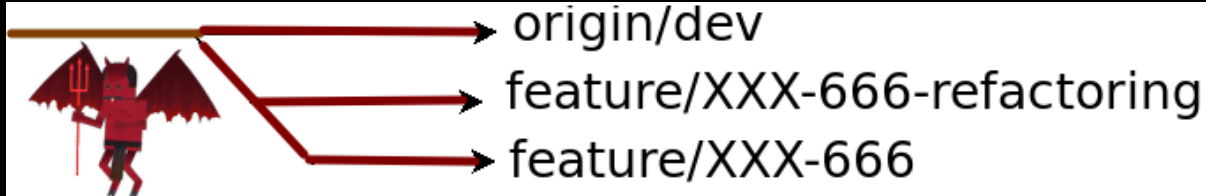
> assuming not pushed yet

DRIVE-BY REFACTORING



Refactoring Branch

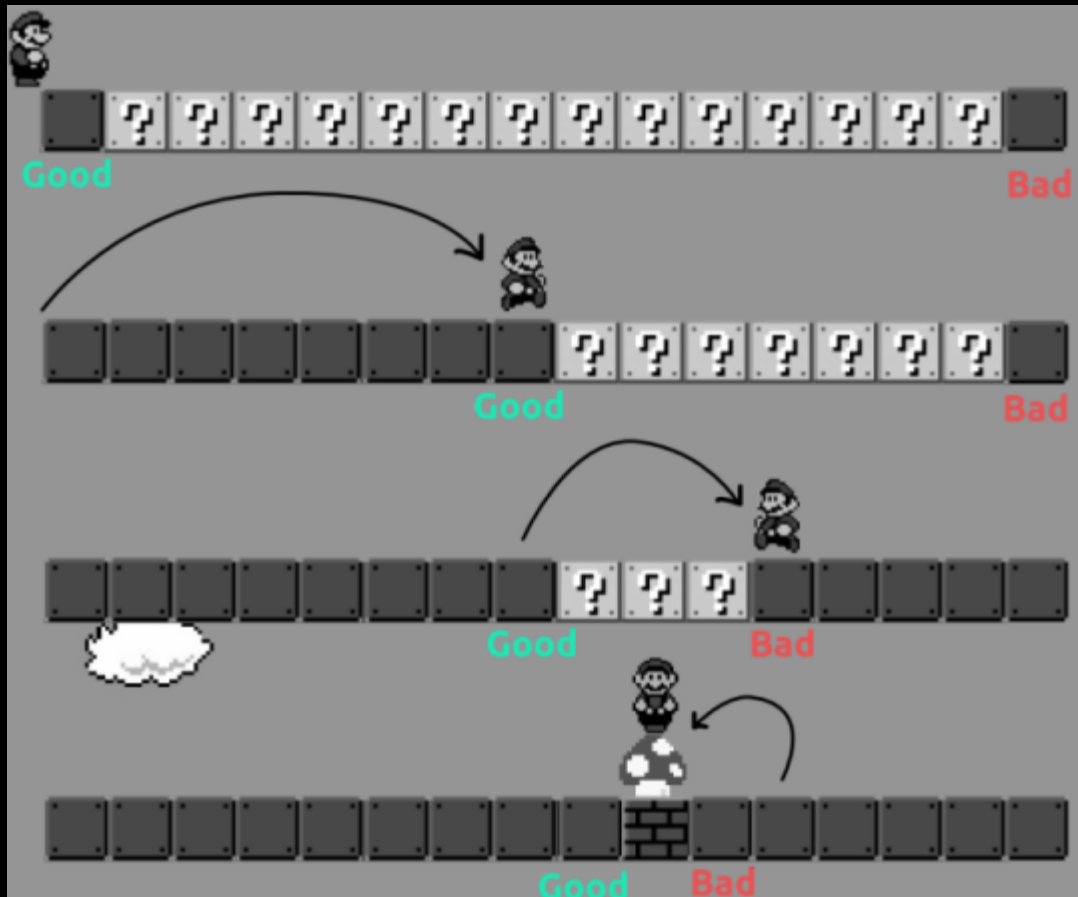
- `feature/XXX-666-refactoring` -> `dev`
- `feature/XXX-666` -> `feature/XXX-666-refactoring`



Interactive Rebasing

- `git commit --amend` on steroids
- a range of unpushed commits
- pretend to be perfect

Bisect



Bisect Ideas

- (old, new) == (good, bad)
- Mark >1 good commit
- Bisect broken test on CI

Git Attributes

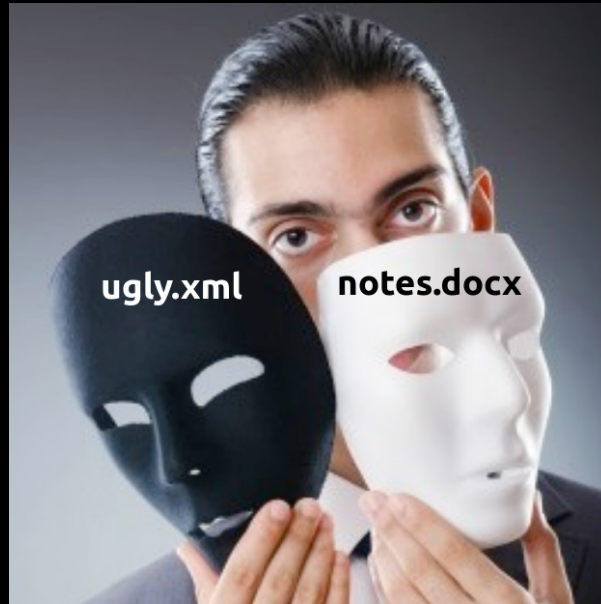
- tell Git how to treat files
- Like `gitconfig`:
 - `system`
 - `global`
 - `local`

Git Worldview

- Text Files
- Binary Files

```
diff --git a/binary/notes.docx b/binary/notes.docx
index 638f1d4..9934c87 100644
Binary files a/binary/notes.docx and b/binary/notes.docx differ
```

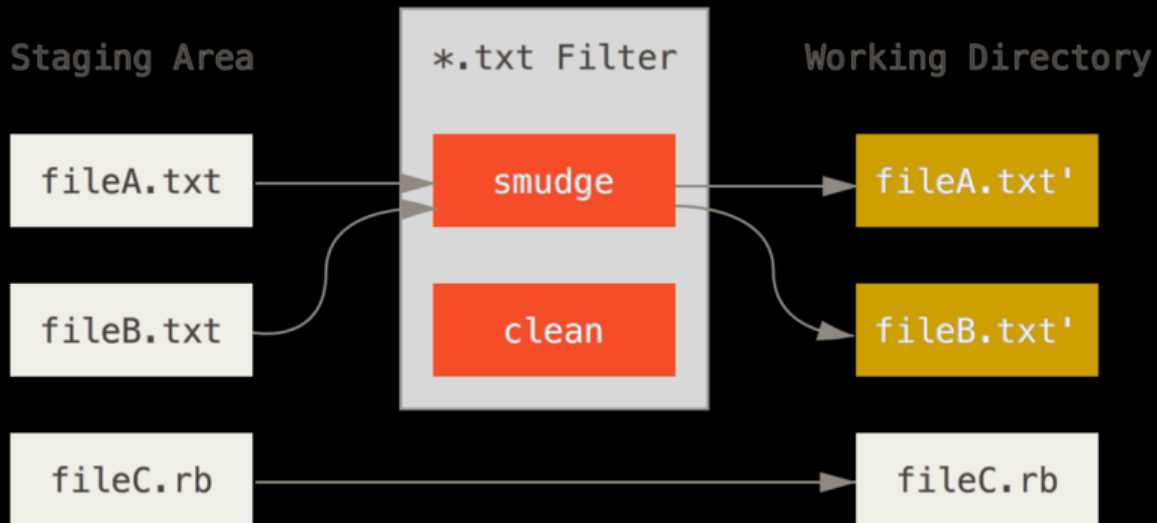
Binary Deception



- some **text** is as good as binary
- some **binaries** are just text

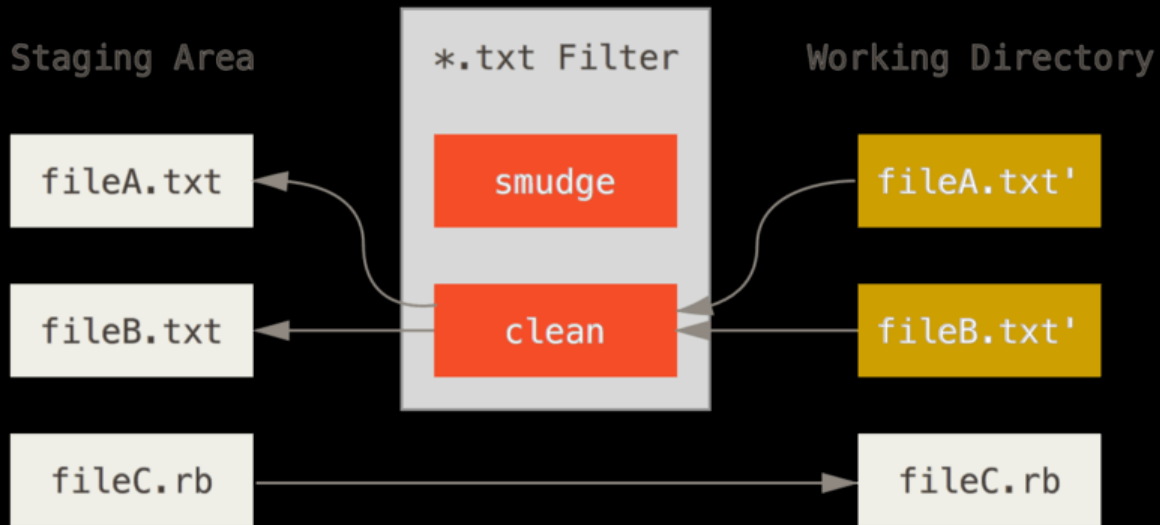
Filters

Smudge (on checkout)



Filters

Clean (on `add`)



Filters: Diff Images

```
*.png diff=exif
git config diff.exif.textconv exiftool
diff --git a/image.png b/image.png
```

```
index 88839c4..4afcb7c 100644
--- a/image.png
+++ b/image.png
@@ -1,12 +1,12 @@
  ExifTool Version Number      : 7.74
  -File Size                   : 70 kB
  -File Modification Date/Time : 2009:04:21 07:02:45-07:00
  +File Size                   : 94 kB
  +File Modification Date/Time : 2009:04:21 07:02:43-07:00
  File Type                   : PNG
  MIME Type                   : image/png
  -Image Width                 : 1058
  -Image Height                : 889
  +Image Width                 : 1056
  +Image Height                : 827
  Bit Depth                   : 8
  Color Type                   : RGB with Alpha
```


Filters: Mask Markdown As Word



- smudge: markdown to Word
- clean: Word to markdown

Filters: Transparent Encryption



- smudge: decrypt
- clean: encrypt
- **required** keyword

Miscellaneous Tips

- The power of -
 - `git checkout -`
 - `git merge -`
- Git Stash
 - `git stash save "Jim asked to fix F00 while I was doing BAR"`
 - `git stash list`
 - `git stash pop`

Miscellaneous Tips (continued)

- insta-checkout file from another branch with

```
git checkout <branch> -- filename
```

- git replace

```
git grep -l $1 | xargs sed -i  
's/$1/$2/g'
```

~~Conclusion~~ Call to Action

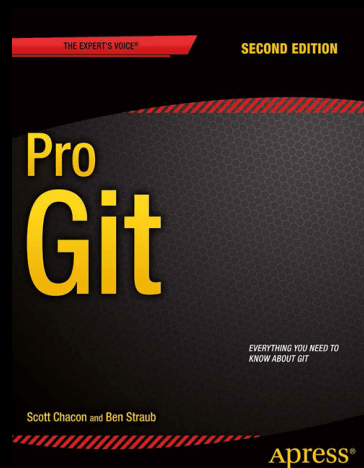
- a couple of aliases

~~Conclusion~~ Call to Action

- a couple of aliases
- make and share a `pre-push` hook

~~Conclusion~~ Call to Action

- a couple of aliases
- make and share a `pre-push` hook
- download Pro Git (it's free!)



¿Questions?