

Title

Alex Phillips

February 25, 2016

Git is about version control. Changes made to files are called commits.

1. To set git account locally

```
$ git config --global user.email "user.email@email.com"
```

1. To initialize folder as git repository.

```
$ git init
```

```
$ git status
```

```
On branch master
```

```
Initial commit
```

```
Untracked files:
```

```
(use "git add <file>..." to include in what will be committed)
```

```
.git_workshop_notes.md.swp
git_workshop_notes.md
git_workshop_notes.pdf
git_workshop_notes.tex
git_workshop_notestex
git_workshop_notestex.aux
git_workshop_notestex.log
git_workshop_notestex.out
git_workshop_notestex.pdf
```

```
nothing added to commit but untracked files present (use "git add" to track)
```

- By default, first commit goes to Master.
- Here I have created files, and they are UNTRACKED!!
- As soon as you make git aware of them, git will begin tracking their changes.
- git add stages the changes.
- Some people add each file separately, then commit in bulk.

```
$ git add *
```

```
$ git status
```

```
On branch master
```

```
Initial commit
```

```
Changes to be committed:
```

```
(use "git rm --cached <file>..." to unstage)
```

```

new file:   git_workshop_notes.md
new file:   git_workshop_notes.pdf
new file:   git_workshop_notes.tex
new file:   git_workshop_notestex
new file:   git_workshop_notestex.aux
new file:   git_workshop_notestex.log
new file:   git_workshop_notestex.out
new file:   git_workshop_notestex.pdf

```

```
Untracked files:
```

```
(use "git add <file>..." to include in what will be committed)
```

```
.git_workshop_notes.md.swp
```

- Now it's tracking them, but they have not been committed yet.
- git commit

```
$ git commit
```

```
[opens vi editor and wants message with the stuff]
```

```
[master (root-commit) 0e0d499] Start notes on this tutorial folder
```

```
8 files changed, 1280 insertions(+)
```

```

create mode 100644 git_workshop_notes.md
create mode 100644 git_workshop_notes.pdf
create mode 100644 git_workshop_notes.tex
create mode 100644 git_workshop_notestex
create mode 100644 git_workshop_notestex.aux
create mode 100644 git_workshop_notestex.log
create mode 100644 git_workshop_notestex.out
create mode 100644 git_workshop_notestex.pdf

```

- vi editor is default, but I prefer vim:

```
$ git config --global user.editor "vim"
```

- git log tells use the history of changes that we have made. “‘bash \$ git log commit 0e0d499b750312cf8fe371799803dfdb5637d
Author: alexprp alexprphillips@gmail.com Date: Thu Feb 25 13:32:33 2016 -0600

Start notes on this tutorial folder

```
$ git log --oneline 0e0d499 Start notes on this tutorial folder “‘
```

- This is why it’s super important to leave good commit messages. Otherwise, the message in the log is meaningless.
- I have made more changes to this notes file and its derivatives since we last committed. Let’s see what that does “‘bash \$ git status On branch master Changes not staged for commit: (use “git add ...” to update what will be committed) (use “git checkout - ...” to discard changes in working directory)

modified: git_workshop_notes.md modified: git_workshop_notes.pdf modified: git_workshop_notes.tex

Untracked files: (use “git add ...” to include in what will be committed)

```
.git_workshop_notes.md.swp
```

no changes added to commit (use “git add” and/or “git commit -a”) - git diff tells us the changes specifically that have been made. - I don't put that here because it's the same text and
pandoc/latex/markdown get confused.bash \$ rm notestex \$ git add * \$ git commit -m ‘deleted
accidental files and added more notes’ [master 904dcea] deleted accidental files and added more notes 3 files
changed, 203 insertions(+), 3 deletions(-) rewrite git_workshop_notes.pdf (62%) \$ git log

commit 904dceaff5ee685c3a52d11e673068e9c22baed0 Author: alexprp alexprphillips@gmail.com Date: Thu
Feb 25 13:58:12 2016 -0600

deleted accidental files and added more notes

commit 0e0d499b750312cf8fe371799803dfdb5637d321 Author: alexprp alexprphillips@gmail.com Date: Thu Feb 25 13:32:33 2016 -0600

Start notes on this tutorial folder

```
$ git log --oneline
```

904dcea deleted accidental files and added more notes 0e0d499 Start notes on this tutorial folder “ - ‘git --staged’ only shows staged changes

Branches

- Master always points to latest commit; it’s a pointer
- HEAD is a special pointer that points to Master, not directly to the latest commit. HEAD points to the **parent of the next commit**.
- If changes are made off of another instance of the previous commit, a new master is created. If there is a dangling Master, it will be deleted periodically, automatically by git if it has not been committed.

Folders

```
$ mkdir folder1
```

```
$ git add folder1
```

[nothing happens]

- git does not track empty folders

```
$ echo 'qwer' > folder1/asdf.txt
```

```
$ git add folder1
```

- nested git repositories are dangerous. don’t do it.
- To make git forget about a directory, simply:

```
$ rm -rf .git
```

.git

```
$ ls .git
```

```
branches
COMMIT_EDITMSG
config
description
HEAD
hooks
index
info
logs
objects
refs
```

- contains many files, including a local instance of `~/.config`

```
$ cat COMMIT_EDITMSG
```

```
# Please enter the commit message for your changes. Lines starting
# with '#' will be ignored, and an empty message aborts the commit.
# On branch master
# Changes to be committed:
#   modified:   git_workshop_notes.md
#   modified:   git_workshop_notes.pdf
#   modified:   git_workshop_notes.tex
#
# Changes not staged for commit:
#   deleted:    git_workshop_notestex
#   deleted:    git_workshop_notestex.aux
#   deleted:    git_workshop_notestex.log
#   deleted:    git_workshop_notestex.out
#   deleted:    git_workshop_notestex.pdf
#
# Untracked files:
```

```
# .git_workshop_notes.md.swo  
# .git_workshop_notes.md.swp  
#
```

- This is what will be shown upon commit; contains stuff from last stage.