

Fundamentals of Git

Illya Starikov

December 25, 2016

1 `Git`ting Good

Let's start with a blank directory¹. After opening your terminal emulator of choice,

1. Navigate to your desktop directory via `cd`.
2. Make a new directory named `git-tac-toe` for your git repository (`mkdir git-tac-toe`) and change into said directory (`cd git-tac-toe`).
3. Create a git repository within this directory (`git init`).

We are now ready to start using Git! Initialized empty Git repository in `...` or something of that nature should appear. If you receive a warning/error message, changes are git was not set up correctly.

A common convention is to have a `README.md` file in your directory. We will create one now.

1. Create a new file named `README.md` with your text editor of choice (for this demo, I will be using `vim`).
2. Insert the following text into the file — if you're not familiar with Markdown, the syntax is easy. Check it out [here](#).

```
# Git Tac Toe

This is just a demonstration on the fundamentals of git!
```

¹We don't always have to do this, especially if we already have a git repository somewhere else.

3. Save your changes. At this point, we have enough to commit our changes. Add the markdown file with `git add README.md`.
4. Commit your files with `git commit`. A text editor should appear asking you for your commit message (if you've never used Git nor done anything with your environmental variables, it will probably be nano).
 - If you want to change the Git text editor, `git config --global core.editor "EDITOR"`.
 - If you want to change the global text editor, `export VISUAL=vim` followed by `export EDITOR="$VISUAL"`.
5. Your commit message should appear like so:

```
Added a read me to describe project
# Please enter the commit message for your changes. Lines starting
# with '#' will be ignored, and an empty message aborts the commit.
# On branch master
#
# Initial commit
#
# Changes to be committed:
#   new file:   README.md
```

Note the commit message isn't particularly long; that's because the intent is very clear.

2 Branching And Merging