

Brief GIT Instructions

Fabio Zanini

May 25, 2012

1 Starting Up

1.1 Create a Remote Repository

1. create a Github account
2. create a local SSH key with ssh-keygen (Google this for details) and add it to your Github profile
3. click on the **Fork** button in Github

1.2 Create a Local Repository

1. create a folder on your computer
2. clone your online repository into the local folder via the following command:

```
git clone https://XXX@github.com/XXX/bi2-ss12-p6.git
```

where XXX is your Github username.

Now you have 1 local branch (master) and 1 remote branch (origin/master). The latter corresponds to your own online repository. You can see all branches by issuing

```
git branch -a
```

1.3 Track Collaborators' Progress

1. add your collaborators' repository as remote:

```
git remote add NAME git://github.com/YYY/bi2-ss12-p6.git
```

where NAME is an arbitrary name you assign to others' *remote* branches repositories, and YYY is to be filled in according to their addresses.

Now you have 2 more remote branches (3 in total) that point to the others' online repositories. You can see your remote branches with

```
git remote -v
```

Optional, but recommended:

5. create 2 local branches that will track closely what's happening on the 2 remote branches:

```
git checkout -b LOCALNAME NAME/master
```

where LOCALNAME is another arbitrary name you give to these auxiliary local branches.

The local branches will mirror the remote master branch of the others, which makes things slightly easier later on. Now, if you look at all your branches, you should have at least 6. For instance, in my case

```
git branch -a
* master
  taylor
  zarin
remotes/origin/HEAD -> origin/master
remotes/origin/master
remotes/taylor/master
remotes/zarin/master
```

Note: local tracking branches such as the two just created are useful to have a sandbox layer between your fellows' official branches and your own code. For instance, assume I have done some work but you want to inspect it closely before merging it into your master branch. You can merge it into my local branch and, if everything's fine, merge it finally into your master. If you just trust other people's work, you do not need this.

2 Typical Workflow

The typical workflow depends a bit whether you have created the local tracking branches or not, but the basic commands are the same.

1. fetch/pull from other people's repositories. If you have not created tracking branches, issue

```
git fetch NAME
```

where NAME is the name of your 2 remote branches (you must issue the command twice, once for each NAME). If you do have tracking branches, you can use instead

```
git checkout LOCALNAME
git pull
```

where LOCALNAME is either one (in succession) of your local tracking branches.

2. merging other people's changes into your own local master. If you do not have local tracking branches,

```
git checkout master
git merge NAME/master
```

If you have local tracking branches,

```
git checkout master
git merge LOCALNAME
```

Note: merge always works by pulling the information from some other branch *into* the current one. So you first have to checkout to *enter* your local master, then to merge from your colleagues' work.

3. developing/coding (not much to say about this :-))
4. adding files to your local history via

```
git add -A
git commit -m 'COMMIT MESSAGE'
```

You can always use `git status` `n{Verbatm[commandchars={},]` it status to know where you are and what is happenin .

pushing changes online onto your remote repository

git push

GIT will possibly ask you for a password/passphrase, depending on how you configured your SSH key.

Notify your colleagues of your work. You can either use Github's "Pull request" button, or just send us an email.

3 List of useful commands

git status show the current state of affairs

git checkout BRANCHNAME switch to another branch

git add -A add all files to the history

git commit -m 'MESSAGE' engrave your changes into history with a message

git push push your current branch onto its natural remote repository

git fetch BRANCHNAME update remote branches (no merge performed)

git merge BRANCHNAME merge another branch into the current one

git pull update the remote branches AND merge them into the current one

git log show the history

git diff BRANCHNAME compare the current branch with another one (e.g. a colleague's one) and show the differences

git branch -a list all branches

git remote -v list all remote repositories