Minimal Git Reference Guide

Alex Blackwell

January 4, 2021

Abstract

A minimal reference guide of the necessary commands to use Git effectively. Explanations are succinct and uncommon commands are omitted.

Contents

1 Vocabulary

Commit

- A record of what changes you have made since the last commit

Staging environment / index

- The environment to which you add modifications for the next commit

Branch

- A pointer to one commit allowing movement back and forth between "states" of a project
- Work on one part of a project and merge
- It's a good idea to delete branches after merging the pull request

Origin

- When a remote repository is cloned to a local machine, git creates an alias called origin
- Origin is essentially shorthand for the remote repository's URL
- To push to the remote repository, the command **git push git@github.com:git/git/git my-branch-name** can be used, or **git push origin my-branch-name** can be used

Pull request

- A way to alert a repository owner that you want to make some changes to their code
- Allows the owner to review the code and make sure it looks good before putting the changes in the primary branch

Head

- A reference to the current commit
- HEAD˜n is a reference to the commit n commits before the current commit

2 Commands

2.1 Initialization

Create a local git repository in a folder:

```
    cmpt@usr:$ git init
```

Now a new repository has been created locally on your computer and has the name of the folder.

2.2 Staging

Add modifications to the repository:

```
    cmpt@usr:$ git add <file>
```

Now this file is added to the staging environment. Changes in the repository can be added as many times and whenever necessary. Use the "." (dot) command to add all changes in this directory.

2.3 Committing

Commit to the staging environment:

```
    cmpt@usr:$ git commit -m "Commit message"
```

Relate the commit message to what the commit contains, commit messages are difficult to change once merged. To change a commit message before pushing, amend it.

```
    cmpt@usr:$ git commit --amend -m "Updated commit message"
```

Now the commit message is updated.

2.4 Updating

To push your changes from the local master branch to the remote master branch:

```
    cmpt@usr:$ git push
```

Now the remote repository is updated to the last local commit.

To get changes on the remote repository on the local repository, from the master branch:

```
    cmpt@usr:$ git pull origin master
```

This can be shortened to **git pull** in most cases.

2.5 Branches

From the primary branch, create a new branch and move there.

```
cmpt@usr:$ git checkout -b my-branch-name
```

A branch called **my-branch-name** is created and you are moved to that branch. To move to an existing branch, do not include the **-b** flag.

To push changes onto a new branch on GitHub:

```
cmpt@usr:$ git push origin my-branch-name
```

GitHub automatically creates a branch on the remote repository named **my-branch-name**. From GitHub a pull request can be made and the branch can be compared and merged into the primary branch.

2.6 Stashing

To record the current state of the staging environment, but return to a clean staging environment, save it:

```
cmpt@usr:$ git stash push
```

This pushes the current local modifications to a new stash entry that can later be referenced with:

```
cmpt@usr:$ git stash list
```

And add back a stash with:

```
cmpt@usr:$ git stash apply [<stash hash>]
```

This merges a stashed state on top of the current work in the working directory.

2.7 Inspecting

Check the status of your Git repository:

```
1    cmpt@usr:$ git status
2    On branch master
3
4    Initial commit
5
6    Untracked files:
7      (use "git add <file>..." to include in what will be committed)
8
9            file.txt
10
11   nothing added to commit but untracked files present (use "git add" to track)
```

This says that a file named **file.txt** has been added to the repository (line 9), but nothing will be done with it unless it is added.

To see all the new commits:

```
cmpt@usr:$ git log
commit 3e270876db0e5ffd3e9bfc5edede89b64b83812c
Merge: 4f1cb17 5381b7c
Author: Your Name <yourname@example.com>
Date:   Fri Jan 3 17:48:11 2021 -0400

    Merge branch 'master' of https://github.com/<your name>/<your repository>
...
```

This log shows the last few commits and information about them. Notice that line 2 shows the commit hash.

Check all the branches in this repository:

```
1       cmpt@usr:$ git branch
2         master
3         * my-new-branch
```

Line 2 and 3 show the branches of this repository and the asterisk indicates what branch you are currently on.

2.8 Reverting and Resetting

To reset the staging environment with no changes:

```
      cmpt@usr:$ git reset
```

Now the staging environment is clean. To remove a file from staging:

```
      cmpt@usr:$ git reset file.txt
```

Revert back to a commit:

```
      cmpt@usr:$ git revert <hash code number>
```

Useful to refer to specific commits and to undo changes.

References

[1] Git. *Git Documentation.* `https://git-scm.com/docs/`. Jan. 2021.

[2] Lisa Tagliaferri. *How To Use Git: A Reference Guide.* `https://www.digitalocean.com/community/cheatsheets/how-to-use-git-a-reference-guide`. July 2020.

[3] HubSpot Product Team. *An Intro to Git and Github for Beginners (Tutorial).* `https://product.hubspot.com/blog/git-and-github-tutorial-for-beginners`. Dec. 2020.