Constantin Mitides

CS 361-611 Homework 3 – GitHub

Part 3

# What is GitHub?

GitHub is a web-based hosting service for version control using Git. Git is an open source project which manages and stores revisions of projects. It is mostly used for code, but Git can be used to manage other type of files as well. It is essentially a filing system for every draft of a document. GitHub provides a web-based graphical interface for the Git command line tool. GitHub provides access control and several collaboration features in addition to the git commands, for example you may add wikis and basic task management tools for every project.

## Forking

One of GitHub’s flagship functionalities is called *forking* which allows you to copy a repository from one user’s account into another. This feature allows you to take a project which you initially would not have write access to and modify it under your own account.

## Pull Request

You can share your changes made under your account to the original author by sending a notification that is called a *pull request* which is sent to the original owner who can then review the changes you made.

## Merge

The original owner of the repository can then accept your changes into their repository by using *merge*.

# When was it created, why and by who?

GitHub was created in 2008 by Tom Preston-Werner, and Chris Wanstrath. to set up an easy way to host and manage software projects which are using Git as subversion control. It removes the need of having to email one another since all the changes are documented on the website and one can simply view and accept, or decline pull requests all in the interface of the website. There is also a desktop GUI for GitHub as well.

# What similar platforms exist?

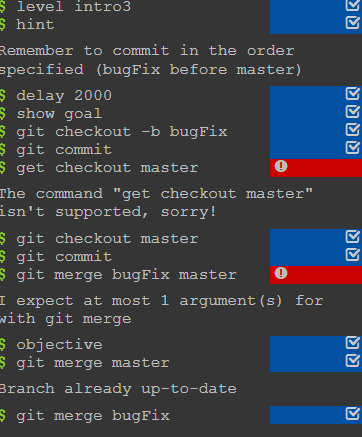
BitBucket

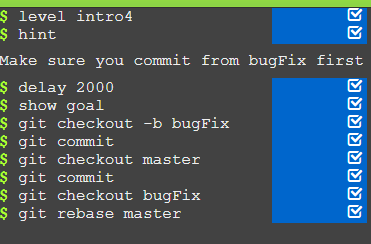
# Why would you use such a platform (5 to 10 lines)

The differences seem to be mostly for those on an enterprise level, or for those who need to keep their code private from the public. One might choose to use Bitbucket over GitHub due to the difference in pricing. If one is also creating a project using the tracking software Jira, they also might want to use Bitbucket instead of GitHub since they are nicely integrated with each other. Another reason to choose Bitbucket over GitHub is that Bitbucket offers a CI/CD solution that is a built in functionality so it does not require any set up for a CI server. CI/CD means Continuous integration and continuous delivery, which allows for frequent releases of software and automation of software production as far as testing, analyzing and versioning.

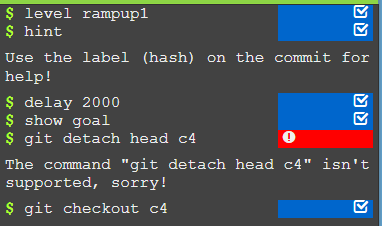
Part 4

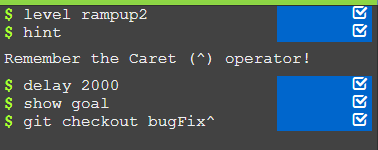
Basic commands

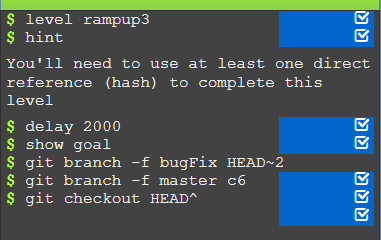


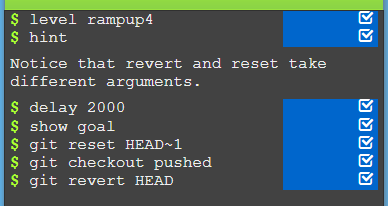


Working with HEAD and relative refs

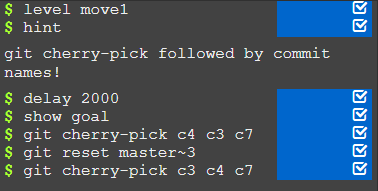


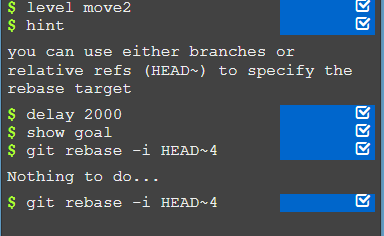




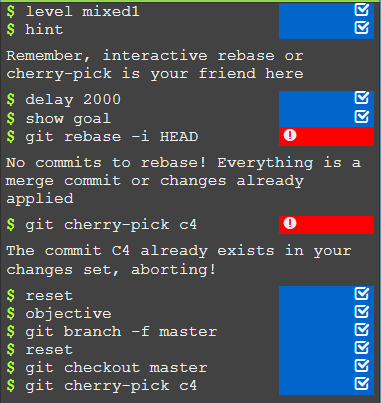


Moving around work

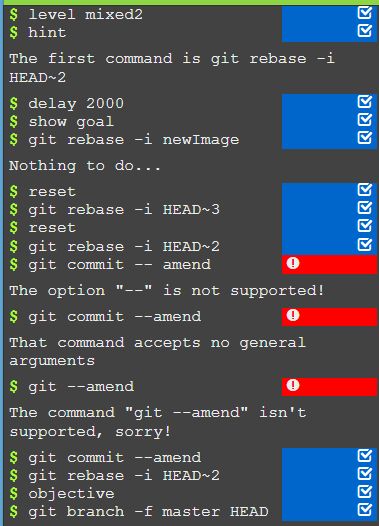




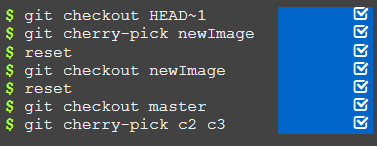
Moving around with specific commits



Grabbing, editing and putting back an old commit and then moving master to current head



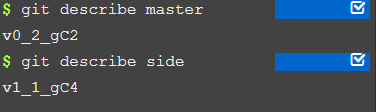
Same as above using cherry-pick



Creating tags and checking out to the tag



Using describe



Part 5

Define the following terms in the context of Git

**Repository**: Data structure to store and manage a project in.

**Commit**: in a git repository, records a snapshot of all the files in the directory.

**Push**: a command that allows you to upload a local repository content to a remote repository

**Branch**: light weight pointers which point to a specific commit, no storage or memory overhead with making branches. Branches include the work that was saved in the commit. *branches refer to commits*

git checkout branchname; get commit – this will record your changes into a new branch

git checkout -b [branchname] will move you to that branch

**Fork:** is a copy of a repository, forking allows you to experiment without affecting original code.

**Merge**: allows you to combine the work from two different branches together which each has a unique commit.

**Clone:** a copy of an existing repository in a new directory, The original repository can be located after cloning the repo.

**Pull:** allows you to bring a local branch up to date with it’s remote version and it also updates your other remote tracking branches.

**Pull request:** notifies others about what changes you have made, once sent the receiver can review your changes.