**Question**

What is GitHub? When was it created? Why? By who? What similar platforms exist? Why would you use such a platform? (Answer between 5 and 10 lines)

**Answer:**

Github is web service that provides online repository for storing and hosting projects, tracking changes(storing versions) and collaborating changes made by different team members. It also allows to contribute to open source project and connect to remote repositories to exchange code using pull/push.

Github was created by Tom Preston Werner, Chris Wanstrath and PJ Hyett. It was designed to make implementation of project changes easier .

Some of the platforms like as [CVS](http://www.nongnu.org/cvs/) and [Subversion](http://subversion.tigris.org/), also provide a central “repository” of all the files associated with a project.

Github should be used because of its features: fork, push and pull.

Forking” – copying a repository from one user’s account to another. This enables you to take a project that you don’t have write access to and modify it under your own account. If you make changes you’d like to share, you can send a notification called a “pull request” to the original owner. That user can then, with a click of a button, merge the changes found in your repo with the original repo.

**Question**

Define the following terms in the context of Git (2 lines maximum):

**Answer:**

* Repository:

A repository is like a folder for your project. Your project's repository contains all of your project's files and stores each file's revision history.

* Commit:

A commit is a record of what files you have changed since the last time you made a commit. Essentially, you make changes to your repo (for example, adding a file or modifying one) and then tell git to put those files into a commit.

COMMAND USED : git commit -m "Commit message"

git push origin my-new-branch

* Push

Send changes to the master branch of your remote repository. This allows other people to see the changes you've made. If they're approved by the repository's owner, the changes can then be merged into the master branch**.**

COMMAND USED : git push origin master or

git push origin my-new-branch

* Branch:

Branches are used to propose changes to GitHub projects.

A branch is a parallel version of the main line of development in the repository, or the default branch (usually master). Use branches to:

* Develop features
* Fix bugs
* Safely experiment with new ideas

Once you're satisfied with the changes in your branch, you can open a pull request between your branch (the *head* branch) and the default branch (the *base* branch).

* Fork

A fork is a copy of a repository that you manage. Forks let you make changes to a project without affecting the original repository. You can fetch updates from or submit changes to the original repository with pull requests.

* You can use a pull request to suggest changes from your fork to the original repository, also known as the *upstream* repository.
* You can bring changes from the upstream repository to your local fork by synchronizing your fork with the upstream repository.
* Merge

Merge a pull request into the upstream branch when work is completed. Anyone with push access to the repository can complete the merge.

If the pull request does not have any merge conflicts, you can merge it on GitHub. If the pull request does have merge conflicts, or if you'd like to test the changes before merging, you can [check out the pull request locally](https://help.github.com/articles/checking-out-pull-requests-locally) and merge it using the command line.

* Clone

If you want to get a copy of an existing Git repository — for example, a project you’d like to contribute to — the command you need is git clone. If you’re familiar with other VCS systems such as Subversion, you’ll notice that the command is "clone" and not "checkout". This is an important distinction — instead of getting just a working copy, Git receives a full copy of nearly all data that the server has. Every version of every file for the history of the project is pulled down by default when you run git clone.

You clone a repository with git clone <url>. For example, if you want to clone the Git linkable library called libgit2, you can do so like this:

$ git clone https://github.com/libgit2/libgit2

* Pull

Git pull is a convenient shortcut for completing both git fetch and git mergein the same command:

* git pull *remotename* *branchname*
* # Grabs online updates and merges them with your local work

Because pull performs a merge on the retrieved changes, you should ensure that your local work is committed before running the pull command. If you run into a merge conflict you cannot resolve, or if you decide to quit the merge, you can use git merge --abort to take the branch back to where it was in before you pulled.

* Pull request

Pull requests let you tell others about changes you've pushed to a repository on GitHub. Once a pull request is opened, you can discuss and review the potential changes with collaborators and add follow-up commits before the changes are merged into the repository.

Part7

Steps to update file in another's repository without cloning

How to Retrieve the README.md file at:

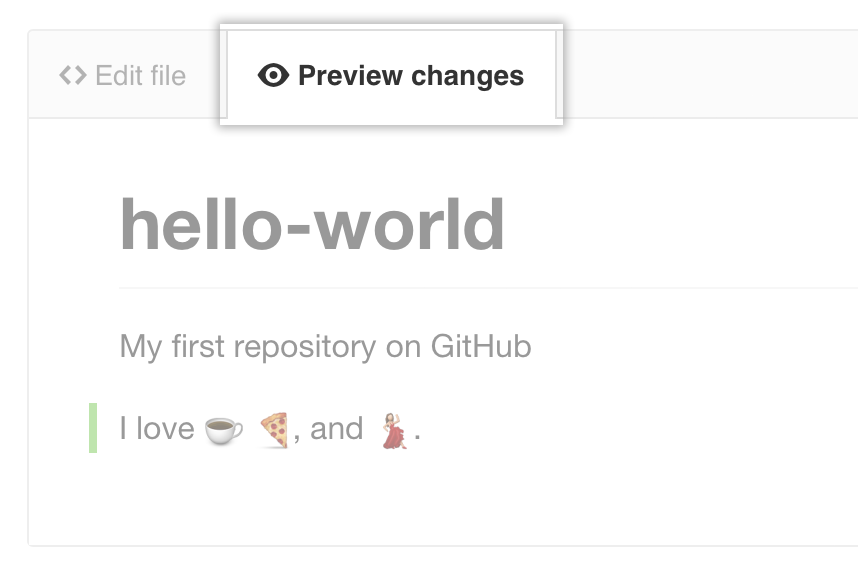
<https://github.com/paceuniversity/courses>

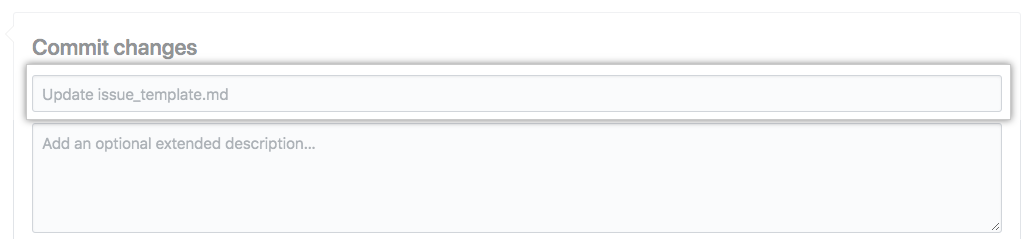
Add your name (lastname, firstname) in the file, **add a comment (date and time) (REQUIRED)**

**Steps**

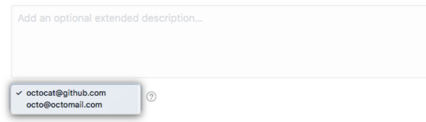
1. In another user's repository, browse to the folder that contains the file you want to edit. Click the name of the file you want to edit.
2. Above the file content, click . At this point, GitHub forks the repository for you.
3. Make any changes you need to the file.



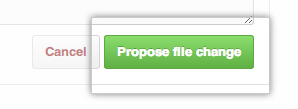
1. Above the new content, click **Preview changes**. 
2. At the bottom of the page, type a short, meaningful commit message that describes the change you made to the file. You can attribute the commit to more than one author in the commit message. For more information, see "[Creating a commit with multiple co-authors](https://help.github.com/articles/creating-a-commit-with-multiple-authors)."



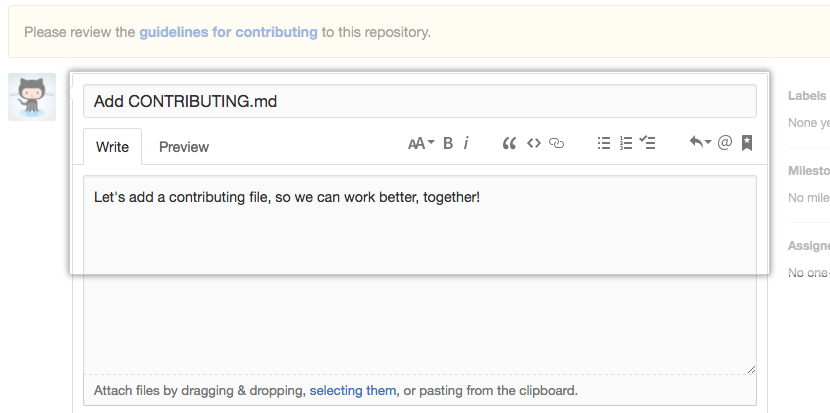
1. Below the commit message fields, click the email address drop-down menu and choose a Git author email address. Only verified email addresses appear in this drop-down menu. If you enabled email address privacy, then username@users.noreply.github.com is the default commit author email address. For more information, see "[About commit email addresses](https://help.github.com/articles/about-commit-email-addresses)."



1. Click **Propose file change**.



1. Type a title and description for your pull request.



1. Click **Create pull request**.

