GIT CHEATSHEE

Configure user information for all local repositories

1. $ git config --global user.name "[name]"

Sets the name you want attached to your commit transactions

1. $ git config --global user.email "[email address]"

Sets the email you want atached to your commit transactions

Start a new repository or obtain one from an existing URL

1. $ git init [project-name]

Creates a new local repository with the specified name

1. $ git clone [url]

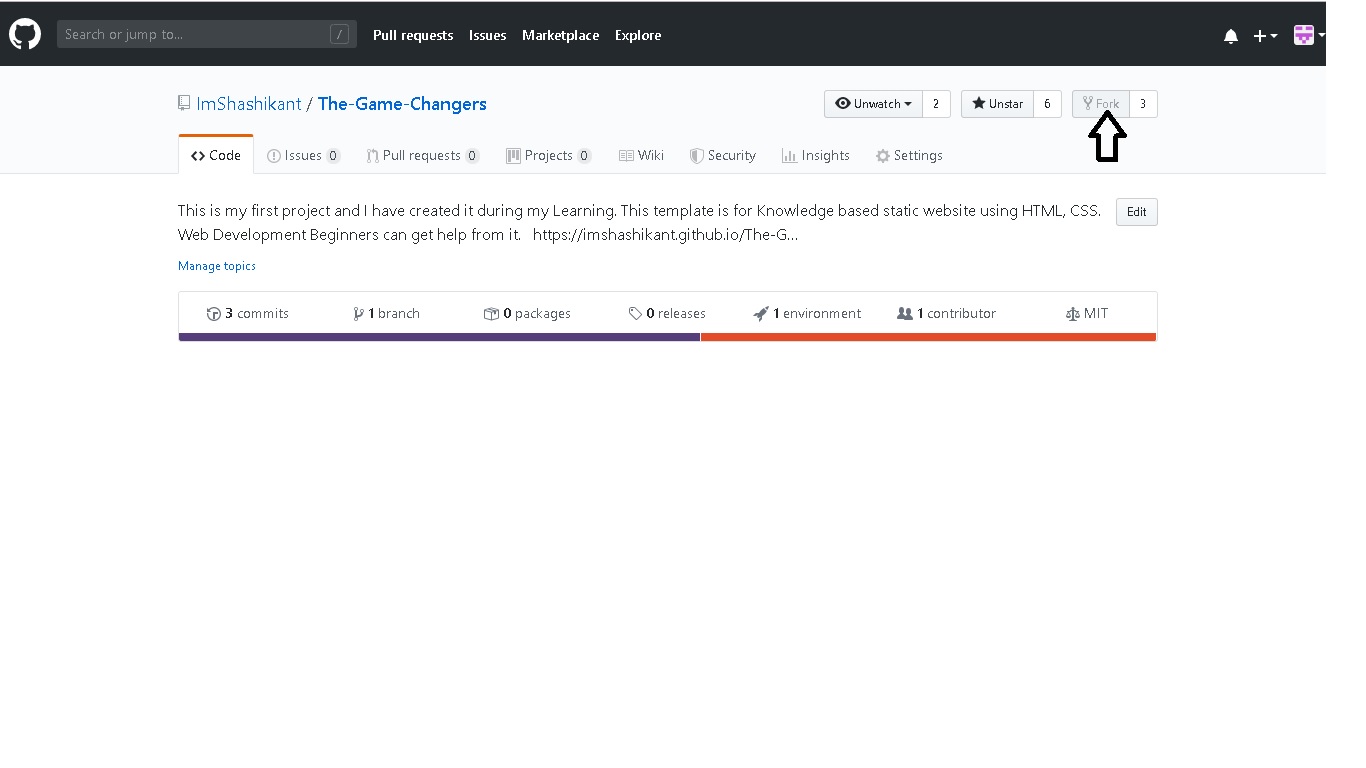
Downloads a project and its entire version history

Make sure you are copying HTTTPS. Use SSH only in case when you RSA SSH have public key uploaded to Github to generate a SSH public key fellow below steps:

* cd /c/users/shash
* mkdir .ssh
* cd /c/Users/shash/.ssh/
* ssh-keygen -t rsa -C “[shashikant.d3698@gmail.com](mailto:shashikant.d3698@gmail.com)” (NO Space between ssh and -keygen)
* cat id\_rsa.pub [Copy id\_rsa.pub key from .ssh folder and go to Authentication in GitHub Account paste it.]

How To Create a Pull Request on GitHub

# Fork the Repository



1. Clone the forked repo.

* $ git clone [url]

Downloads a project and its entire version history

# 

## Create a New Branch

* $ git branch new-branch

Now that our new branch is created, we can switch to make sure that we are working on that branch by using the git checkout command:

* $ git checkout new-branch

Once you enter the git checkout command, you will receive the following output:

Output

Switched to branch 'new-branch'

Alternatively, you can condense the above two commands, creating and switching to a new branch, with the following command and -b flag:

* $ git checkout -b new-branch

If you want to switch back to master, you’ll use the checkout command with the name of the master branch:

* git checkout master

## Make Changes Locally

Once you have modified existing files or added new files to the project, you can add them to your local repository

* $ git add -filename.txt

1. Commit Changes

* $ git commit -m "Fixed documentation typos"
* $ git status

1. push the changes to the current branch of your forked repository:

* $ it push --set-upstream origin new-branch

### Configure a Remote for the Fork

* $ git remote -v

Since we cloned a repository, our output should look similar to this:

Output

origin https://github.com/your-username/forked-repository.git (fetch)

origin https://github.com/your-username/forked-repository.git (push)

* $ git remote add upstream https://github.com/original-owner-username/original-repository.git
* $ git remote -v

Output

origin https://github.com/your-username/forked-repository.git (fetch)

origin https://github.com/your-username/forked-repository.git (push)

upstream https://github.com/original-owner-username/original-repository.git (fetch)

upstream https://github.com/original-owner-username/original-repository.git (push)

1. Sync the Fork

* $ git fetch upstream

Output

From https://github.com/original-owner-username/original-repository

\* [new branch] master -> upstream/master

Let’s switch to the local master branch of our repository:

* $ git checkout master

Output

Switched to branch 'master'

1. merge any changes that were made in the original repository’s master branch, that we will access through our local upstream/master branch, with our local master branch:

* $ git merge upstream/master

## **Create Pull Request**



