

ON GITHUB.COM

- Created an open source repository
- Started and managed a new branch
- Changed a file and committed those changes to GitHub
- Opened and merged a Pull Request

1. To create a new repository
2. Create a Branch
3. Make and commit changes
4. Open a Pull Request
5. Merge your Pull Request
6. Delete the Branch

ON GIT

Configuring and connecting to a remote repository

MAKE DIRECTORY

1. From the command line, [move to the directory](#) you want to contain your Git repository.
2. Type the following command to configure your Git username, where <your name> will be your GitHub username.

```
git config --global user.name "<your name>"
```

3. After entering the above command, you should be returned to the command prompt. Next, enter your e-mail address by typing the following command, where <your e-mail> is your e-mail address.

```
git config --global user.email "<your e-mail>"
```

4. Once the above steps have been completed, you'll be ready to connect to a remote repository. To find the repository address, go to a repository on GitHub and click the **Clone or download repository** link to get the address. For example, we've created a repository called "example" at <https://github.com/Computerhope/example.git> address. Copy the address to your [clipboard](#).
5. Once copied go back to the command line and type the following command, where <URL> is the address you copied. To paste that address into the command line right-click in the command line window and click paste.

`git clone <URL>`

6. Once the Git repository is created, you'll have a new directory in your current directory with the name of the Git repository.
7. Once the Git remote repository is cloned to your local repository, you should have a new folder in the current directory with the name of the Git repository. For example, in our "example" Git we would have a new directory called "example". Use the `cd` command to change into the new directory.
8. Once in the new directory, type the following command to list the remote repositories.

`git remote`

9. If successful, you should see "origin" that is the name of your master Git branch. To see the aliases (URL or path), type the following command.

`git remote -v`

Working in your local repository and pushing files

After following the above steps and cloning a remote repository, you can work on the files as you normally would. You can create new files or edit existing files from the command line or your favorite [text editor](#). Below we will go through the steps in creating a new file and pushing that new file as well as editing an existing file and pushing the update.

Creating a new file and pushing to remote repository

1. Create a new file in the Git directory by typing the following command from either the Bash or Windows command line. The following command opens and creates a file called `example.txt` in [Notepad](#). In Git Bash, you could also use the [touch command](#) to create a blank new file and then type "`start <name of file>`" to open the file in your favorite text editor.

`start notepad example.txt`

2. In the text editor, enter some text into the file and save and exit the file.
3. Back at the command line type the following command to get the current status of your branch and untracked files.

`git status`

4. Git displays a window similar to the example shown below, showing that the file we created is new and untracked by Git.
5. As mentioned in the notes and seen in the picture, we'll now want to add this file to Git to be tracked by typing the following command. If your file is not named "`example.txt`," you'd want to change the text to the name of your file.

`git add example.txt`

6. After entering the above command, the file is added as a new file also known as staging. Typing **git status** again shows you in green that the file is a new file that is ready to be committed.
7. Next, type the following command to commit the changes made in the local workspace to the local repository. In the example below, our notes "First example" should be notes that will make sense to you and anyone else who may be working with you on your project.

```
git commit -m "First example"
```

8. Finally, now that changes have been moved from your workspace into your local repository it is ready to be pushed to the remote repository. Type the following command to push all changes to the remote repository.

```
git push
```

Modifying a file and pushing to remote repository

1. Edit and modify one or more files in your Git.
2. Type **git status** to get see the status of all the files that have not yet been committed from the workspace to the local repository.
3. Type the following command to add *all* files. The single period indicates that you want all files to be added to the local repository. Some people may also use **git add -A** to add all.

```
git add .
```

4. Once the files have been added, type the following command to commit. Change the notes to apply to your commit.

```
git commit -m "Second update"
```

Finally, type **git push** to push the commit to the remote repository.

If it's been awhile since you've committed any work, perform the **git pull** command to get the latest updates from the remote repository and merge them into your local repository.

To get all changes without merging, run the **git fetch** command to grab all of the latest updates from the remote repository without merging any of the new changes.

```
git add .
git commit -m "Fixed the merge conflict"
git push origin master
```

To create a branch in your local repository, follow the steps below.

1. In the Git master directory, type the following command, where "<New Branch>" is where you would put the name of the new branch name. For example, we could call the branch "examplebranch".

```
git branch <New Branch>
```

2. Next, type the following command to switch to the branch.

```
git checkout <Branch Name>
```

3. After entering the above command, the prompt (in Git Bash) changes from "master" to the branch name to indicate you're working in a branch and not the master.
4. From this point, you can continue to use Git and modify the files how you have in the past.
5. To see all available branches, you can use the git branch command. For example, typing **git branch** displays your local repository branches. Typing **git branch -a** displays all local and remote repositories.
6. If you need to push the branch to the remote repository, you can run the following command.

```
git push --set-upstream origin <Branch Name>
```

7. Finally, if you need to switch back to the master branch, you can type the following command.

```
git checkout master
```

How to merge a branch back into the master

After you've completed your work in a branch, you'll want to merge it back into the master or another branch by following the steps below.

1. Move into the branch you want to merge into. For example, if you wanted to merge back into the master, type the following command.

```
git checkout master
```

2. Once in the master, type the following command to merge the branch.

```
git merge <Branch Name>
```

3. Once the merge is performed, add the files.
4. Next, commit the changes.
5. Once merged and committed, push the merge by typing the following command. If you get conflicts during the merge, see our [how to deal with merge conflicts section](#).

```
git push
```

How to delete a local and remote branch

If after merging a branch you no longer want to keep the local or remote branch, you can perform the following commands.

To delete the local branch, type the following command.

```
git branch -d <Branch Name>
```

To delete the remote branch, type the following command.

```
git push origin --delete <Branch Name>
```