

INTRODUCTION TO GIT & GITHUB

by Bobby Ilier



Table of Contents

About the book	5
About the author	6
Sponsors	7
Ebook PDF Generation Tool	9
Book Cover	10
License	11
 Introduction to Git	 12
 Version Control	 14
 Installing Git	 16
 Basic Shell Commands	 19
 Git Configuration	 22
 Introduction to GitHub	 26
 Initializing a Git project	 31
 Git Status	 33
 Git Add	 35
 Git Commit	 37

Git Diff	40
Git Log	42
Gitignore	45
SSH Keys	54
Git Push	59
Git Pull	62
Git Branches	66
Git Merge	73
Reverting changes	80
Git Clone	84
Forking in Git	86
Git Workflow	89
Pull Requests	92
Git And VS Code	95
Installing VS Code	96
Cloning a repository in VS Code	98

Reverting changes

As with everything, there are multiple ways to do a specific thing. But what I would usually do in this case I want to undo my latest commit and then commit my new changes is the following.

- Let's say that you made some changes and you committed the changes:

```
git commit -m "Committing the wrong changes"
```

- After that if you run `git log`, you will see the history of everything that has been committed to a repository.
- To undo the last commit, just run the following:

```
git reset --soft HEAD~1
```

The above command will reset back with 1 point.

Note: the above would undo your commit, but it would keep your code changes if you would like to get rid of the changes as well, you need to do a hard reset: `git reset --hard HEAD~1`

- After that, make your new changes

- Once you are done with the changes, run `git add` to add any of the files that you would like to be included in the next commit:

```
git add .
```

- Then use `git commit` as normal to commit your new changes:

```
git commit -m "Your new commit message"
```

- After that, you could again check your history by running:

```
git log
```

Here's a screenshot of the process:

```

root@do-dev:~/demo# git init .
Initialized empty Git repository in /root/demo/.git/
root@do-dev:~/demo# touch demo.txt
root@do-dev:~/demo# git add .
root@do-dev:~/demo# git commit -m "First commit"
[master (root-commit) 45f651d] First commit
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 demo.txt
root@do-dev:~/demo# echo "Wrong changes..." > demo.txt
root@do-dev:~/demo# git add .
root@do-dev:~/demo# git commit -m "Wrong commit..."
[master 9688e23] Wrong commit...
 1 file changed, 1 insertion(+)
root@do-dev:~/demo# git log
commit 9688e23761a6ccbbfaa4362a391b50a63d4ba39a (HEAD -> master)
Author: Bobby Iliev <bobby@bobbyiliev.com>
Date:   Wed Jan 8 08:49:49 2020 +0000

    Wrong commit...

commit 45f651dd4e83205fe1a72ac16bf0d7a3ecfba904
Author: Bobby Iliev <bobby@bobbyiliev.com>
Date:   Wed Jan 8 08:49:27 2020 +0000

    First commit
root@do-dev:~/demo# git reset --soft HEAD~1
root@do-dev:~/demo# git log
commit 45f651dd4e83205fe1a72ac16bf0d7a3ecfba904 (HEAD -> master)
Author: Bobby Iliev <bobby@bobbyiliev.com>
Date:   Wed Jan 8 08:49:27 2020 +0000

    First commit
root@do-dev:~/demo# echo "Fixed changes..." > demo.txt
root@do-dev:~/demo# git add .
root@do-dev:~/demo# git commit -m "Fixed commit..."
[master 081f0fb] Fixed commit...
 1 file changed, 1 insertion(+)
root@do-dev:~/demo# git log
commit 081f0fbf3d32ad7e7946e1ec4cc5b432fc699fef (HEAD -> master)
Author: Bobby Iliev <bobby@bobbyiliev.com>
Date:   Wed Jan 8 08:50:28 2020 +0000

    Fixed commit...

commit 45f651dd4e83205fe1a72ac16bf0d7a3ecfba904
Author: Bobby Iliev <bobby@bobbyiliev.com>
Date:   Wed Jan 8 08:49:27 2020 +0000

    First commit
root@do-dev:~/demo# █

```

Another approach would be to use `git revert COMMIT_ID` instead.

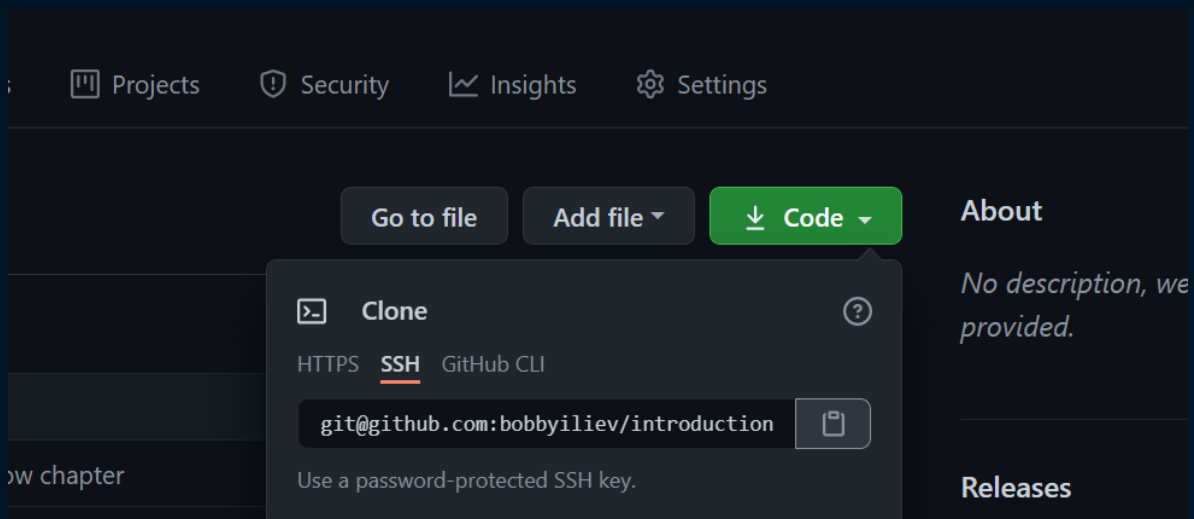
Here is a quick video demo on how to do the above:

[Reverting changes](#)

Git Clone

More often than not, rather than starting a new project from scratch, you would either join a company and start working on an existing project, or you would contribute to an already established open source project. So in this case, in order to get the repository from GitHub to your local machine, you would need to use the `git clone` command.

The most straight forward way to clone your GitHub repository is to first visit the repository in your browser, and then click on the green **Code** button and choose the method that you want to use to clone the repository:



In my case, I would go for the SSH method as I already have my SSH keys configured as per chapter 14.

As I am cloning this repository [here](#), the URL would look like this:

```
git@github.com:bobbyiliev/introduction-to-bash-scripting.git
```


Once you have this in my clipboard, head back to your terminal, go to a directory where you would like to clone the repository to and then run the following command:

```
git clone git@github.com:bobbyiliev/introduction-to-bash-scripting.git
```

The output that you would get will look like this:

```
Cloning into 'introduction-to-bash-scripting'...
remote: Enumerating objects: 21, done.
remote: Counting objects: 100% (21/21), done.
remote: Compressing objects: 100% (16/16), done.
remote: Total 215 (delta 7), reused 14 (delta 4), pack-reused
194
Receiving objects: 100% (215/215), 3.08 MiB | 5.38 MiB/s,
done.
Resolving deltas: 100% (114/114), done.
```

Essentially what the `git clone` command does is to more or less download the repository from GitHub to your local folder.

Now you can start making the changes to the project by creating a new branch, writing some code, and finally committing and pushing your changes!

One important thing to keep in mind is that in case that you are not the maintainer of the repository and do not have the rights to push to the repository, you would need to first fork the original repository and then clone the forked repository from your account. In the next chapter, we will go through the full process of forking a repository!

Setup a commit message template

If you want to speed up the process and have a predefined template for your commit messages, you can create a simple file that will contain this information.

In order to do that, open your terminal if you're on Linux or macOS and create the following file: `.gitmessage` in your home directory. In order to create the file, you can open it in your favorite text editor and then simply put the default content you would like and then just save and exit the file. Example content is:

```
cat ~/.gitmessage
```

```
#Title
```

```
#Summary of the commit
```

```
#Include Co-authored-by for all contributors.
```

To tell Git to use it as the default message that appears in your editor when you run `git commit` and set the `commit.template` configuration value:

```
$ git config --global commit.template ~/.gitmessage
$ git commit
```

Conclusion

If you prefer to code in Visual Studio Code and you also use version control, I will definitely recommend you to give it a go and interact with the repositories in VS code. I believe that everyone has their own style, and they might do things differently depending on their mood as well. As long as you can add/modify your code and then commit your changes to the repository, there is no exactly correct/wrong way to achieve this. For example, you can edit your code in vim and push the changes using the git client in your terminal or do the coding in Visual Studio and then commit the changes using the terminal as well. You're free to do it the way you want it and the way you find it more convenient as well. I believe that using git within VS code can make your workflow more efficient and robust.

Additional sources:

- [Version Control](#) - Read more about integrated Git support.
- [Setup Overview](#) - Set up and start using VS Code.
- [GitHub with Visual Studio](#) - Read more about the GitHub support in VS code
- You can also check this mini video tutorial on how to use the basics of Git version control in Visual Studio Code

Source:

- Contribured by: [Alex Georgiev](#).
- Initially posted [here](#).

GitHub CLI

The GitHub CLI or `gh` is basically GitHub on command-line.

You can interact with your GitHub account directly through your command line and manage things like pull requests, issues, and other GitHub actions.

In this tutorial, I will give a quick overview of how to install `gh` and how to use it!

This is a sample from "Introduction to Git and GitHub" by Bobby Iliev.

For more information, [Click here](#).