SHORT TUTORIAL ON GIT AND GITHUB FOR THE BUSY PROGRAMMER

In the world of software development, people use version control systems to keep track of their source code.
Whether you are a software engineer/Data scientist or just working in any IT related fields this tutorial is for you.
First things first, in an endeavor to version control your code you will need a hosting server in this case GitHub(we have others eg GitLab).

Next you will need to install Git on your machine. Git will help you to communicate with your hosting server. For R programmers you can do the communication via a graphical user interface in Rstudio if you like.

To sign up for a free github account please visit : https://github.com/
To install Git visit : https://git-scm.com/downloads

1.Now that you have an account with Github and you have Git installed on your machine, we can begin.

2.When we host our code files on GitHub, there are hosted in a special directory called a repository.

3.To create your first repositoy on Github just log in and then on your right you will see a + symbol, click on it and it will give you an option to craete a new repository, just give it a name and you can leave other options with the defaults.

4.Just one thing to remember is that this repository is hosted on the server not on your machine.

5.To have a copy on your machine that you will be working with, you have to clone this repository.

6.To clone a repositoy, just open your Git app and then type git clone repository URL.

7.To get the url of any repository just open that repository and on your right you will see an option called code, click on it and then click the save icon.

Working on your own project repository

After cloning your repository, you can now modify your files locally, edit and improve your code.
Once done you need to stage your files for committing on Github.
You use `git add filename.extension` to stage a particular file or `git add .` to stage all files.
Next commit your files with `git commit -m "fix bug"`.
Give an imperative commit message, to help the future you or someone else understand why you made the change .
Finally push your files to the server with `git push origin master`

Working/contributing to some other project/repository

When working on a collaborative project you need to first clone the project repository, work on your files, fetch and merge other peoples work, then finally push your files.

Use the below workflow.

- Clone the repository

```
git clone repository url
```

- Work on your files and then fetch and merge work done by others

```
git fetch origin
git merge origin/master
```

- Finally push your work to the server

```
git push origin master
```

Sometimes you just want to have a copy of other people's repositories either locally on your machine or on your github account.
To do this either clone those repositories or fork them.
To keep these repositories up to date you have to constantly keep them in sync with the original ones. Use the below commands to do that.
First make sure you change your working directory in Git to point to the specific repository you wanna keep in sync, then issue the following,

```
git fetch --all
git fetch --tags
git merge --hard origin/master
```

For a tutorial written by GitHub themselves which is frequently updated please check out this link: https://guides.github.com/activities/hello-world/