Jordan Alexis Caraballo Vega, Paola Alicea, Jose O. Sotero-Esteva
Partnership for Research and Education in Materials
University of Puerto Rico at Humacao
February 06, 2018

Guide on How to Use Git Environment

What is git and why do we use it at PREM-UPRH?

Git is a free and open source distributed version control system that let the users create and manage projects individually and through organizations and teams. We will be using git to keep a control of who commits changes to our projects and to have a remote copy available for everyone. To view, download, contribute, or learn more about any of our projects you can visit https://github.com/compMathUPRH.

How to get started

The first step to start using git is installing it in your local machine (you will need an administrator account to install it). If you already have administrator permissions open a terminal in your Linux distribution and type:

 # sudo apt-get install git-core

If you do not have admin privileges talk to your mentor or with someone who does. Once you have git installed you will need to create an account of your own. Go ahead and in a browser visit: https://github.com/join/ and sign up to github.
Note: Please select an appropriate username since it will be your working username.

Setting username and account details:

git config --global user.name "Your Name"
git config --global user.email "youremail@gmail.com"
git config --global branch.autosetuprebase always

How to clone environment

To clone (download an existing remote environment to your local system) one of our projects you will need to access our repository at https://github.com/compMathUPRH. For this example we will use our wolffia project.

git clone https://github.com/compMathUPRH/wolffia

This will download the directory download the whole directory together with the files that are remotely.

How to initialize your own environment

To initialize a new environment in your local machine create a folder with the name that you want to assign to your project and the files that you want to include. Once that is done, you can simply initialize a git repository locally by typing:

git init

To check your initialized repository and the status of your files you can type:

git status

How to pull

If you want to be up to date with the remote repository you will need to do a pull. To perform a pull from a cloned environment you will just need to type:

git pull

This command will compare both snapshots and will download any missing files from the remote repository.

How to commit locally and remotely

In order to commit any changes you will need to first add the files that you changed and then commit those changes to the local project. You will have two options at the time of adding files: you can either add everything inside a directory or you can add just specific files that you may want to address. The commands are as follow:

Command to add everything
git add .

Command to add a specific file
git add filename

Once the files are added, you can simply type commit and a comment giving a description of the changes that your commit includes.

git commit -m “Your description of the change”

How to create a branch

A branch is a work environment in git that will let the users work in a single environment to later on update the master branch. In order to create a branch you will need to:

git checkout -b branchName

Once you have worked on your branch, you can commit in your own branch following the commit instructions and pushing your changes directly to your branch with:

git push origin branchName

How to push remotely

This will be the last step at the time of submitting any changes performed at your local repository to the outside world. If you are commiting for the first time you will need to specify the remote and the origin addresses by:

git remote add origin your_username@github.com:project.git

If this is not your first time committing to that specific project you can go ahead and push your changes. Make sure that you already commited your files and that everything is ready to go. If you are sure the type:

git push origin master

And that is it. You have pushed your changes to the master environment

References

https://product.hubspot.com/blog/git-and-github-tutorial-for-beginners

https://www.tutorialspoint.com/git/