# Git 101

Download the awesome GitHub Git Cheat Sheet!

Git is currently the most popular tool for collaboration in software development projects. It allows developers to work in an integrated and consistent way even remotely, maintaining a history of all updates made by the team. GitHub (https://github.com/) is a company (owned by Microsoft, as of 2018) that provides hosting for software development projects using git.

## Basic Commands

To copy a repository to your computer, open a shell terminal, move to the folder where you want to save the repository, and then type the following command:

```
git clone https://github.com/folderName/fileName
```

Maintain this local copy updated with the central repository by issuing: `git pull`

Note: Need to be inside the root folder of the repository in order for the pull command to work.

## Collaboration

This section now describes a typical `git` workflow for collaborative software development. The steps described here use the placeholders `<PRJ>` and `<REPO_URL>` to refer to the project's name and its `GitHub` repository (repo for short) `URL`, respectively. Terminal commands are prefixed with a `$` symbol just so you know where to type them.

Also, this link has some naming conventions that you can use as a guide when choosing names for your git projects. As far as naming conventions, some people separate words with dashes and some use underscores. I personally prefer naming my repositories using underscore. The repository should use the same name as the package.

**Step 1: Fork**

`Fork` (copy) the target repo to your own account. On `GitHub` there is a specific `Fork` button as shown below.

**Step 2: Clone**

Clone the repo to your local machine:

- Open a terminal window.
- Change current directory to where you normally save your projects.
- Clone the repo locally.

```
$ git clone <REPO_URL>
$ cd <PRJ>
```

**Step 3: Branch**

Check out a new "topic branch" and make changes. For example, let's say you will be working on a new class called `Snake`. Perhaps you want to create a new branch called "snake". Branches should not have spaces in their name.

```
$ git branch snake
$ git checkout snake
```

**Step 4: Push**

`Push` your "topic branch" to your `Fork`.

```
$ git add .
$ git commit -m "write a meaningful comment about your work"
$ git push origin snake
```

**Step 5: Pull**

Create a pull request (a request to merge your changes with the original project code). Use `GitHub` interface as shown below.