Semester 1 2024
Astroinformatics I

Tutorial 8: GitHub

In this tutorial session, we will learn how to use GitHub.

Why using GitHub?

The main reasons are:

- version control
- make code public
- sometimes required for submitting code for applications (summer schools, jobs...)

1 Git

If you're using Mac or Linux, you likely have git installed already. Open a terminal window and run

```
$ which git
```

If `git` is already installed, the response should be something like:

```
$ which git
/usr/local/bin/git

$ git --version
git version 2.23.0
```

In case `git` isn't found, you have to install it from `https://git-scm.com/download/`.

2 GitHub Account

Once Git is installed, create a GitHub Account on `https://github.com/` if you don't already have one.

Then connect your GitHub account to your Git account.
This is done from the terminal. To set your Git username, type the following in your terminal:

```
$ git config --global user.name "Firstname Lastname"
```

using your own name.

To set your Git email, type this in your terminal:

```
git config --global user.email "youremail@gmail.com"
```

with your own e-mail address.

To confirm that you have set your Git username correctly, type this:

```
$ git config --global user.name
```

Check that Git is configured correctly so that your username and email address match that of your account on GitHub.

To confirm that you have set your Git email correctly, type this:

```
git config --global user.email
```

3 GitHub Repository

First, create and edit your files (source code or other content) locally.
Then create a GitHub repository, following the instructions in the GitHub documentation:
`https://docs.github.com/en/repositories/creating-and-managing-repositories/quickstart-for-repositories`

After you have done this once, we upload (push) files to GitHub. We will repeat the following steps each times we make changes locally.

3.1 Push your local code to GitHub

You can upload files directly into this repository (drag-and-drop), or you can manage your repository directly from your computer. Assuming your GitHub username is `myusername` and you created a repo `mytestrepo`, and your local folder is `repos`, do the following on your local computer:

```
$ cd repos
$ cd mytestrepo/
$ git add mytestfile.txt
$ git commit -m "added"
$ git push
```

When you now browse the repository on GitHhub you will see the file to be uploaded.
You can update your files in the repository by doing:

4 Common Git Commands to Know

They are many Git commands you can use in the terminal, and that can get overwhelming. So I'd suggest focusing on some of the most popular ones first.

`git init` lets you initialize Git in your folder.
`git add Readme.md` lets you add the Readme file, while `git add .` lets you add all files in the present folder.
`git commit` stores the added files. Use -m for message followed by the actual message.
`git branch` creates a new branch which is a new version of the repository as it appears when added, and `-M` to move the name to main.
`git remote add origin` finally connects the local folder to the repository on GitHub. It is followed by the repository's link.
`git push -u origin main` pushes the code to GitHub. The `-u` flag creates a tracking reference for the branch, and origin main puts the code in the main branch.

The more you continue using GitHub, the more comfortable you'll get using these commands. The key is to start small and maintain your momentum.

Those are some of the main commands you'll use all the time. This is a beginner and non-technical guide to help you get started using Git and GitHub, so we won't go into too much more detail here.