Data.Trek 2020: Git Tutorial

Author: Savandara Besse

A. ACKNOWLEDGMENTS

- Thanks to Francis, Gracielle and Gabriel for the Git Demonstration on Zoom! It helped me a lot to write this summary tutorial 😊

B. REFERENCES (for the two first pictures)

- Github documentation
  - https://help.github.com/en/github/using-git/setting-your-username-in-git
  - https://help.github.com/en/github/setting-up-and-managing-your-github-user-account/setting-your-commit-email-address

C. PRE-REQUIREMENTS

- On Windows:
  - Git bash (more user-friendly than your windows terminal)
  - Available here:https://gitforwindows.org/
- On Linux:
  - Git
  - Available by with this command line `sudo apt-get install git`

Main steps

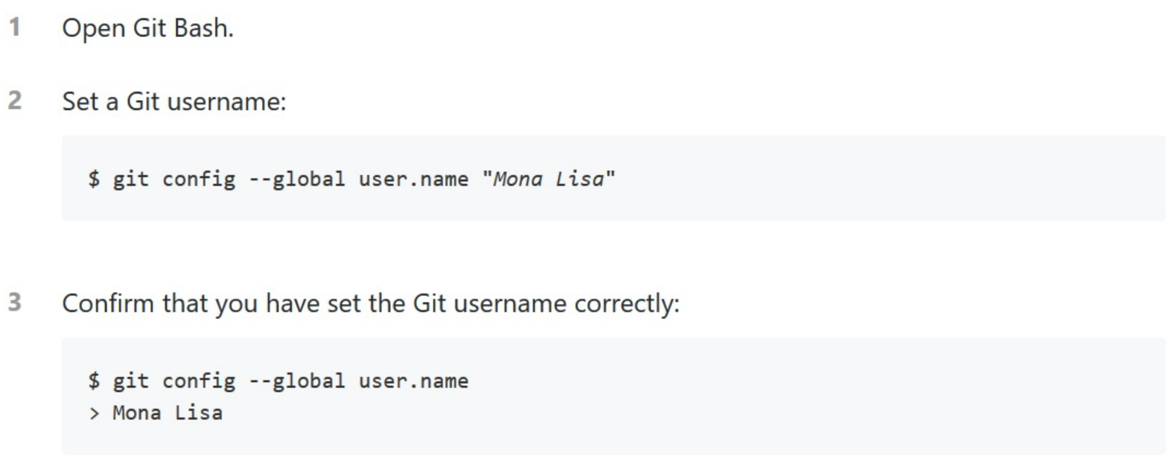
1. Create a GitHub account

- Sign up on https://github.com/

2. Configure your username and email on Git

- Note: `--global` will set this username and email for all your Git repositories

a. user.name



1 Open Git Bash.

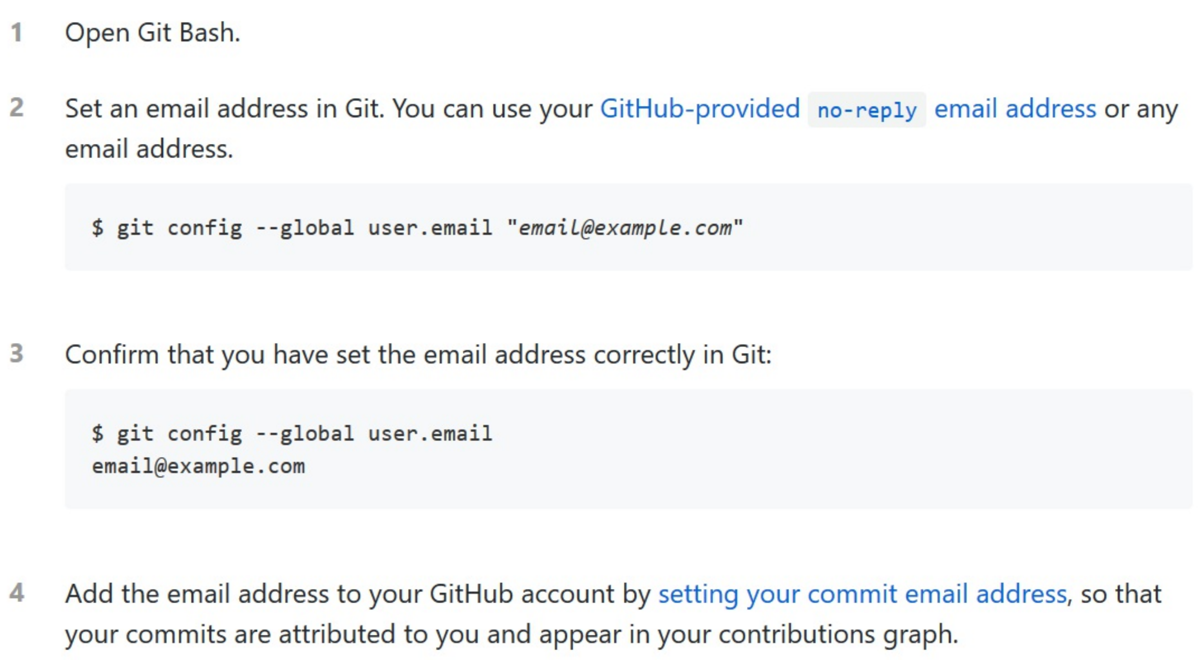
2 Set a Git username:

```
$ git config --global user.name "Mona Lisa"
```

3 Confirm that you have set the Git username correctly:

```
$ git config --global user.name
> Mona Lisa
```

b. user.email



1 Open Git Bash.

2 Set an email address in Git. You can use your GitHub-provided `no-reply` email address or any email address.

```
$ git config --global user.email "email@example.com"
```

3 Confirm that you have set the email address correctly in Git:

```
$ git config --global user.email
email@example.com
```

4 Add the email address to your GitHub account by setting your commit email address, so that your commits are attributed to you and appear in your contributions graph.

3. Create your Git repo and clone it into your local computer

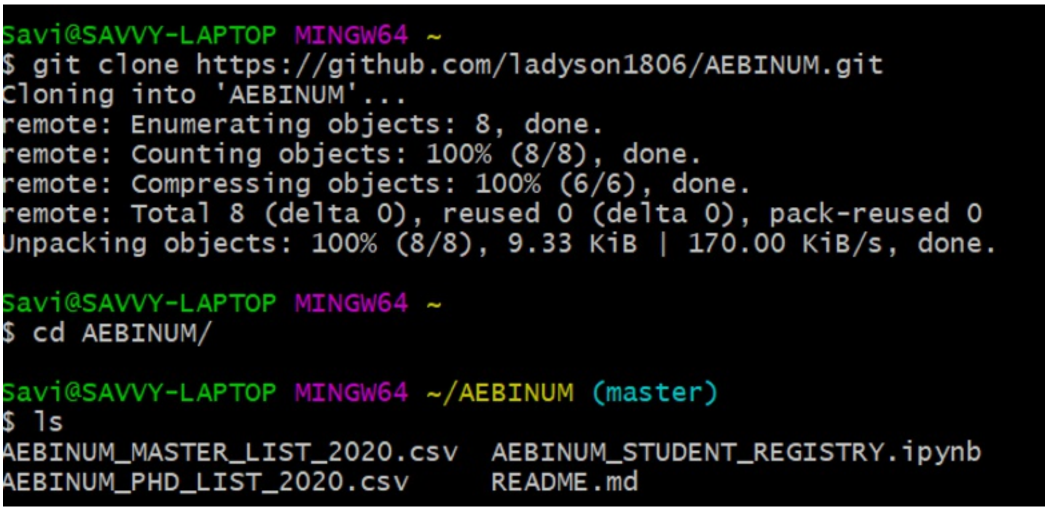
- Creation of your repository:
  - Follow the guidelines here: https://help.github.com/en/github/getting-started-with-github/create-a-repo

    > - Read until this line: “Congratulations! You've successfully created your first repository and initialized it with a README file.”
    > - The next part is about making a commit from the web interface

- Clone your repository on your computer

  i. Open Git Bash
  ii. Write `cd TO/YOUR/PATH/FOLDER`
  iii. Write `git clone http://www.github.com/your_repository_url`
  iv. Enter in the folder using `cd` and check if the content of your report was created with `ls`
     - You should have only an empty README.md file

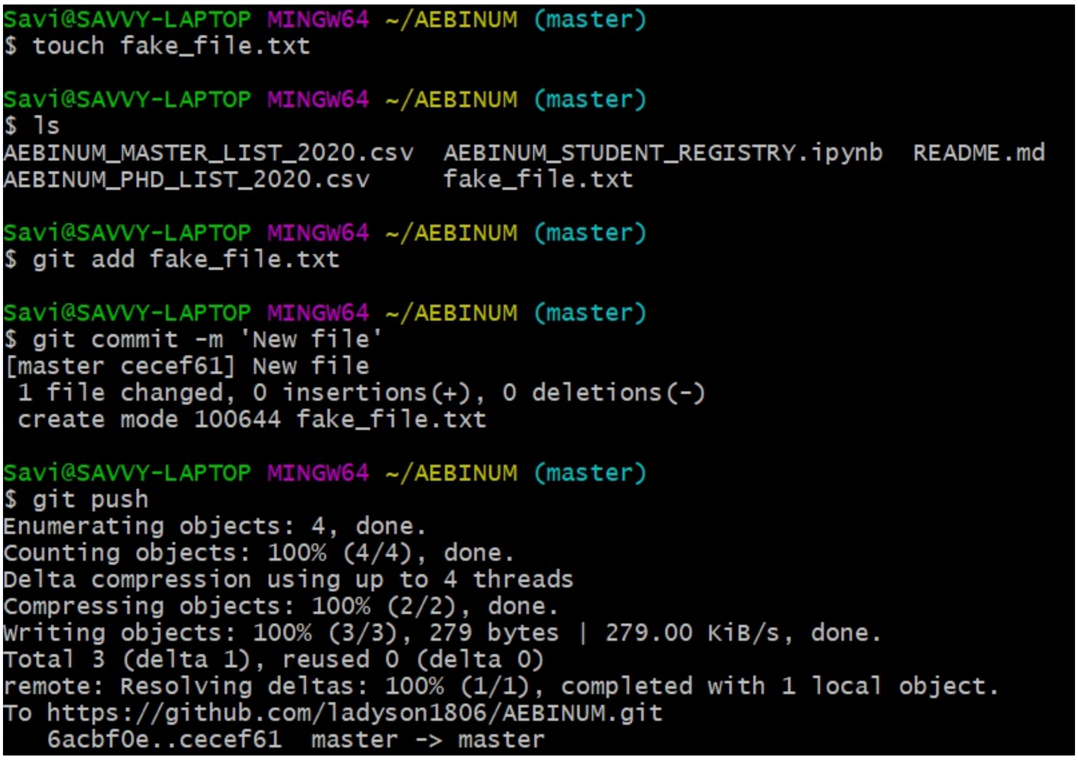
> **Note**: Your will collect this link on the main page of your repository by clicking on the “Clone/Download button”



```
Savi@SAVVY-LAPTOP MINGW64 ~
$ git clone https://github.com/ladyson1806/AEBINUM.git
Cloning into 'AEBINUM'...
remote: Enumerating objects: 8, done.
remote: Counting objects: 100% (8/8), done.
remote: Compressing objects: 100% (6/6), done.
remote: Total 8 (delta 0), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (8/8), 9.33 KiB | 170.00 KiB/s, done.

Savi@SAVVY-LAPTOP MINGW64 ~
$ cd AEBINUM/

Savi@SAVVY-LAPTOP MINGW64 ~/AEBINUM (master)
$ ls
AEBINUM_MASTER_LIST_2020.csv  AEBINUM_STUDENT_REGISTRY.ipynb
AEBINUM_PHD_LIST_2020.csv     README.md
```

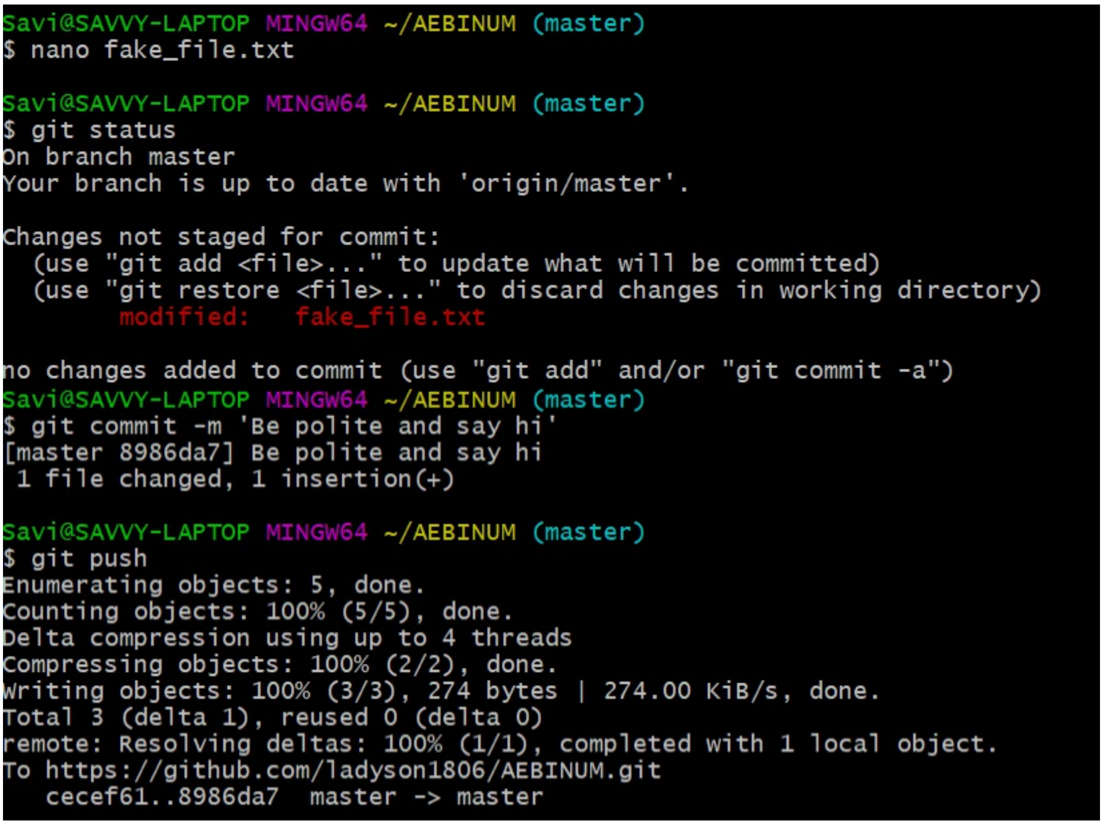
4. Add a new file into your repository and update your Git repo

1. Create a file in your folder
   - In my example, it will be an empty txt file : fake_file.txt
2. Write `git add fake_file.txt` will “warn” your git that you want to commit a change
3. Write `git commit -m “New file”` will add a short description related to your commit
4. Write `git push` to update your git repo



```
Savi@SAVVY-LAPTOP MINGW64 ~/AEBINUM (master)
$ touch fake_file.txt

Savi@SAVVY-LAPTOP MINGW64 ~/AEBINUM (master)
$ ls
AEBINUM_MASTER_LIST_2020.csv  AEBINUM_STUDENT_REGISTRY.ipynb  README.md
AEBINUM_PHD_LIST_2020.csv     fake_file.txt

Savi@SAVVY-LAPTOP MINGW64 ~/AEBINUM (master)
$ git add fake_file.txt

Savi@SAVVY-LAPTOP MINGW64 ~/AEBINUM (master)
$ git commit -m 'New file'
[master cecef61] New file
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 fake_file.txt

Savi@SAVVY-LAPTOP MINGW64 ~/AEBINUM (master)
$ git push
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 4 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 279 bytes | 279.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/ladyson1806/AEBINUM.git
   6acbf0e..cecef61  master -> master
```

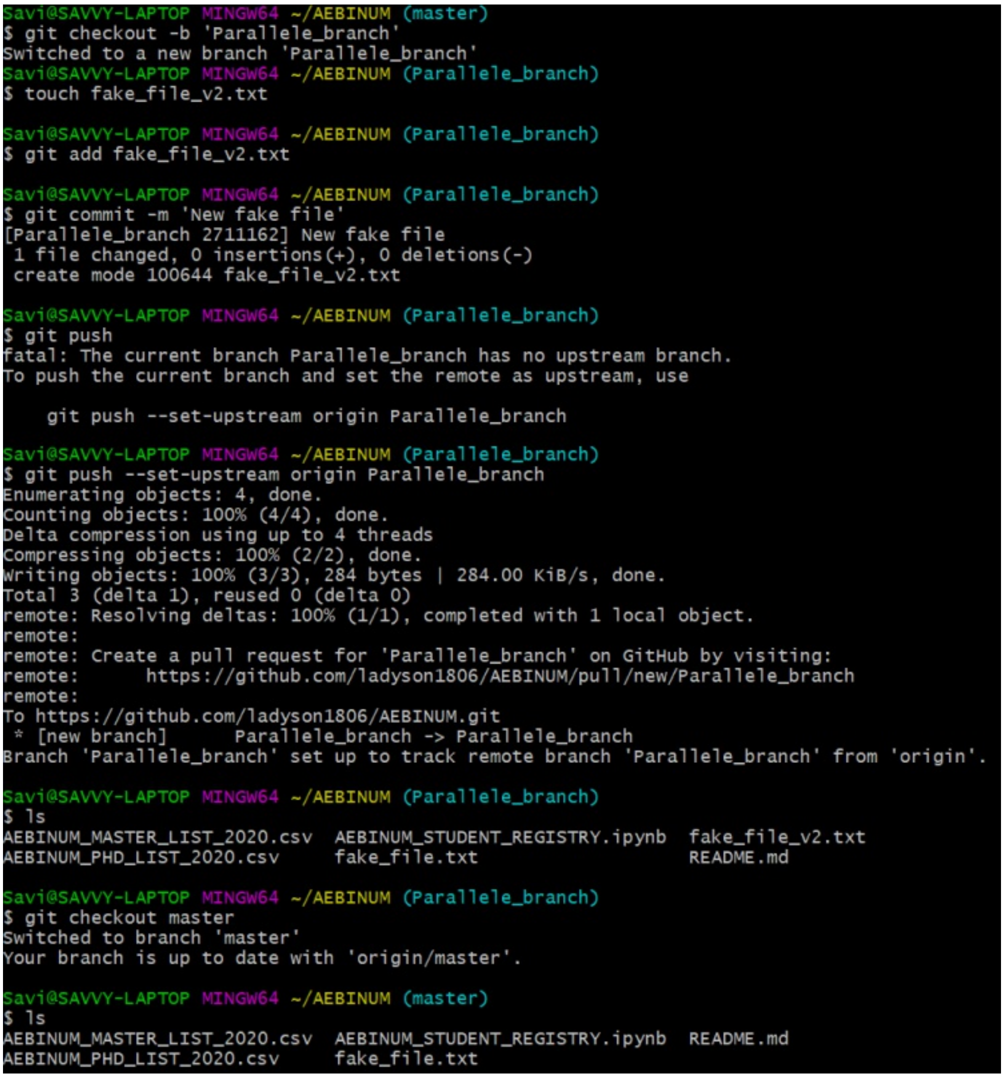
> **Notes**:
>
> - Always think to commit if you are modifying a file! (That's make a historic of your changes aka. versioning)
> - Never forget this order:
>
>   **add / commit / push**
> - `git status` will give you an overview of what you need to commit
>   - If you are working with someone, it is important to always have the last recent version of your Git repo
> - `git pull` will help you to have the last version of the Git repo from Github (not show in my example)
>   - If you have something new in your Git repo (either because you add something through the web interface or someone else add something in the Git repo)
>   - If nothing changed, it will tell you that your repo is up to date



```
Savi@SAVVY-LAPTOP MINGW64 ~/AEBINUM (master)
$ nano fake_file.txt

Savi@SAVVY-LAPTOP MINGW64 ~/AEBINUM (master)
$ git status
On branch master
Your branch is up to date with 'origin/master'.

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   fake_file.txt

no changes added to commit (use "git add" and/or "git commit -a")
Savi@SAVVY-LAPTOP MINGW64 ~/AEBINUM (master)
$ git commit -m 'Be polite and say hi'
[master 8986da7] Be polite and say hi
 1 file changed, 1 insertion(+)

Savi@SAVVY-LAPTOP MINGW64 ~/AEBINUM (master)
$ git push
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 4 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 274 bytes | 274.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/ladyson1806/AEBINUM.git
   cecef61..8986da7  master -> master
```

5. Create and work in a branch

- Creating a branch is useful when you want to make some tests without modifying your master branch aka. *the main one*
- In this example, I will create a branch called Parallele_branch and I will add a new file in this branch
  - You can directly create a branch and be placed in it by writing `git checkout -b Parallele_branch`
  - To switch between your different branches, you will the write `git checkout BRANCHNAME (here master or Parallele_branch)`
- If you are working in team, be sure to create your own branch to not overwrite the code of your team mate!

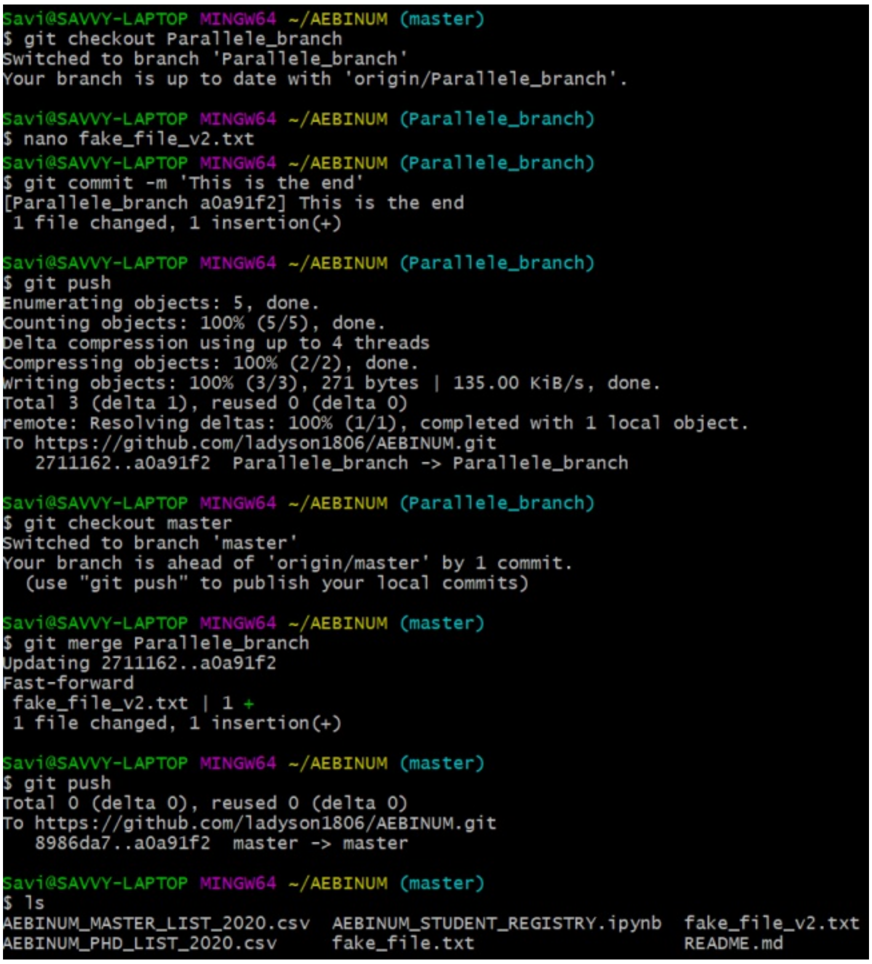


```
Savi@SAVVY-LAPTOP MINGW64 ~/AEBINUM (master)
$ git checkout -b 'Parallele_branch'
Switched to a new branch 'Parallele_branch'
Savi@SAVVY-LAPTOP MINGW64 ~/AEBINUM (Parallele_branch)
$ touch fake_file_v2.txt

Savi@SAVVY-LAPTOP MINGW64 ~/AEBINUM (Parallele_branch)
$ git add fake_file_v2.txt

Savi@SAVVY-LAPTOP MINGW64 ~/AEBINUM (Parallele_branch)
$ git commit -m 'New fake file'
[Parallele_branch 2711162] New fake file
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 fake_file_v2.txt

Savi@SAVVY-LAPTOP MINGW64 ~/AEBINUM (Parallele_branch)
$ git push
fatal: The current branch Parallele_branch has no upstream branch.
To push the current branch and set the remote as upstream, use

    git push --set-upstream origin Parallele_branch

Savi@SAVVY-LAPTOP MINGW64 ~/AEBINUM (Parallele_branch)
$ git push --set-upstream origin Parallele_branch
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 4 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 284 bytes | 284.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
remote:
remote: Create a pull request for 'Parallele_branch' on GitHub by visiting:
remote:      https://github.com/ladyson1806/AEBINUM/pull/new/Parallele_branch
remote:
To https://github.com/ladyson1806/AEBINUM.git
 * [new branch]      Parallele_branch -> Parallele_branch
Branch 'Parallele_branch' set up to track remote branch 'Parallele_branch' from 'origin'.

Savi@SAVVY-LAPTOP MINGW64 ~/AEBINUM (Parallele_branch)
$ ls
AEBINUM_MASTER_LIST_2020.csv  AEBINUM_STUDENT_REGISTRY.ipynb  fake_file_v2.txt
AEBINUM_PHD_LIST_2020.csv     fake_file.txt                   README.md

Savi@SAVVY-LAPTOP MINGW64 ~/AEBINUM (Parallele_branch)
$ git checkout master
Switched to branch 'master'
Your branch is up to date with 'origin/master'.

Savi@SAVVY-LAPTOP MINGW64 ~/AEBINUM (master)
$ ls
AEBINUM_MASTER_LIST_2020.csv  AEBINUM_STUDENT_REGISTRY.ipynb  README.md
AEBINUM_PHD_LIST_2020.csv     fake_file.txt
```

> Have you seen the differences between branches??

6. Merge the content of a branch

- If the file you add in your second branch will be need in the final version of your code, you need to merge to the master branch
  - First you will need to come back to the master branch with this command line `git checkout master`
  - Then, you will merge the branch Parallele_branch to master with this command line `git merge Parallele_branch`



```
Savi@SAVVY-LAPTOP MINGW64 ~/AEBINUM (master)
$ git checkout Parallele_branch
Switched to branch 'Parallele_branch'
Your branch is up to date with 'origin/Parallele_branch'.

Savi@SAVVY-LAPTOP MINGW64 ~/AEBINUM (Parallele_branch)
$ nano fake_file_v2.txt
Savi@SAVVY-LAPTOP MINGW64 ~/AEBINUM (Parallele_branch)
$ git commit -m 'This is the end'
[Parallele_branch a0a91f2] This is the end
 1 file changed, 1 insertion(+)

Savi@SAVVY-LAPTOP MINGW64 ~/AEBINUM (Parallele_branch)
$ git push
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 4 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 271 bytes | 135.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/ladyson1806/AEBINUM.git
   2711162..a0a91f2  Parallele_branch -> Parallele_branch

Savi@SAVVY-LAPTOP MINGW64 ~/AEBINUM (Parallele_branch)
$ git checkout master
Switched to branch 'master'
Your branch is ahead of 'origin/master' by 1 commit.
  (use "git push" to publish your local commits)

Savi@SAVVY-LAPTOP MINGW64 ~/AEBINUM (master)
$ git merge Parallele_branch
Updating 2711162..a0a91f2
Fast-forward
 fake_file_v2.txt | 1 +
 1 file changed, 1 insertion(+)

Savi@SAVVY-LAPTOP MINGW64 ~/AEBINUM (master)
$ git push
Total 0 (delta 0), reused 0 (delta 0)
To https://github.com/ladyson1806/AEBINUM.git
   8986da7..a0a91f2  master -> master

Savi@SAVVY-LAPTOP MINGW64 ~/AEBINUM (master)
$ ls
AEBINUM_MASTER_LIST_2020.csv  AEBINUM_STUDENT_REGISTRY.ipynb  fake_file_v2.txt
AEBINUM_PHD_LIST_2020.csv     fake_file.txt                   README.md
```

> **Notes**:
>
> - To delete a branch, write `git checkout -d BRANCHNAME`
>   - Avoid deleting your master branch, tho'

THE END!
