# Notes on Git Workflow

Chen Heinrich

April 1, 2019

## 1 Clone repo and create a new local branch

- CIone the repository (ideally under a directory called “git” so you will not confuse a git and a non-git directory, e.g. CIP_planck/git/:

```
git clone <the url for the repository>
```

- Checkout master branch:

```
git checkout master
```

- Create a new branch and switch to it:

```
git branch micah/test_something
git checkout micah/test_something
```

  or in one command

```
git checkout -b micah/test_something
```

## 2 Adding, committing and pushing to remote

- Always check which branch you are in before you start working:

```
git branch
```

- After you work some, stage/add particular files or changes to them:

```
git add <filename> or git add .
```

- Commit staged changes:

```
git commit -m ‘‘** NEW ** added ...”
```

- Push to remote:

```
git push
```

- Pull from remote (might need to do so before pushing sometimes if someone also push to the same remote as you do):

```
git pull
```

- The first time you push a new local branch, it's going to asking you to setup upstream of the same name, just use the command it gives you.

- After you do, anyone in the repo will be able to see the upstream of your new branch as well, and I can pull from both your branch and master on my local machine.

## 3 Merging into master

- Master is usually locked, so send me a pull request whenever you need to merge changes from origin/micah/test_something into origin/master.

- Before sending a pull request, try to merge locally first:

  - get the latest from origin/master into the local machine

    ```
    git fetch
    ```

  - when you are ready to merge, make sure you are in your branch (or if you think there might be too many conflicts and you don't want to mess up your current branches, create a new branch use info above and name it for example micah_merge_master_test) :

    ```
    git checkout micah/test_something
    ```

  - test merging locally:

    ```
    git merge master
    ```

  - resolve conflicts and commit:

    ```
    git commit −a −m ‘‘solved merging conflicts”
    ```

  - push to remote:

    ```
    git push
    ```

    (this pushes to origin/micah_testing)
  - go on the website and send a pull request.

# 4 Other useful commands

- Check on current status:

```
git status
```

  (shows you which files are tracked/added, which ones are not)

- Stage only changes to currently tracked files and commit all in one step:

```
git commit -a -m ‘‘<comments>”
```

- The first thing you do before your first commit in a branch is to setup a .gitignore file (touch .gitignore in the top directory), where you name files to never track (e.g.: .*pyc, ./scratch/*; *DS_Store if you work on a Mac; and other scripts that only you should play with but no one else needs to access, etc.) It is important you do this first before you add and commit files you want to ignore, because you can't ignore a file using .gitignore once it's been tracked. But you can do it with some other commands.

- See which branch you are on currently:

```
git branch
```