# Exercise 1b – Setting up your Jupyter Notebook working folder

# Description

In this Part 1 and 2 you will learn more about how to work with Git and how to create a repository either via Github and then clone it locally or create a Github repository from an existing folder. In Part 3 you will learn some useful git commands for managing your repository. At the end of this exercise, you should be able to fully connect the folder you created in Exercise 1a with Github via either of the approaches represented here. The recommended approach is presented in part 1.

## Needed Setup

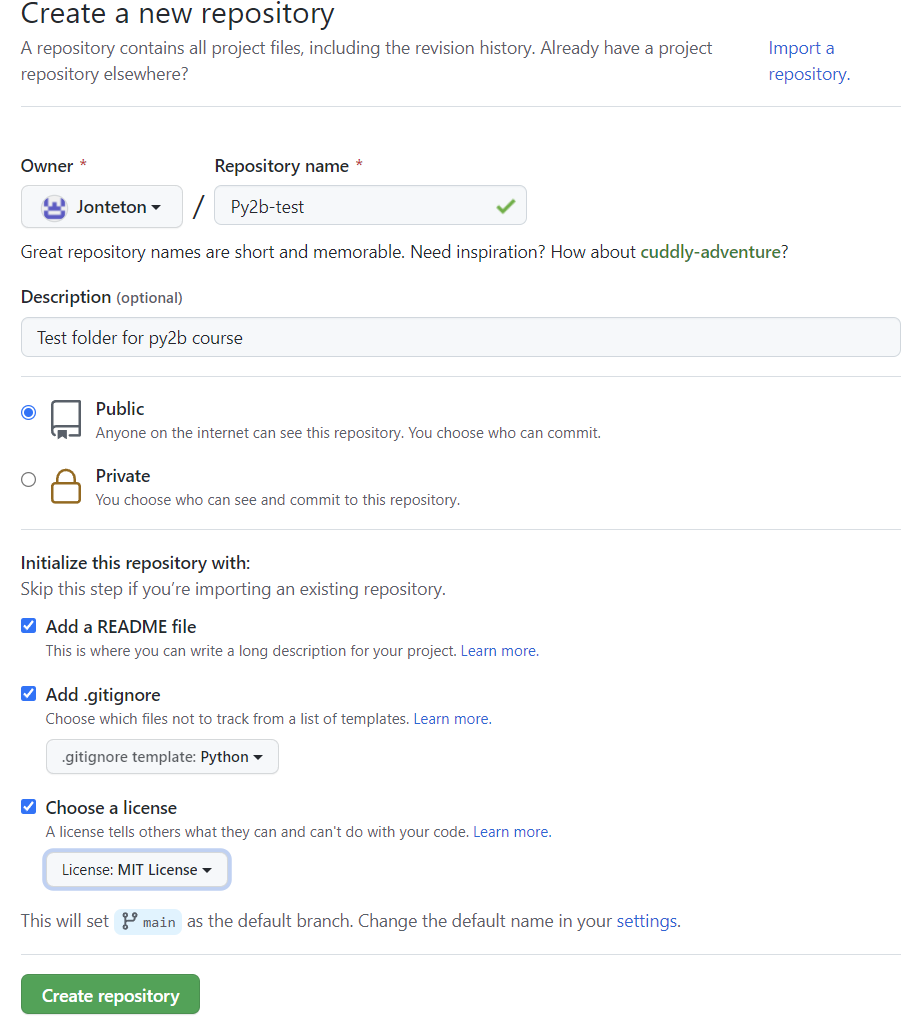
* Working installation of Git on your computer and access to tools such as Git Bash/Windows Terminal/PowerShell/ Terminal

>>> Continues on the next page

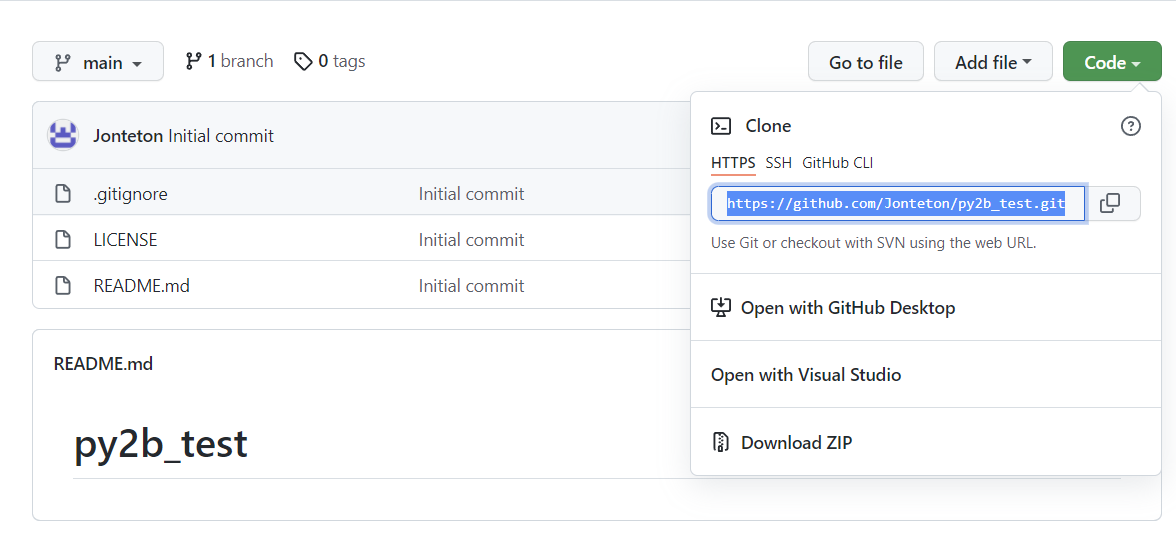
## Part 1 – Creating repository on Github and clone locally

The easiest approach is first creating the repository via Github.

* Log in on Github
* Press your account icon (top right)
* Select Your Repositories
* Press the shiny green New button
* Fill out the following information blocks:
  + Repository name (Go with Py2b\_firstname\_lastname not py2b-test like me)
  + Description
  + Select Add a README file
  + Select Add .gitignore -> Python
  + Select Choose a license -> MIT license



* Go to your repositories and select your freshly created repository
* Press the green Code button
* Copy the HTTPS address (image on next page)



* Open a Command Prompt window
* Use cd to navigate to the folder we created in exercise 1a (Jupyter Notebook Proj was the folder name) and then use git clone paste-ur-url-adress-here
* Use cd again to navigate into the folder you just cloned from github.

Text

Description automatically generated

* Done! This folder is now empty apart from a readme, license and a .gitignore file. Use the file explorer to navigate into the folder we just created and create a new folder called week1, then create an empty text file inside of it

Graphical user interface, text, application

Description automatically generated

* Go back to the command prompt we used earlier, type git status to check for any changes you have made but that is not currently in the repository here we can see that we need to add the folder we just created. Type git add week1 to do this (**OR** git add . to add all files, no need to specify name)

Text

Description automatically generated

* Type git commit -m “Added week 1” and then git push

Text

Description automatically generated

* We can now see week1 in our repository on the website

Graphical user interface, application

Description automatically generated

Repeat the same procedure but create all 5 of the weeks, so week1-week5 and push these to your repository. If you want to see a walkthrough of how I did it live check out the lecture/lecture recording. Whenever you have made a new file, changed a notebook or whatever always go through the same procedure before you wrap up for the day:

1. Git status
2. Git add files (if any new files have been created and need to be added)
3. Git commit -m “some-informative-message here”
4. Git push

## Part 2 – Adding locally existing folder as git repository (Optional/Alternative to part 1)

**This approach is not as thorough in explanation as above and is not the recommended one. The approach in Part 1 is the easiest to use, this is just optional and if you want to learn another way of doing it.**

This approach focuses rather on adding a link between a currently existing folder and github, so instead of having to clone the repo and then add the contents to the newly cloned folder, you instead just need to create a repo and a link between it and your desired folder.

Follow the great guide here: <https://docs.github.com/en/github/importing-your-projects-to-github/importing-source-code-to-github/adding-an-existing-project-to-github-using-the-command-line>

!!IMPORTANT:

Start at the title about halfway through called:  
  
**Adding a project to Github without Github CLI**

## Part 3 – Good to know commands and the next exercise

Some good to know commands (not really an exercise but good to know about):

* git commit -m “your git commit message”
* git add
* git push
* git pull
* git status
* git checkout
* git checkout -b my-new-branch

For ALL the coming exercises and the assignments, we will be working in Jupyter Notebook. Download the notebook you want to work with from the course page and then place it into the correct week in your Py2b folder. To then do the exercise open anaconda prompt, type Jupyter Notebook to run notebook, navigate into the folder you placed the exercise/assignment and open it. From there follow the instructions in the notebook:) When you are done updated your github repo according to the instructions at the end of Part1 to push the updates.