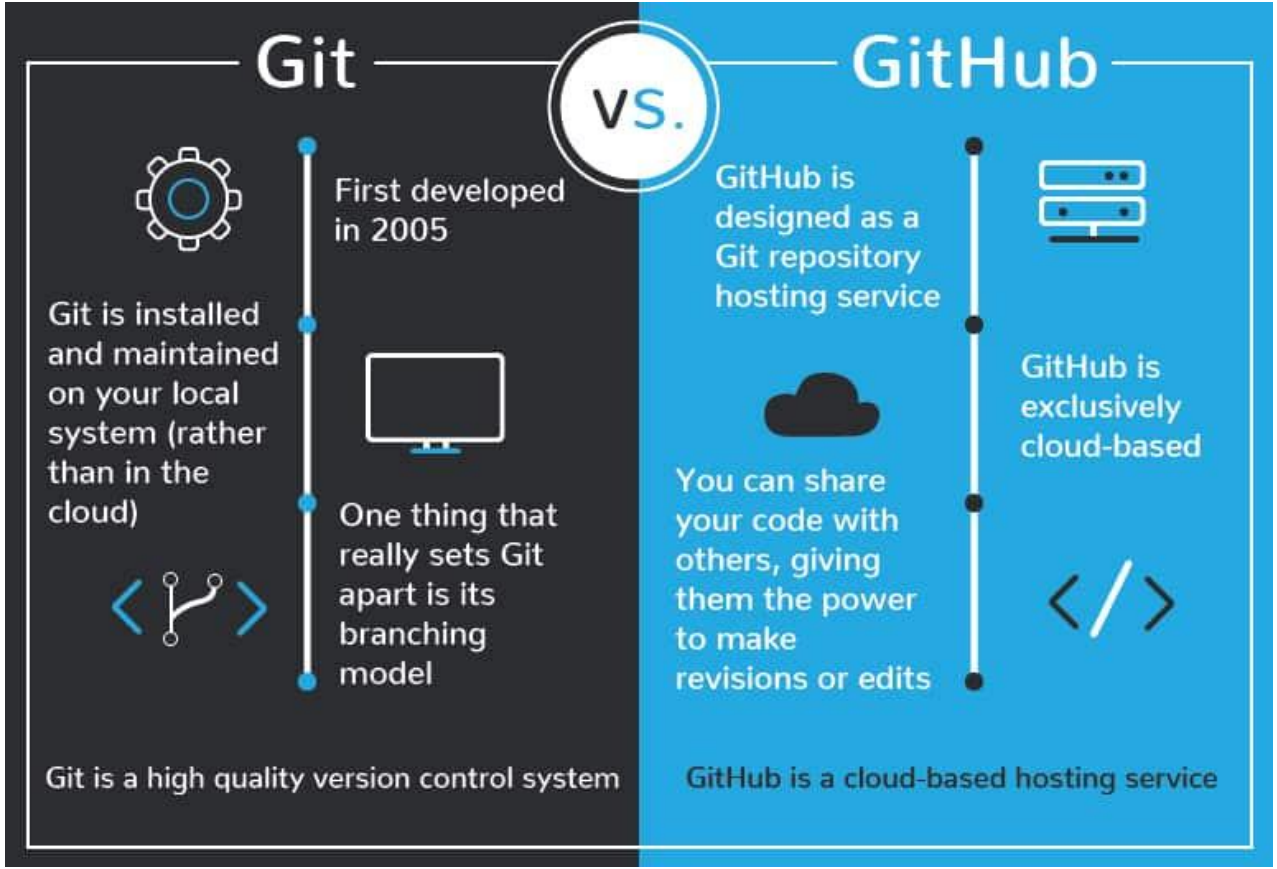
Intro to Git

12/2/2024

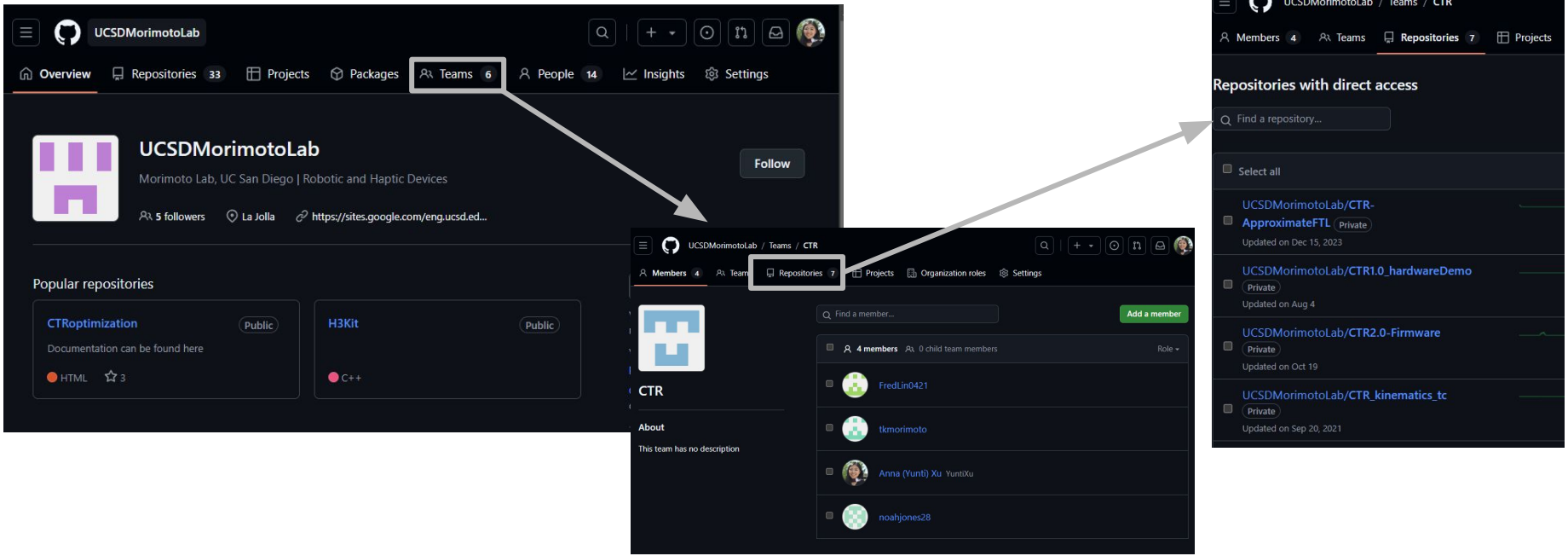
Yunti (Anna) Xu

- *What is code version control? Why should we care about it?*

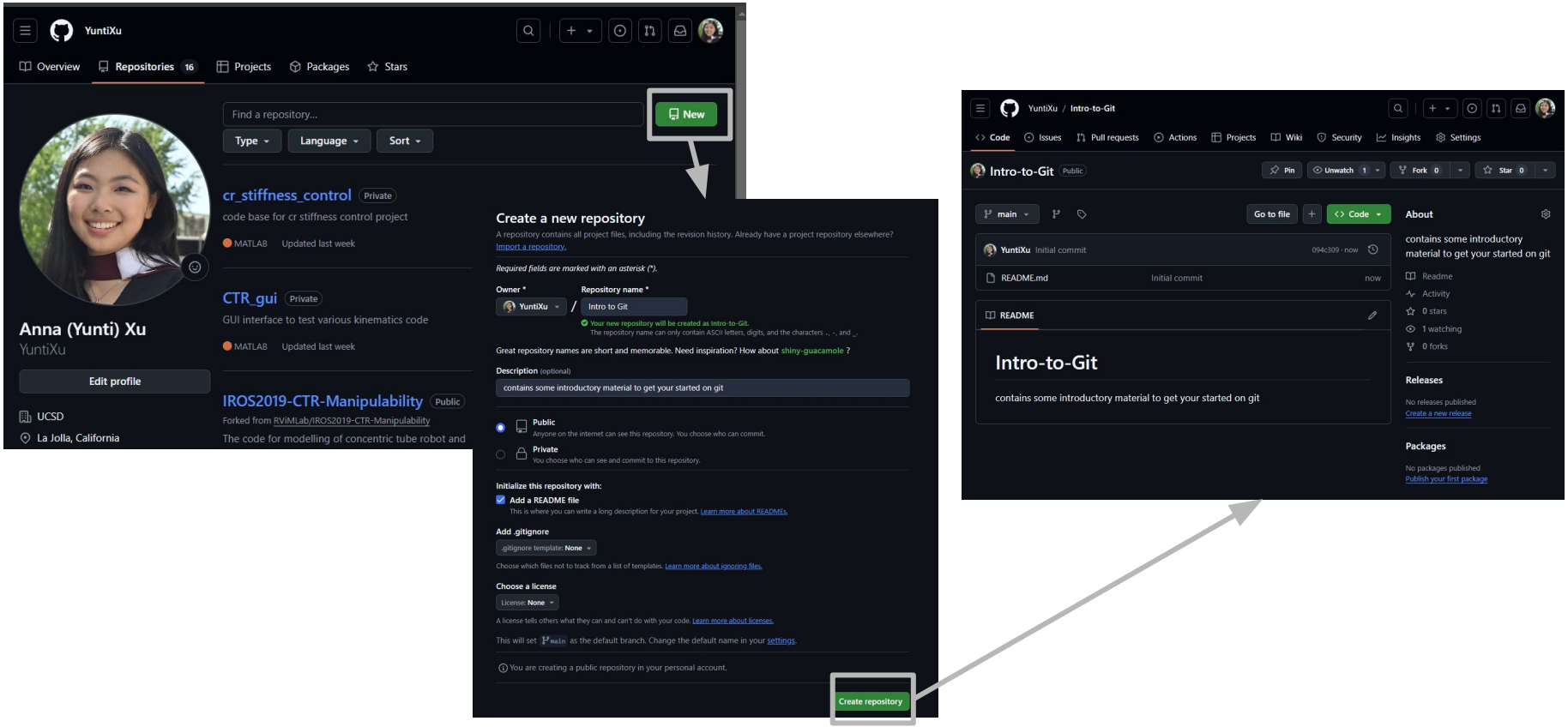


https://devmountain.com/blog/git-vs-github-whats-the-difference/

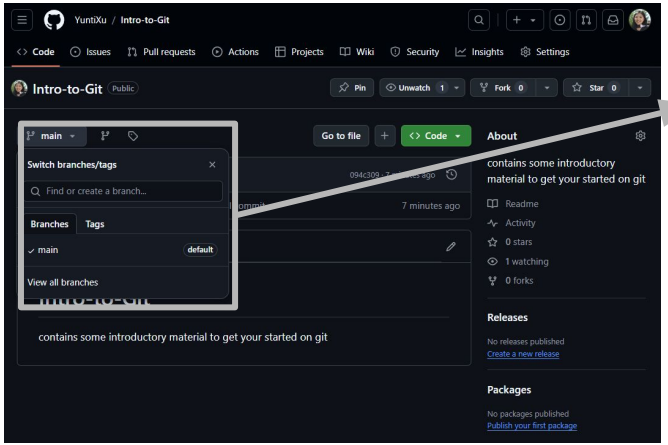
- *Where is our lab's [GitHub repository](#)?*
- *How is our GitHub repository organized?*
  - Morimoto lab members have read access to all the repositories, this means you can clone and pull our repositories (public and private) -> should restrict only to PhDs + PostDoc?
  - Aedan and Anna are have admin access -> if you need to be added to your GitHub these two are who you should go to
  - Each project should have an associated team, members should have write access to the team's repositories (well, this is really up to the PhD/PostDoc leading that particular team)

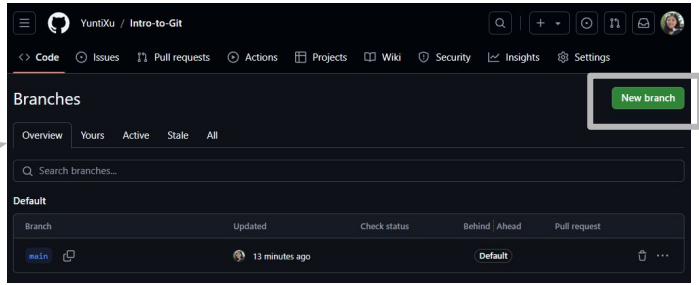


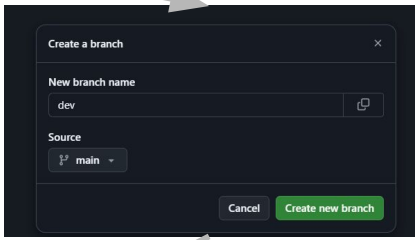
- *How to create your own GitHub repository?*

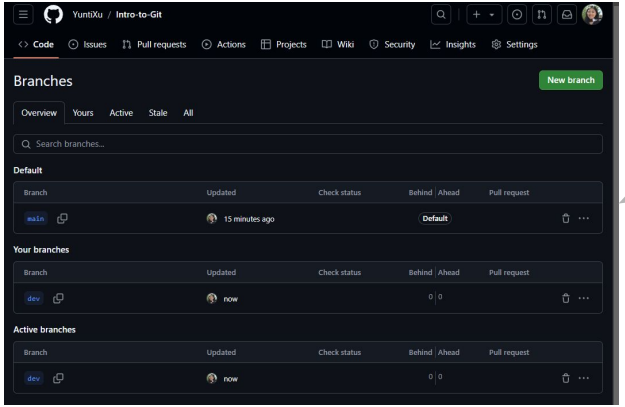


- *How to organize your GitHub repository?*
  - Ultimately it is up to you, it depends on who else might be collaborating with you on the repository. Typically, each repo has at least two branches: **main** and **dev**.
  - The **main** branch contains the “stable” version of your code, this could be the final version of the code used for your paper for instance
  - The **dev** branch is the branch you commit to (ie. update/backup) day-to-day, you will be changing the contents in this branch very frequently!

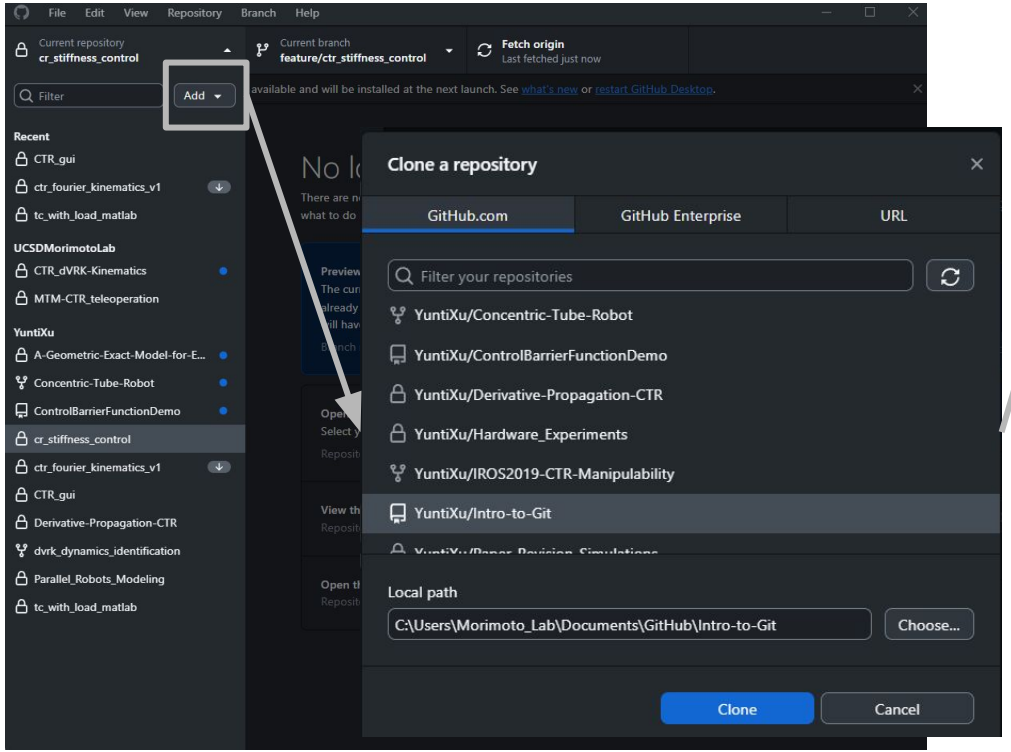


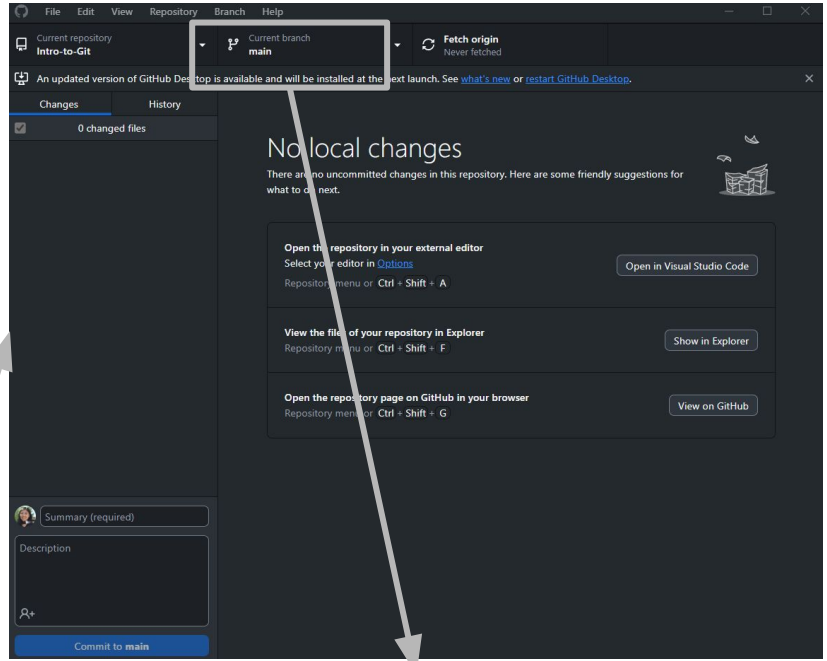


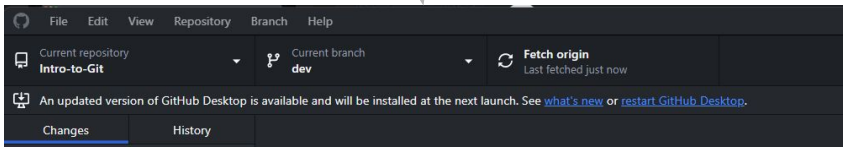




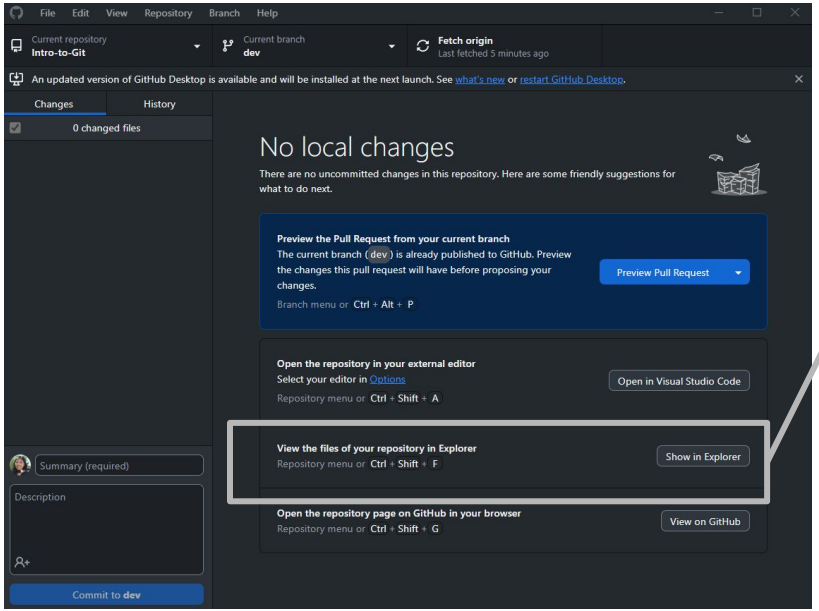
- *How to upload your code to your GitHub repository? (Windows & Mac)*
  - The easiest way is to download [GitHub desktop](), this way you interact with a GUI and don't have to deal with the inner workings of git.
  - Step #1: find and clone your repository from GitHub

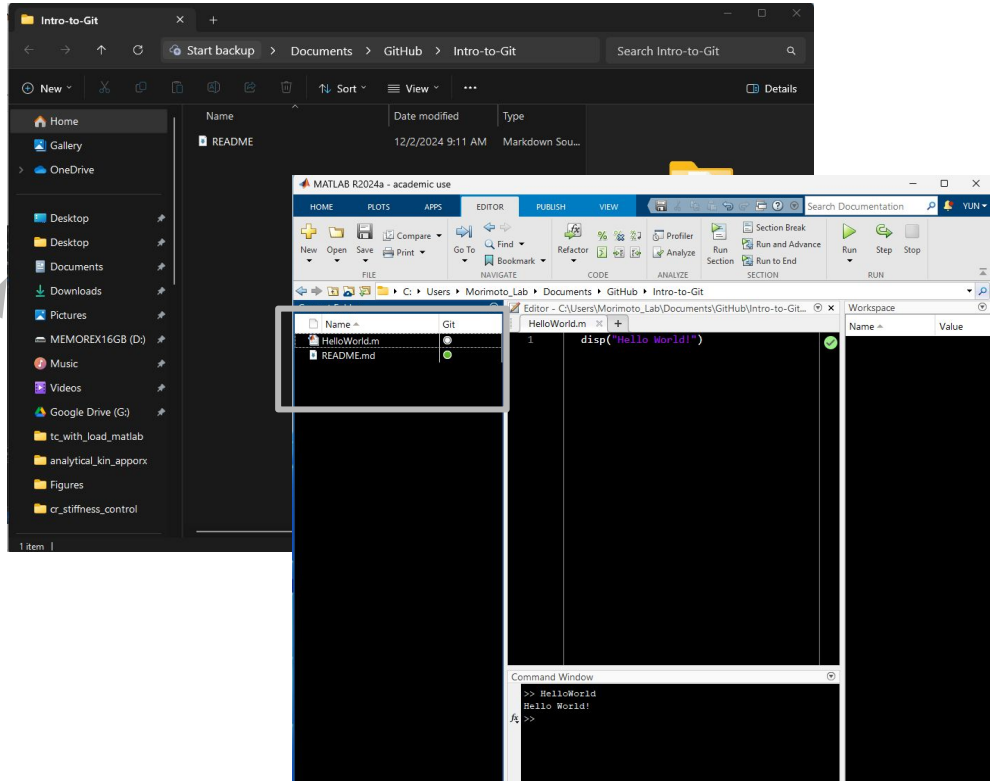




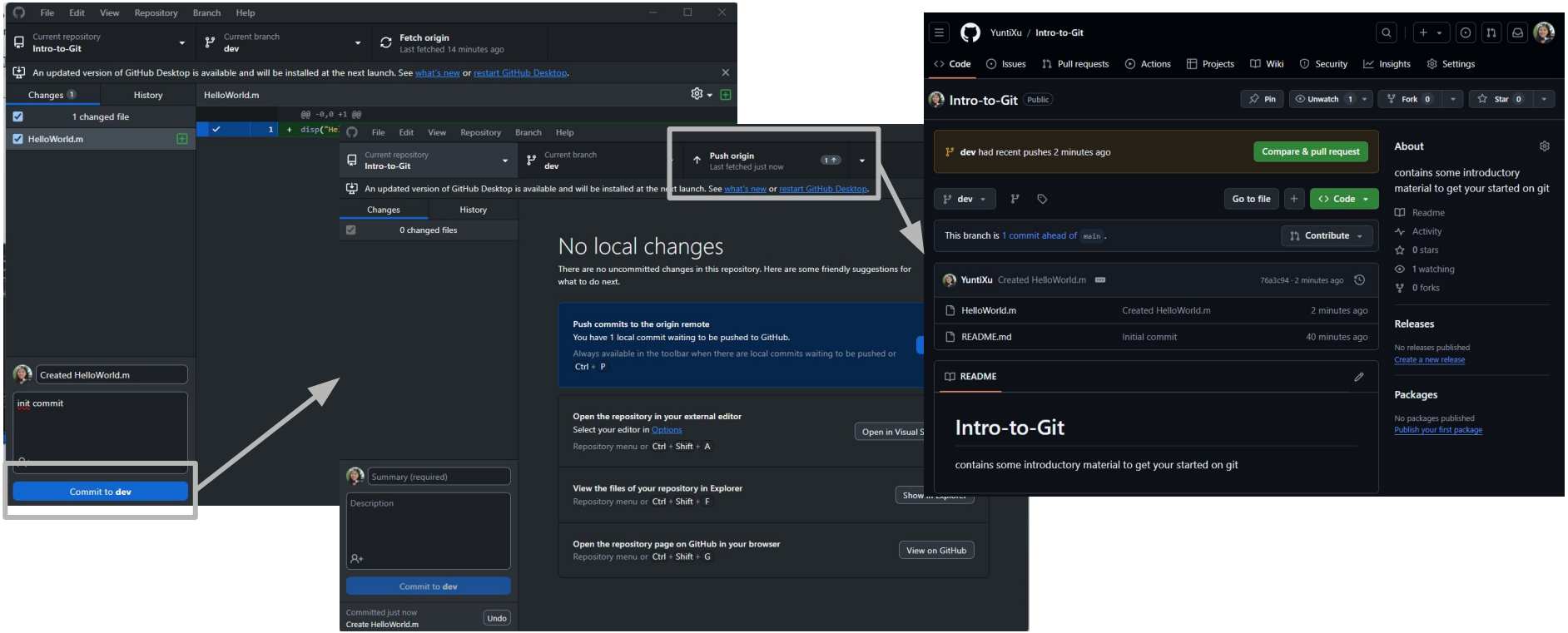


- *How to upload your code to your GitHub repository? (Windows & Mac)*
  - The easiest way is to download [GitHub desktop](), this way you interact with a GUI and don't have to deal with the inner workings of git.
  - Step #1: find and clone your repository from GitHub
  - Step #2: edit your code in your local folder (a clone of the remote GitHub repository)

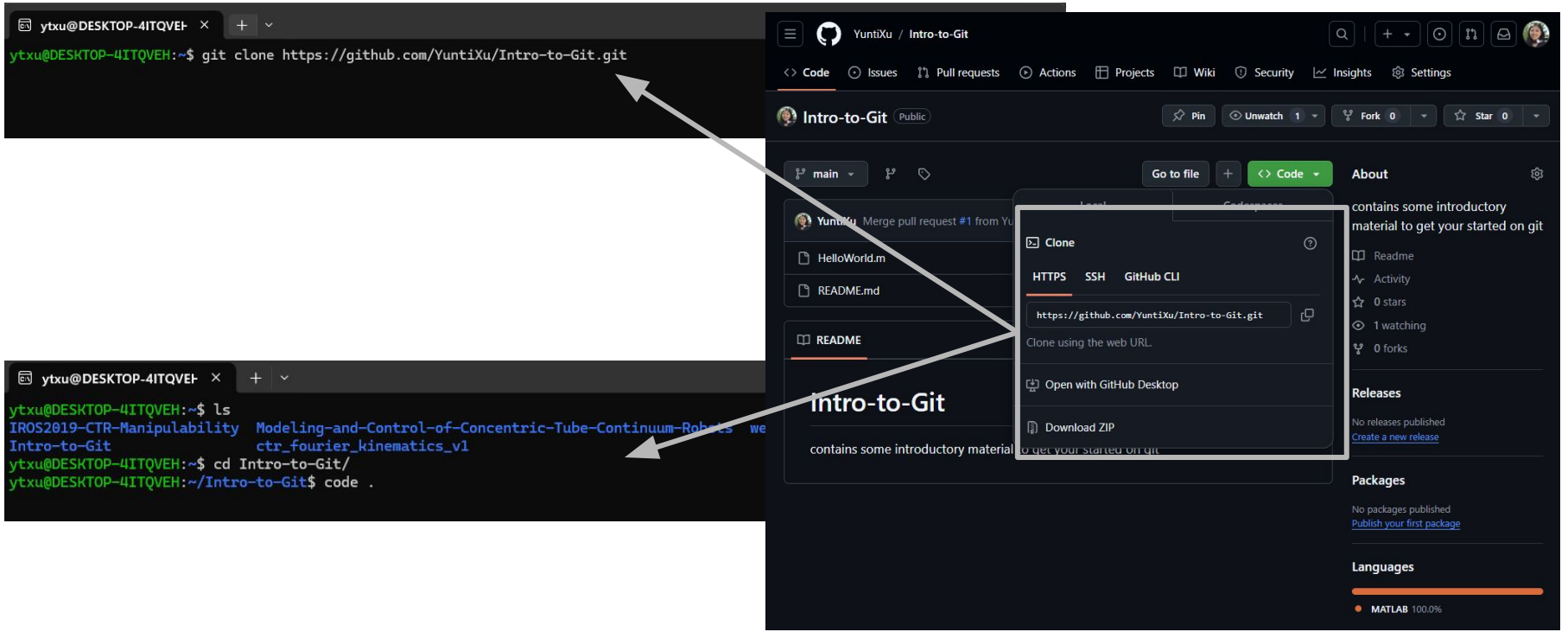




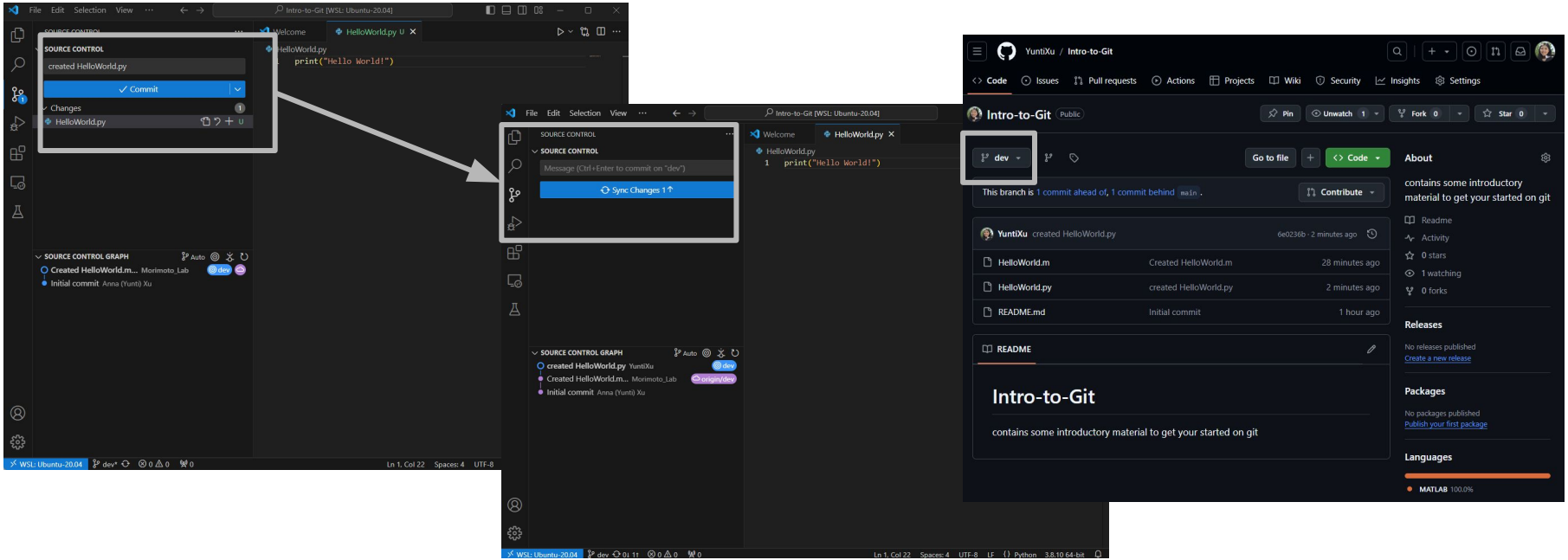
- *How to upload your code to your GitHub repository? (Windows & Mac)*
  - The easiest way is to download [GitHub desktop](), this way you interact with a GUI and don't have to deal with the inner workings of git.
  - Step #1: find and clone your repository from GitHub
  - Step #2: edit your code in your local folder (a clone of the remote GitHub repository)
  - Step #3: commit (ie. update) the repository and then push (ie. sync) with the GitHub website



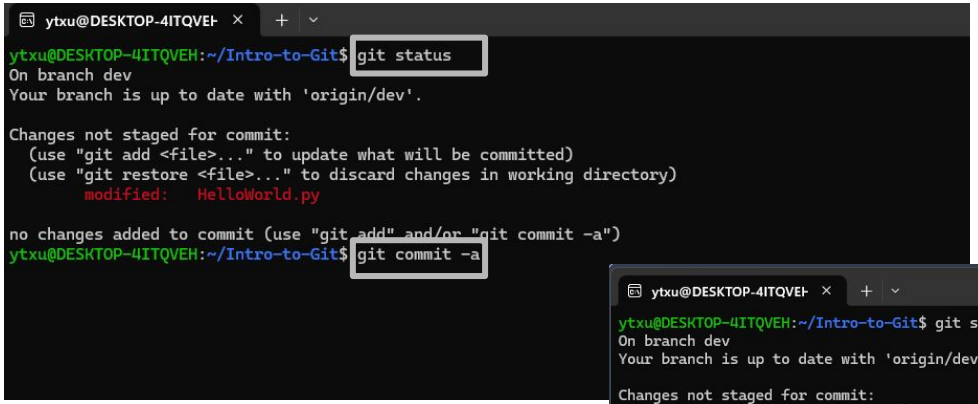
- *How to upload your code to your GitHub repository? (Linux)*
  - There are two ways depending on how hardcore you are as a person (well, using Linux is already making a statement)
  - The easier approach would be to use an [extension] in VScode, if you are using it as your code editor anyways (those of you who are using Python should already be familiar with this IDE)
    - Step #1: clone your git repository via command line

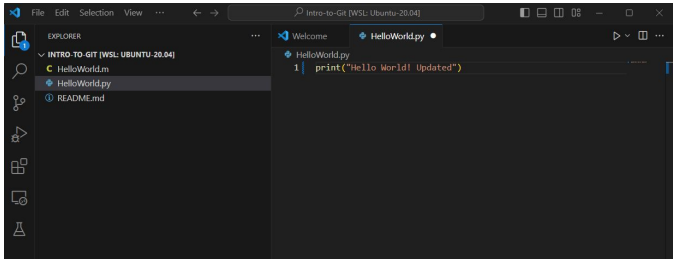


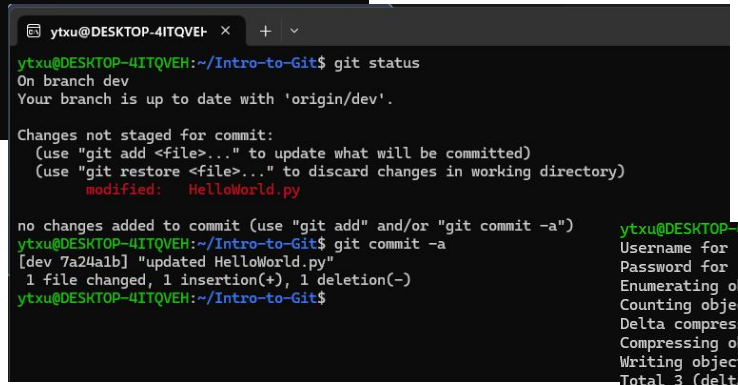
- *How to upload your code to your GitHub repository? (Linux)*
  - There are two ways depending on how hardcore you are as a person (well, using Linux is already making a statement)
  - The easier approach would be to use an [extension] in VScode, if you are using it as your code editor anyways (those of you who are using Python should already be familiar with this IDE)
    - Step #1: clone your git repository via command line
    - Step #2: edit your code in your local folder (a clone of the remote GitHub repository) + commit and push to origin directly in VScode

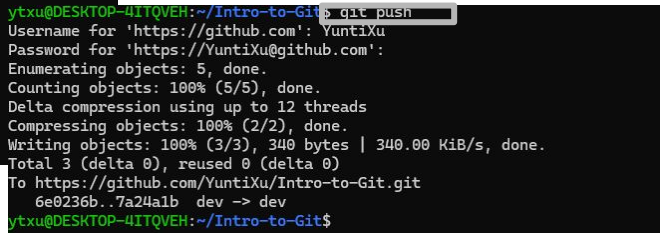


- *How to upload your code to your GitHub repository? (Linux)*
  - There are two ways depending on how hardcore you are as a person (well, using Linux is already making a statement)
  - If you using MATLAB unfortunately, you could attempt the above…but I would suggest directly do your git commands from terminal
    - Remember, you are basically doing the same thing, except instead of clicking buttons on a GUI you are writing the commands in terminal…

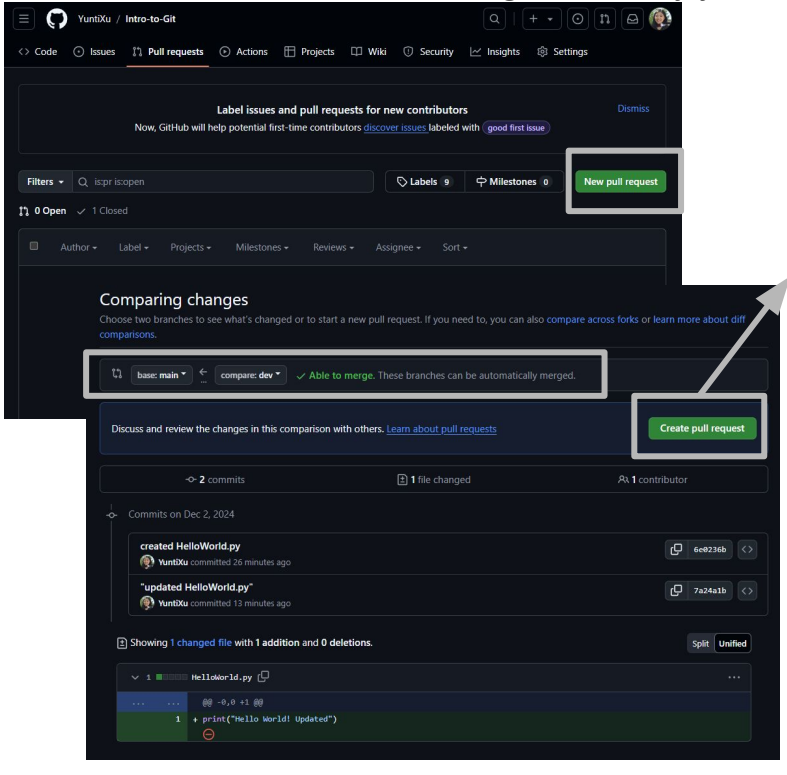


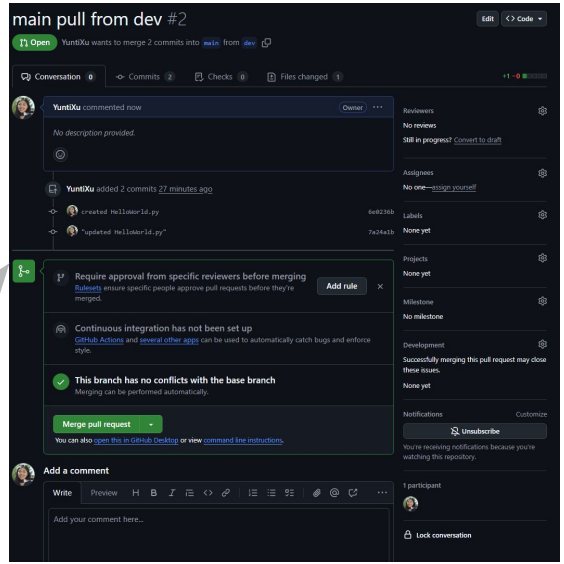


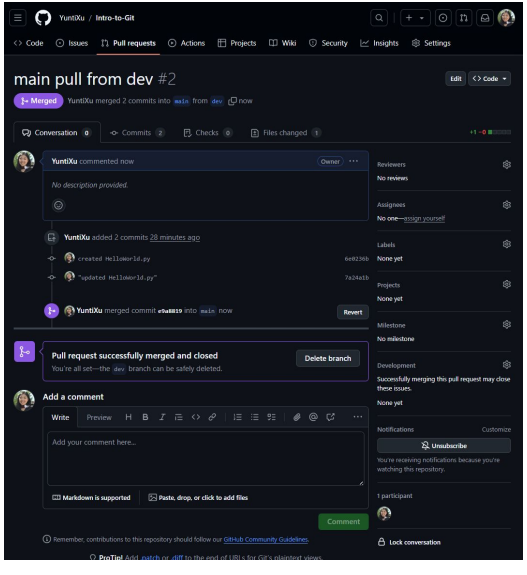




- *What happens when you are done with code development?*
  - So far, we have been working with the **dev** branch, when we are satisfied with our code, we should merge all the changes into the **main** branch
    - It is good practice to do especially in scenarios where you are developing your code in components, once you have tested a “subsystem”, have a stable version of it saved somewhere before making any more changes, that way you can always fall back on a working version!







Questions?