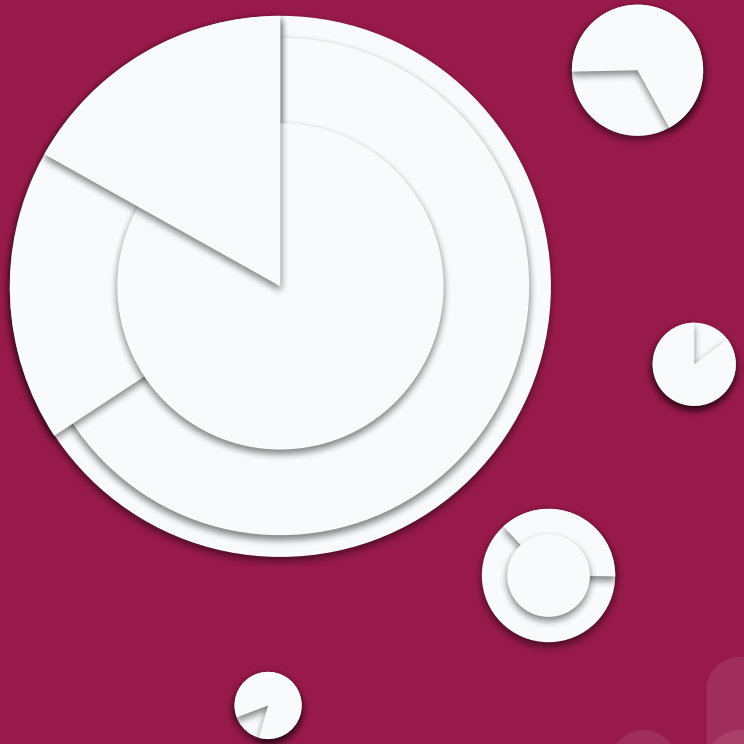
Data ScienceTech Institute

Clean IT

Hanna Abi Akl



Course Summary

- Github
- Git locally
- Git with IDE (Visual Studio Code)

Github

- A code hosting platform that lets you version your code and collaborate on projects with others
- Create a Github account here
- Familiarize yourself with your profile interface (star, pin, contribution activity)

Github

- Create a new repository
- Add a description
- Add a README file.md file
- Markdown cheat sheet here

Github

- Try adding files to your repository
- Try “saving” (committing) your files
- Can you add a jupyter notebook to your repository?
- Essential files to have in every repository
  - README.md (to explain the project)
  - Requirements.txt (to allow others to reproduce project)

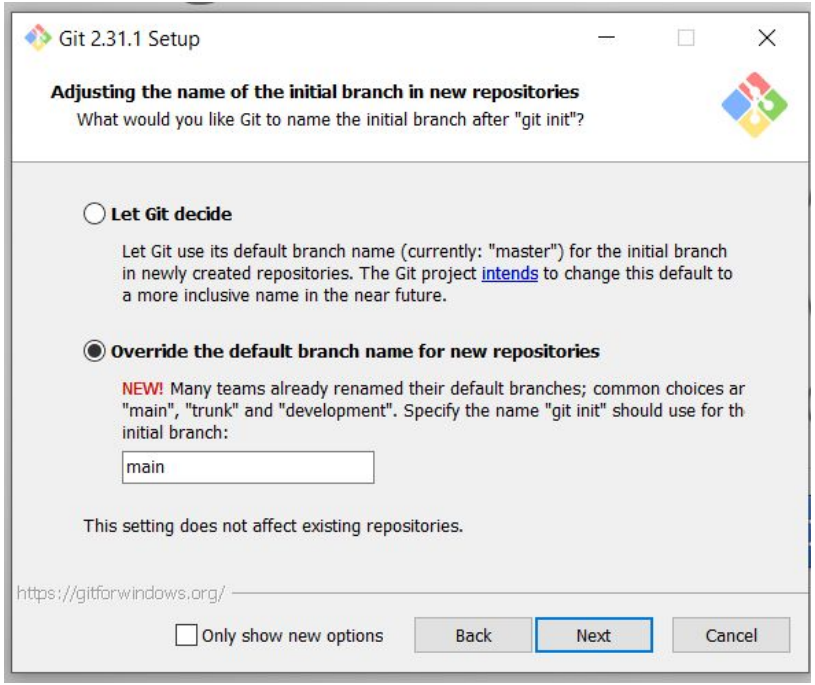
Github

- “pip install -r requirements.txt” command to install project dependencies locally from a requirements.txt file
- “pip freeze > requirements.txt” command to generate all dependencies relevant to a project and write them in the requirements.txt file

Data ScienceTech Institute

Git locally

- Git is a software for tracking changes in any set of files
- Install Git for Windows here
- During installation, override the default branch name to “main” (identically to Github)



Git locally

- Configure global variables (name and email)
  - git config --global user.name “<your_name>”
  - git config --global user.email “<your_email>”
  - git config --list

Git locally

- Create a local project and track it with git
  - git init (inside project directory)
  - git status (to track any changes in the project like new file creation)

Data ScienceTech Institute

Git locally

- Stage a file
  - git add <filename>
  - git add . (to add multiple changes performed instead of one by one)
- Commit a file
  - git commit -m <commit_message>
- Check commit history
  - git log

Git locally

- Git process
  - Modify file -> Stage file -> Commit file
- To reset/cancel your last commit
  - git reset --soft HEAD~1

Git locally

- Git branches
  - By default, your repository has one branch named “main”
  - You can create a branch off main (make a copy of it) to work on without affecting the main branch in production
  - If another person makes changes to main while you work, you can pull in those changes

Git locally

- To create a new branch
  - git branch <branch_name>
- To switch to the new branch
  - git checkout <branch_name>
- To delete a branch
  - git branch -d <branch_name>
- To get a graphical view of all branches and commits
  - git log --graph --oneline --all

Git with IDE (Visual Studio Code)

- Use git functionalities in Visual Studio Code to practice linking a project to Github
- Benefits
  - Show your work
  - Backup your code
  - Version your code
  - Collaborate with others

Git with IDE (Visual Studio Code)

- To remove files you don’t want to push to Github
  - Create a .gitignore file locally
  - Inside the .gitignore add the file types you don’t want to push
  - For example, *.ipynb will prevent Jupyter notebooks from being pushed
  - You can also do this for large data sets or folders: data/*

Scenario 1: Individual Work

- On Github
  - Create a new project repository
  - Get the link of the project in your repository
- On local workstation
  - Clone your forked repository (*git clone projectlink*)
  - Go to the project folder (*cd projectname*)
  - Open project with IDE (e.g. VSCode)

Scenario 1: Individual Work

- **On VSCode**
  - **Install the Github Pull Requests and Issues extension (first time only)**
  - **Sign in with your Github account from the extension (first time only)**
  - **Create remote to read from project repository (*git remote add upstream projectlink*)**
  - **Verify you are on main branch (*git checkout main*)**
  - **Sync your forked copy with original project (*git pull upstream main && git push origin main*)**

Scenario 1: Individual Work

- **On VSCode**
  - **Activate your virtual environment**
  - **Create a branch to work on (***git checkout -b yourbranchname***)**
  - **Track changes (***git status***)**
  - **Add changes to your working branch (***git add filename* **or** *git add .* **to add all changes)**
  - **Save changes on your working branch (***git commit -m "commit message"***)**
  - **Create a Pull request to add changes to original project (***git push -u origin yourbranchname***)**

Scenario 1: Individual Work

- **On Github**
  - **A Pull Request will be created on the project**
  - **You have to validate PR**
- **On VSCode**
  - **Switch to your main branch (*git checkout main*)**
  - **Sync latest changes from original project to your local environment (*git pull upstream main && git push origin main*)**

Scenario 2: Collaborative Work

- On Github
  - Create a fork of an existing project
  - Get the link of the forked project in your repository
- On local workstation
  - Clone your forked repository (*git clone projectlink*)
  - Go to the project folder (*cd projectname*)
  - Open project with IDE (e.g. VSCode)

Scenario 2: Collaborative Work

- **On VSCode**
  - **Install the Github Pull Requests and Issues extension (first time only)**
  - **Sign in with your Github account from the extension (first time only)**
  - **Verify you are on main branch (*git checkout main*)**
  - **Sync your forked copy with original project (*git pull upstream main && git push origin main*)**

Scenario 2: Collaborative Work

Data ScienceTech Institute

- **On VSCode**
  - **Activate your virtual environment**
  - **Create a branch to work on (***git checkout -b yourbranchname***)**
  - **Track changes (***git status***)**
  - **Add changes to your working branch (***git add filename* **or** *git add .* **to add all changes)**
  - **Save changes on your working branch (***git commit -m "commit message"***)**
  - **Create a Pull request to add changes to original project (***git push --set-upstream origin yourbranchname***)**

Scenario 2: Collaborative Work

- On Github
  - A Pull Request will be created on the original project
  - Project maintainers will validate PR
- On VSCode
  - Switch to your main branch (*git checkout main*)
  - Sync latest changes from original project to your forked copy (*git pull upstream main && git push origin main*)

Additional Reading

- Tutorial to contribute to a Github project (here)
- Article summarizing Git branching (here)
- Github documentation on resolving merge conflicts (here)
- Article on using Conda effectively (here)
- Working with Github in VSCode (here)
- Tutorial to connect Github to VSCode (here)