**Exercises: Day 0**

1. *Create an ssh-key and add it to your github account*
2. *Create a repository on github. Choose name and description as you like*
3. *Decide if you want it to be public or private*
4. *Initialize your repository with README, license and gitignore*
5. *Clone your repository*

Solutions:

1. Lecture
2. Lecture
3. Lecture
4. Lecture
5. git clone <yourRepository>

**Exercises: Day 1**

1. *Navigate to your git repository and create a new file (Feel free to chose any file you like, just make sure the file suffix is not within .gitignore)*
2. *Add some code or text, stage your changes and make sure everything worked.*
3. *Commit your changes. Check the history if your can see your commit*
4. *Add some changes to your file. Is it possible to view your changes? Stage and commit the altered file.*
5. *Upload all changes to github. Navigate to github and make sure that your repository is updated.*

Solutions:

1. Lecture
2. Lecture
3. *git add -A / git add /path/to/file  
   git commit -m “Your First Commit”  
   git log*
4. *git diff / git diff /path/to/file  
   git add -A / git add -u / git add /path/to/file  
   git commit -m “Altered corresponding file”*
5. *git push*

**Exercises: Day 2**

1. *Create a new branch called "development" in your private repository*
2. *Add a new file in this branch*
3. *Create a Pull Request for this file (Run all necessary git commands before)*
4. *Merge the Pull Request into main/master*
5. *Clone the official course repository if not already done (*[*https://github.com/Pierian-Data/Git-and-GitHub-Zero-to-Hero*](https://github.com/Pierian-Data/Git-and-GitHub-Zero-to-Hero)*)*
6. *Create a new branch called yourSurname\_test*
7. *Copy the errorneous file from the official repo, fix it and create a PR.*
8. *What is the difference between your own portfolio and the official one?*

Solutions:

1. *git branch development*
2. Lecture
3. *git switch development  
   git add -A  
   git commit -m “Commit for PR”  
   git push / git push origin development*\*Create PR\*
4. Lecture
5. *git clone <URL>*
6. *git branch yourSurname\_test*
7. *git switch yourSurname\_test*\*Fix Error\*  
   *git add /path/to/file*  
   *git commit -m “Fixed error”  
   git push*\*Create PR\*
8. You cannot merge this PR on your own as you do not have the rights to do so

**Exercises: Day 3**

1. Create a commit with an error in the commit message. Fix the commit message.
2. You don't want to push this commit. Delete it.
3. Go back to the first commit you created, continue working from it (add some change)
4. Reset your git environment to the state it was before starting this exercise.

Solutions:

1. \*Change some files / add something new\*  
   *git add -A  
   git commit -m “Changed something”  
   git commit --amend -m "Changed something"*
2. *git rebase -i ~2*\*Drop corresponding commit\* / Lecture  
   Alternative: *git reset --hard HEAD~1*
3. *git revert --no-commit <FirstCommitID>  
   \*Change something\*  
   git commit -m “Back to first Commit”*
4. *git reset --hard HEAD~1 / git revert <CommitID of last commit before this exercise>*

**Exercises: Day 4**

*1. Add a workflow to greet new users:*

*2. Set the text for the first issue to : Hi! Great first Issue!*

*3. Set the text for the first PR to : Hi! Great first PR!*

*1. Add new folders to your reposiotry: "src", "config"*

*2. Add an automatic PR tagger to tag all "root" file changes as Root, all "src" file changes as Src and all "config" file changes as Config*

*3. Add a file to src and config, respectively (Dont forget to change to a new branch first)*

*4. Create the PR and verify that it is tagged correctly.*

Solutions:

name: Greetings

on: [pull\_request\_target, issues]

jobs:

greeting:

runs-on: ubuntu-latest

permissions:

issues: write

pull-requests: write

steps:

- uses: actions/first-interaction@v1

with:

repo-token: ${{ secrets.GITHUB\_TOKEN }}

issue-message: "*Hi! Great first Issue!*"

pr-message: "*Hi! Great first PR!*"

label.yml:

name: Labeler

on: [pull\_request]

jobs:

label:

runs-on: ubuntu-latest

permissions:

contents: read

pull-requests: write

steps:

- uses: actions/labeler@v4

with:

repo-token: "${{ secrets.GITHUB\_TOKEN }}"

labeler.yml

Root:

- '\*'

src:

- src/\*

config;

- config/\*