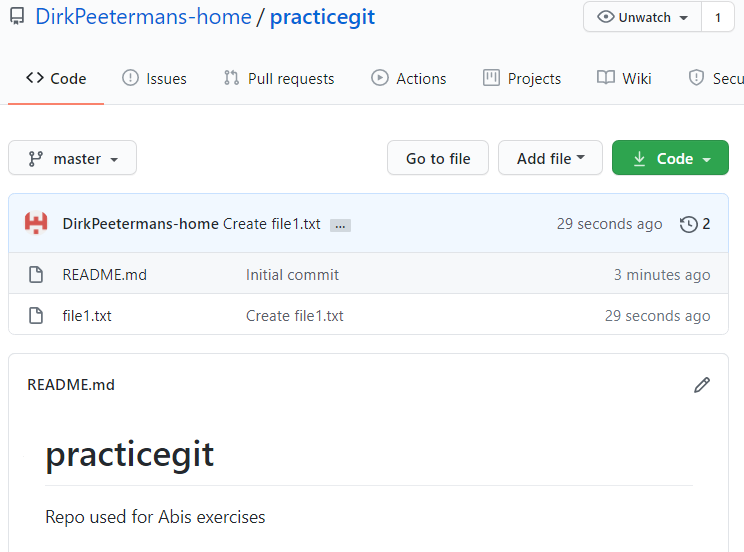
***Terminal and Git – Exercises***

**General Setup**

Create a file called solutionsGit<Yourname>.txt (e.g. solutionsGitSandy.txt). Copy all entered statements together with the number of the exercise in it. Make sure to push it to GitHub at the end of part A, and/or mail it to your instructor.

**A) Basic Git Commands**

1. Create a new repo called “practicegit” in your public GitHub account. Initialize it with a readme file. Add a new file “file1.txt” to it, with content “hello from file 1.”.



2. Open GitBash. Check out which directory you are in.

Create a new directory structure JavaReskilling -> GitProjects on your computer. Use “ls” to check whether the

directories were created correctly. Move to the GitProjects directory, and initialize git there.

dpcvaptp@PC1 MINGW64 ~/Desktop

$ mkdir JavaReskilling

dpcvaptp@PC1 MINGW64 ~/Desktop

$ cd JavaReskilling/

dpcvaptp@PC1 MINGW64 ~/Desktop/JavaReskilling

$ mkdir GitProjects

dpcvaptp@PC1 MINGW64 ~/Desktop/JavaReskilling

$ cd GitProjects/

dpcvaptp@PC1 MINGW64 ~/Desktop/JavaReskilling/GitProjects

$ git config --global user.name "Dirk-Peetermans-home"

dpcvaptp@PC1 MINGW64 ~/Desktop/JavaReskilling/GitProjects

$ git config --global user.email [peetermans.dirk@telenet.be](mailto:peetermans.dirk@telenet.be)

dpcvaptp@PC1 MINGW64 ~/Desktop/JavaReskilling/GitProjects

$ git init

Initialized empty Git repository in C:/Users/dpcvaptp/Desktop/JavaReskilling/GitProjects/.git/

3. Clone the repository created in exercise 1 into this directory.

dpcvaptp@PC1 MINGW64 ~/Desktop/JavaReskilling/GitProjects (master)

$ git clone https://github.com/DirkPeetermans-home/practicegit

Cloning into 'practicegit'...

remote: Enumerating objects: 6, done.

remote: Counting objects: 100% (6/6), done.

remote: Compressing objects: 100% (3/3), done.

remote: Total 6 (delta 0), reused 0 (delta 0), pack-reused 0

Unpacking objects: 100% (6/6), 1.27 KiB | 4.00 KiB/s, done.

4. Create a new branch called “firstbranch” and switch to it. Try to do this in 1 command!

dpcvaptp@PC1 MINGW64 ~/Desktop/JavaReskilling/GitProjects (master)

$ cd practicegit/

dpcvaptp@PC1 MINGW64 ~/Desktop/JavaReskilling/GitProjects/practicegit (master)

$ git branch firstbranch

dpcvaptp@PC1 MINGW64 ~/Desktop/JavaReskilling/GitProjects/practicegit (master)

$ git checkout firstbranch

Switched to branch 'firstbranch'

dpcvaptp@PC1 MINGW64 ~/Desktop/JavaReskilling/GitProjects/practicegit (firstbranch)

$ ls -l

total 2

-rw-r--r-- 1 dpcvaptp 197121 20 sep 18 11:26 file1.txt

-rw-r--r-- 1 dpcvaptp 197121 45 sep 18 11:26 README.md

dpcvaptp@PC1 MINGW64 ~/Desktop/JavaReskilling/GitProjects/practicegit (firstbranch)

$ git status

On branch firstbranch

nothing to commit, working tree clean

5. Add a file called “file2.txt” to the branch (you can make up some content) and stage it. Try to commit it without adding a commit message. Use “git status” regularly to see what is happening as from now!

dpcvaptp@PC1 MINGW64 ~/Desktop/JavaReskilling/GitProjects/practicegit (firstbranch)

$ echo "Content for file2" > ./file2.txt

dpcvaptp@PC1 MINGW64 ~/Desktop/JavaReskilling/GitProjects/practicegit (firstbranch)

$ ls -l

total 3

-rw-r--r-- 1 dpcvaptp 197121 20 sep 18 11:26 file1.txt

-rw-r--r-- 1 dpcvaptp 197121 18 sep 18 11:32 file2.txt

-rw-r--r-- 1 dpcvaptp 197121 45 sep 18 11:26 README.md

dpcvaptp@PC1 MINGW64 ~/Desktop/JavaReskilling/GitProjects/practicegit (firstbranch)

$ git status

On branch firstbranch

Untracked files:

(use "git add <file>..." to include in what will be committed)

file2.txt

nothing added to commit but untracked files present (use "git add" to track)

dpcvaptp@PC1 MINGW64 ~/Desktop/JavaReskilling/GitProjects/practicegit (firstbranch)

$ git add file2.txt

warning: LF will be replaced by CRLF in file2.txt.

The file will have its original line endings in your working directory

dpcvaptp@PC1 MINGW64 ~/Desktop/JavaReskilling/GitProjects/practicegit (firstbranch)

$ git status

On branch firstbranch

Changes to be committed:

(use "git restore --staged <file>..." to unstage)

new file: file2.txt

dpcvaptp@PC1 MINGW64 ~/Desktop/JavaReskilling/GitProjects/practicegit (firstbranch)

$ git commit

Aborting commit due to empty commit message.

dpcvaptp@PC1 MINGW64 ~/Desktop/JavaReskilling/GitProjects/practicegit (firstbranch)

$ git commit -m "Committing file2.txt"

[firstbranch e643bc3] Committing file2.txt

1 file changed, 1 insertion(+)

create mode 100644 file2.txt

dpcvaptp@PC1 MINGW64 ~/Desktop/JavaReskilling/GitProjects/practicegit (firstbranch)

$ git status

On branch firstbranch

nothing to commit, working tree clean

6. Push the result to your remote repository, and check via GitHub whether everything is there.

dpcvaptp@PC1 MINGW64 ~/Desktop/JavaReskilling/GitProjects/practicegit (firstbranch)

$ git push origin firstbranch

Enumerating objects: 4, done.

Counting objects: 100% (4/4), done.

Delta compression using up to 2 threads

Compressing objects: 100% (2/2), done.

Writing objects: 100% (3/3), 340 bytes | 340.00 KiB/s, done.

Total 3 (delta 0), reused 0 (delta 0), pack-reused 0

remote:

remote: Create a pull request for 'firstbranch' on GitHub by visiting:

remote: https://github.com/DirkPeetermans-home/practicegit/pull/new/firstbranch

remote:

To https://github.com/DirkPeetermans-home/practicegit

\* [new branch] firstbranch -> firstbranch

**B) Undoing things**

1. Add two more files file3.txt and file4.txt. Stage both of them. Unstage file4.txt again.

2. Commit and check git log. Change the commit message. Revert the last commit such that file3.txt is deleted from your directory.

3. Try to undo your revert.

4. Stage file4.txt again and commit. Undo the commit, but this time, file4.txt should still stay in your directory.

5. Push to your remote repository again. Check that file4.txt is not present there.

**C) Some more Git**

1. Make some changes to file2.txt directly in the remote repository. Pull the changes into your local repository. Check the contents of file2.txt

2. Create a new branch “secondbranch” in your local repository, and make sure it contains everything from firstbranch.

3. Delete firstbranch both in your local and remote repository.

4. Switch to master. Add a file called file2.txt there too. Add in some text which is different than the content of the file in secondbranch.

5. Stage and commit these changes. Merge the master branch with secondbranch. What happened? Try to solve this.

Hint:

https://openclassrooms.com/en/courses/3321726-manage-your-code-with-git-and-github/3323386-resolving-conflicts

<https://opensource.com/article/20/4/git-merge-conflict>

6. Checkout the following tutorial:

https://openclassrooms.com/en/courses/3321726-manage-your-code-with-git-and-github/3323406-ignoring-files

Create files file5.txt and secret.txt in secondbranch. Stage them and commit. Push the changes to your remote repository, but make sure that secret.txt is ignored.

7. Create a subdirectory “subdir” of the practicegit directory and put some files in there. Make sure that everything is pushed to your remote repository.

**D) Working in a team**

1. Navigate back to your gitprojects directory.

Make a clone of https://github.com/sschillebeeckx/gitex .

Navigate to it, you can keep using the master branch.

2. Edit the hello.txt file, where you add a line containing “Hello from <your name>”.

3. Commit, and try to push your file to the remote repository? Why does (n’t) work? Solve this.

4. Again try to push your solution. At the end we should have a file with a hello message from everybody in the

repository.