

Mathieu Wassmuth, Imanuel Leiserowitz

Info3/SoSe23

Lab01 – Git Repo Setup

In this lab assignment, we set up a Git repository and tried out different operations with it. Git is a powerful version control system that allows you to track changes to your code over time and collaborate with other developers on a project.

During the lab, we learned how to create a Git repository, add files to it, commit changes, and push those changes to a remote repository on GitHub. We also explored different Git workflows, such as the centralized workflow, feature branch workflow, Gitflow workflow, and forking workflow.

In addition, we tested out different Git operations, such as merging, rebasing, and fast-forwarding branches, resolving merge conflicts, and switching between branches. These operations allowed us to manage and organize changes to our codebase effectively.

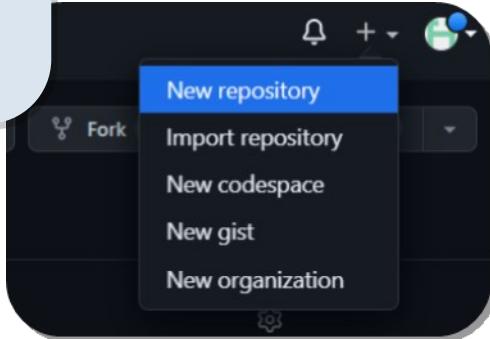
By the end of this lab assignment, we gained a solid understanding of Git and its various features, which will be invaluable in our future development projects.

Setting up the GitHub repository

Setting up the repository with GitHub, and adding our accounts to be able to access the rep using the HTTPS repo URL went quick and without any trouble

Cloning the rep to a local device

Using the Windows cmd we entered the desired Location, where we want our rep to be cloned to. With the command “**git clone <rep URL>**” we cloned the repository to our local machine.



```
I:\HTW\Semester_3\info3\repository>git clone https://github.com/leunami420/info3.git
Cloning into 'info3_lab01_gruppe_BlackMamba'...
info: please complete authentication in your browser...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (3/3), done.

I:\HTW\Semester_3\info3\repository>
```

Creating a local file and pushing it to the rep

Name	Änderungsdatum
.git	12.04.2023 18:07
README.md	12.04.2023 18:07
TESTDATEI.txt	12.04.2023 18:10

We created the file **TESTDATEI.txt** on a local device

Using „**git add TESTDATEI.txt**“ we added the file to our local rep

```
I:\HTW\Semester_3\info3\repository\info3_lab01_gruppe_BlackMamba>git add TESTDATEI.txt
I:\HTW\Semester_3\info3\repository\info3_lab01_gruppe_BlackMamba>git commit -m 'testtesttest'
[main e0723a8] 'testtesttest'
 1 file changed, 1 insertion(+)
 create mode 100644 TESTDATEI.txt

I:\HTW\Semester_3\info3\repository\info3_lab01_gruppe_BlackMamba>git push
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 8 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 281 bytes | 281.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/leunami420/info3_lab01_gruppe_BlackMamba.git
 0000cf0..e0723a8 main -> main
```

With „**git commit -m 'commit message'**“ we created a new commit

After that, we could use „**git add**“ to push the updated rep to GitHub

```
'testtesttest'
└─ main
  └─ leunami420 committed 5 minutes ago

Showing 1 changed file with 1 addition and 0 deletions.

  ▾ 1 █ TESTDATEI.txt
  ... ...
  @@ -0,0 +1 @@
  1 + HI
```

Now we are able to see the updated rep, including the changes that were made in the commit on the GitHub page

Syncing your local files with the rep.

```
I:\HTW\Semester_3\info3\repository\info3_lab01_gruppe_BlackMamba>git pull
remote: Enumerating objects: 8, done.
remote: Counting objects: 100% (8/8), done.
remote: Compressing objects: 100% (3/3), done.
remote: Total 6 (delta 2), reused 3 (delta 1), pack-reused 0
Unpacking objects: 100% (6/6), 946 bytes | 25.00 KiB/s, done.
From https://github.com/leunami420/info3_lab01_gruppe_BlackMamba
  55c9849..83f0c8c main      -> origin/main
 * [new branch]  feature-branch-test -> origin/feature-branch-test
Updating 55c9849..83f0c8c
Fast-forward
 pulltest.txt | 1 +
 1 file changed, 1 insertion(+)
 create mode 100644 pulltest.txt
```

```
I:\HTW\Semester_3\info3\repository\info3_lab01_gruppe_BlackMamba>
```

With “git pull“ we are able to sync our files with the github repository

A merge - conflict occurs!

When we tried to sync a repository where, our main branch had different sets of change that were not present in the other commit, Git did not know how to merge the changes, and gave us 3 options to resolve the merge-conflict

```
hint: You have divergent branches and need to specify how to reconcile them.
hint: You can do so by running one of the following commands sometime before
hint: your next pull:
hint:
hint:   git config pull.rebase false  # merge
hint:   git config pull.rebase true   # rebase
hint:   git config pull.ff only      # fast-forward only
hint:
hint: You can replace "git config" with "git config --global" to set a default
hint: preference for all repositories. You can also pass --rebase, --no-rebase,
hint: or --ff-only on the command line to override the configured default per
hint: invocation.
```

Trying to merge 2 files with different content

2 lines (2 sloc) | 23 Bytes

```
1 hello world  
2 change 1.1
```

[Give feedback](#)

pulltest.txt - Editor

Datei Bearbeiten Format Ansicht Hilfe
hello world
change 1.2

We created the same text file with
different contents

```
Unpacking objects: 100% (3/3), 658 bytes | 32.00 KiB/s, done.  
From https://github.com/leunami420/info3_lab01_gruppe_BlackMamba  
* 83f0c8c..dadb0ef main      -> origin/main  
Auto-merging pulltest.txt  
CONFLICT (content): Merge conflict in pulltest.txt  
Automatic merge failed; fix conflicts and then commit the result.
```

When trying to pull the repository with the other version of our textfile git failed to merge the different commits

pulltest.txt - Editor

Datei Bearbeiten Format Ansicht Hilfe

```
hello world  
<<<<< HEAD  
change 1.2  
=====  
change 1.1  
>>>> dadb0ef44e9e8d4609e217d2a600be209866d9f0
```

Creating a feature branch and merging it

```
leuwasssmuth@Mathieus-MacBook-Air info3_lab01_gruppe_BlackMamba % ls
README.md      TESTDATEI.txt    file100.txt
athieuwasssmuth@Mathieus-MacBook-Air info3_lab01_gruppe_BlackMamba % git checkout -b feature-branch-test
Switched to a new branch 'feature-branch-test'
athieuwasssmuth@Mathieus-MacBook-Air info3_lab01_gruppe_BlackMamba % git add newbranchfile.txt
fatal: pathspec 'newbranchfile.txt' did not match any files
athieuwasssmuth@Mathieus-MacBook-Air info3_lab01_gruppe_BlackMamba % touch newbranchfile.txt
athieuwasssmuth@Mathieus-MacBook-Air info3_lab01_gruppe_BlackMamba % ls
README.md      TESTDATEI.txt    file100.txt    newbranchfile.txt
athieuwasssmuth@Mathieus-MacBook-Air info3_lab01_gruppe_BlackMamba % git commit -m 'Neuer Branch mit einem neuen File names newbranchfile.txt'
[feature-branch-test cb667d4] Neuer Branch mit einem neuen File names newbranchfile.txt
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 newbranchfile.txt
athieuwasssmuth@Mathieus-MacBook-Air info3_lab01_gruppe_BlackMamba % git push -u origin feature-branch-test
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 8 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 310 bytes | 310.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
remote:
remote: Create a pull request for 'feature-branch-test' on GitHub by visiting:
remote:   https://github.com/leunami420/info3_lab01_gruppe_BlackMamba/pull/new/feature-branch-test
remote:
To https://github.com/leunami420/info3_lab01_gruppe_BlackMamba.git
 * [new branch]      feature-branch-test -> feature-branch-test
Branch 'feature-branch-test' set up to track 'origin/feature-branch-test'.
athieuwasssmuth@Mathieus-MacBook-Air info3_lab01_gruppe_BlackMamba %
```

We used **git checkout -b feature-branch-test** to create and work in a new branch called „feature-branch-test“

```
I:\HTW\Semester_3\info3\repository\info3_lab01_gruppe_BlackMamba>git log
commit cb667d4fbe9b94060125e76b7ccda366d6fad6e (HEAD -> feature-branch-test, origin/feature-branch-test)
Author: prodmatzte <wasssmuth.bus@gmail.com>
Date:   Wed Apr 12 18:53:13 2023 +0200

        Neuer Branch mit einem neuen File names newbranchfile.txt
```

Then we created a new file and added it to our branch

After that we switched back to our main branch to merge the 2 branches together

```
I:\HTW\Semester_3\info3\repository\info3_lab01_gruppe_BlackMamba>git checkout main
Switched to branch 'main'
Your branch is up to date with 'origin/main'.

I:\HTW\Semester_3\info3\repository\info3_lab01_gruppe_BlackMamba>git merge --no-ff feature-branch-test
Merge made by the 'ort' strategy.
newbranchfile.txt | 0
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 newbranchfile.txt

I:\HTW\Semester_3\info3\repository\info3_lab01_gruppe_BlackMamba>
```

We used **git merge --no-ff feature-branch-test** to merge the branches without using „fast forward“