

## Teil 1

### 1.1. Checke dein Git-Projektarchiv aus

- `git@gitlab.fh-ooe.at:sib/ebd/SXXXXXXXXXX.git`

```
sarah@DESKTOP-6LEAOT9:~/ebd1_ue$ git clone git@gitlab.fh-ooe.at:sib/ebd/S2310239008.git
Cloning into 'S2310239008'...
remote: Enumerating objects: 58, done.
remote: Counting objects: 100% (33/33), done.
remote: Compressing objects: 100% (32/32), done.
remote: Total 58 (delta 12), reused 0 (delta 0), pack-reused 25
Receiving objects: 100% (58/58), 6.32 KiB | 646.00 KiB/s, done.
Resolving deltas: 100% (15/15), done.
sarah@DESKTOP-6LEAOT9:~/ebd1_ue$ ls
S2310239008
sarah@DESKTOP-6LEAOT9:~/ebd1_ue$
```

- `git clone git@gitlab.fh-ooe.at:sib/ebd/S2310239008.git`  
`git clone <URL>` ist ein Befehl, um eine lokale Kopie eines Remote Git-Repositories zu erstellen. Dabei wird ein neues Verzeichnis erstellt und die Git-Metadaten angelegt. Unterstützt werden die Protokolle SSH und HTTPS.

### 1.2. Lege eine neue Datei „archimedes.txt“ an und übernimm diese in das zentrale Projektarchiv

- Verwende die Commit-Nachricht: `git-1.2`

```
sarah@DESKTOP-6LEAOT9:~/ebd1_ue$ cd S2310239008/
sarah@DESKTOP-6LEAOT9:~/ebd1_ue/S2310239008$ ls
Bassie.txt bassie.txt helloCMakeProject helloEBD.c helloWorld.c
sarah@DESKTOP-6LEAOT9:~/ebd1_ue/S2310239008$ touch archimedes.txt
sarah@DESKTOP-6LEAOT9:~/ebd1_ue/S2310239008$ ls
Bassie.txt archimedes.txt bassie.txt helloCMakeProject helloEBD.c helloWorld.c
sarah@DESKTOP-6LEAOT9:~/ebd1_ue/S2310239008$ git status
On branch master
Your branch is up to date with 'origin/master'.

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    archimedes.txt

nothing added to commit but untracked files present (use "git add" to track)
sarah@DESKTOP-6LEAOT9:~/ebd1_ue/S2310239008$ git add .
sarah@DESKTOP-6LEAOT9:~/ebd1_ue/S2310239008$ git commit -m "git-1.2"
[master 4b89a82] git-1.2
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 archimedes.txt
sarah@DESKTOP-6LEAOT9:~/ebd1_ue/S2310239008$ git push
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 12 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 277 bytes | 277.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0), pack-reused 0
To gitlab.fh-ooe.at:sib/ebd/S2310239008.git
   aeed2a4..4b89a82  master -> master
sarah@DESKTOP-6LEAOT9:~/ebd1_ue/S2310239008$
```

- `touch archimedes.txt`  
Der Befehl `touch <neuer_dateiname.endung>` erstellt eine neue Datei, falls sie noch nicht existiert. Ansonsten wird einfach ihr Zeitstempel aktualisiert.

- `git status`  
Zeigt den aktuellen Status von Arbeitsverzeichnis und Staging Area im Vergleich zum letzten Commit (HEAD). Er informiert darüber was gelöscht/modifiziert/hinzugefügt wurde und was sich in der Staging Area befindet und was nicht.
- `git add .`  
Mittels `git add <Datei/Verzeichnisname>` werden die Änderungen an den angegebenen Dateien und Verzeichnissen in die Staging Area übernommen. Mit dem Punkt statt Datei/Verzeichnisnamen werden einfach alle Änderungen übernommen.
- `git commit -m „git-1.2“`  
Der Befehl `git commit` wird verwendet, um Änderungen in einem lokalen Repository, welche sich in der Staging Area befinden, zu einem neuen Commit zu machen. Ein Commit repräsentiert eine bestimmte Version des Projekts zu einem bestimmten Zeitpunkt. Auf diese Versionen kann jederzeit wieder zurückgegriffen werden. Mit `-m` kann direkt eine Message übergeben werden, welche Informationen zur Version enthält. Wird dies nicht gemacht, öffnet sich ein Editor und die Nachricht muss nachträglich eingegeben werden.
- `git push`  
Die Änderungen (Commits) des aktiven Branches werden auf das zentrale Projektarchiv hochgeladen. Dabei wird standardmäßig das Projektarchiv verwendet, von dem gecloned wurde (Origin). Möchte man nicht den aktiven Branch, in dem man sich befindet pushen, oder in ein anderes Archiv pushen, kann dies im Befehl noch genauer spezifiziert werden mittels `git push <remote-name> <branch-name>`.

### 1.3. Erweitere die Datei „archimedes.txt“ um folgenden Inhalt und übernehme die Änderung in das zentrale Projektarchiv

- “Archimedes was a physicist, engineer and mathematician who invented too many things to list them here.”
- Verwende die Commit-Nachricht: git-1.3

```
sarah@DESKTOP-6LEAOT9:~/ebd1_ue/S2310239008$ nano archimedes.txt
GNU nano 6.2 archimedes.txt *
Archimedes was a physicist, engineer and mathematician who invented too many things to list them here.

sarah@DESKTOP-6LEAOT9:~/ebd1_ue/S2310239008$ git add .
sarah@DESKTOP-6LEAOT9:~/ebd1_ue/S2310239008$ git status
On branch master
Your branch is up to date with 'origin/master'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   archimedes.txt

sarah@DESKTOP-6LEAOT9:~/ebd1_ue/S2310239008$ git commit -m "git-1.3"
[master 7ba268a] git-1.3
 1 file changed, 1 insertion(+)
sarah@DESKTOP-6LEAOT9:~/ebd1_ue/S2310239008$ git push
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 12 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 341 bytes | 341.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0), pack-reused 0
To gitlab.fh-ooe.at:sib/ebd/S2310239008.git
 4b89a82..7ba268a  master -> master
sarah@DESKTOP-6LEAOT9:~/ebd1_ue/S2310239008$
```

- `nano archimedes.txt`

Wenn man nano mit dem Namen einer Datei aufruft, wird die Datei im Nano Editor geöffnet.  
Wenn die Datei nicht vorhanden ist, wird sie erstellt und geöffnet.

## Teil 2

2.1. Erstelle einen Branch mit deiner S-Nummer (SXXXXXXXXX) als Namen und wechsele hinein

```
sarah@DESKTOP-6LEA0T9:~/ebd1_ue/S2310239008$ git checkout -b S2310239008
Switched to a new branch 'S2310239008'
sarah@DESKTOP-6LEA0T9:~/ebd1_ue/S2310239008$ git branch
* S2310239008
  master
sarah@DESKTOP-6LEA0T9:~/ebd1_ue/S2310239008$
```

- `git checkout -b S2310239008`  
Erstellt einen neuen Branch und wechselt sofort in diesen. Ebenso könnte der Branch mit dem Befehl `git branch <branchname>` erstellt werden, dann müsste aber manuell mit `git checkout <branchname>` in den Branch gewechselt werden.
- `git branch`  
Zeig eine Liste aller im lokalen Repository vorhandenen Branches an. Der aktive Branch (der in dem aktuell gearbeitet wird) wird mit einem Stern (\*) markiert.

2.2. In dem neuen Branch; erstelle eine neue Datei „bartik.txt“ mit dem unten stehenden Inhalt und übernahm es in das zentrale Projektarchiv

- Setze den neuen Branch als upstream Branch
- “Jean Bartik, born Betty Jean Jennings, was one of the original programmers for the ENIAC computer.”
- Verwende die Commit-Nachricht: git-2.2

```
sarah@DESKTOP-6LEA0T9:~/ebd1_ue/S2310239008$ git branch
* S2310239008
  master
sarah@DESKTOP-6LEA0T9:~/ebd1_ue/S2310239008$ nano bartik.txt
```

```
sarah@DESKTOP-6LEA0T9: ~/ebd1_ue/S2310239008
GNU nano 6.2                                bartik.txt *
Jean Bartik, born Betty Jean Jennings, was one of the original programmers for the ENIAC computer.
```

```
sarah@DESKTOP-6LEA0T9:~/ebd1_ue/S2310239008$ git add .
sarah@DESKTOP-6LEA0T9:~/ebd1_ue/S2310239008$ git commit -m "git-2.2"
[S2310239008 b7022cc] git-2.2
 1 file changed, 1 insertion(+)
 create mode 100644 bartik.txt
sarah@DESKTOP-6LEA0T9:~/ebd1_ue/S2310239008$ git push
fatal: The current branch S2310239008 has no upstream branch.
To push the current branch and set the remote as upstream, use

    git push --set-upstream origin S2310239008

sarah@DESKTOP-6LEA0T9:~/ebd1_ue/S2310239008$ git push --set-upstream origin S2310239008
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 12 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 357 bytes | 357.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0), pack-reused 0
remote:
remote: To create a merge request for S2310239008, visit:
remote:   https://gitlab.fh-ooe.at/sib/ebd/S2310239008/-/merge_requests/new?merge_request%5Bsource_branch%5D=S2310239008
remote:
To gitlab.fh-ooe.at:sib/ebd/S2310239008.git
 * [new branch]      S2310239008 -> S2310239008
Branch 'S2310239008' set up to track remote branch 'S2310239008' from 'origin'.
sarah@DESKTOP-6LEA0T9:~/ebd1_ue/S2310239008$
```

- `git push --set-upstream origin S2310239008`
  - `set-upstream..` "Upstream" wird beim ersten Mal pushen eines neuen Branches benötigt, damit Git weiß, welcher Remote-Branch mit diesem lokalen Branch in Verbindung steht.
  - `origin..` Das Remote Repository von dem gecloned wurde
  - `S2310239008..` Name des neuen Branches

### 2.3. Erweitere die Datei „archimedes.txt“ um folgenden Inhalt und übernehme die Änderung in das zentrale Projektarchiv

- “Archimedes was a **physicist and mathematician** who invented too many things to list them here.”
- Verwende die Commit-Nachricht: `git-2.3`

```
sarah@DESKTOP-6LEAOT9: ~/ebd1_ue/S2310239008
GNU nano 6.2 archimedes.txt *
Archimedes was a physicist and mathematician who invented too many things to list them here

sarah@DESKTOP-6LEAOT9:~/ebd1_ue/S2310239008$ nano archimedes.txt
sarah@DESKTOP-6LEAOT9:~/ebd1_ue/S2310239008$ git add .
sarah@DESKTOP-6LEAOT9:~/ebd1_ue/S2310239008$ git commit -m "git-2.3"
[S2310239008 ca7497f] git-2.3
1 file changed, 1 insertion(+), 1 deletion(-)
sarah@DESKTOP-6LEAOT9:~/ebd1_ue/S2310239008$ git push
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 12 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 334 bytes | 334.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0), pack-reused 0
remote:
remote: To create a merge request for S2310239008, visit:
remote:   https://gitlab.fh-ooe.at/sib/ebd/S2310239008/-/merge_requests/new?merge_request%5Bsource_branch%5D=S2310239008
remote:
To gitlab.fh-ooe.at:sib/ebd/S2310239008.git
 b7022cc..ca7497f  S2310239008 -> S2310239008
sarah@DESKTOP-6LEAOT9:~/ebd1_ue/S2310239008$
```

## Teil 3

- Wechsle zurück in den Main-Branch

```
sarah@DESKTOP-6LEAOT9:~/ebd1_ue/S2310239008$ git checkout master
Switched to branch 'master'
Your branch is up to date with 'origin/master'.
sarah@DESKTOP-6LEAOT9:~/ebd1_ue/S2310239008$
```

- `git checkout master`  
Mit `checkout <branchname>` kann in einen anderen Branch gewechselt werden

### 3.1. Erweitere die Datei „archimedes.txt“ um folgenden Inhalt und übernehme die Änderung in das zentrale Projektarchiv

- “Archimedes was a **physicist and engineer** who invented too many things to list them here.”
- Verwende die Commit-Nachricht: `git-3.1`

```
sarah@DESKTOP-6LEAOT9: ~/ebd1_ue/S2310239008
GNU nano 6.2 archimedes.txt
Archimedes was a physicist and engineer who invented too many things to list them here.
```

```
sarah@DESKTOP-6LEAOT9:~/ebd1_ue/S2310239008$ nano archimedes.txt
sarah@DESKTOP-6LEAOT9:~/ebd1_ue/S2310239008$ git add .
sarah@DESKTOP-6LEAOT9:~/ebd1_ue/S2310239008$ git commit -m "git-3.1"
[master 0f7d6c0] git-3.1
 1 file changed, 1 insertion(+), 1 deletion(-)
sarah@DESKTOP-6LEAOT9:~/ebd1_ue/S2310239008$ git push
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 12 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 334 bytes | 334.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0), pack-reused 0
To gitlab.fh-ooe.at:sib/ebd/S2310239008.git
 7ba268a..0f7d6c0  master -> master
sarah@DESKTOP-6LEAOT9:~/ebd1_ue/S2310239008$
```

3.2. Wechsle zurück in den Branch mit deiner S-Nummer und merge die Änderungen vom main-Branch in deinen Branch. Löse den auftretenden Merge-Konflikt zu folgendem Text

- “Archimedes was a **physicist, engineer and mathematician** who invented too many things to list them here.”
- Verwende die Commit-Nachricht: git-3.2

```
sarah@DESKTOP-6LEAOT9:~/ebd1_ue/S2310239008$ git checkout S2310239008
Switched to branch 'S2310239008'
Your branch is up to date with 'origin/S2310239008'.
sarah@DESKTOP-6LEAOT9:~/ebd1_ue/S2310239008$ git merge master
Auto-merging archimedes.txt
CONFLICT (content): Merge conflict in archimedes.txt
Automatic merge failed; fix conflicts and then commit the result.
sarah@DESKTOP-6LEAOT9:~/ebd1_ue/S2310239008$ nano archimedes.txt
```

```
sarah@DESKTOP-6LEAOT9: ~/ebd1_ue/S2310239008
GNU nano 6.2 archimedes.txt *
Archimedes was a physicist, engineer and mathematician who invented too many things to list them here.
```

```
sarah@DESKTOP-6LEAOT9:~/ebd1_ue/S2310239008$ git add .
sarah@DESKTOP-6LEAOT9:~/ebd1_ue/S2310239008$ git commit -m "git-3.2"
[S2310239008 a4b55f6] git-3.2
sarah@DESKTOP-6LEAOT9:~/ebd1_ue/S2310239008$ git status
On branch S2310239008
Your branch is ahead of 'origin/S2310239008' by 2 commits.
(use "git push" to publish your local commits)

nothing to commit, working tree clean
sarah@DESKTOP-6LEAOT9:~/ebd1_ue/S2310239008$
```

○ **git merge master**

Gibt man im aktiven Branch *git merge <quelle-branch>* ein, so wird der aktuelle Stand des Quell-Branche in den aktiven Branch integriert. Sofern in den beiden Branches nur unterschiedliche Dateien, oder unterschiedliche Stellen in derselben Datei bearbeitet wurden, sollte dies im Normalfall automatisch ohne Probleme funktionieren. Wenn, wie in unserem Fall, dieselbe Datei an derselben Stelle bearbeitet wurde, so gibt es einen Merge Conflict und man muss sich händisch für eine der beiden Änderungen entscheiden – oder etwas anderes hinschreiben. Diese Manuelle Lösung wird dann committed und der Merge Conflict ist gelöst.

3.3. Lass dir die Log-Einträge anzeigen. Füge den Log-Output in die Textdatei „log.txt“ ein und lade diese in das zentrale Projektarchiv hoch (in dem Branch mit deiner S-Nummer)

- Verwende die Commit-Nachricht: git-3.3

```
sarah@DESKTOP-6LEA0T9:~/ebd1_ue/S2310239008$ git log
commit a4b55f6eeec344bb3e1a4a915fcee6117c95b3c6 (HEAD -> S2310239008)
Merge: ca7497f 0f7d6c0
Author: S2310239008 <S2310239008@students.fh-hagenberg.at>
Date: Fri Dec 29 15:36:04 2023 +0100

    git-3.2

commit 0f7d6c02878ce60915cdad05fd28b1739a16513f (origin/master, origin/HEAD, master)
Author: S2310239008 <S2310239008@students.fh-hagenberg.at>
Date: Fri Dec 29 15:31:53 2023 +0100

    git-3.1
```

- `git log`

Gibt die Commit-Historie eines Git-Repositories aus, beginnend mit dem neuesten Commit und rückwärts in die Vergangenheit.

Folgende Informationen über jeden Commit werden ausgegeben:

- Commit-ID (SHA-1-Hash)
- Autor
- Datum und Uhrzeit des Commits
- Commit-Nachricht

```
sarah@DESKTOP-6LEA0T9:~/ebd1_ue/S2310239008$ git log > log.txt
sarah@DESKTOP-6LEA0T9:~/ebd1_ue/S2310239008$ cat log.txt
commit a4b55f6eeec344bb3e1a4a915fcee6117c95b3c6
Merge: ca7497f 0f7d6c0
Author: S2310239008 <S2310239008@students.fh-hagenberg.at>
Date: Fri Dec 29 15:36:04 2023 +0100

    git-3.2

commit 0f7d6c02878ce60915cdad05fd28b1739a16513f
Author: S2310239008 <S2310239008@students.fh-hagenberg.at>
Date: Fri Dec 29 15:31:53 2023 +0100

    git-3.1

commit ca7497f5f1fa344e0b457315c99fddaea69f2e04
Author: S2310239008 <S2310239008@students.fh-hagenberg.at>
```

- `git log > log.txt`

Mit dem Größer-Zeichen wird der Output von git log zum Input für eine neue Datei namens log.txt

- `cat log.txt`

Schreibt den Inhalt von log.txt auf die Standardausgabe (Terminal)

```
sarah@DESKTOP-6LEA0T9:~/ebd1_ue/S2310239008$ git add .
sarah@DESKTOP-6LEA0T9:~/ebd1_ue/S2310239008$ git commit -m "git-3.3"
[s2310239008 2591807] git-3.3
1 file changed, 146 insertions(+)
create mode 100644 log.txt
sarah@DESKTOP-6LEA0T9:~/ebd1_ue/S2310239008$ git push
Enumerating objects: 10, done.
Counting objects: 100% (10/10), done.
Delta compression using up to 12 threads
Compressing objects: 100% (6/6), done.
Writing objects: 100% (6/6), 1.79 KiB | 1.79 MiB/s, done.
Total 6 (delta 2), reused 0 (delta 0), pack-reused 0
remote:
remote: To create a merge request for S2310239008, visit:
remote: https://gitlab.fh-ooe.at/sib/ebd/S2310239008/-/merge_requests/new?merge_request%5Bsource_branch%5D=S2310239008
remote:
To gitlab.fh-ooe.at:sib/ebd/S2310239008.git
ca7497f..2591807 S2310239008 -> S2310239008
sarah@DESKTOP-6LEA0T9:~/ebd1_ue/S2310239008$
```