

GIT TRAINING FOR IT PROFESSIONALS

KRISTOF WERLING
KWERLINGIT GMBH
1 SEPT 2023





WHO AM I

- Kristof Werling
- 31 years of experience at HP:
DevOp, Developer, Architect
- Started KwerlingIT GmbH
- Focus: IT / Cyber Security
- Security Audits, IT consulting,
Trainings, Software creation



KwerlingIT GmbH

kristof.werling@kwerlingit.com

<https://www.linkedin.com/in/kristof-werling/>

<https://www.kwerlingit.com>

TABLE OF CONTENT

- Disclaimer
- Short Command Line Introduction
- Introduction into GIT
- Common GIT commands
- Working with Branches
- Remote repositories
- Best practices
- GIT Workflows
- Project & Issue Management

DISCLAIMER

- The presentation was created on a MS-Windows-based Computer
- Screen-shots are made, and explanation are based in the MS-Windows environment
- Wherever the author is aware of a significant difference between MS-Windows and Macintosh operating environments the slides will point out these differences.
- The author uses [Github.com](https://github.com) for all demonstration purposes

SHORT COMMAND LINE INTRODUCTION

HERE YOU LEARN HOW TO REACH THE COMMAND LINE AND
HOW TO NAVIGATE IN THE DIRECTORY TREE OF YOUR FILESYSTEM.

ACCESSING THE COMMAND LINE

MS-WINDOWS

1. Open the "Start" menu.
2. Type "cmd" or "Command Prompt" in the search bar and press Enter.
3. This opens the Command Prompt window in your user directory.

MACINTOSH

1. Open the "Applications" folder.
2. Go to the "Utilities" folder.
3. Launch the "Terminal" application.
4. This provides the shell command line in your home directory.

SHOW CONTENT OF A DIRECTORY

MS-WINDOWS

- Type “dir“ and press Enter

MACINTOSH

- Type “ls“ and press Enter
- In order to get the same level of detail as with the “dir“ command type “ls -l“

SHOW CONTENT OF A DIRECTORY

MS-WINDOWS

- There are two special directory names: ":" and ".."
- They exist in every directory and are needed for the OS to finds its way in the filetree.
- How to make use of them is shown in the explanation for relative paths.

MACINTOSH

- Same as for MS-Windows

SHOW CONTENT OF A DIRECTORY

MS-WINDOWS

```
GIT Training > dir
Volume in Laufwerk C: hat keine Bezeichnung.
Volumeseriennummer: 4E37-3A48

Verzeichnis von C:\Users\Kwerling

03.09.2023  00:26    <DIR>      .
09.01.2023  18:37    <DIR>      ..
13.01.2023  02:12    <DIR>      .android
28.08.2023  16:30      53 .git-for-windows-updater
27.01.2023  13:50      59 .gitconfig
13.01.2023  02:12    <DIR>      .gradle
13.01.2023  02:15    <DIR>      .m2
03.09.2023  00:26      14 .minttyrc
09.01.2023  20:24    <DIR>      .vscode
09.01.2023  12:11    <DIR>      3D Objects
09.01.2023  20:47    <DIR>      AndroidStudioProjects
09.01.2023  18:55    <DIR>      Contacts
14.01.2023  22:35    <DIR>      Data
14.01.2023  22:35    <DIR>      Dev
14.01.2023  22:56    <DIR>      Documents
02.09.2023  23:02    <DIR>      Downloads
09.01.2023  18:55    <DIR>      Favorites
09.01.2023  18:55    <DIR>      Links
09.01.2023  18:55    <DIR>      Music
09.01.2023  18:57    <DIR>      OneDrive
30.08.2023  23:42    <DIR>      OneDrive - KwerlingIT GmbH
```

MACINTOSH

```
GIT Training > ls
'3D Objects'/
AndroidStudioProjects/
Anwendungsdaten@
AppData/
Contacts/
Cookies@
Data/
Dev/
Documents/
Downloads/
Druckumgebung@
'Eigene Dateien'@
Favorites/
IntelGraphicsProfiles/
Links/
'Lokale Einstellungen'@
Music/
NTUSER.DAT
NTUSER.DAT{a2332f18-cdbf-11ec-8680-002248483d79}.TM.blf
```

WHERE AM I IN THE DIRECTORY TREE

MS-WINDOWS

- Type "cd" and press Enter
- In MS-Windows the directory names are separated by a backslash ("\")
- In MS-Windows Drives exist. Each drive has a single letter followed by a colon as name ("c:", "f:"). Each drive has its own root directory.

MACINTOSH

- Type "pwd" and press Enter
- In Unix and macOS the directory names are separated by a slash ("/")
- Unix and macOS only have one root directory and the complete directory tree is reachable from there.

WHERE AM I IN THE DIRECTORY TREE

MS-WINDOWS

MACINTOSH

```
GIT Training > cd  
C:\Users\Kwerling
```

```
GIT Training > pwd  
/c/Users/Kwerling
```

MOVING IN THE FILESYSTEM

ABSOLUTE PATH

- An absolute path specifies the exact location of a file or directory from the file system's root directory (the top-level directory).
- It begins with the root directory's name (e.g., C:\ in Windows or / in Unix-like systems) and provides a complete path to the target file or directory.
- Absolute paths are not dependent on the current working directory and can be used to locate a file or directory from anywhere in the file system.
- Examples of absolute paths:
 - Windows: C:\Users\YourUsername\Documents\file.txt
 - macOS/Linux: /home/YourUsername/Documents/file.txt

MOVING IN THE FILESYSTEM

RELATIVE PATH

- A relative path starts either with a directory name or one or two dots ("." or "..").
 "." (dot) represents the current directory, and ".." (dot-dot) represents the parent directory.
- A relative path specifies the location of a file or directory with respect to the current working directory.
- Examples (in Unix notation with "/"):
 - **Reports/file1.txt**: file "file1.txt" in the subdirectory " Reports " of the current directory
 - **../ Reports /file1.txt**: file "file1.txt" in the subdirectory " Reports " of the parent directory

MOVING IN THE FILESYSTEM

MS-WINDOWS

- Type "cd <directory>", where <directory> is the directory, you want to change to, and then press Enter.
- Directories can be named absolute or relative.

MACINTOSH

- Same as for MS-Windows

CREATING NEW DIRECTORIES

MS-WINDOWS

- Type "mkdir <directory>", where <directory> is the directory you want to create, and then press Enter.
- Directories can be named absolute or relative.
- "mkdir" can be abbreviated as "md"

MACINTOSH

- Same as for MS-Windows, but the abbreviation of "md" does not exist.

REMOVE / DELETE A DIRECTORIES

MS-WINDOWS

- Type "rmdir <directory>", where <directory> is the directory, you want to delete, and then press Enter.
- Directories can be named absolute or relative.
- The directory must be empty.
- "mkdir" can be abbreviated as "rd"

MACINTOSH

- Same as for MS-Windows, but the abbreviation of "rd" does not exist.

EXERCISES

MS-WINDOWS

- Open a new Command Prompt window
- Switch to the "AppData\Local\Temp" directory
- The content of the directory will look different for each user, as it is used by the OS and many applications

MACINTOSH

- Open a new Terminal window
- Switch to the "/tmp" directory
- The content of the directory will look different for each user, as it is used by the OS and many applications

EXERCISES

MS-WINDOWS AND MACINTOSH

- Build the directories and subdirectories as shown on the right-hand-side.



EXERCISES

MS-WINDOWS AND MACINTOSH

- Switch to directory "Folder 2"
- Delete directory "Folder 2.2"
- Goto to the parent directory
- Delete "Folder 2" --- What happens and how to delete the folder?
- Do the same for "Folder 1"
- Try to use absolute and relative path names for the directories you work with

INTRODUCTION INTO GIT

HOW TO WORK WITH GITHUB, GITLAB, BITBUCKET AND MANY
OTHER PROVIDERS OF GIT REPOSITORIES

TECHNICAL ENVIRONMENT

Git Repository Service Provider:
Github

IDE: IntelliJ

Command line tools: standard git
programs

TABLE OF CONTENT

- Why a VCS?
- Basic GIT concepts
- Installation of GIT
- Local configuration
- Creation of a repository
- Adding files / directories to a repository
- Working with branches
- Remote repositories
- Tags in GIT
- Git workflows
- Best practises

WHY A VCS?

- Data Protection
(prevent data loss / corruption)
- Collaboration
(Multiple users can work on the same project in a controlled way)
- Auditing
(tracking of changes on each file)

BASIC GIT CONCEPTS

- Repository (local or remote): Store for your files and their version history
- Commit: A new version of files stored to the repository version history.
- Branch: A forked specific version of the files in the repository
- Merge: Combining changes of one branch into another

INSTALLATION OF GIT

MS-WINDOWS

- Windows requires an installation

MACINTOSH

- There is a fair chance, that GIT is already installed on your machine.
- Verify it by opening a Terminal and type "git –version" followed by Enter
- If it is not installed see next slide

INSTALLATION OF GIT

- Goto: <https://git-scm.com/downloads>
- Depending on your OS follow the instructions on the page.
- Make sure that the git-command can be used from the command line
- There are also GUI Clients available from that page. No need to install any of them.

LOCAL CONFIGURATION

- It is necessary to configure your real name and email address:
- On the command line enter this:

```
git config --global user.name "<<Real name of user>>"
```

```
git config --global user.email "somebody@computer.com"
```

LOCAL CONFIGURATION

- There are 3 levels of configuration hierarchy:

Level	Explanation
system	Taken, when there is no global or local configuration
global	Taken, when there is no local configuration
local	Taken, if it exists

LOCAL CONFIGURATION

- If wished, a standard editor (or IDE) can be configured for viewing or editing text and source files:

```
git config --global core.editor "idea"
```

CREATION OF A REPOSITORY

- "`git init`" creates a new git repository in the directory it is executed in
- If there are files or directories in this directory, then these will not be added to the repository automatically.
- After execution there will be a new directory by the name of ".git". It stores the content of the repository as well as the needed information for the repository management.

EXERCISE: CREATE A NEW LOCAL REPOSITORY

- Open a Terminal window
- Goto an empty directory
- Verify that it is empty
- Run the "git init" command
- Verify that the ".git" directory exists
- Have a look into that directory

EXERCISE: CREATE A NEW LOCAL REPOSITORY

COMMAND PROMPT

```
GIT Training > cd Git_Training

GIT Training > dir
 Volume in Laufwerk C: hat keine Bezeichnung.
 Volumeseriennummer: 4E37-3A48

 Verzeichnis von C:\Temp\Git_Training

06.09.2023 00:00 <DIR> .
06.09.2023 00:00 <DIR> ..
 0 Datei(en),          0 Bytes
 2 Verzeichnis(se), 1.626.398.474.240 Bytes frei

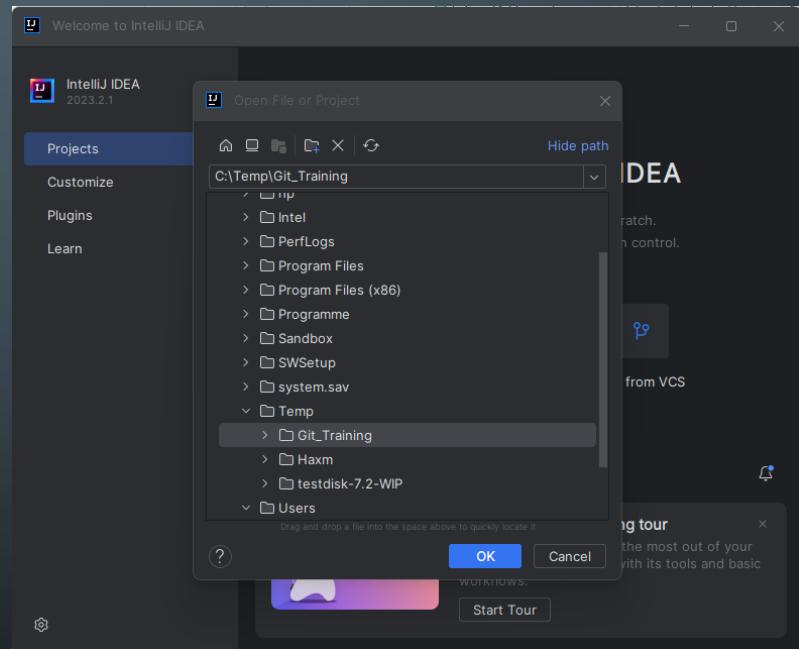
GIT Training > git init
Initialized empty Git repository in C:/Temp/Git_Training/.git/

GIT Training > dir
 Volume in Laufwerk C: hat keine Bezeichnung.
 Volumeseriennummer: 4E37-3A48

 Verzeichnis von C:\Temp\Git_Training

06.09.2023 00:00 <DIR> .
06.09.2023 00:00 <DIR> ..
 0 Datei(en),          0 Bytes
 2 Verzeichnis(se), 1.626.398.175.232 Bytes frei
```

INTELLIJ



EXERCISE: CREATE A NEW LOCAL REPOSITORY

COMMAND PROMPT

```
GIT Training > dir /AH
Volume in Laufwerk C: hat keine Bezeichnung.
Volumeseriennummer: 4E37-3A48

Verzeichnis von C:\Temp\Git_Training

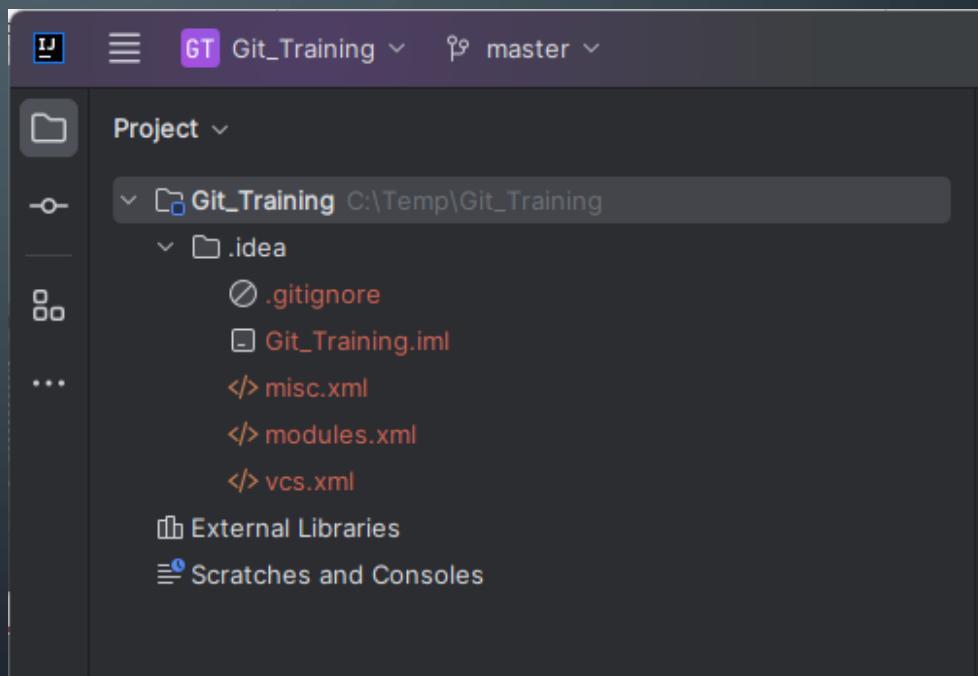
06.09.2023  00:00    <DIR>          .git
    0 Datei(en),           0 Bytes
    1 Verzeichnis(se), 1.626.402.197.504 Bytes frei

GIT Training > dir .git
Volume in Laufwerk C: hat keine Bezeichnung.
Volumeseriennummer: 4E37-3A48

Verzeichnis von C:\Temp\Git_Training\.git

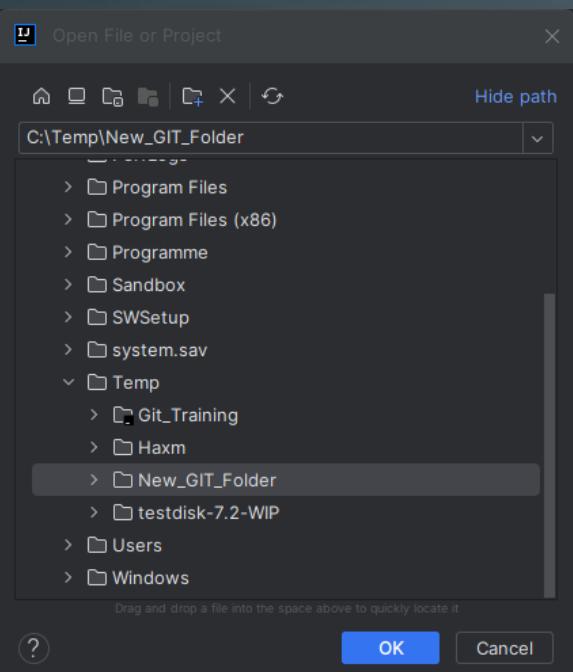
06.09.2023  00:00    <DIR>          ..
06.09.2023  00:00            130 config
06.09.2023  00:00            73 description
06.09.2023  00:00            23 HEAD
06.09.2023  00:00    <DIR>          hooks
06.09.2023  00:00    <DIR>          info
06.09.2023  00:00    <DIR>          objects
06.09.2023  00:00    <DIR>          refs
    3 Datei(en),           226 Bytes
    5 Verzeichnis(se), 1.626.402.078.720 Bytes frei
```

INTELLIJ

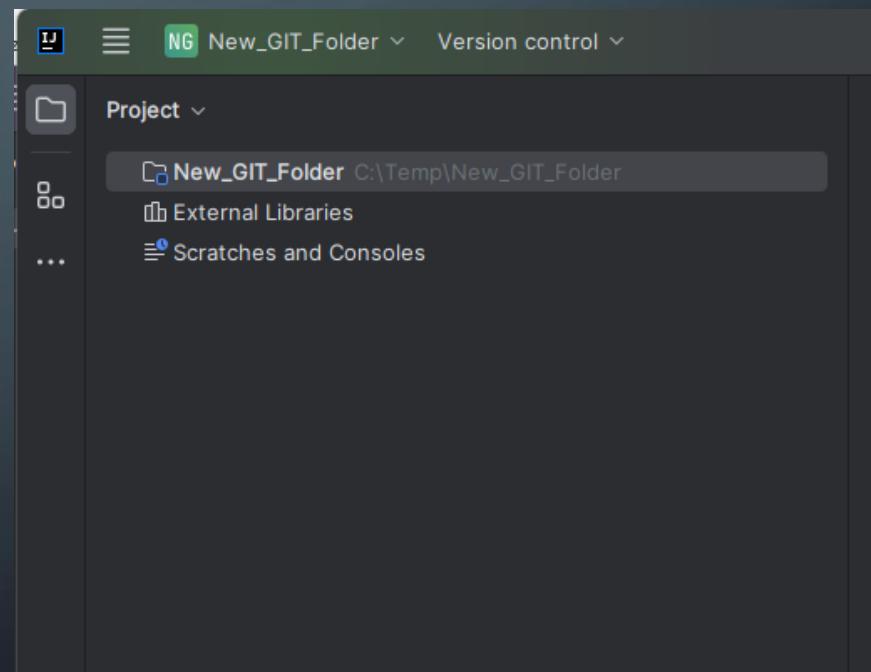


EXERCISE: CREATE A NEW LOCAL REPOSITORY

INTELLIJ

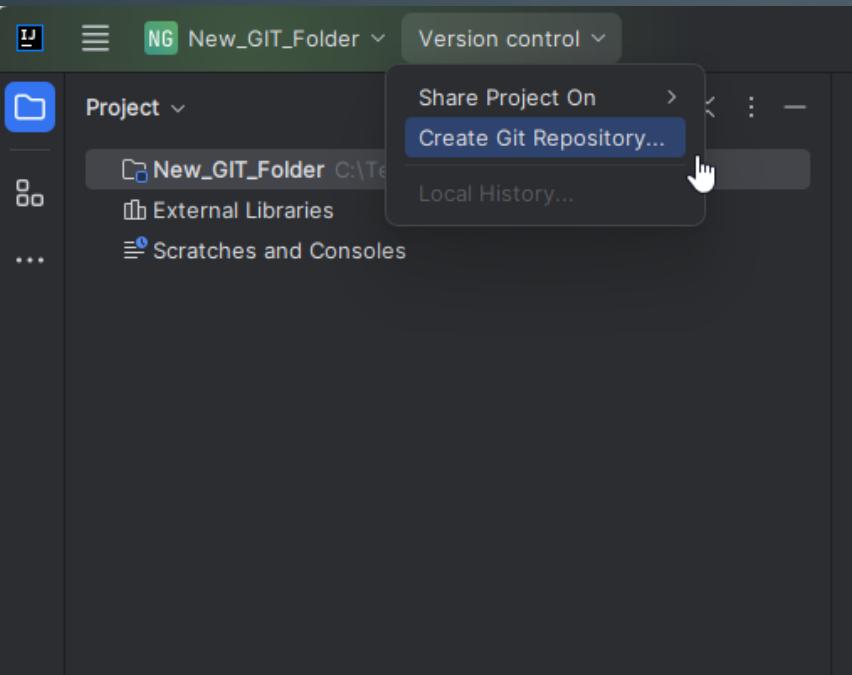


INTELLIJ

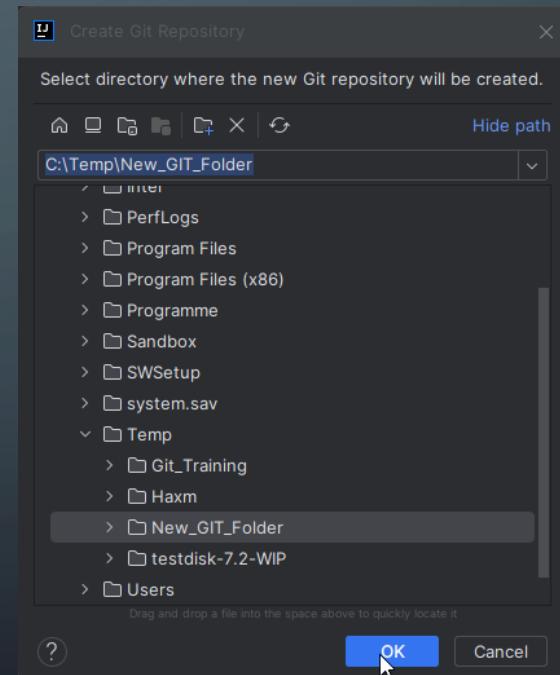


EXERCISE: CREATE A NEW LOCAL REPOSITORY

INTELLIJ

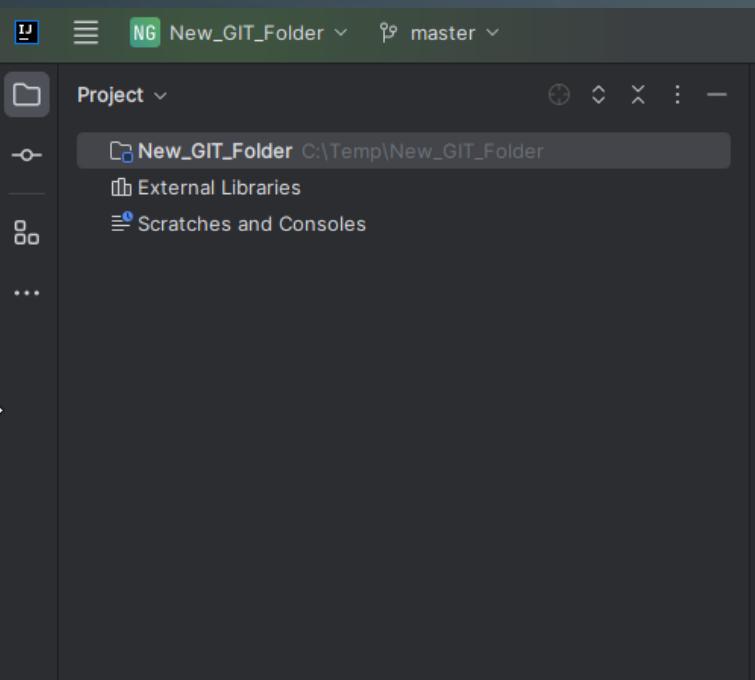


INTELLIJ



EXERCISE: CREATE A NEW LOCAL REPOSITORY

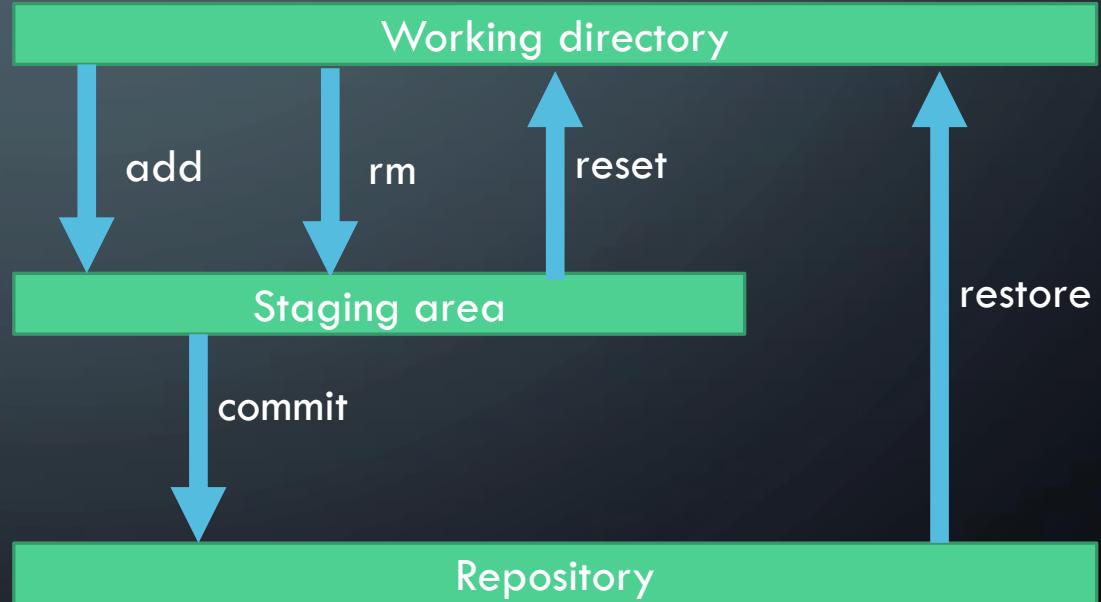
INTELLIJ



ADDING FILES / DIRECTORIES TO A REPOSITORY

- Adding files / directories to a repository is a 2 phase process:

1. Adding to a staging area
2. Committing to the repository



ADDING FILES / DIRECTORIES TO A REPOSITORY

- Here are the command you'll need in that context:

Command	Explanation
git status	Shows changes since last commit and the content of the staging area
git add <<file/dir . --all>>	Adds to the staging area if content is newer than last commit
git commit -m "Commit comment"	Adds files from staging into the repository
git reset HEAD <<file/dir name>>	Remove file/dir from staging
git rm <<file>> -r <<dir>>	Remove file/dir and add it to staging
git restore <<file/dir>>	Restore file/dir from the repository
git log	Shows the commit history

ADDING FILES / DIRECTORIES TO A REPOSITORY

What	.gitignore	.git/info/exclude
Location	Every directory in the project can have one, but mostly found in the root directory.	Only one file on the above shown location in the project.
Function	Contains the information on what to ignore for /exclude from commits as of the directory it is in.	Contains the information on what to ignore for / exclude from commits for the project.
Format	<ul style="list-style-type: none">• #• *• /• <>file/dir>>• **/• /**• [a-zA-Z0-9]	<p>Start a comment line</p> <p>Stands for any character but a slash</p> <p>Separates directories</p> <p>Matches the file or dir, absolute or relative</p> <p>Matches any directory + subdirectories</p> <p>Matches all files and sub-dirs in a dir</p> <p>Matches a range(s) of characters</p>

ADDING FILES / DIRECTORIES TO A REPOSITORY

Pattern in .gitignore	Explanation
node_modules	Ignore the node_modules dir (or file)
node_modules/	Ignore the content of the dir, but not the dir itself
# All bin files	Comment, has no consequence
/*/temp	Ignore the temp dir in each of the subdirs of the dir the .gitignore is in (1 level)
/**/readme	Ignore the file readme in any of the subdirs of the dir the .gitignore file is in
d/**/k	Matches d/k, d/x/k, d/x/y/k, and so on

EXERCISE: ADDING TO THE REPOSITORY

(TRY TO DO THE EXERCISE ON THE COMMAND LINE AND, IF POSSIBLE, IN INTELLIJ)

- Goto an empty directory, where you created a new GIT reposito and which you already opened as a project in IntelliJ.
- Look at the GIT status
- You should see an .idea directory, which is not comitted yet
- Add the directory to staging
- Look at the GIT status
- Remove the directory and its files from staging again
- Look at the GIT status

EXERCISE: ADDING TO THE REPOSITORY

(TRY TO DO THE EXERCISE ON THE COMMAND LINE AND, IF POSSIBLE, IN INTELLIJ)

- Make sure that the .idea directory is never considered for staging again
- Look at the GIT status
- Use IntelliJ to create file1.txt with this content:
This is the first file I am going to check in
 - Save the file
 - Look at the GIT status
 - Add the file to staging
 - Look at the GIT status

EXERCISE: ADDING TO THE REPOSITORY

(TRY TO DO THE EXERCISE ON THE COMMAND LINE AND, IF POSSIBLE, IN INTELLIJ)

- Commit to the repository
- Look at the GIT status
- Add a second line to file1.txt and save it:
Here is a modification
- Look at the GIT status
- Add all modified files to staging and commit them to the repository
- Look at the GIT status

EXERCISE: ADDING TO THE REPOSITORY

(TRY TO DO THE EXERCISE ON THE COMMAND LINE AND, IF POSSIBLE, IN INTELLIJ)

- Use the OS GUI to delete file1.txt from the directory
- Look at the GIT status
- Get the latest version of the file from the repository
- Look at the GIT status
- Have a look at the commit history

EXERCISE: ADDING TO THE REPOSITORY

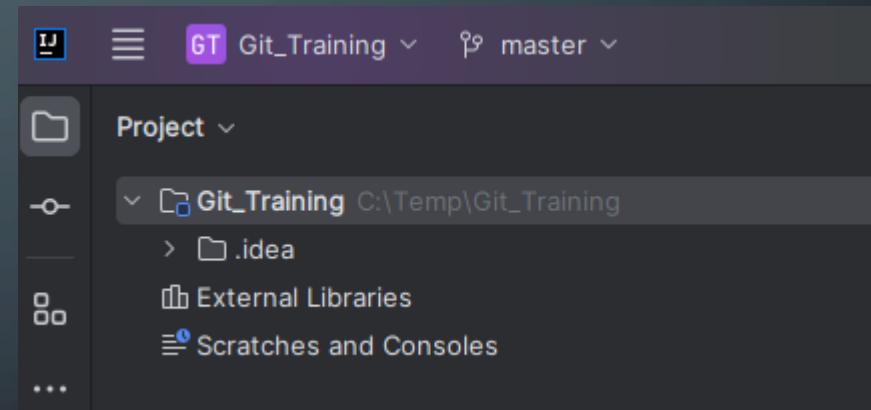
COMMAND PROMPT

```
GIT Training > dir
Volume in Laufwerk C: hat keine Bezeichnung.
Volumeseriennummer: 4E37-3A48

Verzeichnis von C:\Temp\Git_Training

06.09.2023  00:07    <DIR>          .
06.09.2023  00:43    <DIR>          ..
06.09.2023  00:07    <DIR>          .idea
                                0 Datei(en),           0 Bytes
                                3 Verzeichnis(se), 1.620.425.322.496 Bytes frei
```

INTELLIJ



EXERCISE: ADDING TO THE REPOSITORY

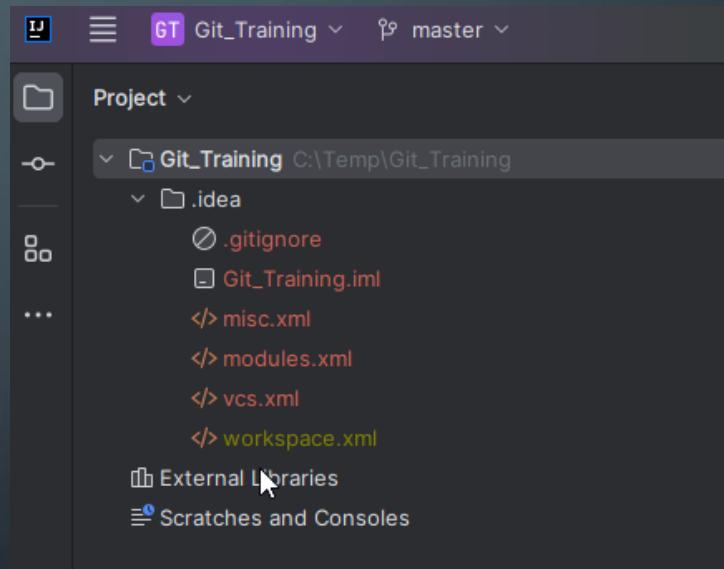
COMMAND PROMPT

```
GIT Training > git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    .idea/
nothing added to commit but untracked files present (use "git add" to track)
```

INTELLIJ

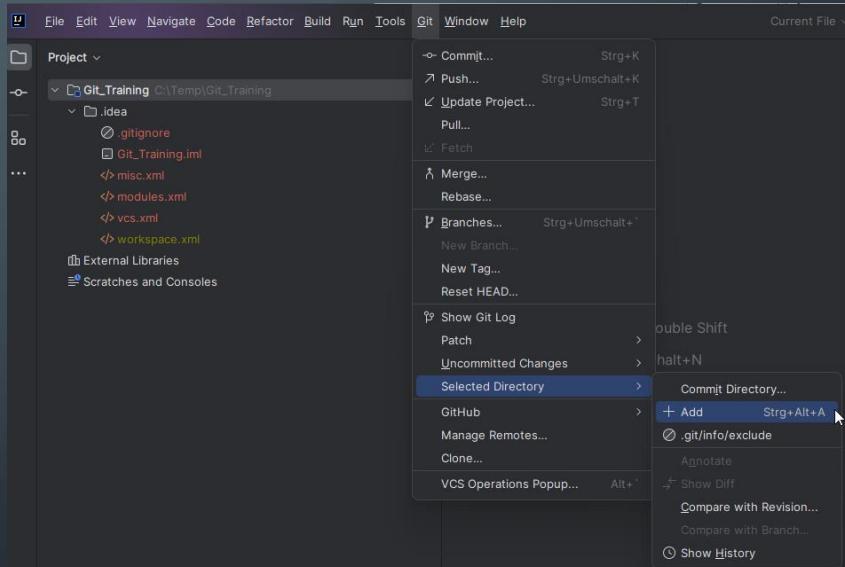


EXERCISE: ADDING TO THE REPOSITORY

COMMAND PROMPT

```
GIT Training > git add .idea  
  
GIT Training > git add .  
  
GIT Training > git add --all  
warning: in the working copy of '.idea/misc.xml', LF will  
be replaced by CRLF the next time Git touches it
```

INTELLIJ



EXERCISE: ADDING TO THE REPOSITORY

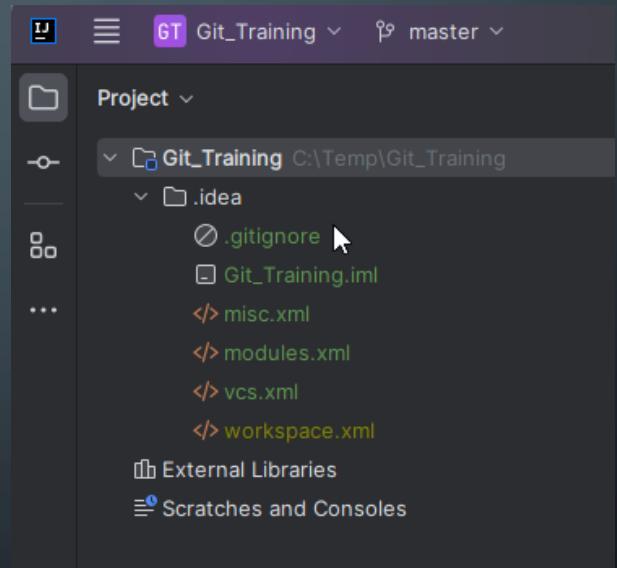
COMMAND PROMPT

```
GIT Training > git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
    new file:   .idea/.gitignore
    new file:   .idea/Git_Training.iml
    new file:   .idea/misc.xml
    new file:   .idea/modules.xml
    new file:   .idea/vcs.xml
```

INTELLIJ

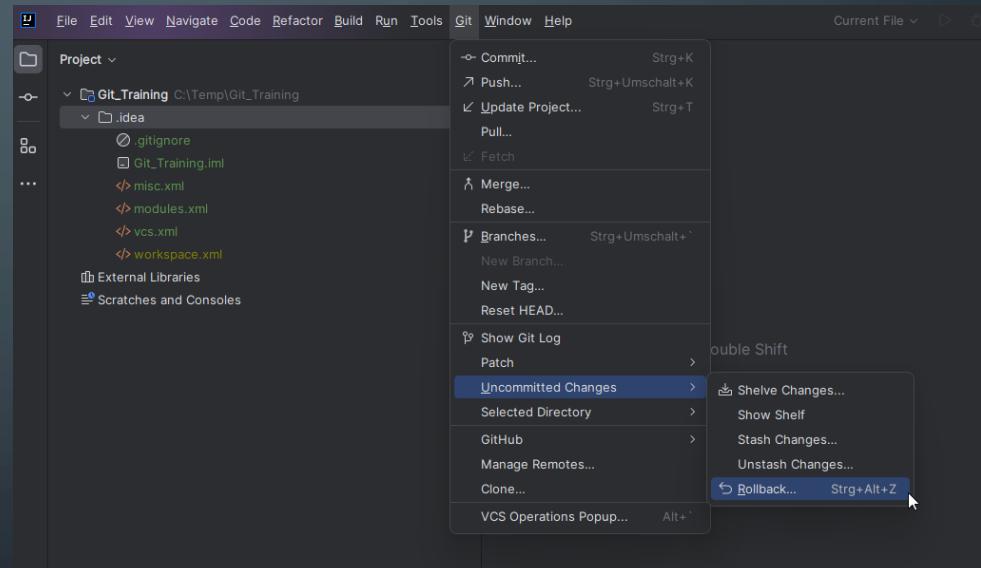


EXERCISE: ADDING TO THE REPOSITORY

COMMAND PROMPT

```
GIT Training > git reset .idea
```

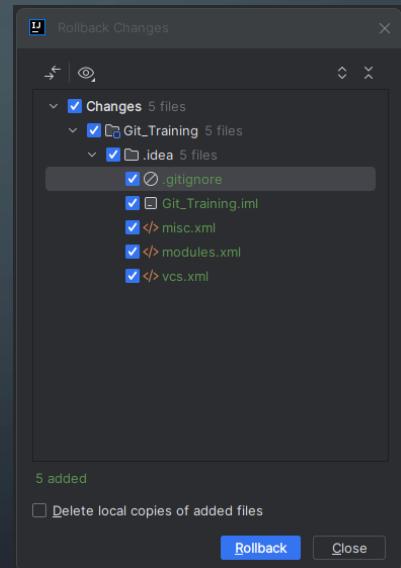
INTELLIJ



EXERCISE: ADDING TO THE REPOSITORY

COMMAND PROMPT

INTELLIJ



EXERCISE: ADDING TO THE REPOSITORY

COMMAND PROMPT

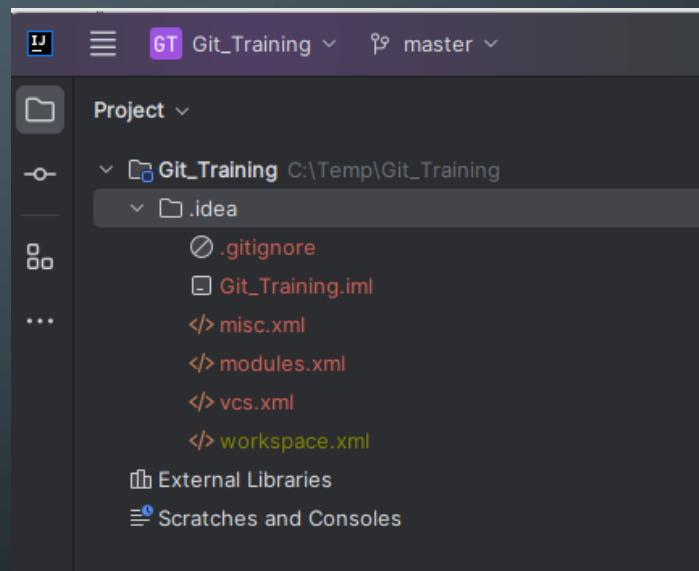
```
GIT Training > git status
On branch master

No commits yet

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

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

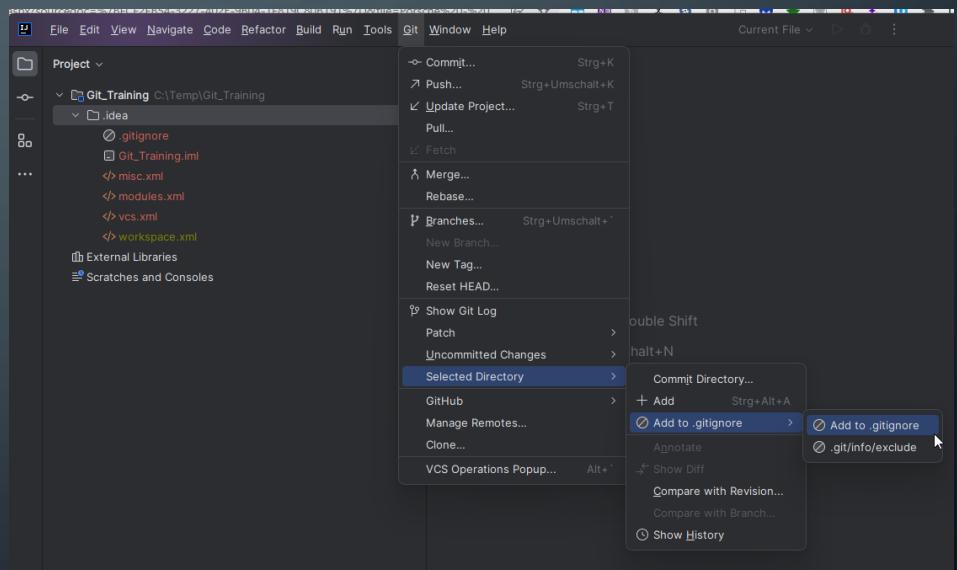
INTELLIJ



EXERCISE: ADDING TO THE REPOSITORY

COMMAND PROMPT

INTELLIJ



EXERCISE: ADDING TO THE REPOSITORY

COMMAND PROMPT

```
GIT Training > dir
Volume in Laufwerk C: hat keine Bezeichnung.
Volumeseriennummer: 4E37-3A48

Verzeichnis von C:\Temp\Git_Training

07.09.2023 23:37    <DIR>        .
06.09.2023 00:43    <DIR>        ..
07.09.2023 23:37            9 .gitignore
07.09.2023 23:29    <DIR>        .idea
    1 Datei(en),          9 Bytes
    3 Verzeichnis(se), 1.620.403.814.400 Bytes frei

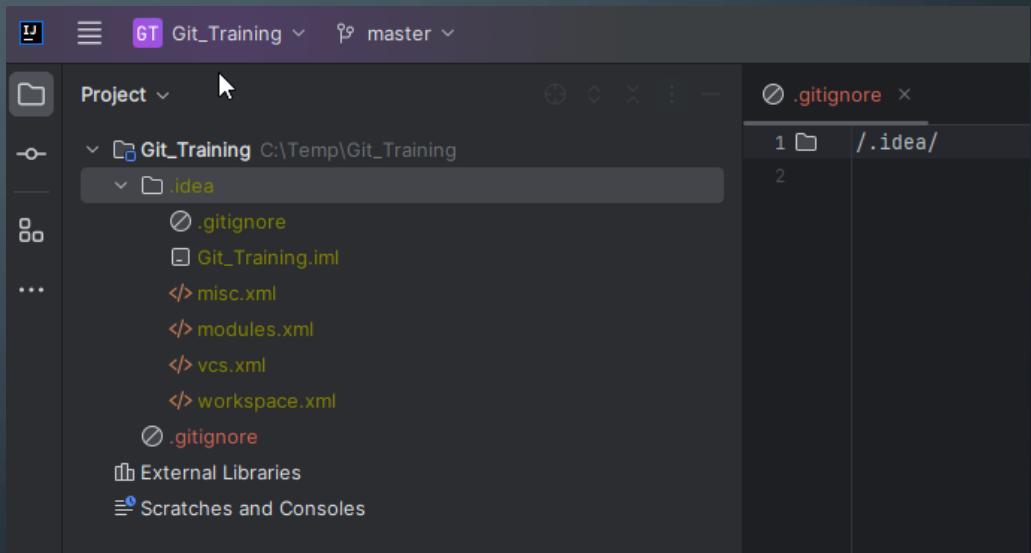
GIT Training > git status
On branch master

No commits yet

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

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

INTELLIJ



EXERCISE: ADDING TO THE REPOSITORY

COMMAND PROMPT

```
GIT Training > dir
Volume in Laufwerk C: hat keine Bezeichnung.
Volume Seriennummer: 4E37-3A48

Verzeichnis von C:\Temp\Git_Training

07.09.2023 23:42    <DIR>        .
06.09.2023 00:43    <DIR>        ..
07.09.2023 23:37            9 .gitignore
07.09.2023 23:29    <DIR>        .idea
07.09.2023 23:42            45 file1.txt
                           2 Datei(en),      54 Bytes
                           3 Verzeichnis(se), 1.620.402.503.680 Bytes frei

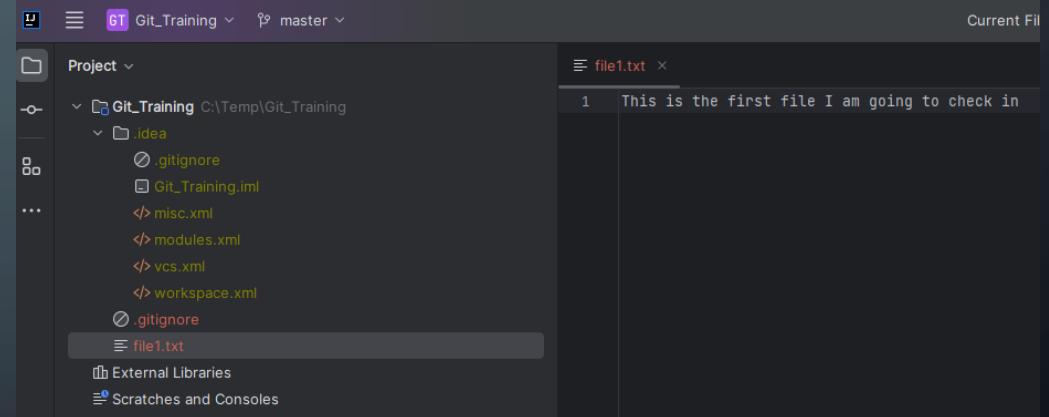
GIT Training > git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    .gitignore
    file1.txt

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

INTELLIJ

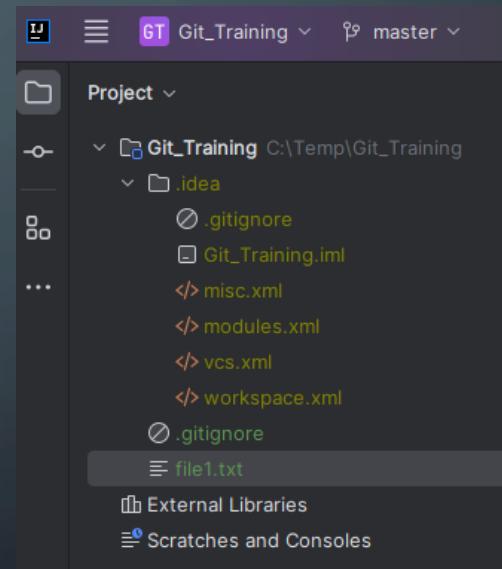


EXERCISE: ADDING TO THE REPOSITORY

COMMAND PROMPT

```
GIT Training > git add --all  
  
GIT Training > git status  
On branch master  
  
No commits yet  
  
Changes to be committed:  
(use "git rm --cached <file>..." to unstage)  
  new file:  .gitignore  
  new file:  file1.txt
```

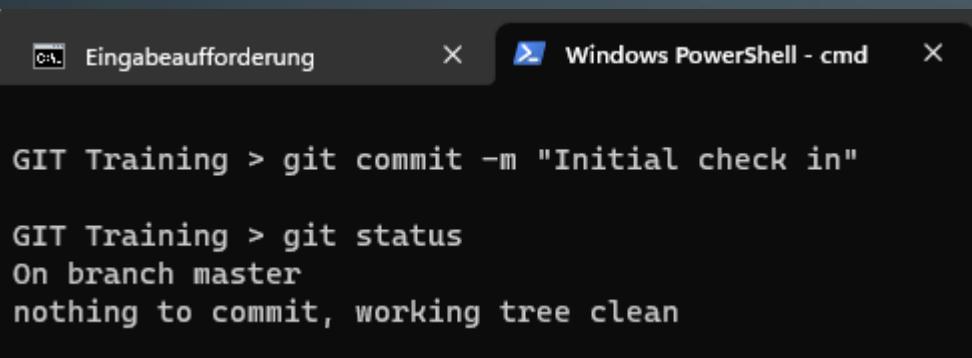
INTELLIJ



EXERCISE: ADDING TO THE REPOSITORY

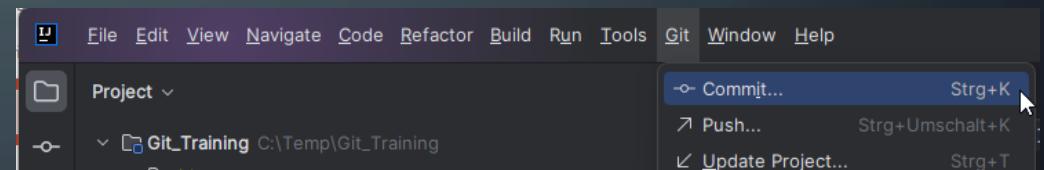
COMMAND PROMPT

INTELLIJ



```
Eingabeaufforderung
Windows PowerShell - cmd

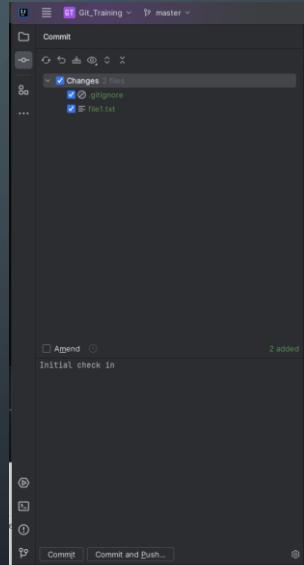
GIT Training > git commit -m "Initial check in"
GIT Training > git status
On branch master
nothing to commit, working tree clean
```



EXERCISE: ADDING TO THE REPOSITORY

COMMAND PROMPT

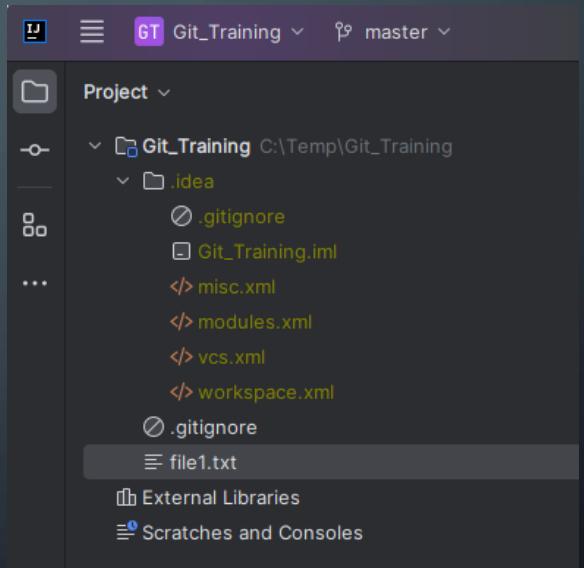
INTELLIJ



EXERCISE: ADDING TO THE REPOSITORY

COMMAND PROMPT

INTELLIJ



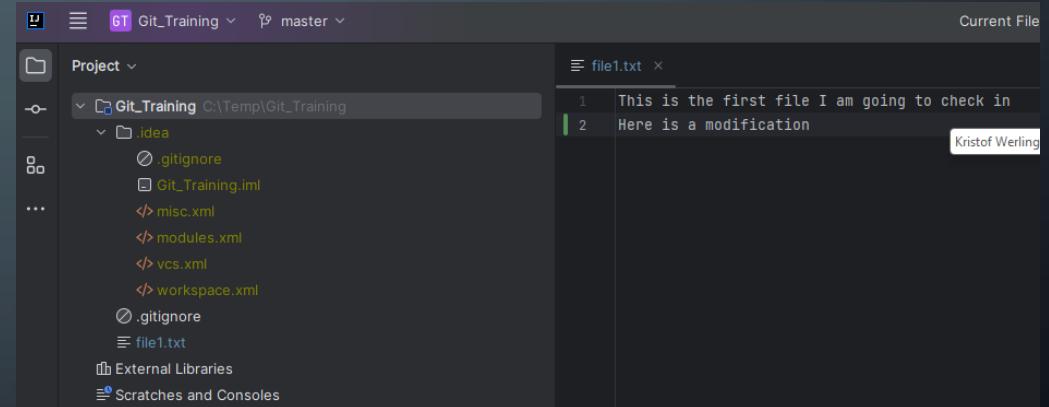
EXERCISE: ADDING TO THE REPOSITORY

COMMAND PROMPT

```
GIT Training > git status
On branch master
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
    modified:   file1.txt

no changes added to commit (use "git add" and/or "git commit -a")
```

INTELLIJ

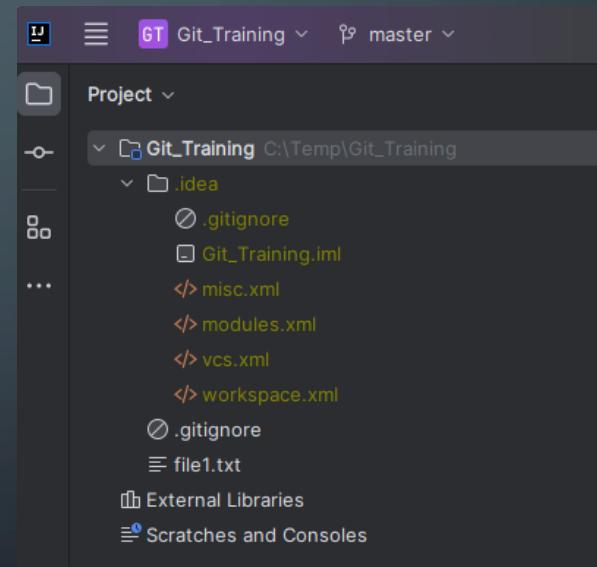


EXERCISE: ADDING TO THE REPOSITORY

COMMAND PROMPT

```
GIT Training > git add --all  
  
GIT Training > git status  
On branch master  
Changes to be committed:  
  (use "git restore --staged <file>..." to unstage)  
    modified:   file1.txt  
  
GIT Training > git commit -m "Added a second line to file1.txt"  
[master 1a2cf6d] Added a second line to file1.txt  
 1 file changed, 2 insertions(+), 1 deletion(-)  
  
GIT Training > git status  
On branch master  
nothing to commit, working tree clean
```

INTELLIJ



EXERCISE: ADDING TO THE REPOSITORY

COMMAND PROMPT

```
GIT Training > del file1.txt

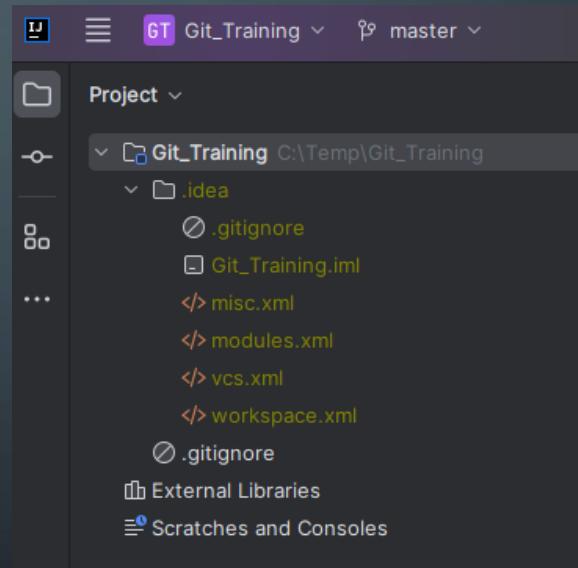
GIT Training > git status
On branch master
Changes not staged for commit:
  (use "git add/rm <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
    deleted:    file1.txt

no changes added to commit (use "git add" and/or "git commit -a")

GIT Training > git add --all

GIT Training > git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    deleted:    file1.txt
```

INTELLIJ



EXERCISE: ADDING TO THE REPOSITORY

COMMAND PROMPT

```
GIT Training > git restore file1.txt

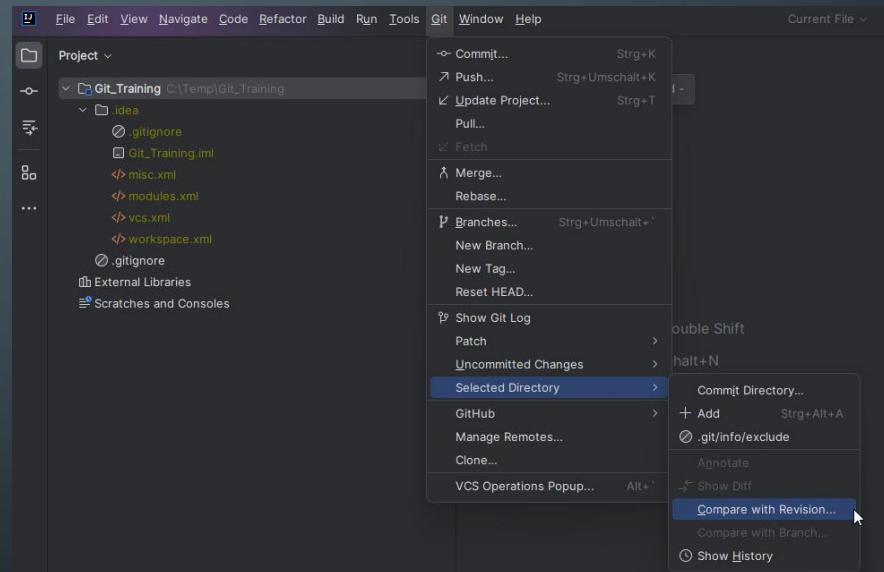
GIT Training > git status
On branch master
nothing to commit, working tree clean

GIT Training > dir
Volume in Laufwerk C: hat keine Bezeichnung.
Volumeseriennummer: 4E37-3A48

Verzeichnis von C:\Temp\Git_Training

08.09.2023  00:07    <DIR>          .
06.09.2023  00:43    <DIR>          ..
07.09.2023  23:37          9 .gitignore
08.09.2023  00:00    <DIR>          .idea
08.09.2023  00:07          69 file1.txt
                           2 Datei(en),           78 Bytes
                           3 Verzeichnis(se), 1.620.389.797.888 Bytes frei
```

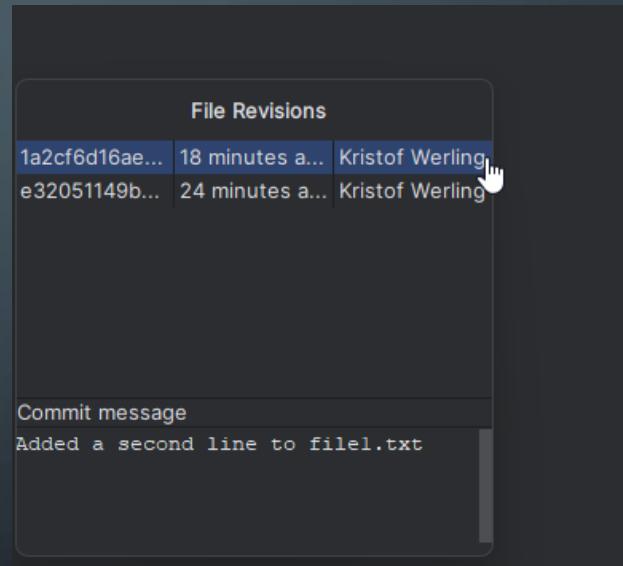
INTELLIJ



EXERCISE: ADDING TO THE REPOSITORY

COMMAND PROMPT

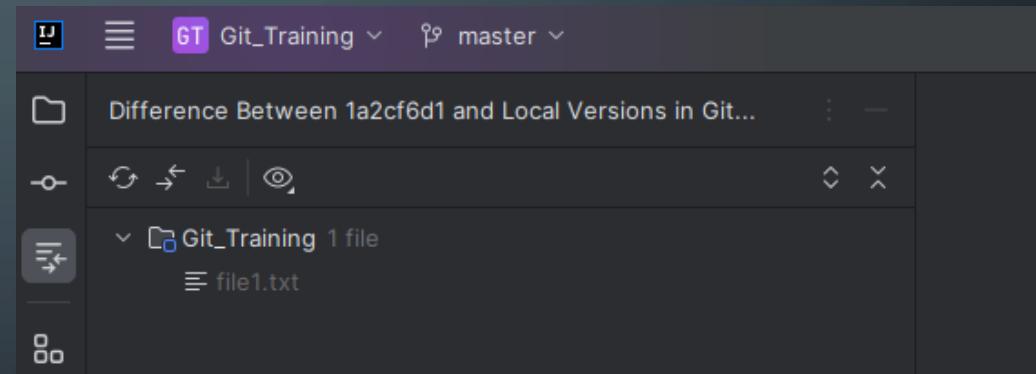
INTELLIJ



EXERCISE: ADDING TO THE REPOSITORY

COMMAND PROMPT

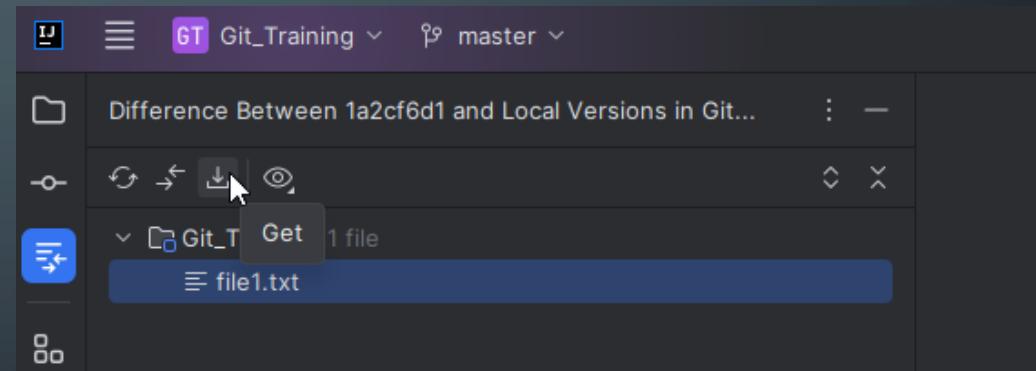
INTELLIJ



EXERCISE: ADDING TO THE REPOSITORY

COMMAND PROMPT

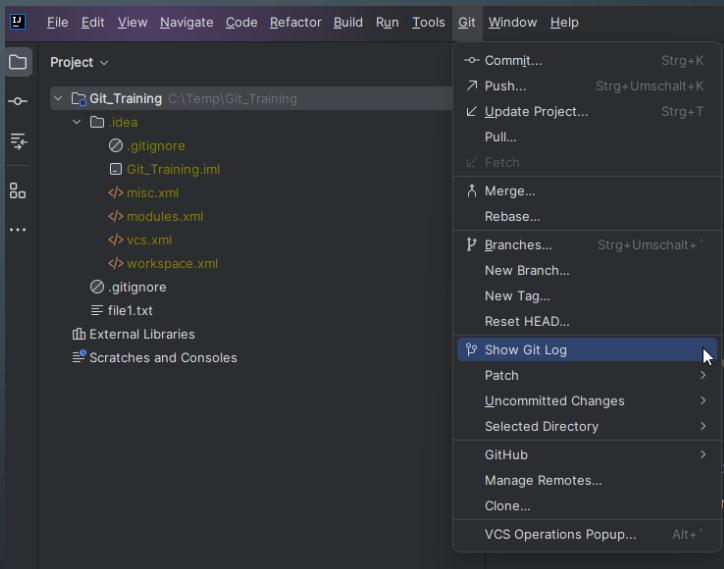
INTELLIJ



EXERCISE: ADDING TO THE REPOSITORY

COMMAND PROMPT

INTELLIJ



EXERCISE: ADDING TO THE REPOSITORY

COMMAND PROMPT

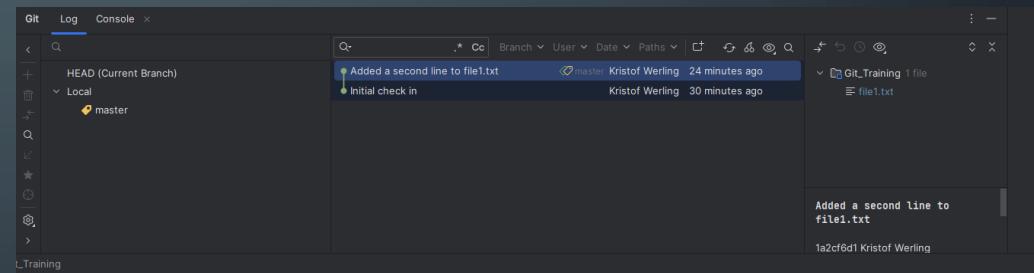
INTELLIJ

```
GIT Training > git log
commit 1a2cf6d16ae94bd00a6dc9b2429bd73d077d23be (HEAD -> master)
Author: Kristof Werling <kwerling@gmail.com>
Date:   Thu Sep 7 23:57:00 2023 +0200

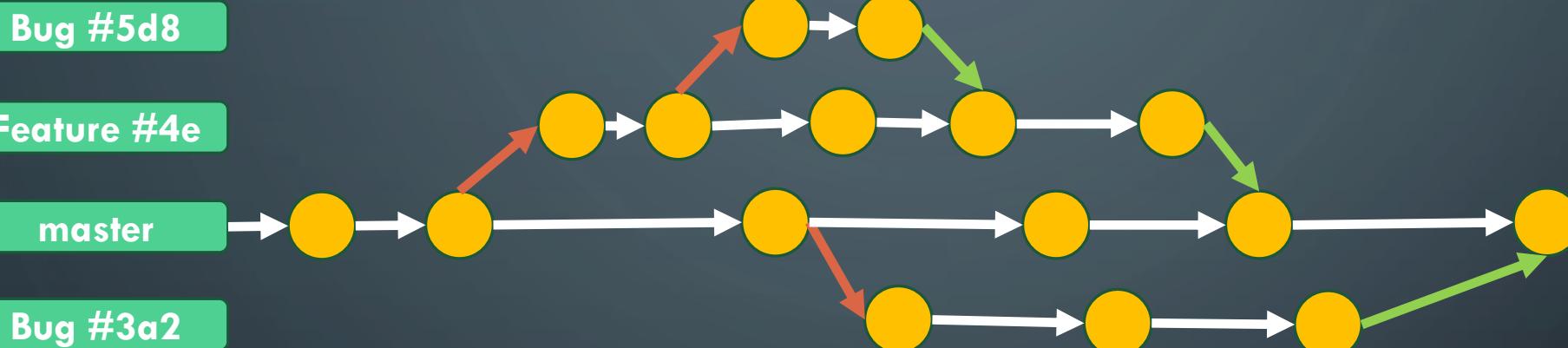
    Added a second line to file1.txt

commit e32051149b71207410f5c4f07b01053b233618db
Author: Kristof Werling <kwerling@gmail.com>
Date:   Thu Sep 7 23:50:59 2023 +0200

    Initial check in
```



WORKING WITH BRANCHES



Legend	master	Branch name	→	Same branch
			↗	New branch
	Commit		↘	Merge

WORKING WITH BRANCHES

- Here are the command you'll need in that context:

Command	Explanation
<code>git branch <<branch name>></code>	Creates a new branch
<code>git switch <<branch name>></code>	Switch to another branch
<code>git merge <<branch name>></code>	Merges the active branch into <<branch name>>
<code>git mergetool</code>	UI to resolve merge conflicts
<code>git stash</code>	Saves changes to prevent a commit
<code>git stash apply</code>	Re-apply the saved changes
<code>git stash list drop</code>	List all saved changes removes a saved change
<code>git branch -[dD] <<branch name>></code>	Deletes branch (-D) or only when merged (-d)

WORKING WITH BRANCHES

- Merge conflicts:
 - They occur when merging overlapping changes from two branches
 - One can resolve them manually (with an editor) or use the "git mergetool" command, IntelliJ, or one of the many other available tool
 - Once resolved the modified file needs to be committed to the repository
 - The modifications to the file(s) only happen in the actual branch.

EXERCISE: WORKING WITH BRANCHES

- Create a branch called "feature"
- Switch to it
- Create a file, called "mergefile.txt" and enter the following text:

This is a change to the feature branch
- Commit the changes to the repository
- Switch back to the master branch

EXERCISE: WORKING WITH BRANCHES

- Create a file, called "mergefile.txt" and enter the following text:

```
This is a change to the master branch
```
- Commit the changes to the repository
- Merge the "feature" branch into the "master" branch
- Resolve the merge conflict.
- Delete the "feature" branch