

Git Fundamentals

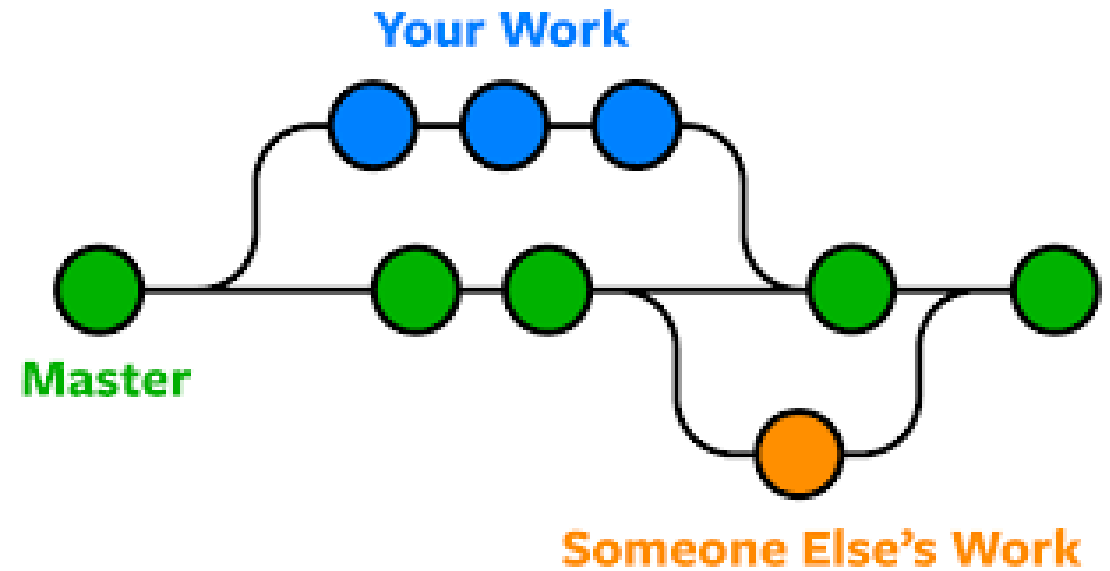
Day 2:

Changes



Today:

- ✓ Day 1 Recap
- ✓ Working area VS Staging area
- ✓ Adding and discarding changes
- ✓ Simple commits



Recap: Day 1

- We motivated the need for a version control system (VCS) in our projects
- Installed tools: VS Code, CLI and GitHub
- CLI: First commands in a Linux like environment
- Git: Settings + Initialize and Clone a repository



Recap: Git configuration

Must have configurations:

- user.name
- user.email

(Recommended same as GitHub)

```
git config --global user.name "John Doe"
```

```
git config --global user.email "johndoe@email.com"
```



.gitconfig

miro

Recap: Git configuration

Optionals (Highly recommended):

- VS Code as default Git editor
- Rename default branch to main

```
git config --global core.editor "code --wait"
```

```
git config --global init.defaultBranch main
```

Somewhere in your file system!
(Usually: ~/.gitconfig)



.gitconfig

miro

Recap: Initialize and Clone

```
alfre@LAPTOP-HOCGNJ8U MINGW64 ~/Desktop/test
$ git init
Initialized empty Git repository in C:/Users/alfre/Desktop/test/.git/

alfre@LAPTOP-HOCGNJ8U MINGW64 ~/Desktop/test (main)
$ git status
On branch main

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        f1.txt
        f2.txt

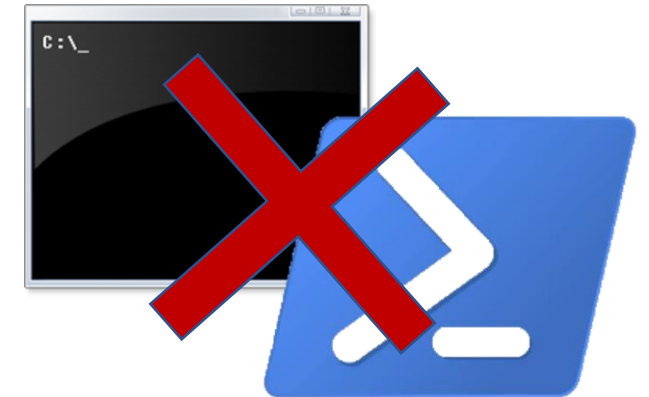
nothing added to commit but untracked files present (use "git add" to track)

alfre@LAPTOP-HOCGNJ8U MINGW64 ~/Desktop/test (main)
$ ls -la
./  ../ .git/ f1.txt f2.txt
```

Recap: Things to keep in mind

- For Windows users (Git Bash) the path format differs.
 - Slash instead of Backslash (/ instead of \)

```
alfre@LAPTOP-HOCGNJ8U MINGW64 ~  
$ pwd  
/c/Users/alfre  
  
alfre@LAPTOP-HOCGNJ8U MINGW64 ~  
$ cd Desktop/git-practice/  
  
alfre@LAPTOP-HOCGNJ8U MINGW64 ~/Desktop/git-practice  
$ ls  
day1/  day2/  day3/  day4/
```



Recap: Things to keep in mind

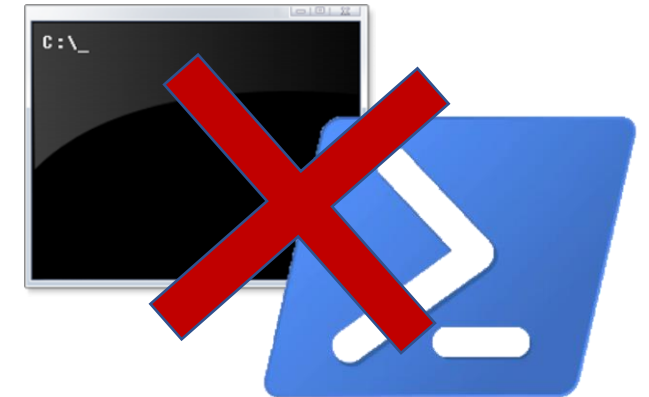
- Unlike in Windows, it is not recommended to have spaces in our directory names. (They require `"` or `\`)

```
alfre@LAPTOP-HOCGNJ8U MINGW64 ~/Desktop
$ cd "dir with space"/

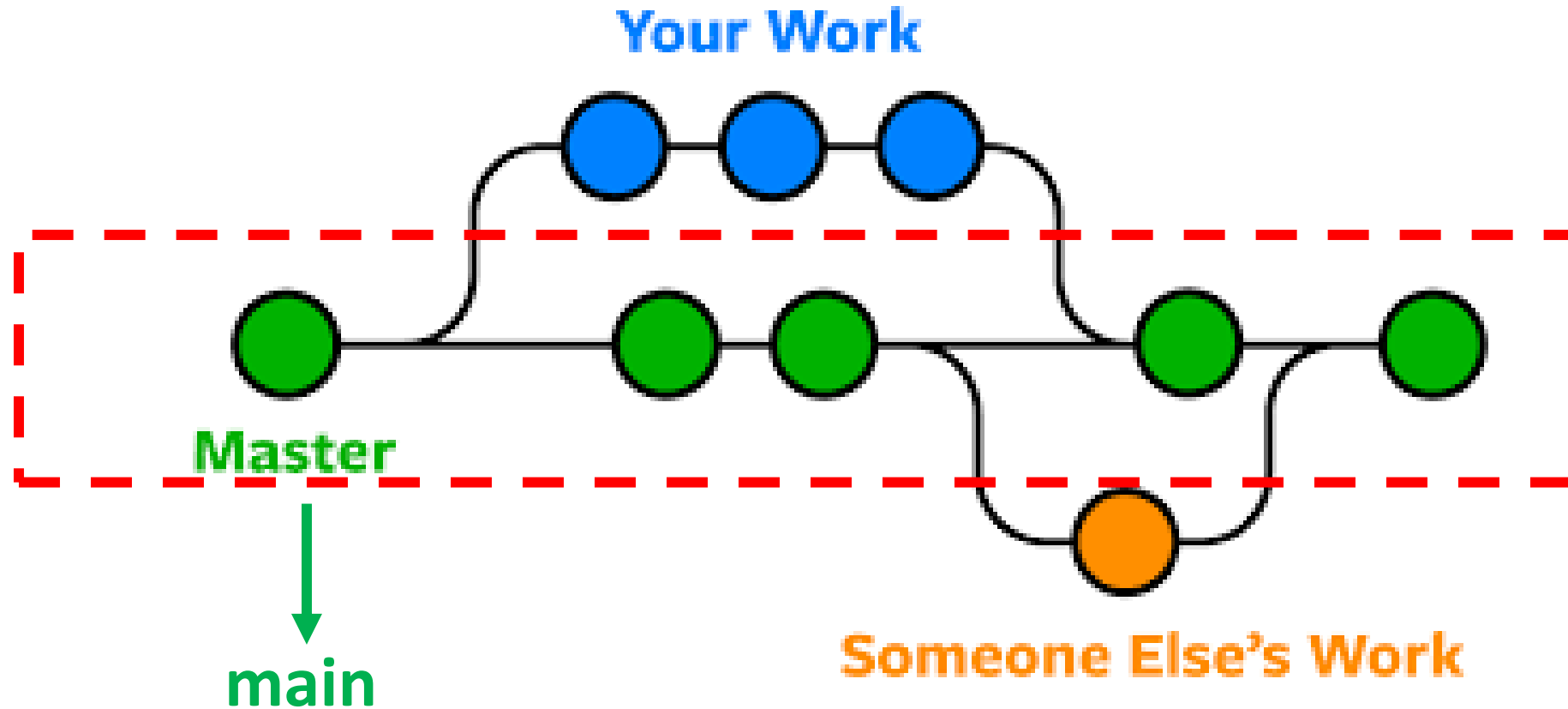
alfre@LAPTOP-HOCGNJ8U MINGW64 ~/Desktop/dir with space
$ cd ..

alfre@LAPTOP-HOCGNJ8U MINGW64 ~/Desktop
$ cd dir\ with\ space/

alfre@LAPTOP-HOCGNJ8U MINGW64 ~/Desktop/dir with space
$ pwd
/c/Users/alfre/Desktop/dir with space
```

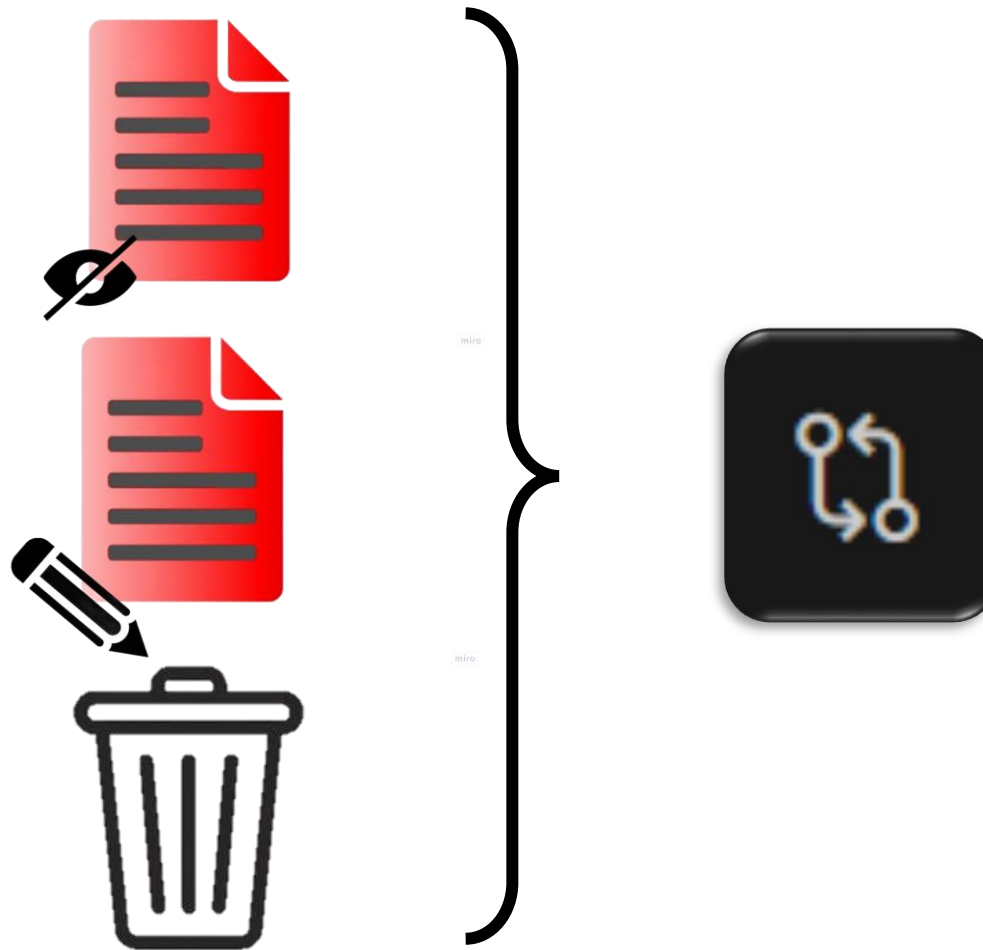


Note: Single timeline (for now)



Changes: Types

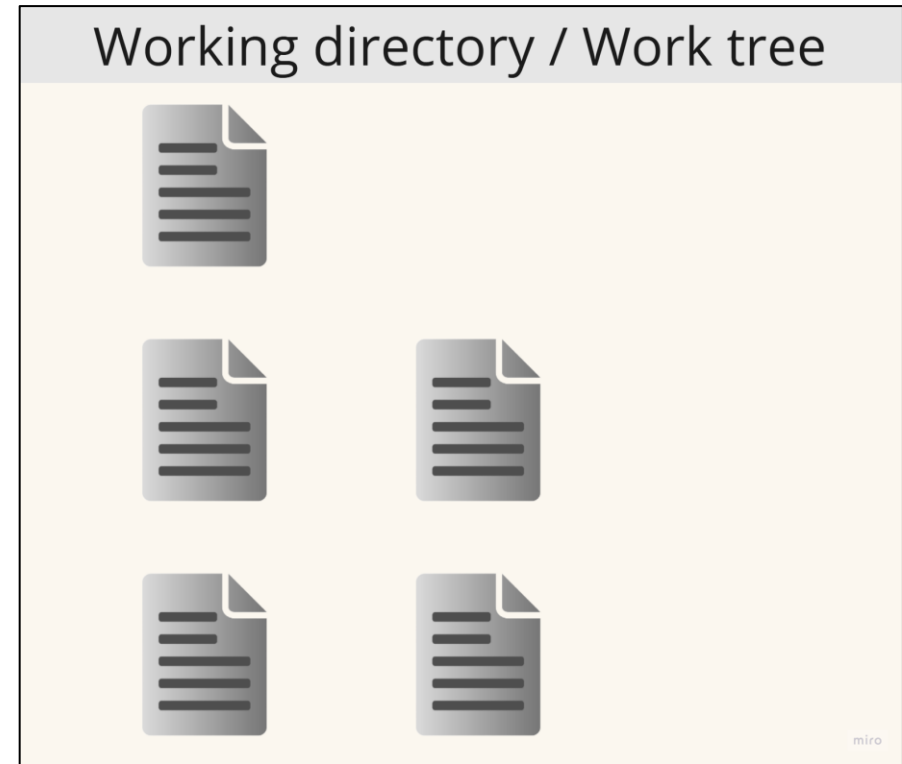
- New untracked files
- Edit tracked files
- Delete tracked files



Changes: Working area

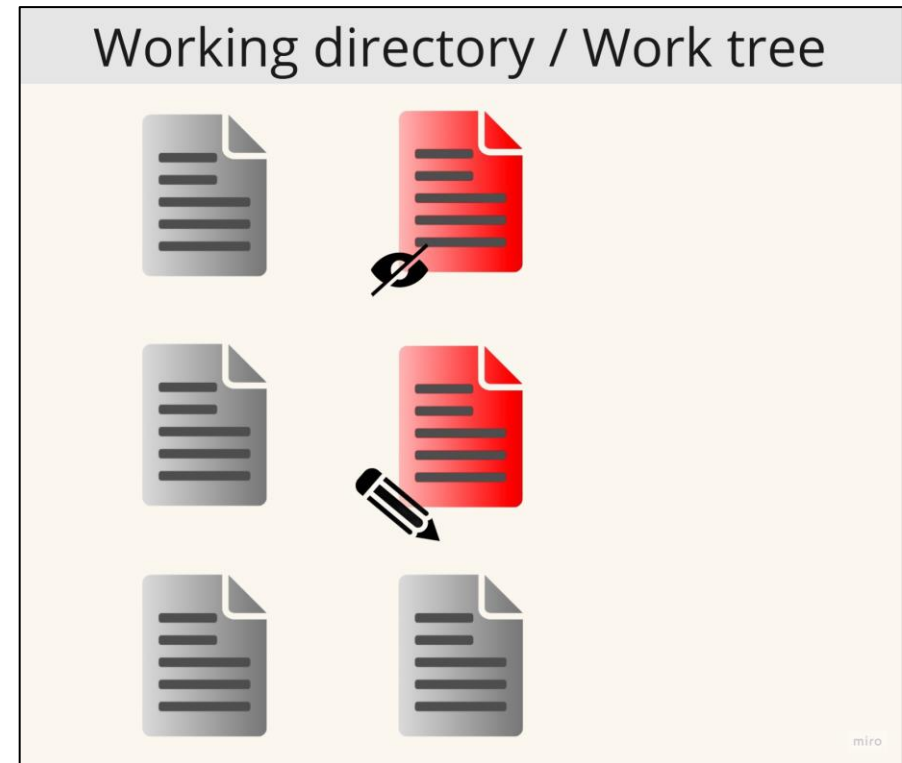
- Local directory where our files live
- Tracked by Git = Existence of .git folder
- Also called Work tree

(From now on we assume hidden .git folder)



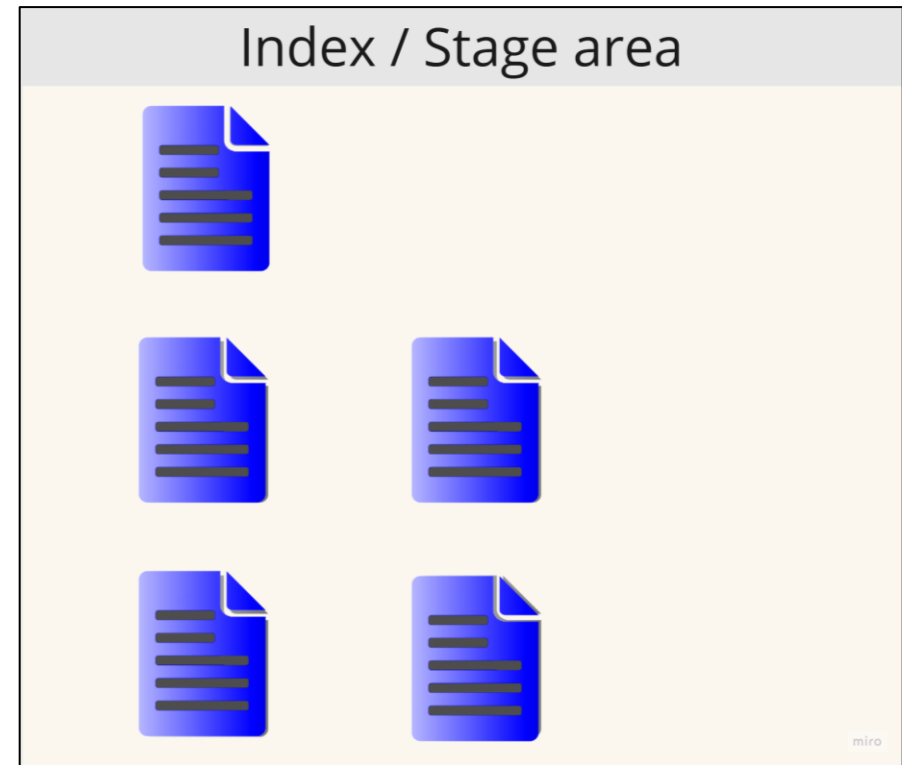
Adding changes to Work tree

- The working area is our changes entryptoint
- To modify the working area either:
 - ✓ Create new files with `echo` or `touch`
 - ✓ Edit existent files (we use `code`)



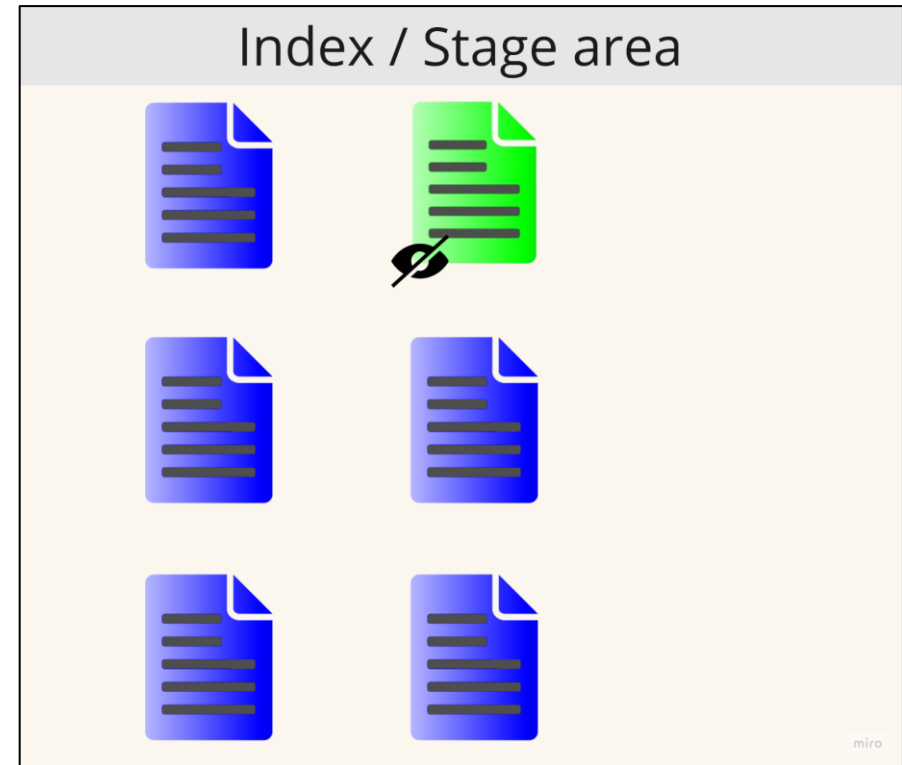
Changes: Staging area

- Preview of changes we want to keep
- Also called Index
- Screenshot of changes we want to keep
- In reality, references to Working area files



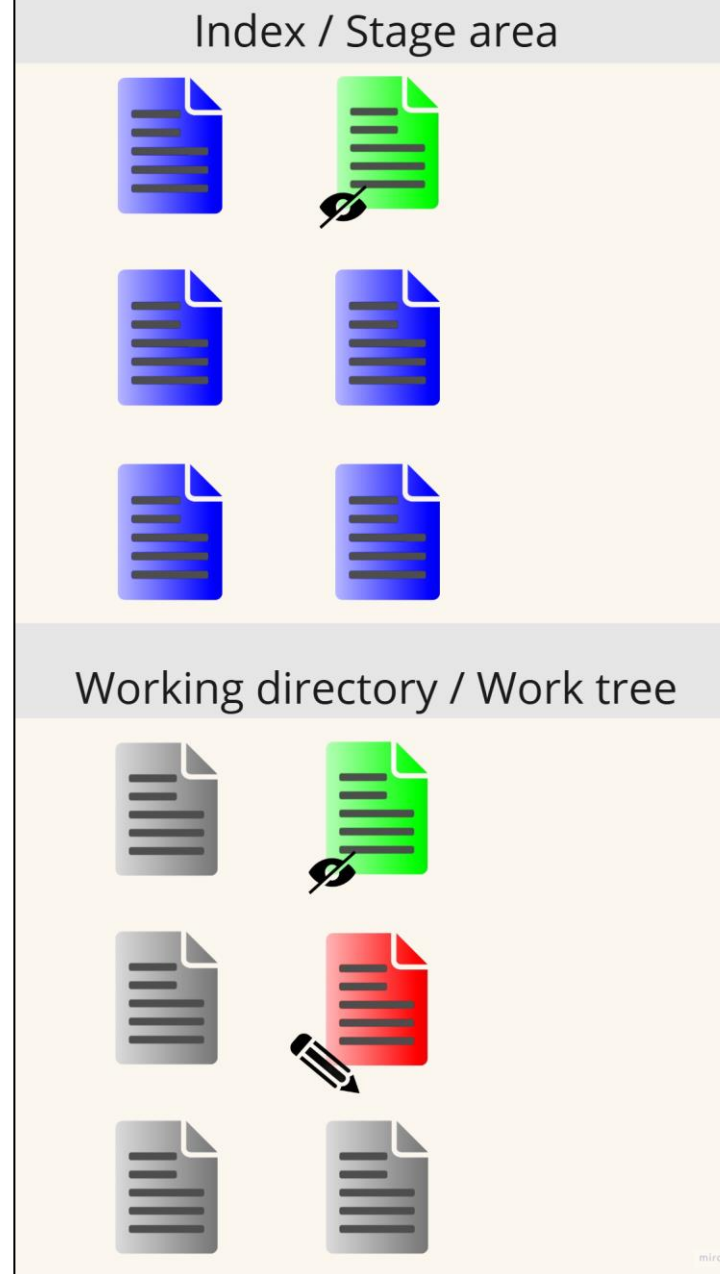
Adding changes to Index

- Before creating a new version we select changes we want to keep
- “We add them to a preview” (Staging area)
- Command: `git add`



Changes

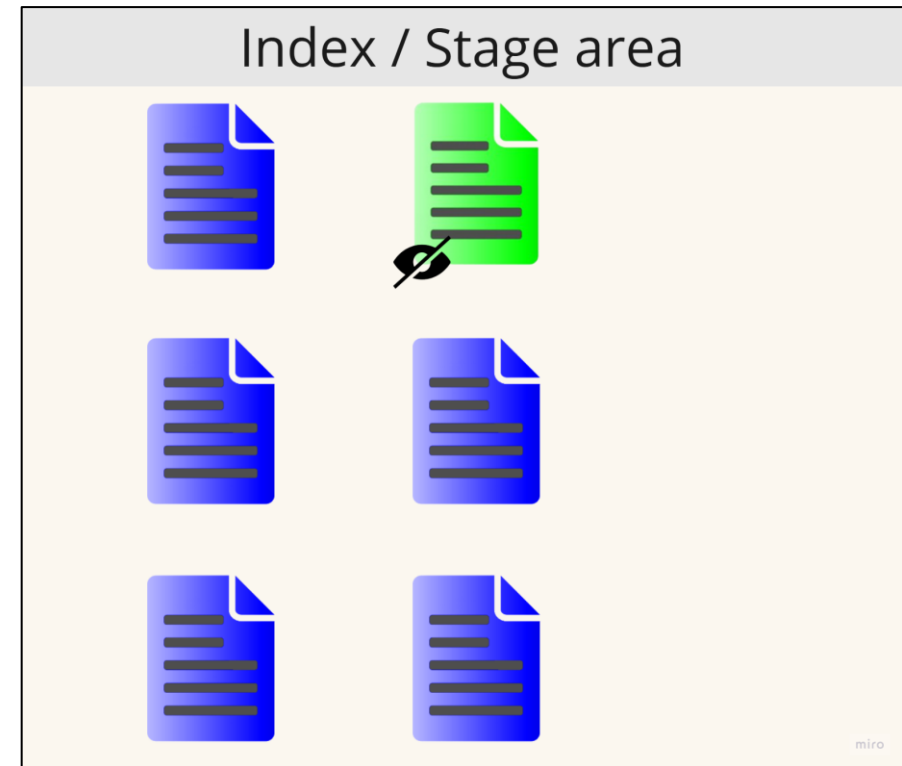
We modify the Stage area by selecting changes to keep



A piece of cake!

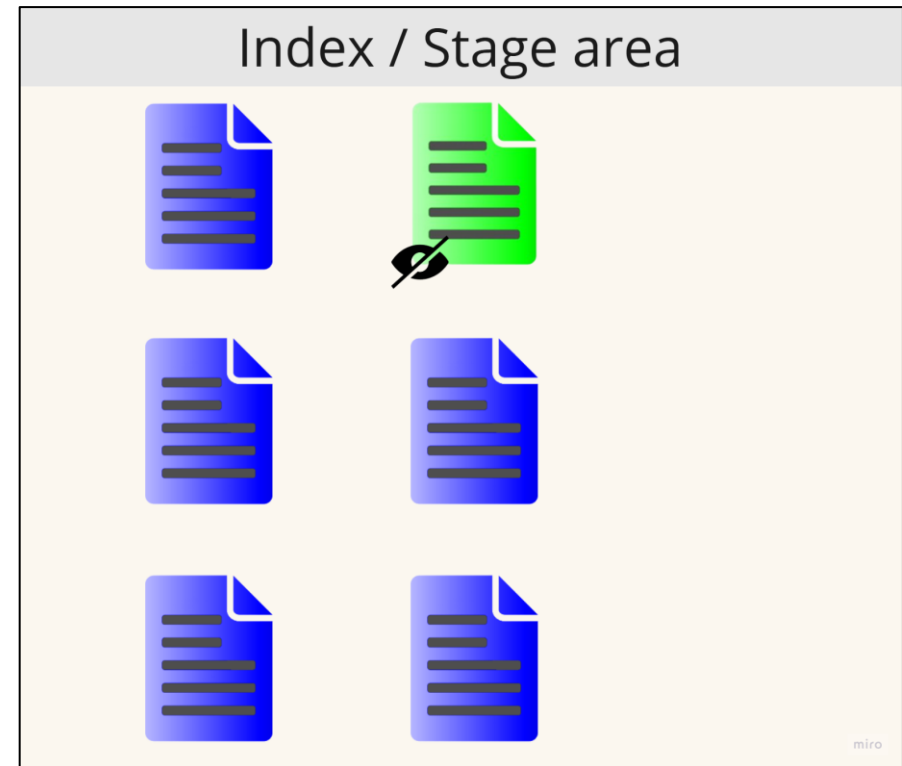
Discarding changes from Index

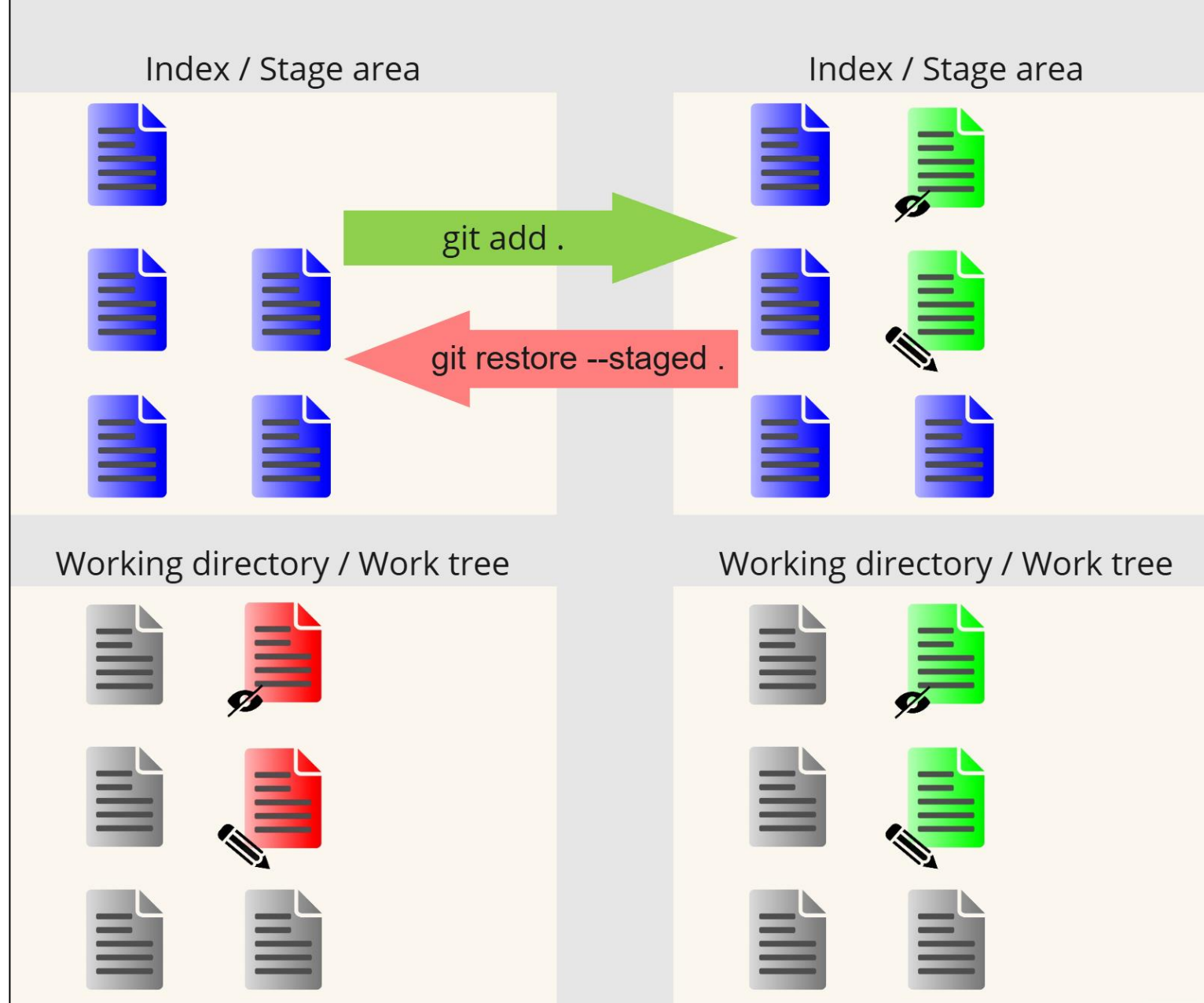
- There may be changes we added that we realized **no longer want in a version**
- To deselect them and restore the Index **to the previous commit** we use `git restore --staged`



Discarding changes from Index

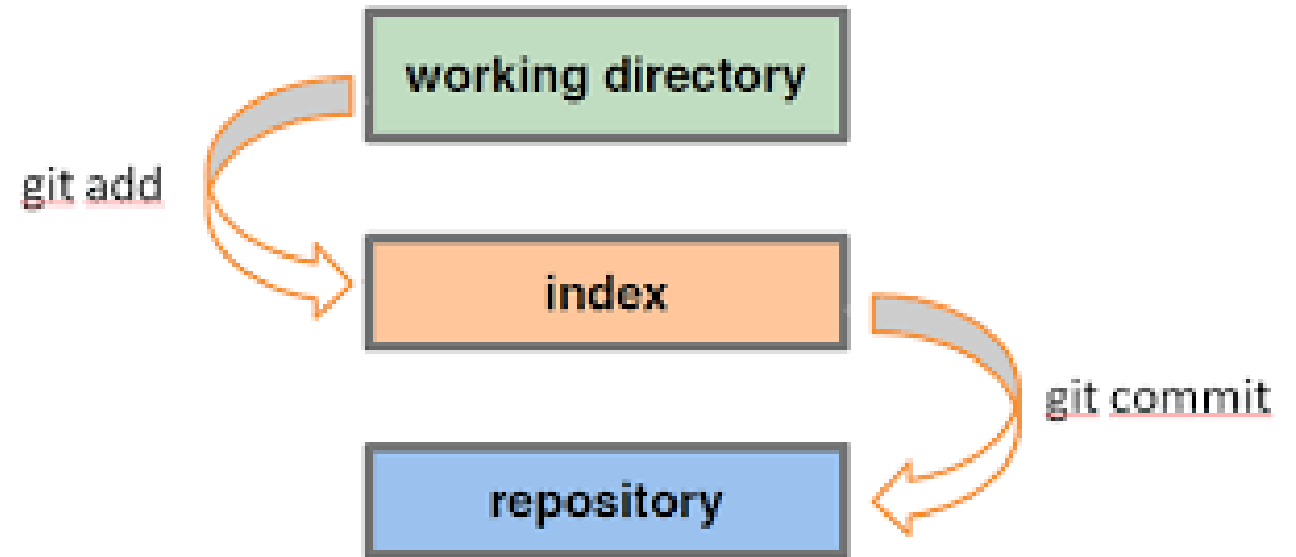
- The command `git restore --staged` doesn't modify our working directory
- Safe... just don't use index as single backup





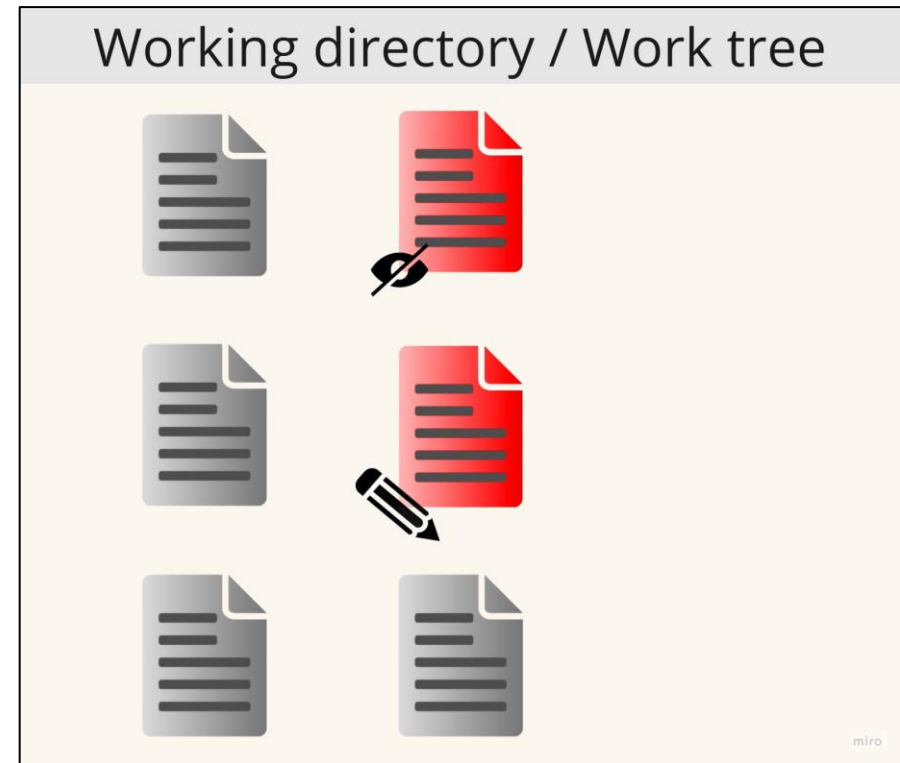
Commits: Introduction

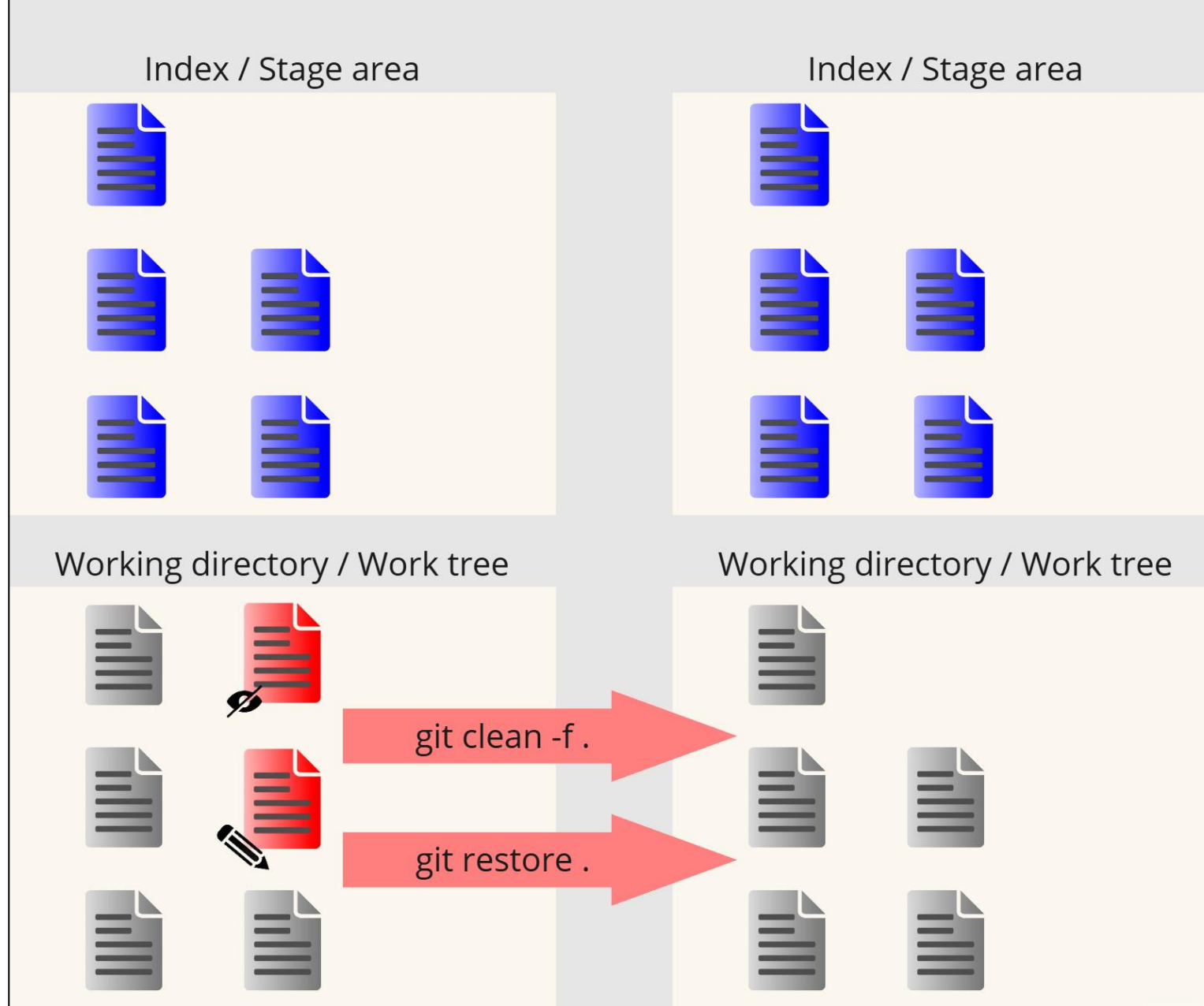
- Snapshots of changes in a repository
- They include:
 - ✓ Username and email
 - ✓ Date and Commit ID
 - ✓ Message



Discarding changes from Work tree

- There may be changes we added that we realized **no longer want to work with**
- We want to mimic the previous commit
- To deselect changes and restore the Work tree we use `git restore` and `git clean`
 - ✓ `git clean` removes untracked files
 - ✓ `git restore` for changes in tracked files





Previous Commit



Working directory



Staged files



Committed files



miro

TODO: Practice

- ✓ Initialize a repository
- ✓ Generate changes (commands)
- ✓ Select / Deselect them
- ✓ Create simple commits

