# Git Basic Staging

With this exercise we will examine the staging area of git.

In git we are working with three different areas:

- The working directory where you are making your changes.
- The staging area where all changs you have added through `git add` will stay.
- The repository where every commit ends up, making your history. To put your staged changes in here you issue the `git commit `command.

A file can have changes both in the working directory and staging area at the same time. These changes do not have to be the same.

We will also work with `git reset` to reset the staged changes of a file, and `git checkout` to return a file to a previous state.

## The task

In your documents folder create a new folder called `basic-staging`. After creating the new folder, `cd` inside your folder and create a new file called `file.txt`.

1. What is the content of `file.txt`?
2. Overwrite the content in `file.txt`: Add your name and surname to file.txt.
3. What does `git diff` tell you?
4. What does `git diff --staged` tell you? Why is this blank?
5. Run `git add file.txt `to stage your changes from the working directory.
6. What does `git diff` tell you?
7. What does `git diff --staged` tell you?
8. Overwrite the content in `file.txt` by adding your github username to the file.
9. What does `git diff `tell you?
10. What does `git diff` --staged tell you?
11. Make sure you understand what is going on.
12. Run `git status` and observe that `file.txt `is present twice in the output.
13. Run `git reset HEAD file.txt `to unstage the change.
14. What does `git status` tell you now?
15. Stage the changes and make a commit.
16. What does the log look like?
17. Overwrite the content in file.txt by adding today's date next to your github username.
18. What is the content of file.txt?
19. What does `git status` tell us?
20. Run `git checkout file.txt`
21. What is the content of file.txt?
22. What does `git status` tell us?

## Useful commands

```
git add
```

```
git commit
```

```
git commit -m "My commit message"
```

```
git reset
```

```
git checkout
```

```
git log
```

```
git log -n 5
```

```
git log --oneline
```

```
git log --oneline --graph
```

```
git reset HEAD
```

```
git checkout
```