

# git and GitHub for beginners

Mitja Kleider

# Introduction

## Why?

Do you ever edit text-based files

- ▶ on multiple devices, or
- ▶ as a team
- ▶ and want to view changes
- ▶ and need to have backups

## What you will learn:

- ▶ basics to contribute to projects
- ▶ git command line basics (there are GUI options)
- ▶ a few 'good practice' tips

# Getting started

- ▶ Install `git`.
  - ▶ Using your system's package manager,
  - ▶ from the official download page or
  - ▶ as Windows bundle including shell.
- ▶ Create a GitHub account at [github.com](https://github.com)

## More than git hosting

GitHub is just the most popular one of many options, we will be using it today. GitHub additionally offers

- ▶ an issue tracker
- ▶ file rendering (e.g. Markdown, STL)
- ▶ Wiki
- ▶ GitHub Pages

# Configuration

Open a terminal now.

```
git config --global user.name "First Last"
```

```
git config --global user.email "mail@example.com"
```

```
git config --global color.ui auto
```

Every command offers help, just call `git <command> --help`.

# First repository

- ▶ Create your working directory: `mkdir my_project/`
- ▶ Initialize a local git repository: `git init.`
- ▶ Create and edit a README.md.

## Advanced note: Distributed version control

There can be an infinite number of clones from any repository. If your server burns down, you can continue from any copy. Repositories can be cloned from local

```
git clone /path/to/repository
```

or remote

```
git clone username@host:/path/to/repository
```

sources.

# File trees

Call `git status`. There are multiple file trees:

1. working directory
2. index (staging area)
3. HEAD

## Staging and committing

Add files from working directory to staging area:

```
git add <file1> <file2>
```

Create commit from staging area:

```
git commit
```

# Pushing local copy to original repository

```
git push origin master
```

Will push your commits to the remote repository called `origin`, which is created by default when cloning.

## Add remote repository

If no remote is configured:

```
git remote add origin <server>
```

Instructions are given when you create an empty repository on GitHub. Do that now and push your stuff.

# Branching

Develop features isolated from other changes. Create a new branch for a feature and switch to it:

```
git checkout -b feature_unicorns
echo "Unicorns!!" >> changed-file.txt
git add changed-file.txt
git commit
```

Switch branch back to default branch (*master*):

```
git checkout master
git merge feature_unicorns master
```



# History

```
git log
```

## Read more

There are many tutorials, the official website **git-scm.com** is really good.

GitHub has non-commandline beginner tutorials:

- ▶ Hello World
- ▶ Understanding the GitHub Flow

on [guides.github.com](https://guides.github.com)