

In this exercise you will get in touch with real git for the first time. You will replay the LEGO session, using the git client SmartGit that have been installed earlier.

- 11. [Open your git client](#)
  - 1.1 [Step 1: Open the GitShell provided with SmartGit](#)
- 22. [Create your own local lego repo](#)
  - 2.1 [Step 1: Initialize your local lego repo](#)
  - 2.2 [Step 2: Create files](#)
  - 2.3 [Step 3: Create a commit](#)
  - 2.4 [Step 4: Create additional changes](#)
- 33. [Create a feature branch](#)
  - 3.1 [Step 1: Create and check out your feature branch](#)
  - 3.2 [Step 2: Make commits on your feature branch](#)
- 44. [Switch back to master](#)
  - 4.1 [Step 1: Switch back to the master branch](#)
  - 4.2 [Step 2: Make additional commits on your master branch](#)
- 55. [Inspect your history](#)
  - 5.1 [Step 1: Check your log](#)

To be able to go through this tutorial it is necessary that you have installed SmartGit as described in [lecture IV](#).

## 1. Open your git client

---

### Step 1: Open the GitShell provided with SmartGit

Browse to the location of your first repository that you have created in lecture IV.

(D:\casdev\ContinuousIntegrationWithGitHub)

Do a right click In your ContinuousIntegrationWithGitHub repository folder and select **Open in SmartGit**.

Select in the **Tools** menu at the top **Open Git-Shell**.

Move with your context to D:\casdev with executing:

```
$ cd ..
```

## 2. Create your own local lego repo

---

### Step 1: Initialize your local lego repo

\$ mkdir lego - create a directory to place your git repository in on your disk with the name lego.

\$ cd lego - change your current position to the lego directory.

\$ git status - git status returns your current repository status to you. Read the feedback from this command carefully. This time, you will receive an error message, because you have not initialized the repository yet.

\$ git init - initializes a new git repository inside the lego folder.

\$ git status - read the feedback from this command carefully. This time, you will receive information about your new repository that you have created with the git init command,

## Step 2: Create files

\$ echo "S" > gear - create the file gear with the content S.

\$ git status - read the feedback from this command carefully. Can you map the feedback into the lego model from the lego session video?

\$ echo "M" > wheel - create the file wheel with the content M.

\$ git status - read the feedback from this command carefully. Can you map the feedback into the lego model from the lego session video?

## Step 3: Create a commit

\$ git add gear - add the gear file to the index area.

\$ git status - read the feedback from this command carefully. Can you map the feedback into the lego model from the lego session video?

\$ git add wheel - add the wheel file to the index area.

\$ git status - read the feedback from this command carefully. Can you map the feedback into the lego model from the lego session video?

\$ git commit - create a new commit object.

\$ git status - read the feedback from this command carefully. Can you map the feedback into the lego model from the lego session video?

## Step 4: Create additional changes

And now it is your turn: Create new files, change already existing files, add your changes and commit them. 😊

# 3. Create a feature branch

---

## Step 1: Create and check out your feature branch

\$ git branch feature - create a new branch with the name feature at your current HEAD position.

\$ git status - read the feedback from this command carefully. On which branch are you currently working?

\$ git checkout feature - check out the feature branch to continue working on this branch.

\$ git status - read the feedback from this command carefully. On which branch are you currently working?

## Step 2: Make commits on your feature branch

And now it is your turn: Create new files, change already existing files, add your changes and commit them. 😊

## 4. Switch back to master

---

### Step 1: Switch back to the master branch

\$ git checkout master - check out the master branch and continue working on it.

\$ git status - read the feedback from this command carefully.

### Step 2: Make additional commits on your master branch

And now it is your turn: Create new files, change already existing files, add your changes and commit them. 😊

## 5. Inspect your history

---

### Step 1: Check your log

\$ git log - check the commits in your log view. Leave the log view with "q" like quit.