

Exercise 1: Command line and Git

General Information:

- Exercises must be handed in individually.
- Submission deadline is **24.10.2018, 11:55 PM** through **Moodle**.
- Each exercise in this course contains 50 points. So from all six exercises you can achieve 300 points. One of the requirements to pass this course is that you achieve at least 150 points. Some exercises also contain bonus points.
- **Read all instructions carefully!**

Documents for submission on Moodle:

- **PDF file containing the documentation** of how you solved the exercises and the answers to the questions. Name it according to “*ex01_yourname_matriclenumber.pdf*” e.g. *ex01_MelanieMüller_1234556.pdf*.

Note: Include your name, matriculation number and GitHub name in the beginning of your documentation file. Your work on GitHub is part of the grading!

In this exercise, you will practice working in the command line and using git to track your work progress. Read the instructions carefully and document your answers in a PDF document that you submit to Moodle.

Some of these exercises require you to work in the command line. In this case, provide all the commands that you used to solve the task and add a brief comment what it does as explained in the seminar (see slides). If the question requires you to provide the output of a command that is shown in the command line, copy and paste it from the command line into your documentation file along with the command that was used to create it and a brief comment.

Introduction

Let's assume we found a GitHub Repository online which contains data that might be useful for us. So we would like to download the repository and explore it a bit. Therefore, we will fork the repository, clone it to our local computer, explore the data sets and save our results. The following exercises will guide you through this task.

Exercise 1

10 points

1. Create a GitHub account and register for [GitHub Education](#) to get unlimited free private repositories. Provide your GitHub user name in the documentation.
2. Fork the following GitHub repository using your GitHub account:

https://github.com/redfrexx/OpenSourceGIS_exercise01.git

Now you should have a new GitHub repository called “OpenSourceGIS_exercise01” in your GitHub account. Provide the link to this repository in the documentation and answer the following questions:

- a) What does “forking” mean in GitHub and why is it useful?
- b) What does “cloning” mean in GitHub?

Exercise 2

10 points

Now we want to get a copy of the repository on our local computer, so we can explore the data. For solving the following tasks (exercises 2-4) **you must use the command line** (“Eingabeaufforderung” or OSGeo4W shell as indicated in the exercise) and **provide all the commands you used along with a brief comment** about what it does.

1. Open the command line (“Eingabeaufforderung”). Check in which directory you are located and list all files. Provide the command.
2. Create a folder called “exercise01” and navigate into this folder. Provide the commands.
3. Clone your repository using the url of your forked repository. It should look something like this but you **have to fill in your github user name**:

`https://github.com/your_github_user_name/OpenSourceGIS_exercise01.git`

Provide the command.

4. Now we want to check whether all files were downloaded. So navigate into the top folder of your repository and list all files contained in it. Provide the command and answer the following questions:
 - a. How many raster files are displayed?
 - b. How many vector files are displayed?

Exercise 3

10 points

Now we would like to find out more about the raster file in the repository using a command called “*gdalinfo*”, which is part of the GDAL library

1. Find out what the command *gdalinfo* does using the GDAL documentation (<https://www.gdal.org>) or further online research. Describe it in the documentation.

This command is part of the OSGeo distribution and not a standard Windows program. So, if you execute it in the normal command line, you will get a message saying that the command was not found. **The command is only available in the OSGeo4W Shell.** Therefore, do the following:

2. Open the OSGeo4W shell and print your current working directory. By default you are located in C:\Windows\System32. Navigate to your user directory.

Hint 1: at the university PC Pool your user directory is “M:\”.

Hint 2: When changing hard drives (from C:\ to M:\) you need to execute “M:\” after cd.

3. Navigate to the main folder of your repository and list all files contained in it. There is a file called “relief_san_andres.tif”. You want to find out more about it by using *gdalinfo*. Therefore, execute the following command in the OSGeo4W shell:

`gdalinfo relief_san_andres.tif`

Based on the output of this command, answer the following questions:

- a. What is the coordinate reference system of the raster file?
 - b. What is the pixel size? Don’t forget to provide the units!
4. What happens if you add the flag “-json” to the command and execute it? What is the difference to the output that is generated without it (see previous command)? (hint: use the GDAL documentation to get some more information)

gdalinfo relief_san_andres.tif -json

5. Redirect the output of the last command to a new file called *"relief_gdalinfo.json"*. List all files of the directory to check if the new file was created. Then display the contents of the file in the command line to check whether it is not empty. Provide all commands and along with brief comments in the documentation.

Exercise 4

10 points

For later reference, we want to store our results (relief_gdalinfo.json) in the repository. (If you didn't manage to create the .json file in exercise 3, create new file using a text editor (e.g. Notepad++) with the same name to continue with this exercise)

The git commands are not available in the OSGeo4W shell. Therefore, we need to switch back to the normal command line. If the normal command line is not open anymore, open it again and navigate to the top folder of your repository.

1. Check the status of your repository. Provide the command along with the output in the documentation.
2. In the section "Untracked files" of the output you should see the newly created "relief_gdalinfo.json" marked in red. (If not, make sure you are located in the right folder and that it exists.) What does this mean?
3. As a final step, add "relief_gdalinfo.json" to your GitHub repository by staging the file, creating a commit and pushing your changes to the remote repository. Provide all commands along with brief comments in the documentation. Verify that "relief_gdalinfo.json" has been successfully added to the remote repository by opening your repository on github.com using an internet browser.

Exercise 5

10 points

1. You are given a data set containing the locations of trees in Heidelberg along with information about their species, height and crown diameter. Come up with a geographic question or geospatial service that could be answered/provided using this data set. Explain your topic in one paragraph in your documentation file. If you need additional datasets for your topic, mention them in your description.
2. You are the proud owner of a new food truck in Heidelberg. For the next "Schlossbeleuchtung" (fireworks at the castle), you want to place your food truck at a location where you will have a large number of customers. Therefore, the location should fulfill the following requirements:
 - a. It needs to have a good view of the castle.
 - b. It should be located in an area where many potential customers pass by.
 - c. It should be located in an area where there are no or only a few other options to eat.

Which data sets do you need to answer this question?

Bonus exercise:

5 Points

Do an online research for data sets suitable to find a location for your food truck and list them here explaining why you chose them and why you assume they are useful. Only requirement is that the data is free. Check out the links in the slides for further advice of where to find spatial data sets.