

# Git & GitHub


## Handout

### Contents

Installation .....	2
Recup Git & GitHub Basics.....	5
Git Bash.....	7
git clone.....	8
git branch & git checkout branch.....	9
Git in RStudio .....	9
git status & git restore .....	13
git config.....	13
git add & git commit & git push .....	13
git fetch & git pull.....	15
git merge.....	16
Git & GitHub Workflow .....	17

# Installation

Website: <https://git-scm.com/downloads>




 **git** --local-branching-on-the-cheap

Type / to search entire site...

[About](#)  
[Documentation](#)  
**[Downloads](#)**  
    [GUI Clients](#)  
    [Logos](#)  
  
[Community](#)

The entire **Pro Git book** written by Scott Chacon and Ben Straub is available to [read online for free](#). Dead tree versions are available on [Amazon.com](#).

## Downloads

 [macOS](#)     [Windows](#)  
  
 [Linux/Unix](#)

Latest source Release  
**2.50.0**  
[Release Notes \(2025-06-16\)](#)  
[Download for Windows](#)

Older releases are available and the Git source repository is on GitHub.

### GUI Clients

Git comes with built-in GUI tools (**git-gui**, **gitk**), but there are several third-party tools for users looking for a platform-specific experience.

[View GUI Clients →](#)

### Logos

Various Git logos in PNG (bitmap) and EPS (vector) formats are available for use in online and print projects.

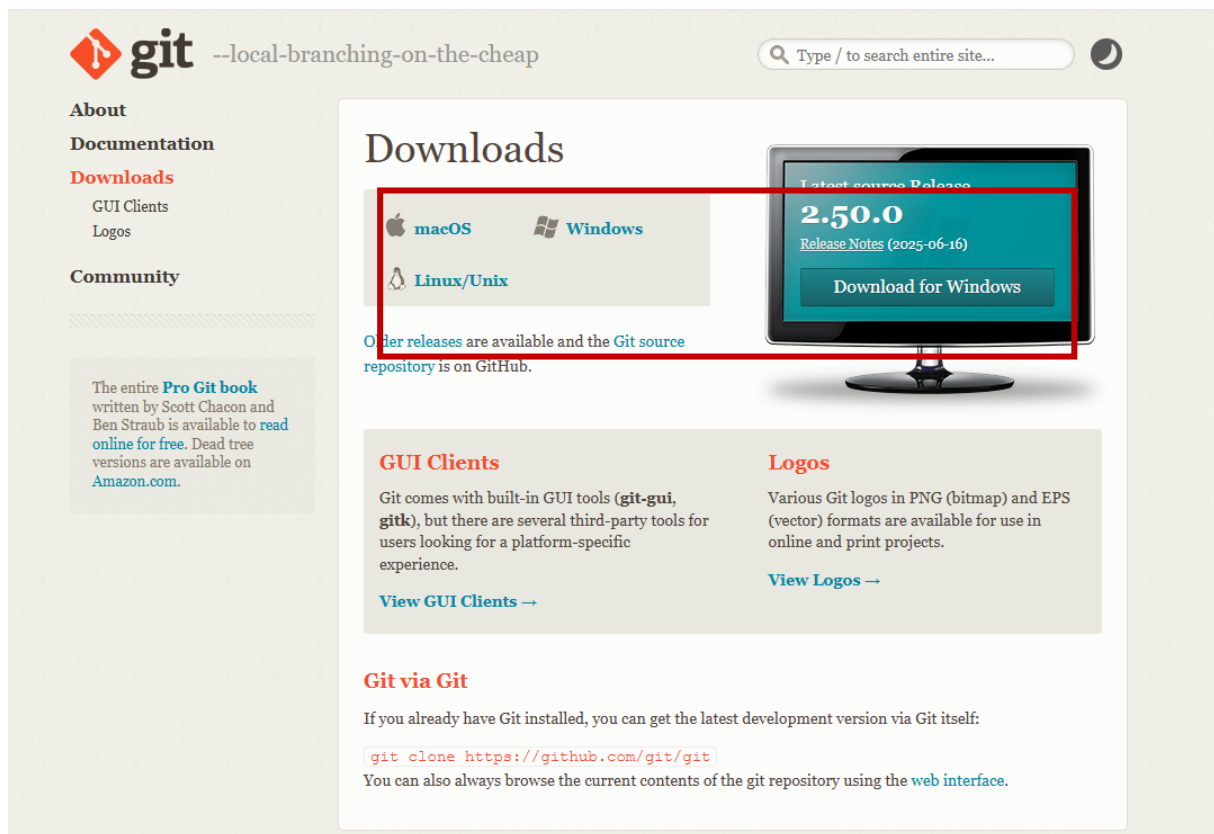
[View Logos →](#)

### Git via Git

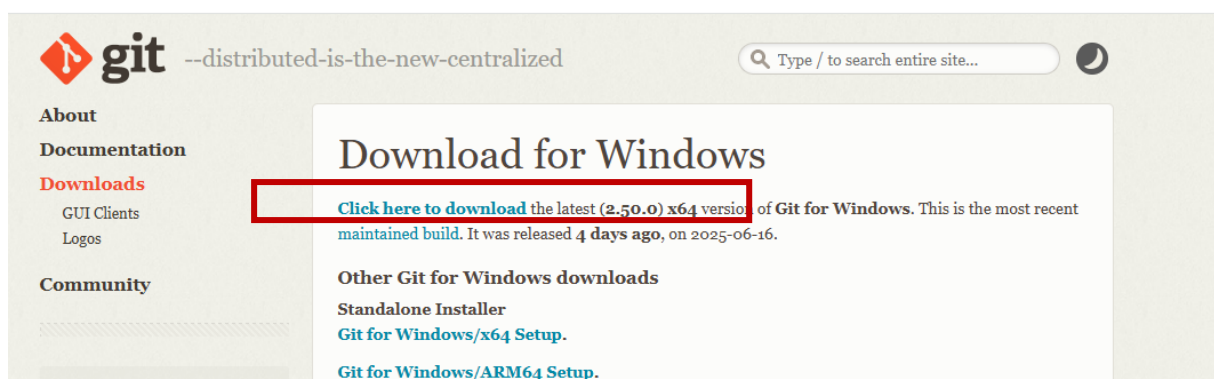
If you already have Git installed, you can get the latest development version via Git itself:

```
git clone https://github.com/git/git
```

You can also always browse the current contents of the git repository using the [web interface](#).



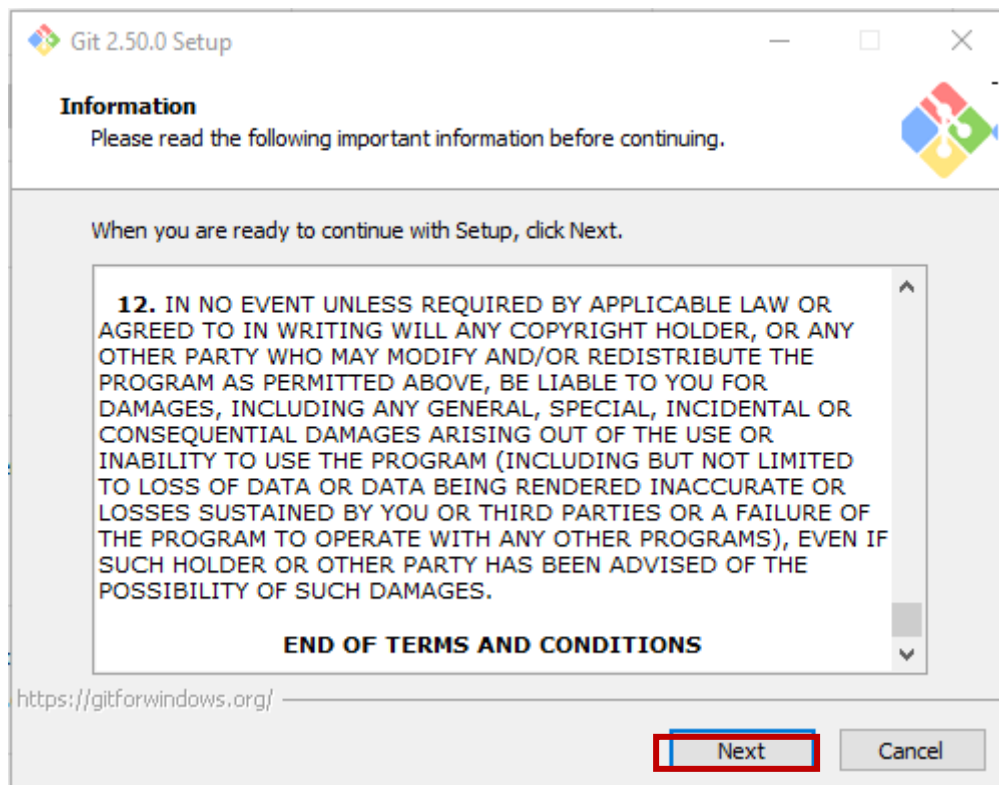
Download installation file



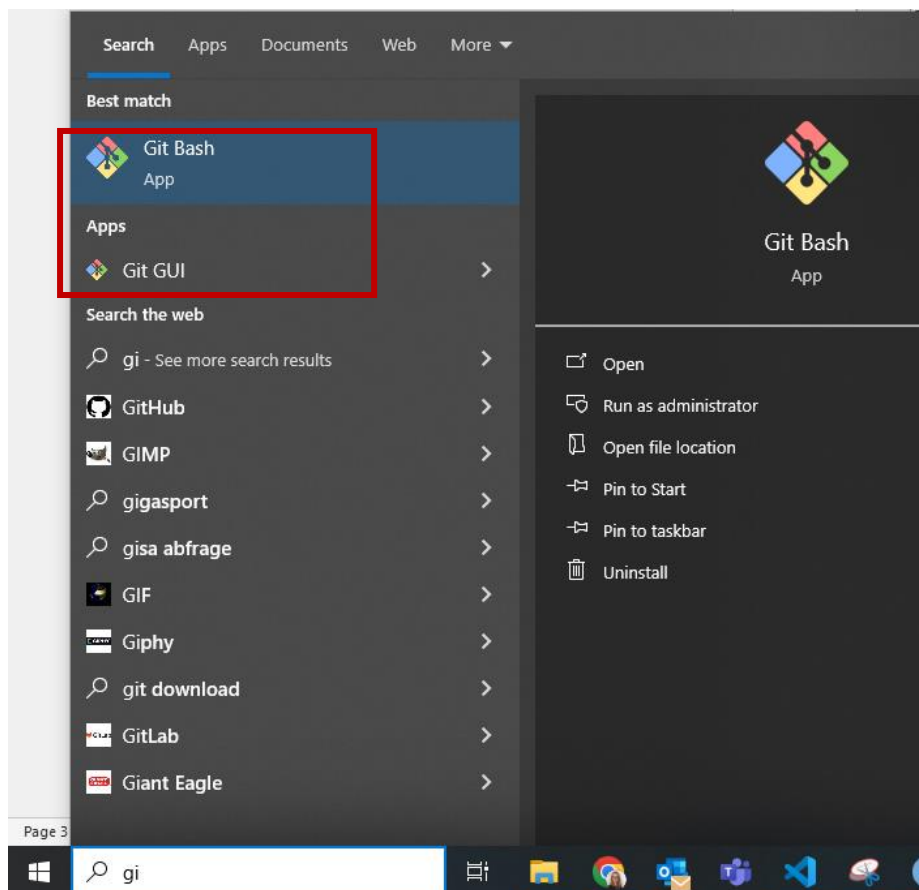
Open installation file from your Downloads

Name	Date modified	Type	Size
▼ Today (1)			
Git-2.50.0-64-bit.exe	20.06.2025 09:54	Application	69 141 KB
▼ Earlier this week (1)			
OfficeSetup (2).exe	18.06.2025 12:23	Application	7 470 KB

Install (Next -> "Leave Default Settings" -> Install)



Check for Git Apps after installation



# Recall: Git & GitHub Basics

## Create a new repo


### Create a new repository

[Preview](#)[Switch back to classic experience](#)

Repositories contain a project's files and version history. Have a project elsewhere? [Import a repository](#).  
Required fields are marked with an asterisk (\*).

#### 1 General

Owner \*

 sagrub ▾

Repository name \*

wcl-lake-data-sg

✔ wcl-lake-data-sg is available.

Great repository names are short and memorable. How about **symmetrical-funicular**?

Description


Repository for publishing physio-chemical data from the Lunz lake

66 / 350 characters

#### 2 Configuration

Choose visibility \*

Choose who can see and commit to this repository

 Public ▾

Add README

READMEs can be used as longer descriptions. [About READMEs](#)

On ☒

Add .gitignore

.gitignore tells git which files not to track. [About ignoring files](#)

 R ▾

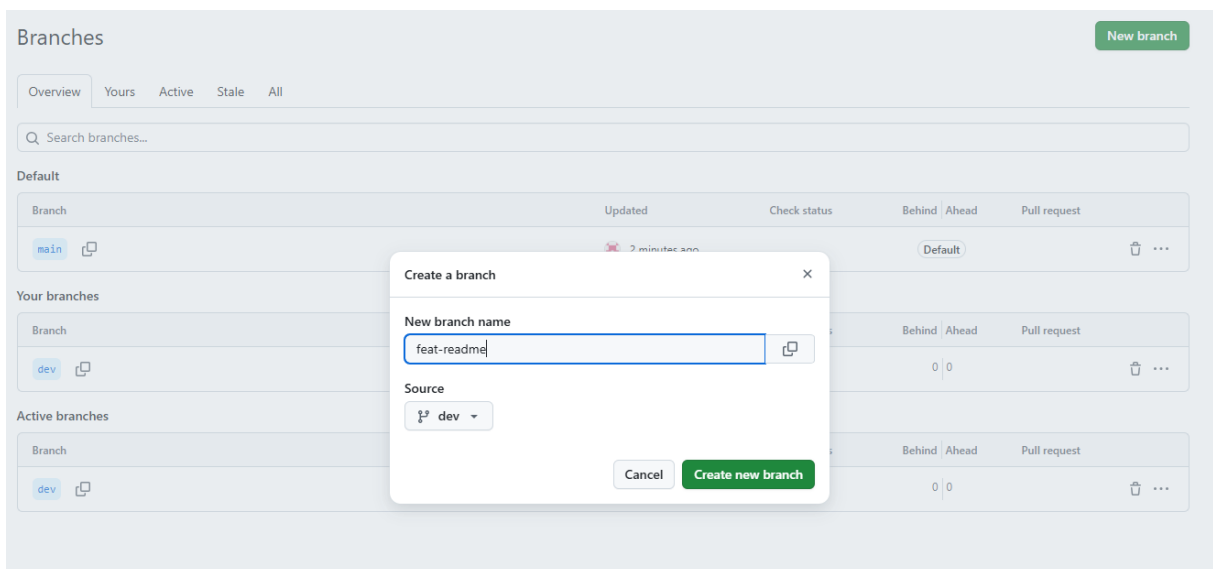
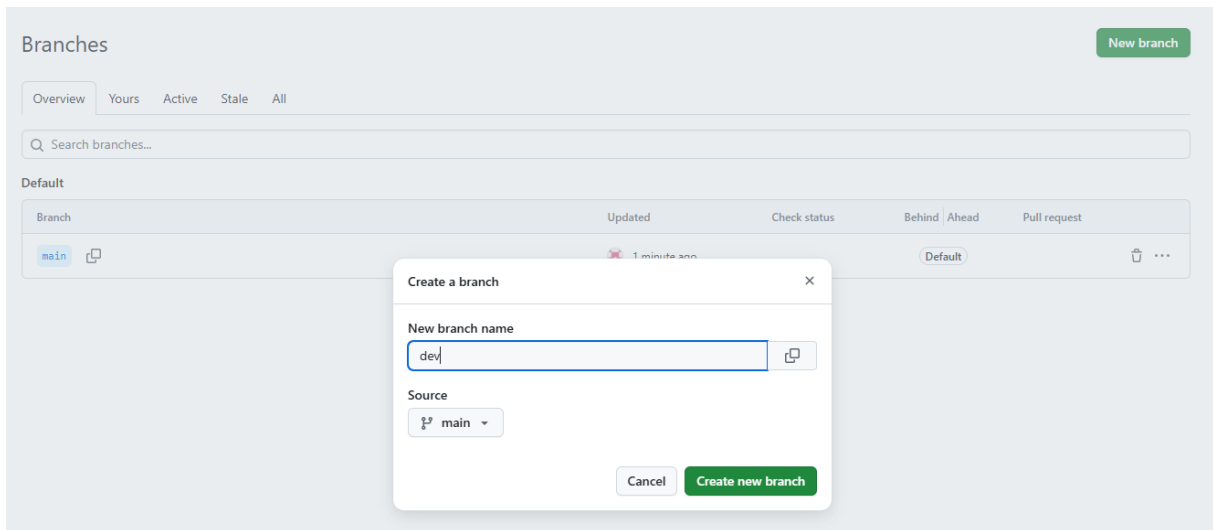
Add license

Licenses explain how others can use your code. [About licenses](#)

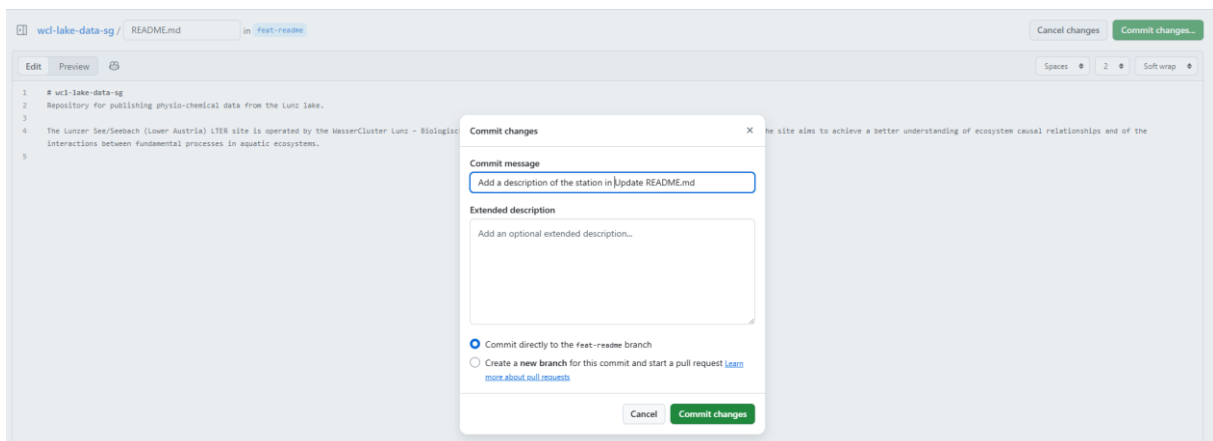
 MIT License ▾

Create repository

## Create a new branch (dev and feat-branch)



## Commit



## Create a pull request

### Comparing changes

Choose two branches to see what's changed or to start a new pull request. If you need to, you can also [compare across forks](#) or [learn more about diff comparisons](#).

base: dev ← compare: feat-readme

Discuss and review the changes in this comparison with others. [Learn about pull requests](#) [Create pull request](#)

1 commit 1 file changed 1 contributor

Commits on Aug 19, 2025

Add a description of the station in Update README.md

Verified 37057f0

sagrub authored now

Showing 1 changed file with 2 additions and 0 deletions. [Split](#) [Unified](#)

2 README.md

@@ -1,2 +1,4 @@

1 1 # wcl-lake-data-sg

2 2 Repository for publishing physio-chemical data from the Lunz lake.

3 +

4 + The Lunzer See/Seebach (Lower Austria) LTER site is operated by the WasserCluster Lunz - Biologische Station GmbH and serves as a long-term research and monitoring area. The site aims to achieve a better understanding of ecosystem causal relationships and of the interactions between fundamental processes in aquatic ecosystems.

## Git Bash

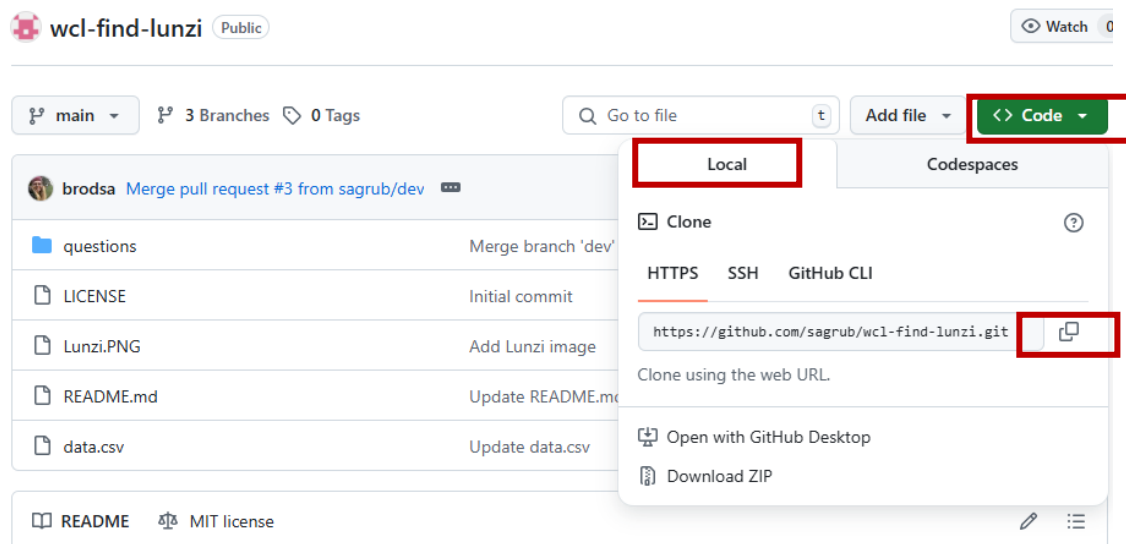
Open Git Bash and type following commands

- ls ... to list files
- cd Documents or cd Dokumente ... to go to Documents
- ls ... to list files in Documents
- mkdir git ... to create git folder
- cd git ... to change the working directory to git
- touch file.txt ... to create a new file
- rm file.txt ... to remove the file

```
MINGW64/c/Users/grubner
grubner@WCLNB17 MINGW64 ~
$ ls
'3D Objects'/' ntuser.dat.LOG1
Anwendungsdaten@ ntuser.dat.LOG2
AppData/' NTUSER.DAT{53b39e88-18c4-11ea-a811-000d3aa4692b}.TM.blf
Contacts/' NTUSER.DAT{53b39e88-18c4-11ea-a811-000d3aa4692b}.TMContainer000000000000000001.regtrans-ms
Cookies@ NTUSER.DAT{53b39e88-18c4-11ea-a811-000d3aa4692b}.TMContainer000000000000000002.regtrans-ms
Desktop/' ntuser.ini
Documents/' OneDrive/
Downloads/' 'OneDrive - WasserCluster Lunz - Biologische Station GmbH'/
Dropbox/' Pictures/
Druckumgebung@ Recent@
'Eigene Dateien'@ 'Saved Games'/
Favorites/' Searches/
Links/' SendTo@
'Lokale Einstellungen'@ Startmenu@
Microsoft/' Videos/
Music/' Vorlagen@
Netzwerkumgebung@ 'WasserCluster Lunz - Biologische Station GmbH'/
NTUSER.DAT NTUSER.DAT
grubner@WCLNB17 MINGW64 ~
$
```

# git clone

Copy the url adresse of your remote git repository



Clone your remote repository – git clone

```
grubner@WCLNB17 MINGW64 ~/Documents/github
$ git clone https://github.com/sagrub/wcl-find-lunzi.git

grubner@WCLNB17 MINGW64 ~/Documents/github
$ git clone https://github.com/sagrub/wcl-find-lunzi.git
Cloning into 'wcl-find-lunzi'...
remote: Enumerating objects: 49, done.
remote: Counting objects: 100% (49/49), done.
remote: Compressing objects: 100% (40/40), done.
remote: Total 49 (delta 14), reused 0 (delta 0), pack-reused 0 (from 0)
Receiving objects: 100% (49/49), 61.70 KiB | 30.85 MiB/s, done.
Resolving deltas: 100% (14/14), done.
```

Check the local repository – use ls and cd command

```
grubner@WCLNB17 MINGW64 ~/Documents/github
$ ls
dwh_wcl/  loopitoy/  synchro_local_to_cloud/  test/  wcl-find-lunzi/

grubner@WCLNB17 MINGW64 ~/Documents/github
$ cd wcl-find-lunzi/

grubner@WCLNB17 MINGW64 ~/Documents/github/wcl-find-lunzi (main)
$ ls
data.csv  LICENSE  Lunzi.PNG  questions/  README.md
```



## git branch & git checkout branch

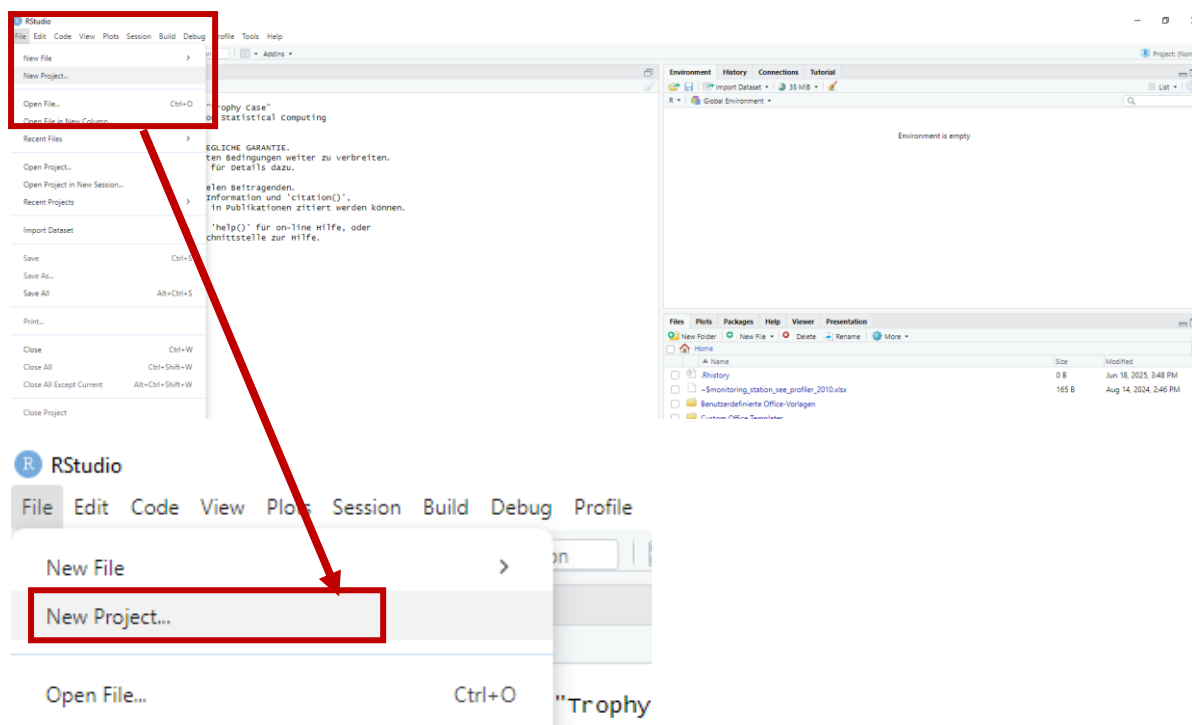
```
grubner@WCLNB17 MINGW64 ~/Documents/git/find-lunzi-sg (dev)
$ git branch -r
origin/HEAD -> origin/main
origin/dev
origin/feat-question-2
origin/main

grubner@WCLNB17 MINGW64 ~/Documents/git/find-lunzi-sg (dev)
$ git checkout feat-question-2
Switched to a new branch 'feat-question-2'
branch 'feat-question-2' set up to track 'origin/feat-question-2'.

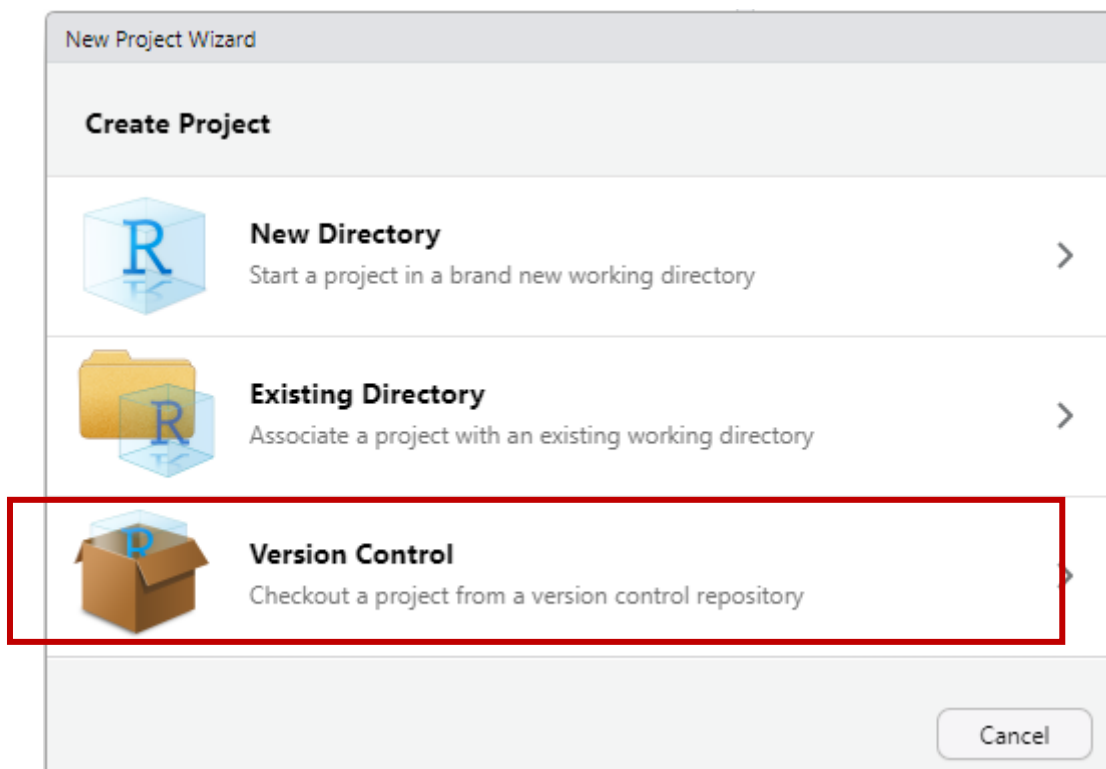
grubner@WCLNB17 MINGW64 ~/Documents/git/find-lunzi-sg (feat-question-2)
$
```

## Git in RStudio

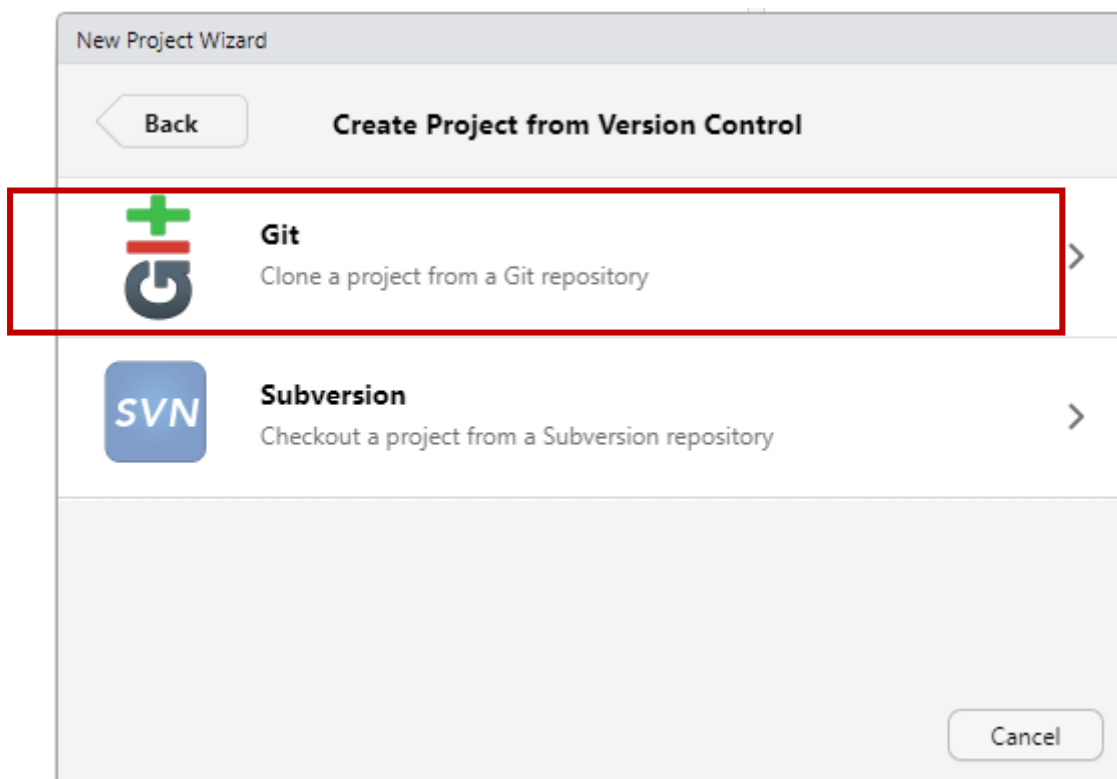
Open RStudio and Create New Project (File -> New Project)



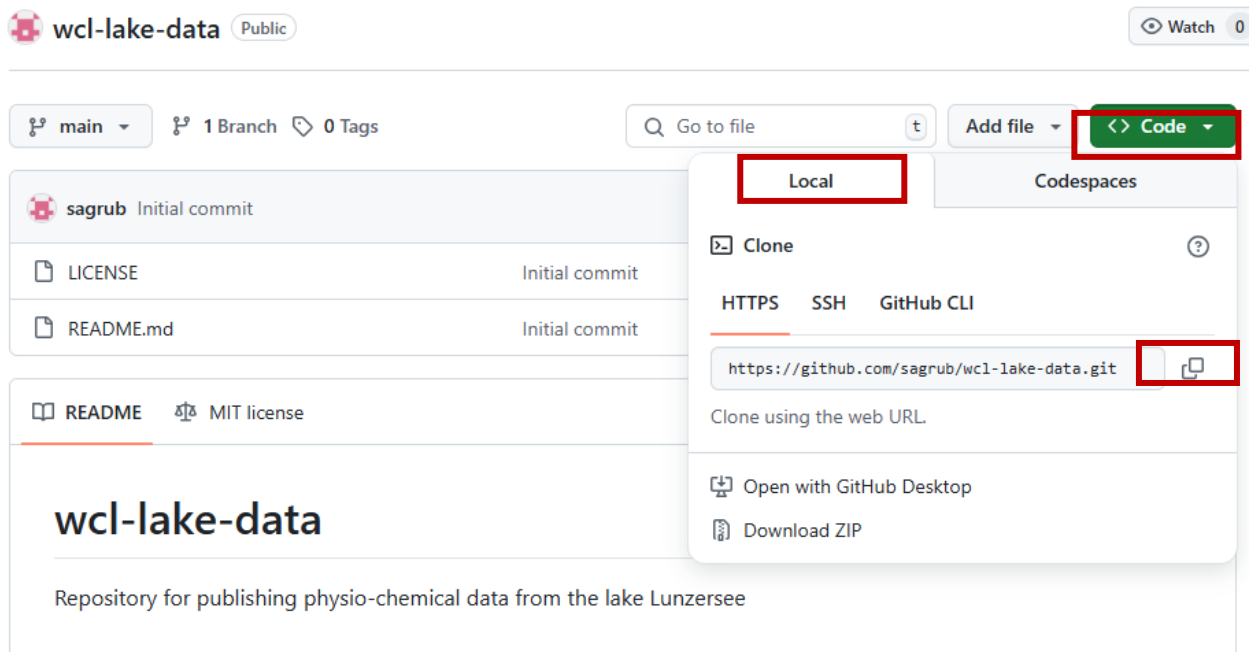
## Select Version Control



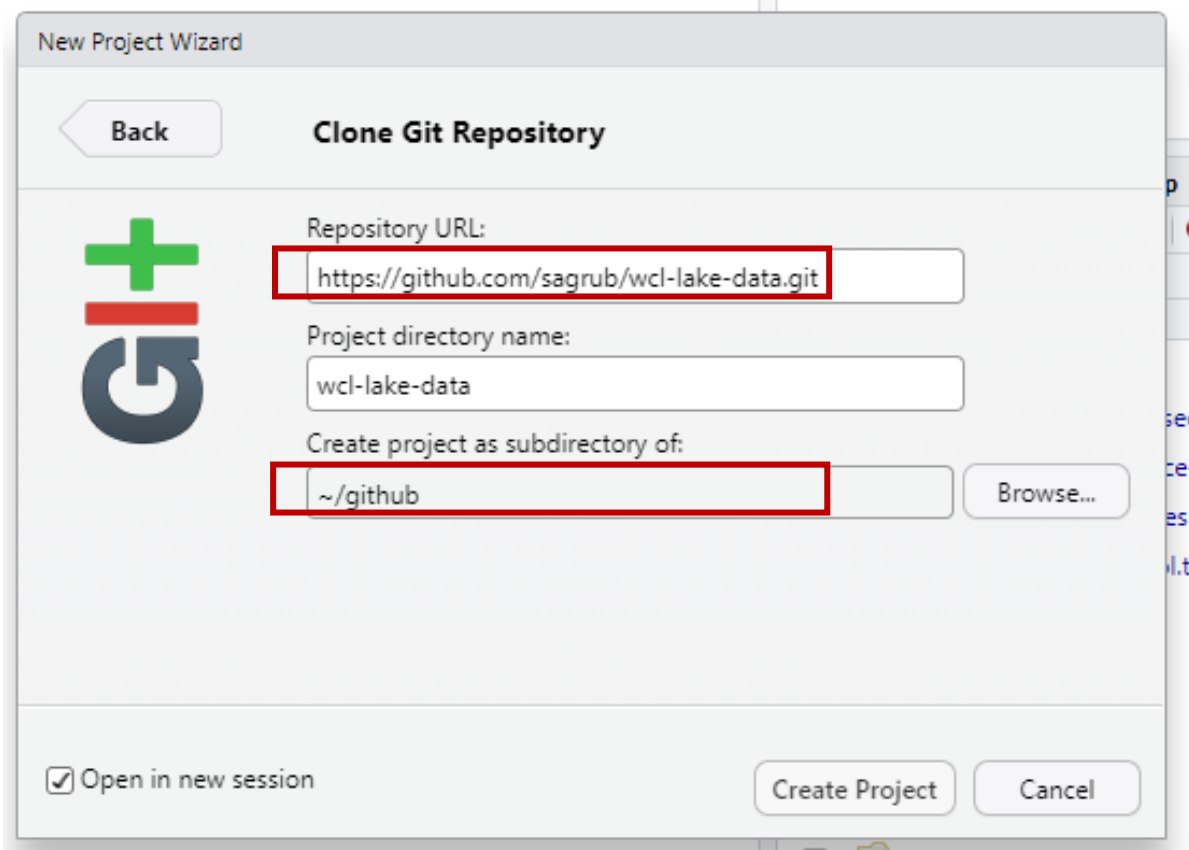
## Chose Git



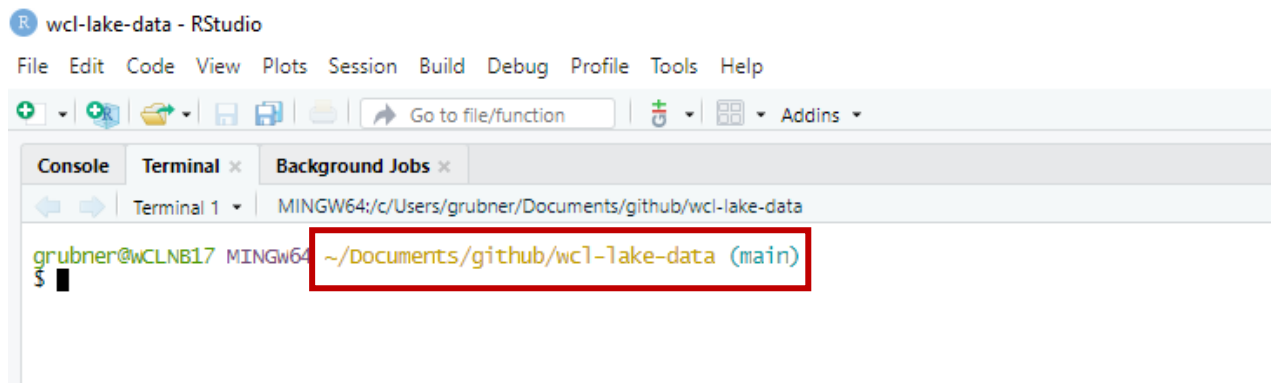
Copy the URL address of your remote GitHub repository (web-browser)



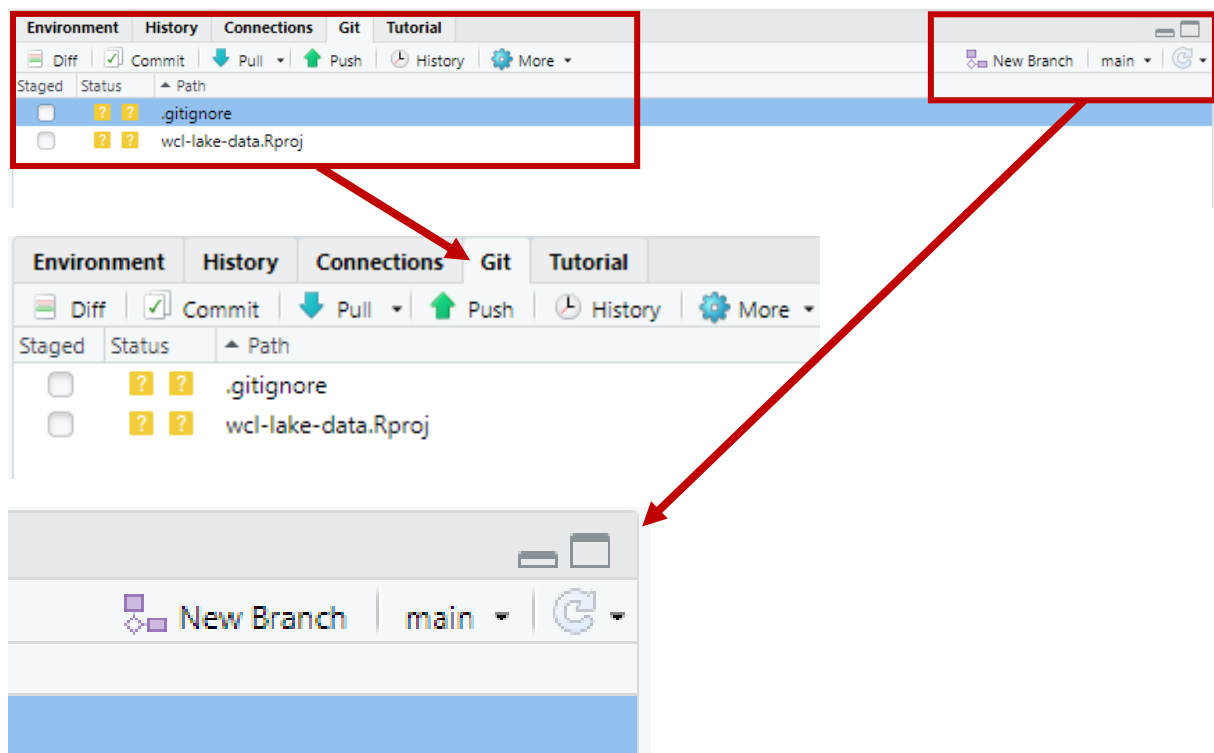
Insert the URL and select subdirectory:



## Terminal Window



## Git & Terminal Window



## git branch & git checkout in Terminal Window

```
grubner@WCLNB17 MINGW64 ~/Documents/git/wcl-lake-data-sg (main)
$ git branch -r
origin/HEAD -> origin/main
origin/dev
origin/feat-readme
origin/main

grubner@WCLNB17 MINGW64 ~/Documents/git/wcl-lake-data-sg (feat-readme)
$ git checkout feat-readme
Already on 'feat-readme'
Your branch is up to date with 'origin/feat-readme'.

grubner@WCLNB17 MINGW64 ~/Documents/git/wcl-lake-data-sg (feat-readme)
$ git checkout -b feat-get-data
Switched to a new branch 'feat-get-data'
```

## git status & git restore

1. Make a small change in README.md
2. Check the status – git status
3. Restore the changes – git restore README.md
4. Check the status – git status

```
grubner@WCLNB17 MINGW64 ~/Documents/git/wcl-lake-data-sg (feat-readme)
$ git status
On branch feat-readme
Your branch is up to date with 'origin/feat-readme'.

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   README.md

no changes added to commit (use "git add" and/or "git commit -a")

grubner@WCLNB17 MINGW64 ~/Documents/git/wcl-lake-data-sg (feat-readme)
$ git restore README.md

grubner@WCLNB17 MINGW64 ~/Documents/git/wcl-lake-data-sg (feat-readme)
$ git status
On branch feat-readme
Your branch is up to date with 'origin/feat-readme'.

nothing to commit, working tree clean

grubner@WCLNB17 MINGW64 ~/Documents/git/wcl-lake-data-sg (feat-readme)
$
```

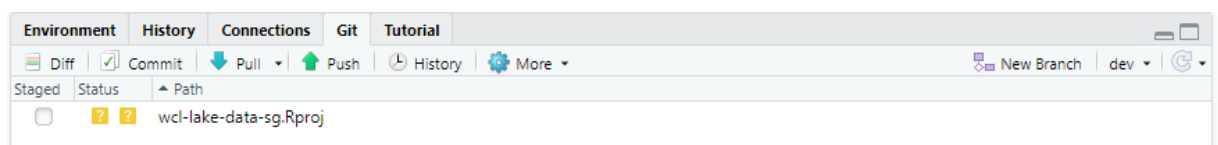
## git config

```
git config --global user.name "Your Name"
git config --global user.email you@example.com
```

## git add & git commit & git push

Make sure you have created the branch **feat-get-data** on GitHub before cloning the repository. If not, proceed with **git pull** to get the feature branch to your local repository.

1. Switch to dev branch from the Git Window



2. Switch to feat-get-data from the Terminal using **git checkout feat-get-data**

```
Console Terminal x Background Jobs x
Terminal 1 MINGW64/c/Users/grubner/Documents/git/wcl-lake-data-sg
$ git branch -r
origin/HEAD -> origin/main
origin/dev
origin/feat-get-data
origin/feat-readme
origin/main

grubner@WCLNB17 MINGW64 ~/Documents/git/wcl-lake-data-sg (dev)
$ git checkout feat-get-data
Switched to a new branch 'feat-get-data'
branch 'feat-get-data' set up to track 'origin/feat-get-data'.

grubner@WCLNB17 MINGW64 ~/Documents/git/wcl-lake-data-sg (feat-get-data)
$
```

3. Copy get\_data.R and .env into your project
4. Update gitignore file to exclude .env from the git repository

```
.gitignore x get_data.R x .env x
1 # Custom files
2 .env
3
4
5 # History files
6 .Rhistory
7 .Rapp.history
8
```

5. Check if it was excluded by checking the status of staging are using **git status**

```
grubner@WCLNB17 MINGW64 ~/Documents/git/wcl-lake-data-sg (feat-get-data)
$ git status
on branch feat-get-data
changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   .gitignore

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        get_data.R
        wcl-lake-data-sg.Rproj

no changes added to commit (use "git add" and/or "git commit -a")
```

6. Another check of excluding .env file with credentials is looking at the Git Window (M-modified, ?-new file, D-deleted file, A-added file into staging)

Environment	History	Connections	Git	Tutorial
Diff	Commit	Pull	Push	History
More				
Staged	Status	Path		
<input type="checkbox"/>	M	.gitignore		
<input type="checkbox"/>	? ?	get_data.R		
<input type="checkbox"/>	? ?	wcl-lake-data-sg.Rproj		

7. Git add all files by using **git add .** and check the status of staging area by **git status**

```
grubner@WCLNB17 MINGW64 ~/Documents/git/wcl-lake-data-sg (feat-get-data)
$ git commit -m "create the code to get raw lake data"
[feat-get-data a5c5024] create the code to get raw lake data
Committer: Sarka Grubner <grubner@wcl.local>
Your name and email address were configured automatically based
on your username and hostname. Please check that they are accurate.
You can suppress this message by setting them explicitly:

    git config --global user.name "Your Name"
    git config --global user.email you@example.com
```

After doing this, you may fix the identity used for this commit with:

```
git commit --amend --reset-author

3 files changed, 117 insertions(+)
create mode 100644 get_data.R
create mode 100644 wcl-lake-data-sg.Rproj
```

8. Use **git push** to upload the changes to your remote repository

```
grubner@WCLNB17 MINGW64 ~/Documents/git/wcl-lake-data-sg (feat-get-data)
$ git push
Enumerating objects: 7, done.
Counting objects: 100% (7/7), done.
Delta compression using up to 12 threads
Compressing objects: 100% (5/5), done.
Writing objects: 100% (5/5), 2.39 KiB | 2.39 MiB/s, done.
Total 5 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/brodsa/wcl-lake-data-sg.git
   a5a615f..28985a9 feat-get-data -> feat-get-data

grubner@WCLNB17 MINGW64 ~/Documents/git/wcl-lake-data-sg (feat-get-data)
```

9. Check the branch on GitHub (i.e. the remote repository)
10. Pull the changes from feature branch into dev on GitHub

## git fetch & git pull

Make sure you have changed the readme file on GitHub

1. Switch to feat-readme branch on your local computer

```
grubner@WCLNB17 MINGW64 ~/Documents/git/wcl-lake-data-sg (dev)
$ git checkout feat-readme
Switched to branch 'feat-readme'
Your branch is behind 'origin/feat-readme' by 1 commit, and can be fast-forwarded.
(use "git pull" to update your local branch)

grubner@WCLNB17 MINGW64 ~/Documents/git/wcl-lake-data-sg (feat-readme)
$
```

2. Get the changes using **git fetch**. Something has changed but your files are untouched.

```
grubner@wCLNB17 MINGW64 ~/Documents/git/wcl-lake-data-sg (feat-readme)
$ git fetch
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Compressing objects: 100% (3/3), done.
remote: Total 3 (delta 2), reused 0 (delta 0), pack-reused 0 (from 0)
Unpacking objects: 100% (3/3), 927 bytes | 132.00 KiB/s, done.
From https://github.com/brodsa/wcl-lake-data-sg
476d508..494dec4 feat-readme -> origin/feat-readme
```

```
1 # wcl-lake-data-sg
2 Repository for publishing physio-chemical data from the lake Lunzersee.
3
4 ## Site description
5 The Lunzer See/Seebach (Lower Austria) LTER site is operated by the wasserCluster Lunz - Biolog
6 |
```

3. Check for changes
  - a. git log feat-readme..origin/feat-readme
  - b. git log --online feat-readme..origin/feat-readme
  - c. git diff feat-readme..origin/feat-readme
  - d. git diff feat-readme..origin/feat-readme README.md – if you know the file
  - e. git diff commit-key
4. If everything is fine, you can pull the changes using **git pull**

## git merge

Prepare a feat-readme for pull (merge) request. Make sure you have pushed all changes from your local feat-readme to remote repository.

1. On git
  - a. Switch to dev - **git checkout dev**
  - b. Fetch and check for changes - **git fetch & git log & git diff**
  - c. Get the changes – **git pull**
  - d. Switch to feat-readme – **git checkout feat-readme**
  - e. Merge dev into feat-readme – **git merge dev**
  - f. Resolve conflicts if needed
    - i. To get the list of files with conflicts
    - ii. git diff --name-only --diff-filter=U
2. on GitHub
  - a. create pull request
  - b. merge feat-readme into dev



## Git & GitHub Workflow

1. Create new feature/fix/doc branch from dev branch on GitHub
2. Get & Check for the remote changes (git fetch & git log & git diff)
3. Pull the remote changes to your local repository
4. Switch to the new feature branch (git checkout)
5. Develop and commit your changes (git add & git status & git commit & git push)
6. Switch to the dev (source) branch of the feature branch
7. Get & Check the latest version (git fetch & git log & git diff)
8. Switch back to the feature branch
9. Merge the feature branch with the source branch
10. Resolve conflicts if necessary (git diff --name-only --diff-filter=U)
11. Push your feature branch to GitHub and create a Pull Request
12. Merge Pull request and provide a clear merge commit message
13. Check GitHub if the remote source branch contains your changes