

universität freiburg

Introduction into GIT

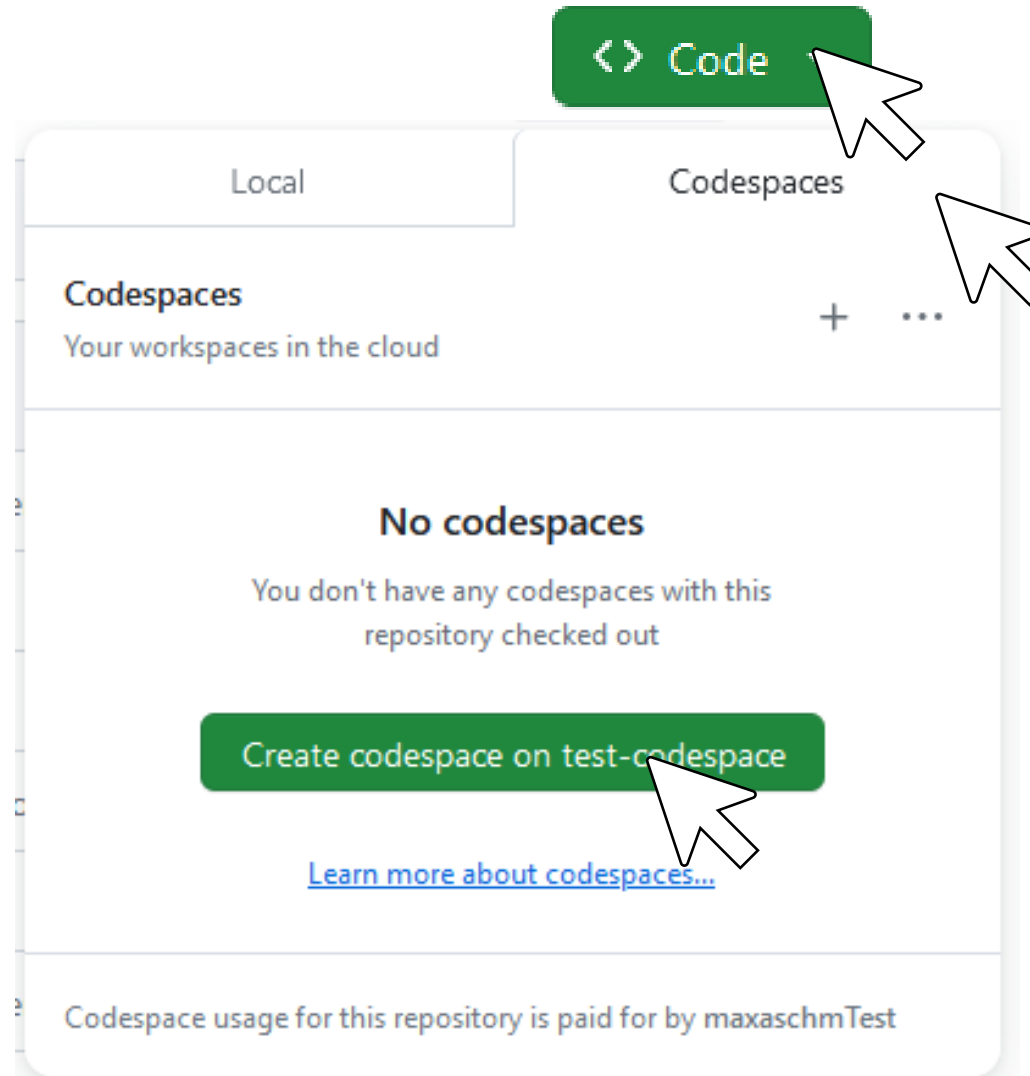
Environmental Monitoring, Data Analysis and Visualization

Professur für Hydrologie
M. Sc. Max Schmit
Freiburg, 10th of January 2024



Prepare Group Work

Go to: https://github.com/data-hydrnv/git_example



About Git



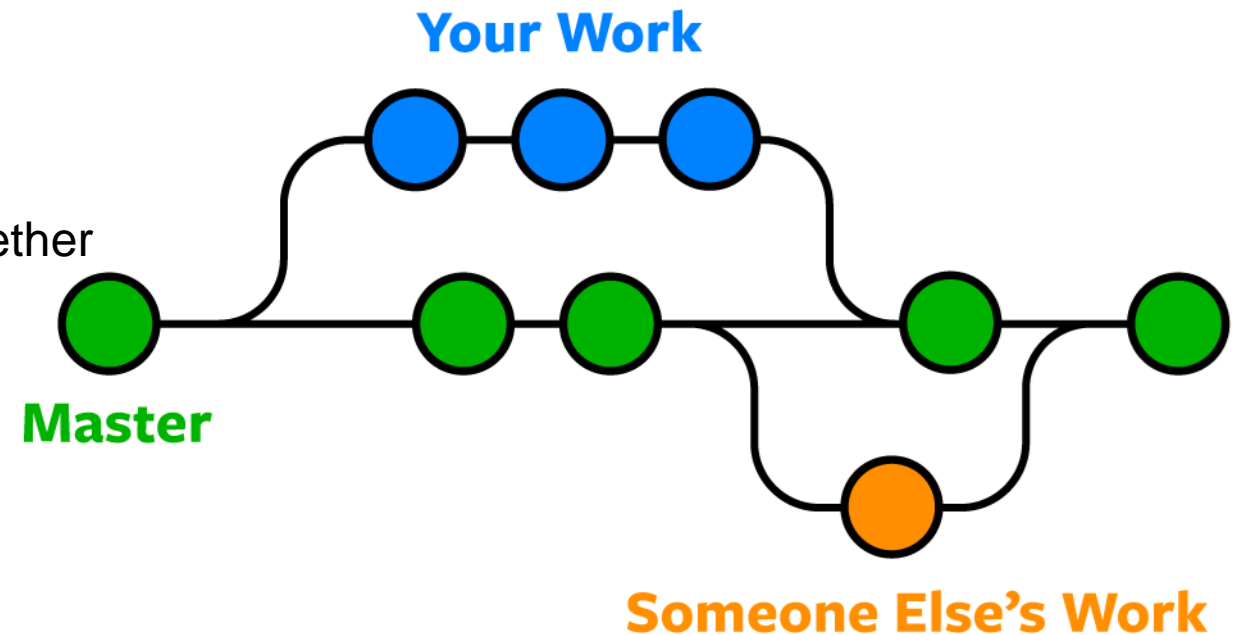
- Created by Linus Torvalds (creator of Linux), in 2005
 - Came out of Linux development community
 - Designed to do version control on Linux kernel
- Naming
 - 3 letters not used in UNIX, similar to get
 - In the US "git" means stupid, dumbass
 - Linus: " I'm an egotistical bastard, and I name all my projects after myself. First 'Linux', now 'Git'."
 - "**G**lobal information tracker,,: when it works
 - "**G**oddamn idiotic truckload of sh*t": when it breaks



Version control?



- Every change is saved in a **commit**
 - Every commit has a **commit message** (explanation)
 - Possibility to restore complete code for every commit
- several **branches**
- **Merge** multiple **branches**
 - Detects the changes and tries to bring them together
 - Throws **merge conflict** when unable
→ resolve manually



Install GIT



- Download from <https://git-scm.com/download>
- Install
- First setup
 - Open terminal
 - Setup user.name and e-mail:

```
1 git config --global user.name "[name]"  
2 git config --global user.email "[email address]"
```

Initiate repository



- Initiate current folder as empty GIT-repository:

```
1 cd C:/directory/where/my/code/lives
2 git init
```

→ Will add a “.git” folder → GIT repository

- Clone online repository

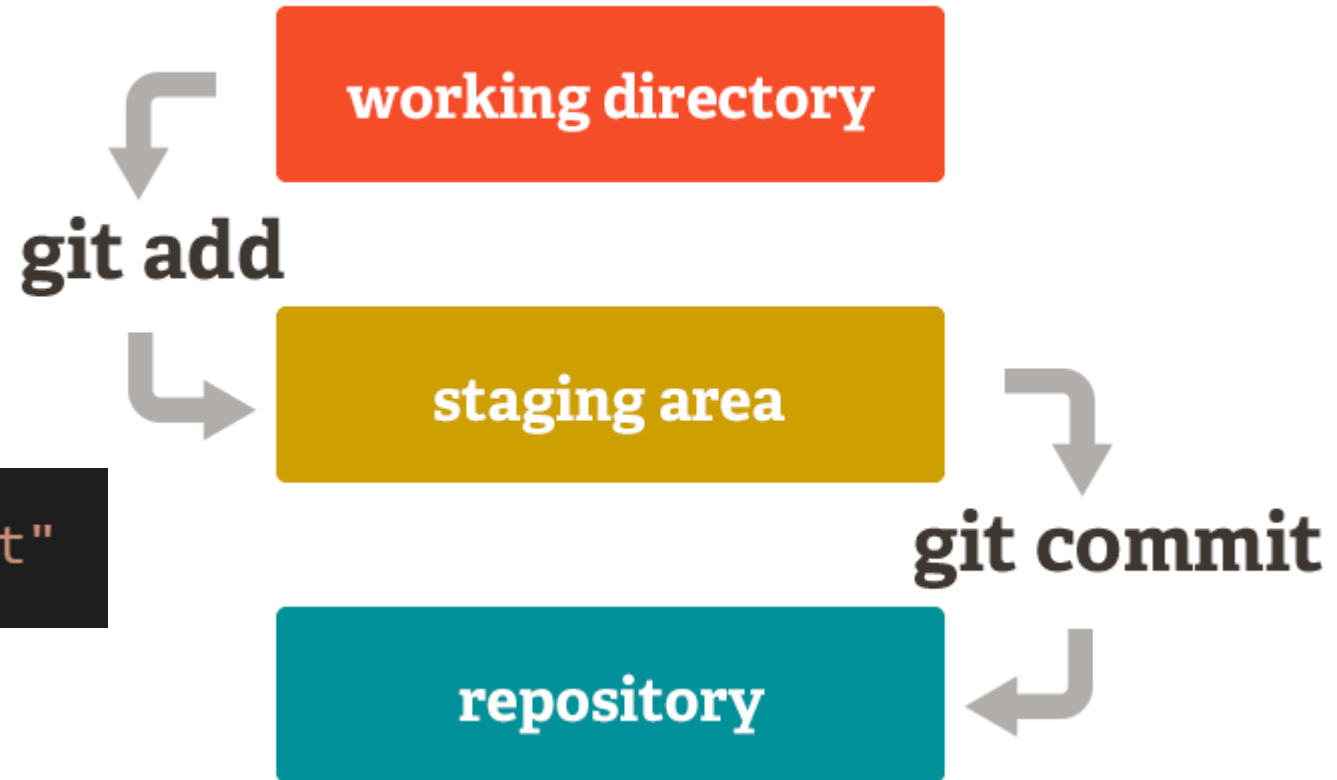
```
1 git clone https://github.com/data-hydv/v/git_example
```

→ Will copy a remote repository to local folder

GIT Workflow

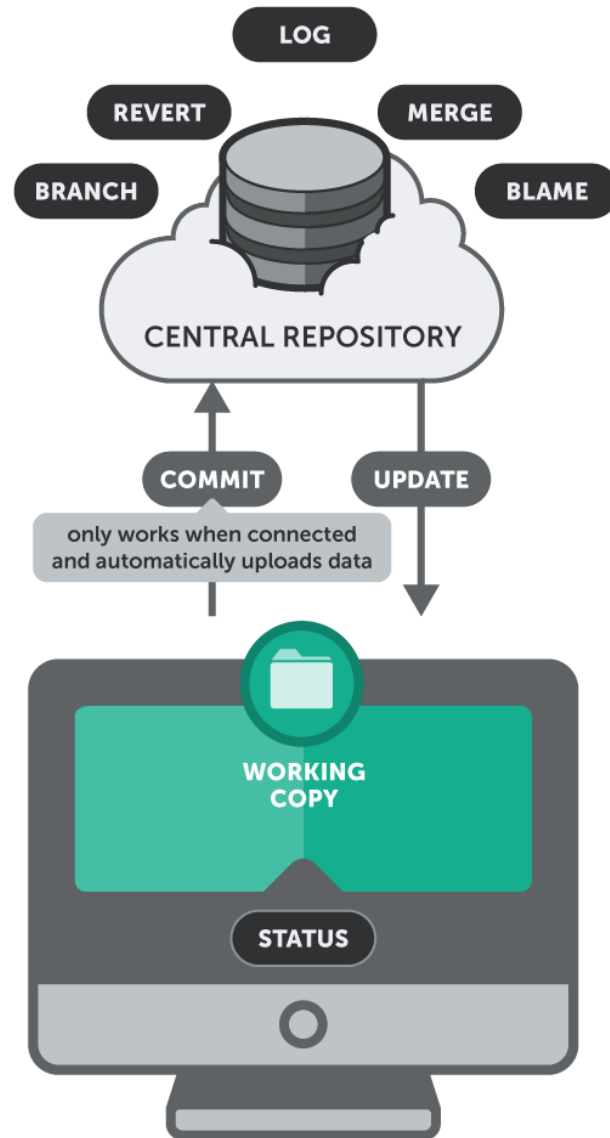
```
git add my_file.Qmd
```

```
git commit -m "first commit"
```

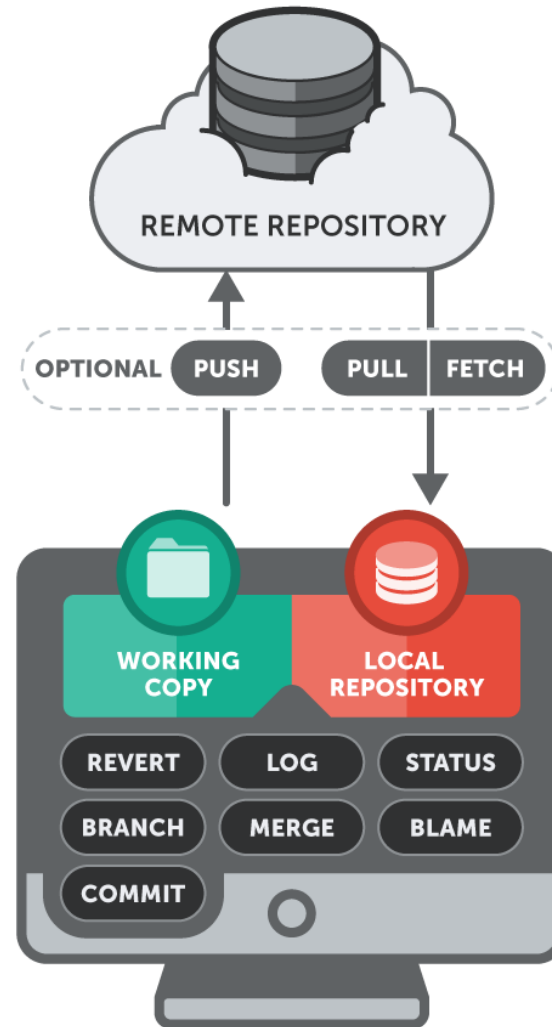


GIT: distributed

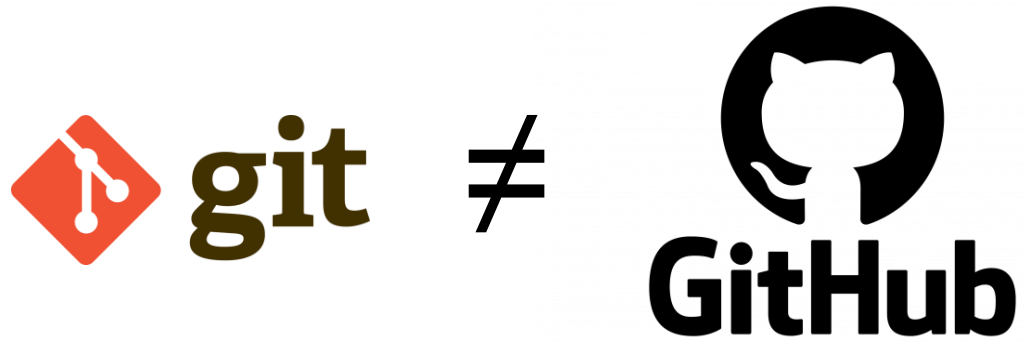
SUBVERSION



GIT



GitHub



- GitHub remote repository for git (“cloud for GIT”)
 - Stores a copy of GIT-repository “origin/...”
 - Exchange GIT-repository
- “social programming”
 - ability to chat, discuss
 - Track and discuss about issues
- Workflows
- Releases
- Graphical user interface (GUI)
- Many more additional features

Similar service:



GIT commands



command	description
<code>git clone <i>url</i> [<i>dir</i>]</code>	copy a Git repository so you can add to it
<code>git add <i>file</i></code>	adds file contents to the staging area
<code>git commit</code>	records a snapshot of the staging area
<code>git status</code>	view the status of your files in the working directory and staging area
<code>git diff</code>	shows diff of what is staged and what is modified but unstaged
<code>git help [<i>command</i>]</code>	get help info about a particular command
<code>git pull</code>	fetch from a remote repo and try to merge into the current branch
<code>git push</code>	push your new branches and data to a remote repository
others: <code>init</code> , <code>reset</code> , <code>branch</code> , <code>checkout</code> , <code>merge</code> , <code>log</code> , <code>tag</code>	

nice Cheatsheet: <https://ndpsoftware.com/git-cheatsheet.html>

GIT CHEATSHEET

escape a git mess, step-by-step
discover character entities at &what

de en es fr it pt 한국어 vn 简中 繁中

AN INTERACTION FROM NDP SOFTWARE

STASH

WORKSPACE

INDEX

LOCAL
REPOSITORY

UPSTREAM
REPOSITORY

status

diff

diff *<commit or branch>*

add *<file... or dir...>*

add -u

rm *<file(s)...>*

mv *<file(s)...>*

commit -a [-m '*msg*']

checkout *<file(s)... or dir>*

reset --hard

reset --hard *<remote>/<branch>*

switch *<branch>*

checkout -b *<name of new branch>*

merge *<commit or branch>*

rebase *<upstream branch>*

cherry-pick *<commit>*

revert *<commit>*

clone *<repo>*

pull *<remote> <refspec>*

clean

stash push [*msg*]

stash pop

stash apply
[*stash*]



Download
Manual
Now



View Manual
(PDF) Instantly



© Andrew Peterson 2009-2022 All Rights Reserved.

GIT GUI

- There are many GUI tools to work with GIT: <https://git-scm.com/downloads/guis/>



Visual Studio Code



GitHub Desktop



Partly free Software:



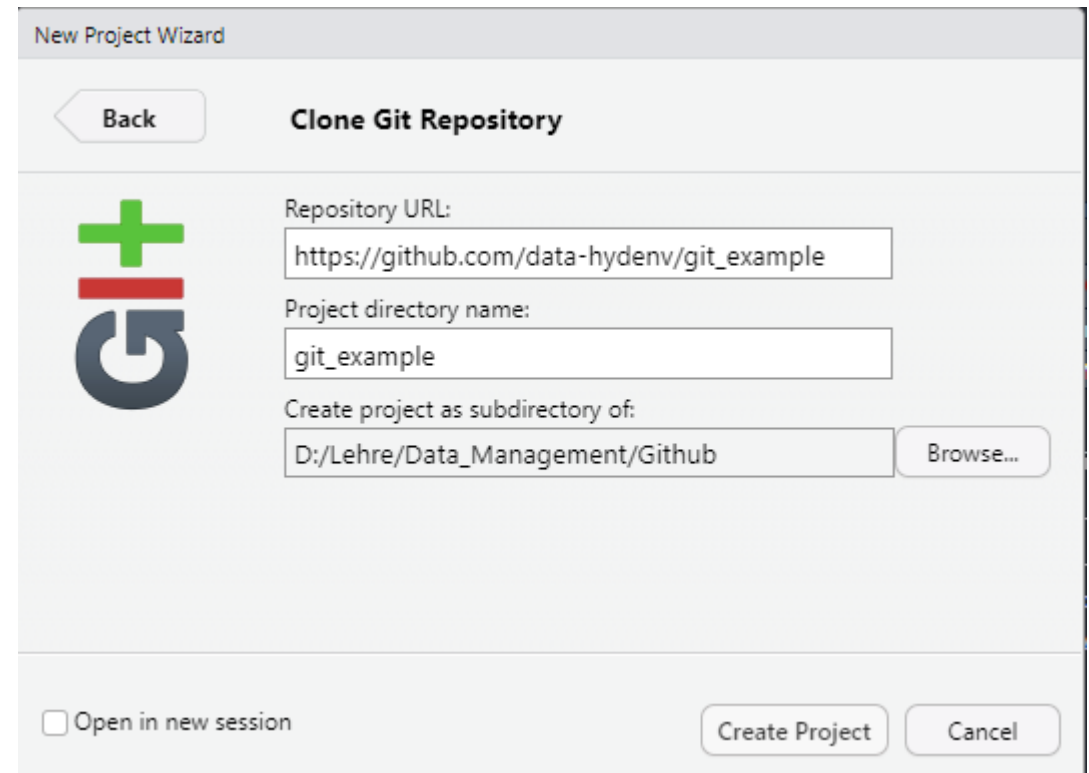
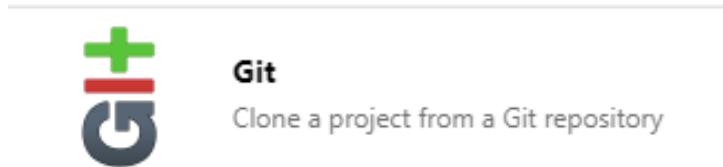
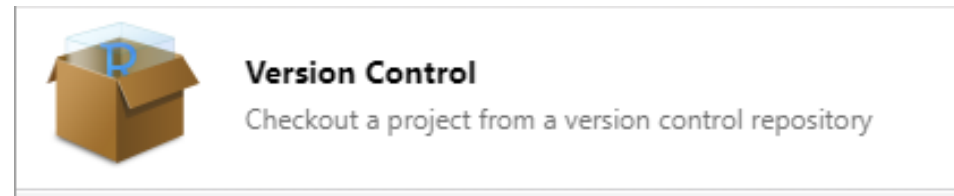
GitKraken



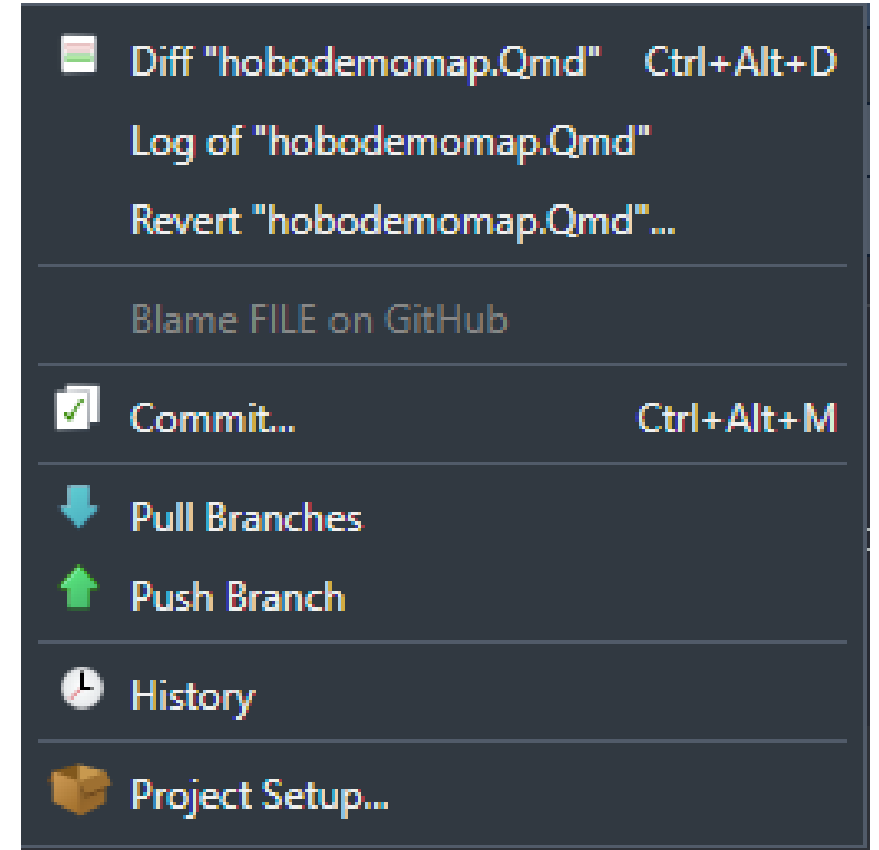
GIT in RStudio



- Enable GIT: *Tools* → *Global Options* → *Git/SVN*
- Open a repository:
 - Open a file inside a local Git-Repository
 - Clone an online repository:
File → *New Project* → *Version Control* → *GIT* →



- Enable GIT: *Tools* → *Global Options* → *Git/SVN*
- Open a repository:
 - Open a file inside a local Git-Repository
 - Clone an online repository:
File → *New Project* → *Version Control* → *GIT* → *create Project*
- Adds GIT Button
 - Diff ... and Commit...
 - see differences between local file and the repository
 - Opens Window to commit the changes
 - Pull/Push → sync with remote repository



GIT in RStudio



Changes History test-codespace | Stage | Revert | Ignore | Pull | Push

Staged	Status	Path
<input checked="" type="checkbox"/>	M	hobo_demo_map.Qmd

Commit message

Amend previous commit

Commit

Show Staged Unstaged Context 5 lines Ignore Whitespace Unstage All

@@ -14,11 +14,11 @@ execute: Unstage chunk

Old Line	New Line	Content
14	14	
15	15	## Access HOB0 meta data
16	16	
17	17	This R Markdown document loads the HOB0 meta data from the bwSync&Share and stores it as `meta_data`.
18	18	
19		This tibble can then be used to plot an interactive map with the `leaflet` package.
	19	This tibble can then be used to plot an interactive map with the `leaflet` package..
20	20	
21	21	```{r}
22	22	# label: libs

GIT in RStudio



The screenshot shows the Git History panel in RStudio. At the top, there are tabs for 'Changes' and 'History', with 'History' selected. Below the tabs, there are dropdown menus for 'master' and '(all commits)', a search bar, and a 'Pull' button. The main area displays a list of commits with columns for Subject, Author, Date (UTC), and SHA. The commit 'add a Readme file' is selected and highlighted in blue. Below the list, there are navigation buttons and a summary of the selected commit.

Subject	Author	Date (UTC)	SHA
origin/master origin/codespace-supreme-tribble-7pp56gw6	maxschmi <maxschm@y>	2024-01-05	274b8226
ignore HTML files	maxschmi <maxschm@y>	2024-01-05	225f735b
add a Readme file	maxschmi <maxschm@y>	2024-01-05	0181e25b
add VSCode setup	maxschmi <maxschm@y>	2024-01-05	8961ae20
initialize RStudio project	maxschmi <maxschm@y>	2024-01-05	05c799a3

Commits 1-5 of 5

SHA 0181e25b15f48cca4b4ce5c8d91fb1610bb0549c
Author maxschmi <maxschm@yahoo.de>
Date (UTC) 2024-01-05 11:19
Subject **add a Readme file**
Parent 8961ae20385a25fcf163d592b1655e3af4b2d3c7

MD README.md

Group Work

Group 1

Change the code chunks background color.

Group 2

Add labels to the code chunks.

Group 3

Add a figure caption to the plot.

Group 4

Write the meta file to an xlsx file using *write.xlsx* from *openxlsx*. Prevent the file from being committed. (.gitignore)

Group 5

Remove entries from meta with unrealistic values.

Group 7

Use variables to store the HTML colors.

Group 6

Add the exposition and altitude to the plots popup



Questions?

Max Schmit

Professur für Hydrologie

Telefon +49 (0)761 – 203 3519

max.schmit@hydrology.uni-freiburg.de

