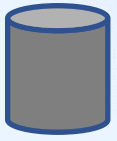
1. Organizing your project on GitHub

1. Create a project **repository** on GitHub (https://www.github.com). Ideally, use a repository template to start.



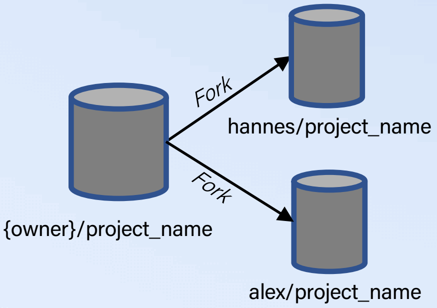
{owner}/project_name

2. Choose to work privately (e.g., for confidential projects), or publicly (e.g., for open-source projects other can contribute to)

We use GitHub "repositories" to collaborate on empirical research projects.

- **Code**: Find the source code written for your project (e.g., data preparation, analysis)
- **Issues**: Keep a to-do list and assign items to team members
- **Projects**: Prioritize issues, define deadlines and organize your team meetings.

2a) Working on an open-source project
Team members fork (=copy) the repository to their *own* GitHub account. The fork can be synchronized with its origin.



2b) Working on a (privately shared) project
The repository owner adds team members as collaborators (settings → manage access → add people)

3a) Team members clone (= download) forks to their computers.
Go to a directory where you want the project to reside → open terminal → run commands below (use *your* own username to refer to the fork).

```
git clone [your url]

git clone github.com/
alex/{project_name}
```

3b) Team members check out the owner's repository to their computers.
Go to a directory where you want the project to reside → open terminal --> run commands below (use the original repository's username).

```
git clone [owner url]

git clone github.com/
owner/{project_name}
```

4. Follow the Git (see below)

5) Review Pull Request and Close Issue
Team members make **pull requests** to ask for code changes to be integrated in the main project.

To review and accept these requests, go to the Pull Requests tab, and review submitted requests. Finally, merge them into your original project, and close the corresponding issue.

2. The Git Workflow

1. Find an issue to work on (i.e., select one from the issues or project page)

- If forked (2a above) - check if fork is up-to-date using sync fork on GitHub.com, and git pull to retrieve the latest changes locally.

Whenever working on a project, we follow the Git workflow.

2. Create a branch
Create a branch with a specific name for the issue you will be working on. This helps others to review your code and avoids conflicts when pulling and pushing changes. Do this always – even if you work in a privately shared project.

- `git branch {name}` → *Creates new branch* *(*Alternatively, click on "Create a branch for this issue" in the respective issue on GitHub.com)
- `git checkout {name}` → *Switches to this branch* (Alternatively, follow the instructions provided on GitHub.com after creating the branch)
- Use `git pull {name}` to get latest changes from this branch

3. Start working on the issue
Make the necessary changes in the source code of the files in the locally cloned repository. Working on an issue usually entails multiple commits, which can be done through Git Bash or editors such as Rstudio or Visual Studio Code.

1. Git Bash	2. RStudio
• `git status` • `git add {file name that was changed}` • `git commit –m [message of what you did, in quotation marks]` • `git push (if first time: git push –u origin {branch name}`	• In the top right, click on "Git" • Select files with changes you want to commit and click on "commit" • Enter a brief commit message and click "commit" • Finally, click on "push" to push your changes

The `git push` command uploads the content of your local repository to a remote repository on GitHub. Your team members will likely not see these changes yet (e.g., because they don't monitor the branch, or because the changes are in a fork).

4. Update the Git "Issue" by letting others know what you changed, inform about what still needs to be done, or request feedback. Tips on how to write good issues can be found at Tilburg Science Hub: tilburgsciencehub.com/write/issues.

5) Make a pull request
Happy with the changes, and ready to ask team members to integrate these changes with the main branch of your (original) repository? Make a pull request. See tilburgsciencehub.com/contribute/pullrequests to find out how.

Your team members now can review and integrate your changes.

>_ Tilburg ScienceHub