

# 1. Organizing your project on GitHub

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



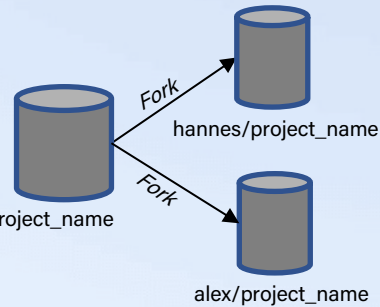
{owner}/project\_name

**2. Next, decide on whether to work privately** (e.g., for confidential projects), or publicly (e.g., for open-source projects)

**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.



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

**3a) Team members check out forks to their computers.**  
Go to a directory → right click → "Git Bash here". Finally, type `cd {project_name}`. Now, clone the repository!

```
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 → right click → "Git Bash here". Finally, type `cd {project_name}`. Now, clone the repository!

```
git clone [owner url]
git clone github.com/owner/{project_name}
```

**5a) Review Pull Request and Close Issue**  
Changes at this point are only visible in the local repository. Go to the Pull Requests tab, and review submitted requests. Finally, merge them into your main project, and close the corresponding issue.

**4. Follow the Git Workflow** (see below)

**5b) Close Issue**  
Since you work in the owner's repository, all changes are already visible online. 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 – 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, specific for the issue you will be working on. This helps others to review your code, and avoids conflicts when pulling and pushing changes.

- `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)
- In the future, 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 repository. Working on an issue usually entails multiple commits, which can be done through Git Bash or editors such as RStudio.

1. Git Bash	2. RStudio
<ul style="list-style-type: none"> <li>• <code>git status</code></li> <li>• <code>git add {file name that was changed}</code></li> <li>• <code>git commit -m [message of what you did, in quotation marks]</code></li> <li>• <code>git push</code> (if first time: <code>git push -u origin {branch name}</code>)</li> </ul>	<ul style="list-style-type: none"> <li>• In the top right, click on "Git"</li> <li>• Select files with changes you want to commit and click on "commit"</li> <li>• Enter a brief commit message and click "commit"</li> <li>• Finally, click on "push" to push your changes</li> </ul>

The `git push` command uploads the contents of your local repository to a remote repository on Github! If you are working on a fork, your team members won't be able to see these changes yet.

**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/](http://tilburgsciencehub.com/write/issues/)

If you worked on a forked repository, follow step 5A above to contribute your changes to the main repository.

**5) Make a pull request**  
Happy with your changes and ready to ask team members to integrate them with your main project? Make a pull request!

See [tilburgsciencehub.com/contribute/pullrequests](http://tilburgsciencehub.com/contribute/pullrequests) to find out how to perform pull requests!