



# Exercise: GitHub Projects, Unit Testing

19/20 November 2019 | Guido Trens (JSC, SimLab Neuroscience)

**In this exercise, we will use the GitHub functionality to organize our software development work into projects and a Kanban-like workflow. We will extend our small Python example source code with new functionality, using the test-driven software development approach.**

**Work with a partner and form a two-developers team.**

1. Update your master to be in sync with the GitHub workshop repository:  
<https://github.com/gtrensch/SoftwareDevInScience2019>
2. Create a new project on GitHub, e.g., for adding new math-functions and unit tests. Chose the basic-Kanban template. Explore the possibilities of setting up a project!
3. Create a ToDo list with at least two activity cards for the implementation of a unit test and a new function. The unit test should describe test cases corresponding to the function you are planning to implement.
4. (optional) Enable “Issues” under the repository settings to integrate lightweight task tracking into your repository. This allows you to track issues within your project.

# Exercise IV: Test-driven Development

---

1. Developer 1: Create a new feature-branch for your project that you have set up in exercise III.
2. Developer 1: Implement the unit test cases that you have defined in exercise III.
3. Add and commit your changes.
4. Developer 2: Pull the feature-branch from Developer 1.  
Hint: use the “git remote add <dev1> <repo>” command.
5. Developer 2: Create a working-branch.
6. Developer 2: Implement the functionality that your test cases define. Use the unit tests to verify the correctness of your implementation.
7. Add and commit your changes.
8. Developer 2: Issue a pull request against the feature-branch.
9. Developer 1: Code-review the pull request and merge it.
10. Developer 1: Issue a pull request against the workshop repository.