Workflow

1. Clone the remote repository onto your local machine.
2. Create a remote branch for the work you are about to do. Be mindful of which branch you create the new branch from.
   (a) Pull the changes to the remote to your local repository.
3. Checkout the new branch.
4. Work on the files.
5. Add and commit the files you changed.
   (a) Pull any changes to the remote to your local repository.
   (b) Resolve any merge conflicts.
6. Push the changes to the local repository to the remote.
7. Create a pull request.
   (a) (Optional) Assign those who have worked on the changes to the pull request, and add those who haven't worked on the changes as reviewers.
8. After the changes are approved by the rest of the team, merge the pull request.

Commands

Command	**Explanation**
`git clone <url>`	Creates a local copy of the remote repository at `<url>`.
`git status`	Provides information about the status of the current repository.
`git add <filename>`	Adds `<filename>` to the commit.
`git commit`	Commits the changes to the local repository. You can use the option `-m "commit msg"` to specify the commit message via the command line.
`git push`	Updates the remote repository with your changes.
`git pull`	Updates your repository with changes from the remote repository.
`git branch`	List all local branches. You can use the option `-d <branchname>` to delete a local branch.
`git checkout <branchname>`	Switch to work on the branch `<branchname>`.

Glossary

Word	Explanation
Branch	See `https://www.atlassian.com/git/tutorials/using-branches`.
Local repository	The copy of the files that is stored on your (local) machine.
Merge conflict	When two branches which are to be merged contains work on the same code, a merge c
Remote repository	The cope of the files that is stored on GitHub's (remote) server.

Setting up a branch

On GitHub, create a branch via the drop-down menu. Name it appropriately - a branch is usually named after the work you are planning on doing in the branch.

A created branch is an extension of the branch you have active when you create the new branch. So, if you want to work on a copy of the master branch, ensure that you are on the page corrensponding to the master branch when creating the new branch.

After the branch is created, you need to pull the changes to your local repository, and checkout the new branch to start working in it.

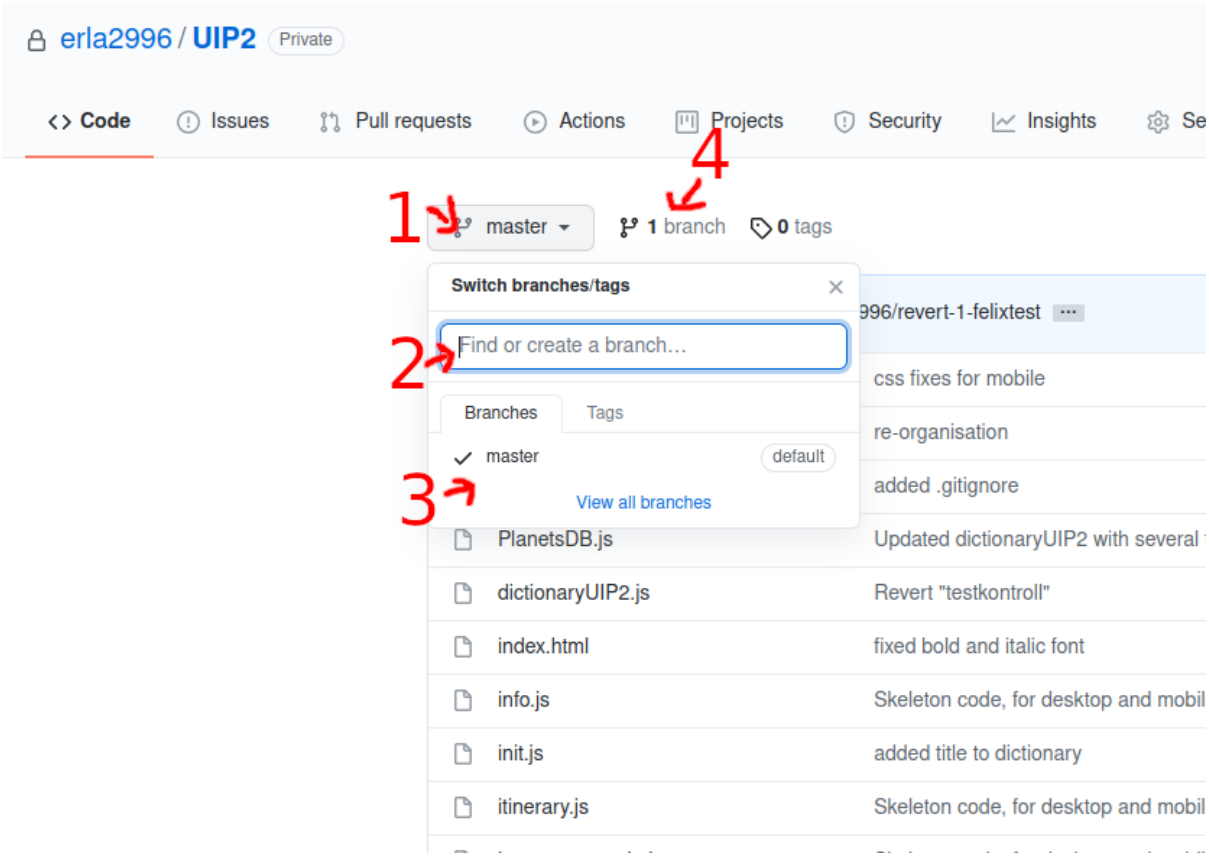

Figure 1: **1**: Button to click to open the shown dropdown menu of branches. **2**: Type the name of the new branch here to create a branch. A button "Create branch: " should appear below the text-field - click that button to confirm. **3**: List of all branches. If there are many branches in the project, the list may be shortened. **4**: Click here to open a list of all branches, where you can delete and rename branches.

Resolving a merge conflict

When there is a conflict in a file, git will insert code indicating where in the file the conflict happened. The code looks as follows:

```
<<<<<<< HEAD
content in current branch
=======
content in the branch to merge with
>>>>>>> new_branch_to_merge_later
```

In order to resolve the merge conflict, remove the code inserted by git, and fix the code in the file where the conflict occurs. Then, once every conflict in the file is fixed, add the file to the commit, and commit the merge conflict resolution (using `git add` and `git commit`, as usual).

For more information on merge conflicts, see `https://www.atlassian.com/git/tutorials/using-branches/merge-conflicts`.