

Git: Branching

ToDo

- Explain what branches are (use slides)
- Usecases:
 - to work on features
 - to fix past mistakes easily

View branches

1. `git branch`

Create branches

1. `git branch branch-name`
2. `git branch`

switch between branches

1. `git checkout branch-name`

Make some changes to branch-name

1. edit file
2. `git commit -am`
3. `git status`
4. `git checkout other-branch` (keep text file open)

Make changes comming from a point in the past

1. `git checkout master`
2. `git branch fix-this`
3. `git branch`
4. Make changes in fix-this
5. `git commit -am`

Merge into master

Note: You have to go to master and merge fix-this into it

1. `git checkout master`
2. `git merge fix-this`
3. Solve conflicts

See merged vs. unmerged branches

1. `git branch --merged`
2. `git branch --no-merged`