

03. Ignoring Things

Basics

1. Make many annoying files
2. `git status`
3. create `.gitignore`
4. Add entries for annoying files
5. `git status`
6. `git add .gitignore`
7. `git commit -m 'Added .gitignore'`
8. `git status --ignored`

Add ignored files

1. `git add ignored-file`
2. `git add -f ignored-file`

Exceptions to `.gitignore`

1. Add exception for a single file via `!file-name`
2. `git status`
3. switch the order

What have we learned

1. **.gitignore**: Helps to keep Git from tracking files or folders
2. **git add -f ignored-file**: Add otherwise ignored files anyway

Exercise

You wrote a script that creates many intermediate log-files of the form `log_01`, `log_02`, `log_03`, etc. You want to keep them but you do not want to track them through git.

1. Write one `.gitignore` entry that excludes files of the form `log_01`, `log_02`, etc.
2. Test your “ignore pattern” by creating some dummy files of the form `log_01`, etc.
3. You find that the file `log_01` is very important after all, add it to the tracked files without changing the `.gitignore` again.
4. Discuss with your neighbor what other types of files could reside in your directory that you do not want to track and thus would exclude via `.gitignore`.