

## ***GIT Basics homework***

### ***Part 1***

1. Install GIT on your workstation
2. Setup git: change your global configs (add name and email, setup core text editor).
3. Create account on GitLab
4. Generate ssh-key and integrate it with GitLab.
5. Create new project on GitLab.
6. Clone project to your workstation.
7. Create .gitignore file. Include .gitignore in its own list. Try to use different patterns in .gitignore file (see in lecture) to make invisible for git some local files that should not be committed. Check that files mentioned in .gitignore file shouldn't be visible via executing git status command.

### ***Part 2***

1. Open git console in root directory of your project and make next steps.
2. Do all your experiments in folder "task1 – git practice I".
3. Create empty readme.txt file.
4. Make init commit.
5. Create develop branch and checkout on it.
6. Create index.html empty file. Commit.
7. Create branch with name "images". Checkout on it. Add images folder with some images inside it. Commit.
8. Change your index.html. Add images source inside it. Commit.
9. Go back to develop branch.
10. Create branch with name "styles". Checkout on it. Add styles folder with styles source inside it. Commit.

11. Change your index.html. Commit.
12. Go to develop branch.
13. Merge two new branches into develop using git merge command. Resolve conflict if it appear. Do it in next sequence:
  - merge “images” into “develop”
  - merge “styles” into “develop”
14. Do not delete any branches!
15. Merge develop into master.
16. Checkout to develop branch and repeat 7-15 items once more (use names “images2”, “styles2” for new branches). Now use **git rebase** command instead of git merge in step 13. Do rebase in next sequence:
  - rebase “images2” onto “develop”
  - rebase (fast-forward) “develop” onto “images2”
  - rebase “styles2” onto “develop”
  - rebase (fast-forward) “develop” onto “styles2”
17. Merge develop into master.

NOTE: As a result master/develop branches should contain all commits that were done in process

### **Part 3**

1. Try to inspect your repository with git log command. Use different options with this command (git log --help).
2. Push all your changes **with all** your branches to origin (git push origin --all).
3. **Execute command “git reflog” and save it content somewhere (not in repository) with filename “GIT\_Basics\_homework\_1.txt”.**

### **Additional task (\*)**

Write common sides and differences between *merge* and *rebase* commands.

Write pros and cons for *merge* and *rebase*. Describe situations for preferably usage of these commands.

Use additional materials for this task.