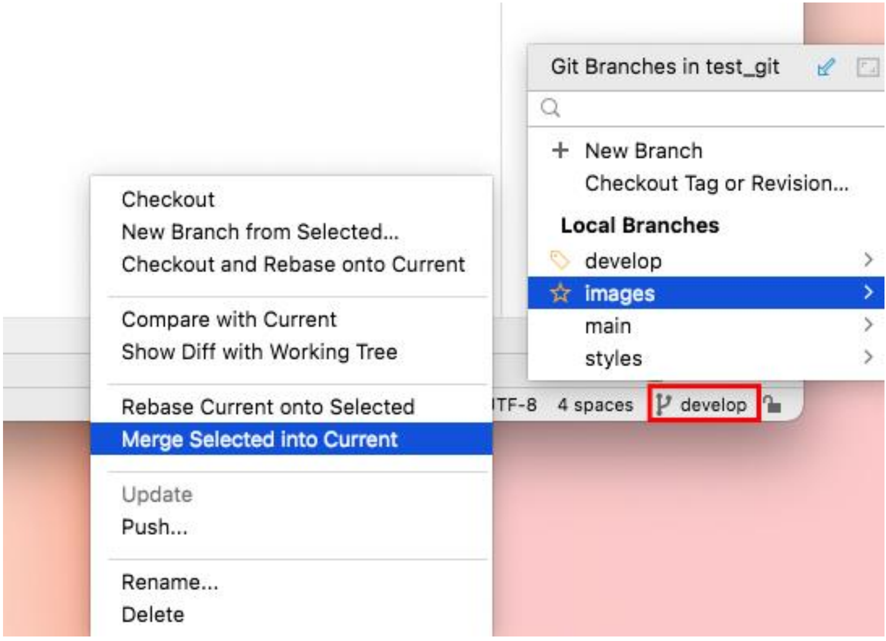
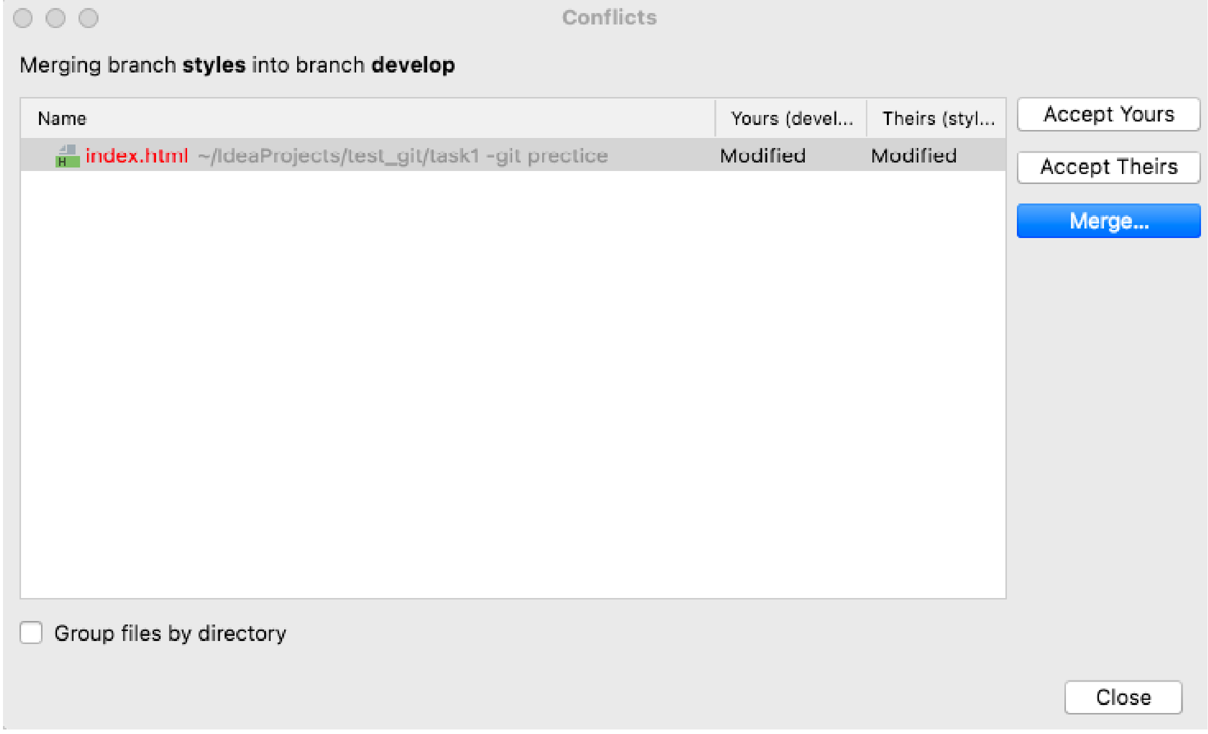
GIT Basics homework

1. Install GIT on your workstation
2. Setup git: change your global configs (add name and email, setup core text editor).
3. Generate ssh-key and integrate it with GitHub.
4. Create new "git_practice" project on GitHub.
5. Clone project to you workstation.
6. Provide access for mentor* to your repository on GitHub (as Developer).
7. Create .gitignore file. Include .gitignore in its own list. Try to use different patterns in .gitignore file (see here for more information) to make invisible for git some local files that should not be commited. To read more about .gitignore file follow the link.

Part 2
1. Do all your experiments in folder “git practice”.
2. Create empty readme.txt file.
3. Make init commit.
5. Create develop branch and checkout on it.
6. Create index.html empty file (for more information about html link). Commit.
7. Create branch with name“images”. Checkouton it. Add images folder with some images inside it. Commit.
8. Change your index.html. Add images source inside it (how to add image in html file link). Commit.
9. Go back to develop branch.
10. Create branch with name“styles”.Checkout on it. Add styles folder with styles source inside it (change the font-size for the images in your html file, link, look at external CSS part). Commit.
11. Change your index.html (add link to the styles file). 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” (you should be on develop branch and do next


Git Branches in test_git
+ New Branch
Checkout Tag or Revision...
Local Branches
develop
images
main
styles
Checkout
New Branch from Selected...
Checkout and Rebase onto Current
Compare with Current
Show Diff with Working Tree
Rebase Current onto Selected
Merge Selected into Current
Update
Push...
Rename...
Delete
TF-8
4 spaces
develop
)

- merge “styles” into “develop” (if you have a conflict, the new window will be opened

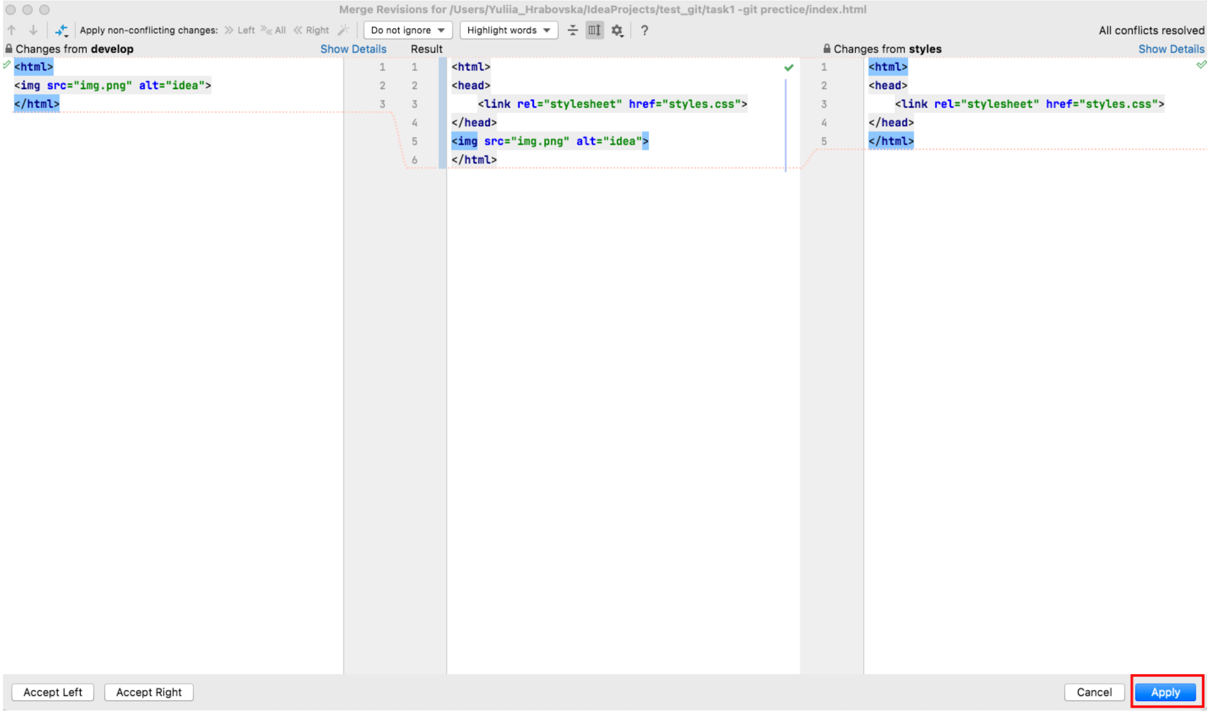


Merging branch **styles** into branch **develop**

Name	Yours (devel...	Theirs (styl...
index.html ~/IdeaProjects/test_git/task1 -git prectice	Modified	Modified

☐ Group files by directory

click merge button, and manually fix the conflicts and click on Apply button


Merge Revisions for /Users/Yuliia_Hrabovska/IdeaProjects/test_git/task1 -git prectice/index.html
Apply non-conflicting changes:
Left
All
Right
Do not ignore
Highlight words
?
Show Details
Show Details
Accept Left
Accept Right
Cancel
Apply

All conflicts resolved

Changes from **develop**

```
<html>
<img src="img.png" alt="idea">
</html>
```

Result

```
<html>
<head>
    <link rel="stylesheet" href="styles.css">
</head>
<img src="img.png" alt="idea">
</html>
```

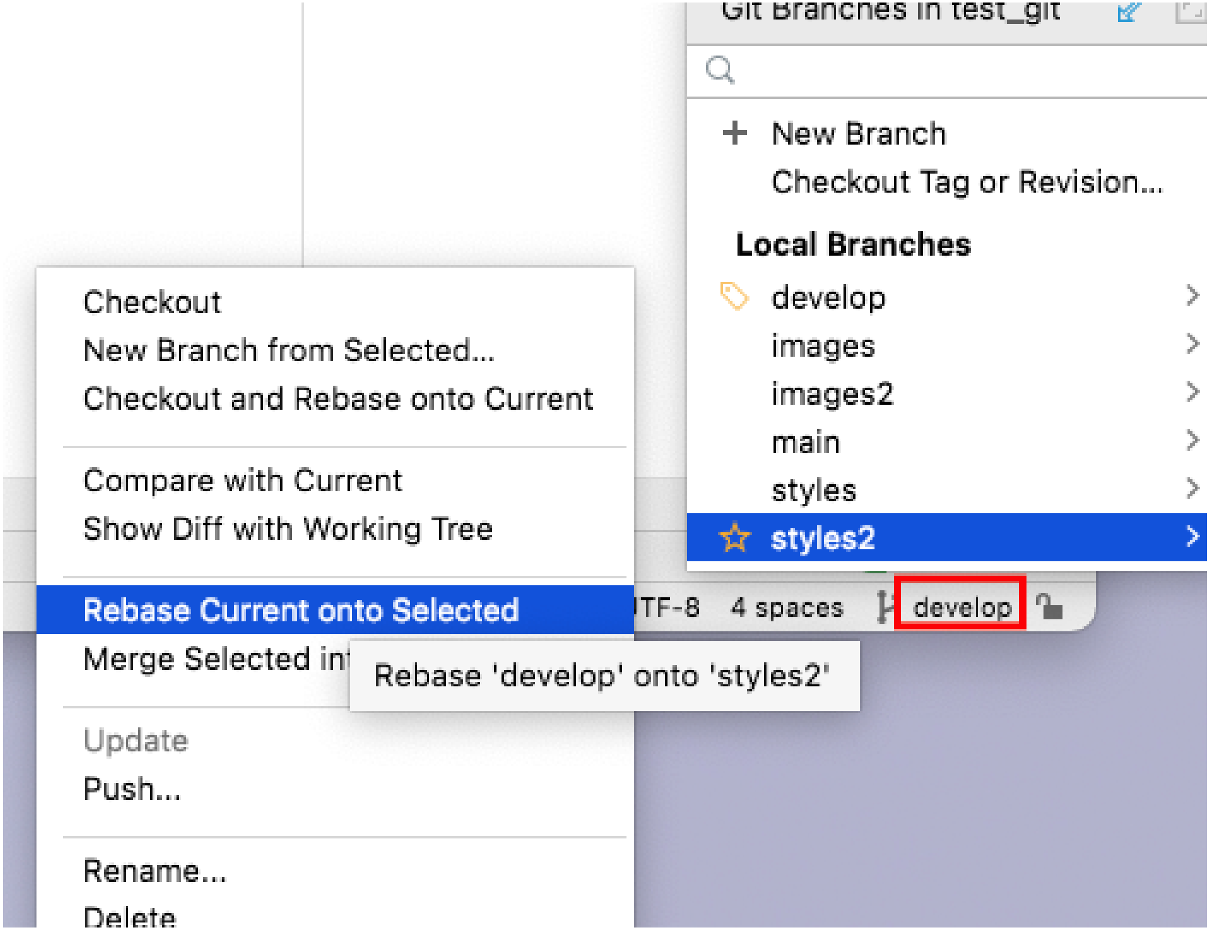
Changes from **styles**

```
<html>
<head>
    <link rel="stylesheet" href="styles.css">
</head>
</html>
```

)

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”(the same as we did with merge, but for rebase



)

- rebase “develop” onto “images2”
- rebase “styles2” onto ‘’develop’
- rebase “develop” onto “styles2”

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

18. Compare results of merging using merge command and rebase.

Part 3

1. Push your changes to remote repository

*mentor** - juliaHrabovska