Basic git commands

You can perform git commands for example in:

- cmd console (widnows start -> cmd. It should oppened Command Prompt app. To go to the root directory, enter cd \ . Next enter cd + path to directory with your project, e.g. cd Users\Joanna\Desktop\my-project

- git bash console, which you can get from https://git-scm.com/downloads . After installing console go to directory with your project, right click and select: Git Bash Here (then console opens).

If you have your console with path to your project, you can enter first command (step one).

Basic git commands:

I - Local repository:

1. Creating a new local repository:

-> git init

In your project catalog it should be created hidden folder .git

2. If you have in your project catalog some files, which you don't want to add to repository (because you don't want to send it to github later), you can create .gitignore file, in which you can write paths to this files. Git will skip them when adding files to the repository.

-> touch .gitignore

3. Shows changes that you made in project, e.g. files names, which have changed the content, but have not been added to the stage yet, are displayed in red, and those that have been added to the stage, but no commit has been made, are displayed in green.

-> git status

Basic git commands

4. Adding file to stage

-> git add file-name, e.g. git add index.html

-> git add .   - dot means, that you add to stage all files from project.

5. Removing file from stage:

-> git rm --cached file-name, np: git rm --cached index.html

6. Adding commit:

-> git commit -m ”commit message”

7. Viewing commit history (if you want to go back to any previous commit, you need its number. By displaying the history you will see all commits and its numbers):

-> git log   <- shows more details

-> git log --oneline   <- less details, but more readable

8. Undoing commit

-> git checkout commit_number, e.g. git checkout af6b84c

Shows code from commit in editor. It not removes history and any commits.
If you want to go back to your code enter: git checkout master (if you work on master branch).

-> git revert commit_number, e.g. git revert af6b84c

Deletes changes from the selected commit (e.g. af6b84c) by adding a new commit (e.g. bg6b92d), in which code entered after commit number af6b84c was deleted. It not removes history.

-> git reset commit_number, e.g. git reset af6b84c   - deletes commits from repository and removes history (it is not recomended). If you choose to delete third commit from the end, it will remove the last 3 commits from editor, use command:
-> git reset commit_number --hard

Basic git commands

II - Remote repository:

As an example, we will use github (github.com)

Create an account on github. You will need to create a new remote repository and provide its name, and description (optional).

Option 1 - You already have a project in the local repository and you want to add it to the remote repository:

9. Github -> create new project -> enter its name and description -> DO NOT check "Initialize this repository with a README" option -> create repository.

10. Adding path to remote repository in local repository. Enter in console:

-> git remote add origin + path to repository on github, egzample:
https://github.com/NazwaKontaGitHub/NazwaProjektu.git
Together:
-> git remote add origin https://github.com/AccountNameOnGithub/ProjectName.git

Path to the repository you can find on github.

11. Checking path to which are sent files to remote repository:

-> git remote -v

Option 2 - You do NOT have a project in any repository yet:

9. Github -> create new project -> enter its name and description -> CHECK "Initialize this repository with a README" option -> create repository.

Basic git commands

10. Clone a github repository on your computer (green button "clone or download" on github). Set path to your project in console and enter:

-> git clone + path to repository, e.g.
git clone https://github.com/AccountNameOnGithub/ProjectName.git

Folder with project from github should appear in location that you set in console.

11. Getting changes from github (from remote to local repository). Switch on branch that you want to pull changes. If you use main branch, which is master branch, you do not have to switch.
-> git pull origin branch_name, from which you pull files, e.g. git pull origin master

To options 1 i 2:

12. Add files from local project to remote repository (from computer to github):

-> git push -u origin branch_name, on which you send files. If you have only one branch, the command is:
-> git push -u origin master

You only need to set the -u flag when you send files to remote repository for the first time, or first time after merge. This helps define path to pull files in later time. Each next files upload to github is command:
-> git push origin master

13. Creating a separate branch - locally:

-> git branch branch_name, e.g. git branch features/searchBar

14. Show all branches:

-> git branch -a

15. Switch to branch:

-> git checkout branch_name, e.g. git checkout features/searchBar or: git checkout master

Basic git commands

16. Creating a branch and switching to it (instead steps 13 i 15):

-> git checkout -b branch_name, np: git checkout -b features/searchBar

17. Commit made on branch:
-> git checkout features/searchBar
-> git add .
-> git commit -m "commit message"

Adding a commit from the local repository to the remote one on branch other than master:

-> git push origin features/searchBar

18. Deleting branch. First, switch on master branch:

-> git checkout master

Next enter:

-> git branch -d branch_name, e.g. git branch -d features/searchBar <- this removes branch only if merge with master was made.

-> git branch -D branch_name, e.g. git branch -D features/searchBar <- this removes branch even if merge with master wasn't made.

19. Merge branch. First, switch on master branch:

-> git checkout master

Next enter:

-> git merge branch_name, np: git merge features/menu