Training material of Control Study

In the below there exist useful material that you can read it before the study. In the study you have to clone a code from a GitHub repository and fix the issues in the repository. You need to follow the 3 bellow steps at the beginning of the study.

STEP1: IDE TUTORIALS

For increasing your productivity please choose one IDE that you are more fluent on it, then read only its related tutorial.

1. WebStorm: https://www.jetbrains.com/help/webstorm/using-git-integration.html?keymap=primary_default_for_macos&section=macOS
2. Visual Studio Code: https://code.visualstudio.com/docs/introvideos/basics
3. Atom-IDE: https://blog.atom.io/2017/09/12/announcing-atom-ide.html
4. Brackets: https://css-snippets.com/brackets-course/

STEP 2: GITHUB BASIC TUTORIAL

In this study, you will be using GitHub to clone, commit, and push code to a GitHub repository. Here is a quick review of how to do each of these in GitHub:

checkout a repository

create a working copy of a local repository by running the command

```
git clone /path/to/repository
```

when using a remote server, your command will be

```
git clone username@host:/path/to/repository
```

add & commit

In order (add it to the Index) using

```
git add <filename>
git add *
```

This is the first step in the basic git workflow. To commit these changes to use

```
git commit -m "Commit message"
```

Now the file is committed to the HEAD, but not in your remote repository yet.

pushing changes

Your changes are now in the HEAD of your local working copy. To send those changes to your remote repository, execute

```
git push origin master
```

Change master to whatever branch you want to push your changes to.

If you have not cloned an existing repository and want to connect your repository to a remote server, you need to add it with

```
git remote add origin <server>
```

Now you can push your changes to the selected remote server

update & merge

To update your local repository to the newest commit, execute

```
git pull
```

in your working directory to fetch and merge remote changes.

to merge another branch into your active branch (e.g. master), use

```
git merge <branch>
```

In both cases, git tries to auto-merge changes. Unfortunately, this is not always possible and results in conflicts. You are responsible to merge those conflicts manually by editing the files shown by git. After changing, you need to mark them as merged with

```
git add <filename>
```

before merging changes, you can also preview them by using

```
git diff <source_branch> <target_branch>
```

cheat-sheet

A cheat-sheet is also attached to this document that might be useful.

STEP3: UNIT TEST TUTORIAL

As you write your code, you may choose to test your work by writing unit tests. You can find information about how to get started writing unit tests using Mocha in the link below:

https://codeburst.io/how-to-test-javascript-with-mocha-the-basics-80132324752e