**Short introductions to the core concepts of Git and GitHub**

**Git** – the name of version control software that GitHub is built on.

**Repository** in GitHub (на български хранилище) - A repository is usually used to organize a single project. Repositories can contain folders and files, images, videos, spreadsheets, and data sets - anything your project needs. Often, repositories include a README file, a file with information about your project. README files are written in the plain text Markdown language (Markdown syntax: <https://www.markdownguide.org/cheat-sheet/>).

**Branching** - Branching lets you have different versions of a repository at one time. By default, your repository has one branch named *main* that is considered to be the definitive branch. You can create additional branches off of *main* in your repository. You can use branches to have different versions of a project at one time. This is helpful when you want to add new features to a project without changing the main source of code. The work done on different branches will not show up on the *main* branch until you **merge** it, which we will cover later in this guide. You can use branches to experiment and make edits before committing them to *main*.

When you create a branch off the *main* branch, you're making a copy, or snapshot, of main as it was at that point in time. If someone else made changes to the *main* branch while you were working on your branch, you could pull in those updates.

Branches accomplish similar goals in GitHub as saving a couple of different versions of the same file with different edits.

**Commits** - on GitHub, saved changes are called **commits**. Each commit has an associated commit message, which is a description explaining why a particular change was made. Commit messages capture the history of your changes so that other contributors can understand what you’ve done and why.

**Pull requests** - pull requests are the heart of collaboration on GitHub. When you open a pull request, you're proposing your changes and requesting that someone review and pull in your contribution and merge them into their branch. Pull requests show **diffs**, or differences, of the content from both branches. The changes, additions, and subtractions are shown in different colors.

**Merging** pull request – allowing the merge of the secondary branch to the main branch. Sometimes, a pull request may introduce changes to code that conflict with the existing code on *main*. If there are any conflicts, GitHub will alert you about the conflicting code and prevent merging until the conflicts are resolved. You can make a commit that resolves the conflicts or use comments in the pull request to discuss the conflicts with your team members.

After the pull request is merged and the changes are on *main*, the secondary branch can be safely deleted. If you want to make more changes to your project, you can always create a new branch and repeat this process.

**For more information check out the GitHub documentation:**

<https://docs.github.com/en>