Проектная работа

На тему «Git, GitHub»

Работу выполнили:

Фазылов Владислав

Колпаков Артём

Парфёнов Никита

Преподаватель:

Мингажева Елена Александровна

2022

**Содержание**

1. О Git …………………………………………………………………………...
2. О GitHub ………………………………………………………………………
3. Что такое репозиторий Git? .............................................................................
4. Что такое коммит Git? ......................................................................................
5. Как выполнить Git Commit ..............................................................................
6. Что такое Git Remote? ......................................................................................
7. Как создать ветку Git .......................................................................................
8. Как удалить ветку Git.......................................................................................
9. Как переименовать ветку Git ...........................................................................
10. Как работает Git SSH ......................................................................................
11. Как работает Git Diff .......................................................................................
12. Что такое Git Checkout? ..................................................................................
13. Что такое Git Pull? ...........................................................................................
14. Как объединить Git .........................................................................................
15. Как сделать Git Stash ......................................................................................
16. Как создать Git-хуки .......................................................................................
17. Как использовать Git Squash ...........................................................................
18. Что такое запрос на извлечение в Git? ...........................................................
19. Как использовать Git Cherry Pick ...................................................................
20. Что такое Git Rebase? .......................................................................................
21. Как разрешить конфликт слияния в Git ..........................................................
22. Что такое Git LFS? ............................................................................................
23. Что такое подмодуль Git? ...............................................................................

**About Git**

The Git feature that really makes it stand apart from nearly every other SCM out there is its branching model.

Git allows and encourages you to have multiple local branches that can be entirely independent of each other. The creation, merging, and deletion of those lines of development takes seconds.

This means that you can do things like:

Frictionless Context Switching. Create a branch to try out an idea, commit a few times, switch back to where you branched from, apply a patch, switch back to where you are experimenting, and merge it in.

Role-Based Codelines. Have a branch that always contains only what goes to production, another that you merge work into for testing, and several smaller ones for day to day work.

Feature Based Workflow. Create new branches for each new feature you're working on so you can seamlessly switch back and forth between them, then delete each branch when that feature gets merged into your main line.

Disposable Experimentation. Create a branch to experiment in, realize it's not going to work, and just delete it - abandoning the work—with nobody else ever seeing it (even if you've pushed other branches in the meantime).

Notably, when you push to a remote repository, you do not have to push all of your branches. You can choose to share just one of your branches, a few of them, or all of them. This tends to free people to try new ideas without worrying about having to plan how and when they are going to merge it in or share it with others.

There are ways to accomplish some of this with other systems, but the work involved is much more difficult and error-prone. Git makes this process incredibly easy and it changes the way most developers work when they learn it.

**About github**

**GitHub** is an [Internet hosting service](https://en.wikipedia.org/wiki/Internet_hosting_service) for [software development](https://en.wikipedia.org/wiki/Software_development) and [version control](https://en.wikipedia.org/wiki/Version_control) using [Git](https://en.wikipedia.org/wiki/Git). It provides the [distributed version control](https://en.wikipedia.org/wiki/Distributed_version_control) of Git plus [access control](https://en.wikipedia.org/wiki/Access_control), [bug tracking](https://en.wikipedia.org/wiki/Bug_tracking_system), [software feature](https://en.wikipedia.org/wiki/Software_feature), [task management](https://en.wikipedia.org/wiki/Task_management), [continuous integration](https://en.wikipedia.org/wiki/Continuous_integration), and [wikis](https://en.wikipedia.org/wiki/Wiki) for every project. Headquartered in [California](https://en.wikipedia.org/wiki/California), it has been a subsidiary of [Microsoft](https://en.wikipedia.org/wiki/Microsoft).

It is commonly used to host [open source](https://en.wikipedia.org/wiki/Open_source) software development projects. As of June 2022, GitHub reported having over 83 million developers and more than 200 million [repositories](https://en.wikipedia.org/wiki/Repository_(version_control)) including at least 28 million public repositories. It is the largest [source code](https://en.wikipedia.org/wiki/Source_code) .

The main purpose of GitHub.com is to facilitate the [version control](https://en.wikipedia.org/wiki/Version_control) and [issue tracking](https://en.wikipedia.org/wiki/Issue_tracking_system) aspects of software development. Labels, milestones, responsibility assignment, and a search engine are available for issue tracking. For version control, Git (and by extension GitHub.com) allows [pull requests](https://en.wikipedia.org/wiki/Pull_request) to propose changes to the source code. Users with the ability to review the proposed changes can see a diff of the requested changes and approve them. In Git terminology, this action is called "committing" and one instance of it is a "commit." A history of all commits is kept and can be viewed at a later time.

In addition, GitHub supports the following formats and features:

* Documentation,[[68]](blob:https://en.wikipedia.org/90505ad3-11d7-4b36-911e-eb1ee86a94ec#cite_note-69) including automatically rendered [README](https://en.wikipedia.org/wiki/README) files in a variety of [Markdown](https://en.wikipedia.org/wiki/Markdown)-like file formats (see [README § On GitHub](https://en.wikipedia.org/wiki/README#On_GitHub))
* [Wikis](https://en.wikipedia.org/wiki/Wikis)[[69]](blob:https://en.wikipedia.org/90505ad3-11d7-4b36-911e-eb1ee86a94ec#cite_note-70)
* GitHub Actions,[[70]](blob:https://en.wikipedia.org/90505ad3-11d7-4b36-911e-eb1ee86a94ec#cite_note-71) which allows building [continuous integration](https://en.wikipedia.org/wiki/Continuous_integration) and [continuous deployment](https://en.wikipedia.org/wiki/Continuous_deployment) pipelines for testing, releasing and deploying software without the use of third-party websites/platforms
* Graphs: pulse, contributors, commits, code frequency, punch card, network, members
* Integrations Directory[[71]](blob:https://en.wikipedia.org/90505ad3-11d7-4b36-911e-eb1ee86a94ec#cite_note-72)
* Email notifications[[72]](blob:https://en.wikipedia.org/90505ad3-11d7-4b36-911e-eb1ee86a94ec" \l "cite_note-73)
* Discussions[[73]](blob:https://en.wikipedia.org/90505ad3-11d7-4b36-911e-eb1ee86a94ec" \l "cite_note-74)
* Option to subscribe someone to notifications by [@ mentioning](https://en.wikipedia.org/wiki/Mention_(blogging)) them.[[74]](blob:https://en.wikipedia.org/90505ad3-11d7-4b36-911e-eb1ee86a94ec#cite_note-75)
* [Emojis](https://en.wikipedia.org/wiki/Emoji)[[75]](blob:https://en.wikipedia.org/90505ad3-11d7-4b36-911e-eb1ee86a94ec#cite_note-76)
* Nested [task-lists](https://en.wikipedia.org/wiki/To_do_list" \o "To do list) within files
* Visualization of [geospatial](https://en.wikipedia.org/wiki/Geospatial_analysis" \o "Geospatial analysis) data
* 3D render files that can be previewed using a new integrated STL file viewer that displays the files on a "3D canvas."[[76]](blob:https://en.wikipedia.org/90505ad3-11d7-4b36-911e-eb1ee86a94ec#cite_note-3d-77) The viewer is powered by [WebGL](https://en.wikipedia.org/wiki/WebGL" \o "WebGL) and [Three.js](https://en.wikipedia.org/wiki/Three.js).
* Photoshop's native PSD format can be previewed and compared to previous versions of the same file.
* PDF document viewer
* Security Alerts of known [Common Vulnerabilities and Exposures](https://en.wikipedia.org/wiki/Common_Vulnerabilities_and_Exposures) in different packages

GitHub's Terms of Service do not require public software projects hosted on GitHub to meet the [Open Source Definition](https://en.wikipedia.org/wiki/Open_Source_Definition). The [terms of service](https://en.wikipedia.org/wiki/Terms_of_service) state, "By setting your repositories to be viewed publicly, you agree to allow others to view and fork your repositories."[[77]](blob:https://en.wikipedia.org/90505ad3-11d7-4b36-911e-eb1ee86a94ec#cite_note-78)