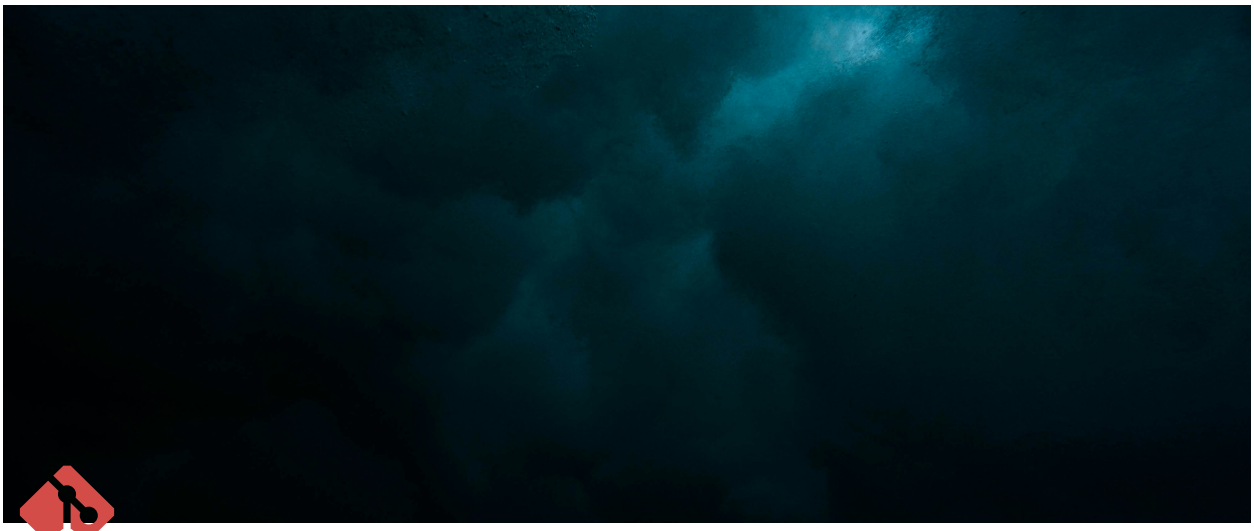


Контрольная работа. GIT

Control questions for today's lesson:

1. Что такое гит?
2. Зачем он нужен?
3. Как инициализировать локальный репозиторий?
4. Как связать локальный и удалённый репозиторий?
5. Как мне создать и отправить файл с моего локального реп. в удалённый?
6. Как стянуть последние обновления с удалённой ветки?
7. Что такое ветки в git?
8. Как создать ветку на удалённом репозитории?
9. Как переключиться на ветку?
10. Как влить ветку на удалённый репозиторий?
11. Как слить 2 ветки в 1?

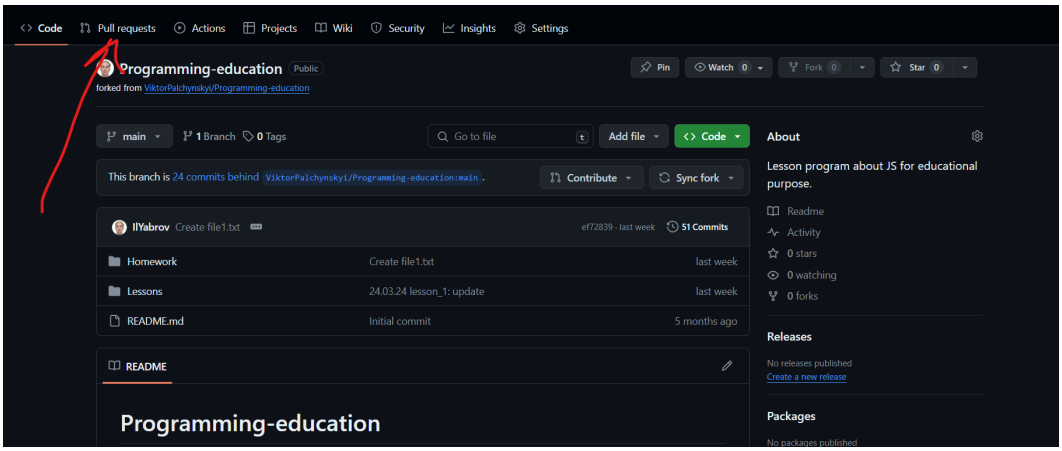
12. Что такое merge conflict?
13. Как зарезолвить merge conflict?
14. Что такое пул реквест?
15. Как создать пр?
16. Как откатить commit?

Answers:

1. GIT — это распределенная система контроля версий.
2. С помощью него легко организовать работу многих людей над каким-либо проектом. Он нужен для отслеживания версий проекта, работе с удаленными репозиториями. К тому же гит используют и для сохранности проекта, потому что он хранится не только локально, но и на серверах, что увеличивает безопасность.
3. Инициализация локального репозитория происходит с помощью команды `git init`
4. Чтобы связать локальный и удаленный репозиторий, используется команда `git remote add НАЗВАНИЕ_РЕПОЗИТОРИЯ АДРЕС_РЕПОЗИТОРИЯ` . В `НАЗВАНИЕ_РЕПОЗИТОРИЯ` указывается, как удаленный репозиторий будет называться локально, а в `АДРЕС_РЕПОЗИТОРИЯ` указывается его http-адрес или ssh-ключ.
5. Для создания файла в локальном репозитории используется либо графический интерфейс, либо командная строка. После создания статус файла — неотслеживаемый (unstaged). Его нужно перевести в отслеживаемый с помощью команды `git add НАЗВАНИЕ_ФАЙЛА` — теперь он имеет статус отслеживаемый (staged). Затем нужно сделать коммит с помощью команды `git commit -m"СООБЩЕНИЕ_КОММИТА"` , статус файла — commited, что значит, что изменение файла было зафиксировано в локальной базе GIT. Для отправки изменений на удаленный репозиторий используется команда `git push НАЗВАНИЕ_УДАЛЕННОГО_РЕПОЗИТОРИЯ ВЕТКА` .
6. С помощью команды `git pull НАЗВАНИЕ_УДАЛЕННОГО_РЕПОЗИТОРИЯ ВЕТКА` .
7. Ветки (branches) — это параллельные линии разработки проекта. Есть главная ветка — `master`, а есть ответвления от нее, где происходит разработка тестовых функций. Если функция нас устраивает, ты мы сливаем ее с главной веткой (делаем merge).

В ветке `master` хранится стабильная версия проекта, а новые функции разрабатываются в отдельных ветках, потому что мы не хотим внедрять нестабильные функции в стабильную версию проекта.

8. Чтобы создать ветку на удаленном репозитории нужно создать ветку и выполнить команду `git push НАЗВАНИЕ_ВЕТКИ`
9. С помощью команды `git checkout НАЗВАНИЕ_ВЕТКИ` можно переключиться на ветку `НАЗВАНИЕ_ВЕТКИ` .
10. Чтобы залить ветку на удаленный репозиторий нужно выполнить команду `git push НАЗВАНИЕ_УДАЛЕННОГО_РЕПОЗИТОРИЯ ВЕТКА`
11. Чтобы влить две ветки в одну, нужно выполнить команду `git merge НАЗВАНИЕ_ВЕТКИ`
12. Merge conflict — это ситуация, при которой гит не может сделать мердж. Рассмотрим ситуацию: мы редактируем какой-то файл в своей ветке, и в это же время кто-то другой уже подредактировал этот файл и замерджил его в основную ветку. Тогда при попытке замерджить наши исправления, гит не будет знать, какую версию файла использовать: нашу, или шаловливого коллеги, который успел нас опередить.
13. Чтобы зарезоливить merge conflict, нужно ручками внести изменения в файлах, в которых возник конфликт. Гит сам выдаст нам места, где есть несостыковки, и предложит оставить одну из версий файла.
14. Пул реквест — это механизм для предложения изменений из одной ветки в другую. В проекте участвуют много людей. Например, нужно ввести какую-нибудь функцию. Для этого один из разработчиков создает новую ветку, в которой будет разрабатываться эта функция, локально разрабатывает эту функцию, заливает в свой удаленный репозиторий и делает пул реквест, то есть отправляет запрос на соединение разрабатываемой ветки к основному проекту (запрос на мердж).
15. Чтобы создать пул реквест, нужно на Гитхабе перейти в своем репозитории во вкладку pull request



И сделать пул реквест, соответственно.

15. Чтобы откатить коммит, нужно выполнить команду `git reset --hard HEAD` или с помощью хэша комита `git reset  --hard ХЭШ_КОММИТА`