



# Обзор работы с git

## ▼ Основные действия

### ☐ git init



В VScode можно создать git репозиторий, зайдя в раздел source control. Редактор использует аккаунт гитхаба, который был введён при начале работы с git. Данная кнопка служит альтернативой консольной гит команды git init.

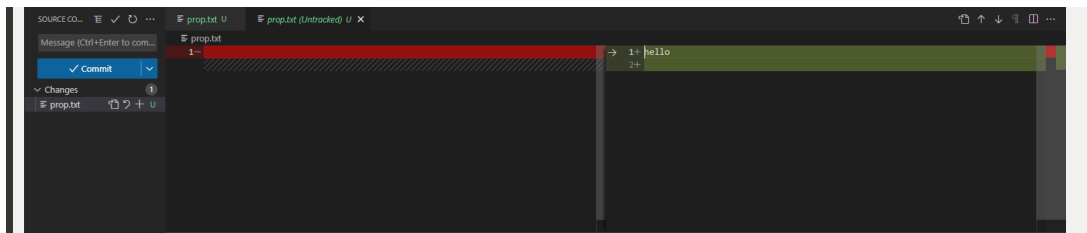
The folder currently open doesn't have a git repository. You can initialize a repository which will enable source control features powered by git.

Initialize Repository

### ☐ git status

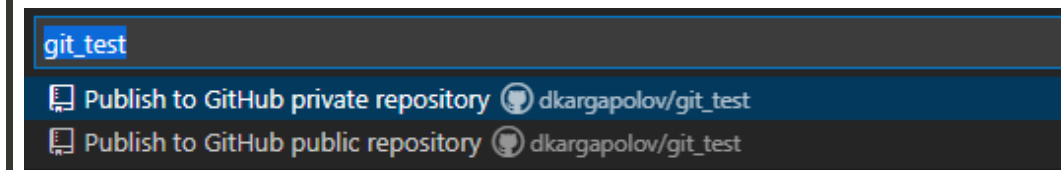
Редактор заменяет команду git status визуальным отображением некоторых вещей: так помечаются untracked файлы, текущая ветка и количество КОММИТОВ...


 master  0↓ 1↑





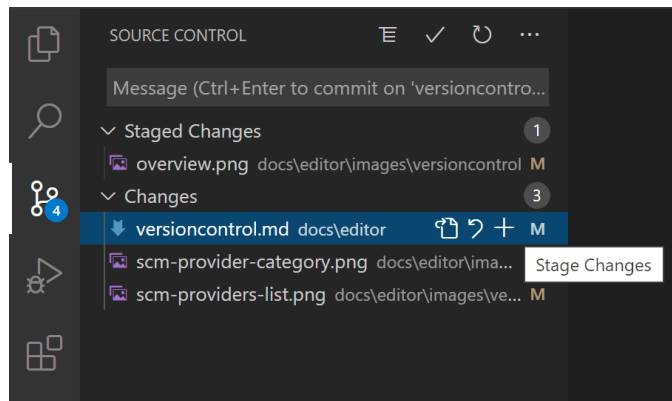
☐ `git commit -m "First file"`

При нажатии кнопки commit, которую можно увидеть в пункте выше. Программа спросит как именно мы хотим опубликовать данную папку. Стоит отметить, что редактор знает аккаунт заранее из-за требования информации о гитхаб аккаунте при скачивании git bash...



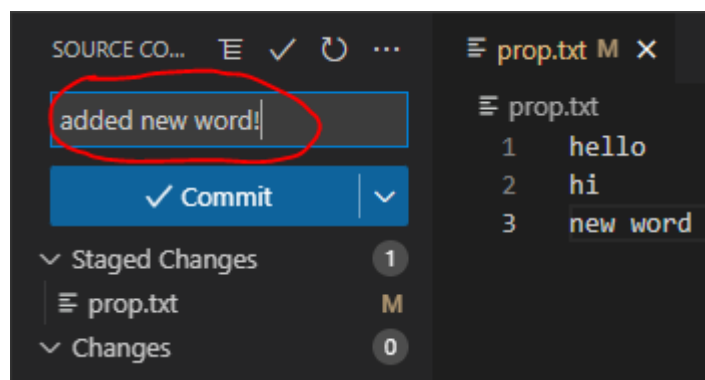
☐ `git add` (имя файла или  для всех файлов)

Значок  перенесёт выбранный файл в staged changes. Отмечу что M с правой стороны файла означает Modified (есть изменения). Чтобы перенести все файлы в staged changes, нужно нажать на Changes и там также выбрать .



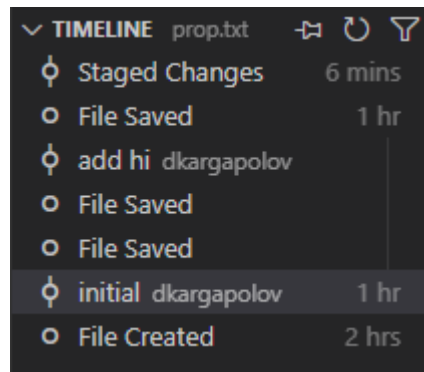
☐ `git commit -m "comment"`

Чтобы вносить комментарий вместе с комитом, достаточно просто вписать необходимый текст на панель, выделенную красным в нижнем скрине.

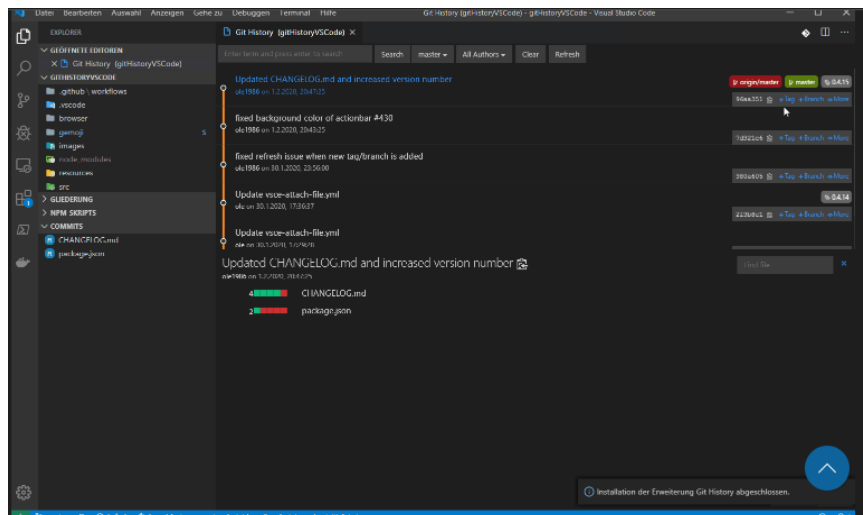


☐ `git log` ( история всех коммитов ) & `git log —summary`

В разделе explorer (работа с директориями файлов) в самом низу есть вкладка timeline, которая отображает некоторые действия, реализуемые через гит.

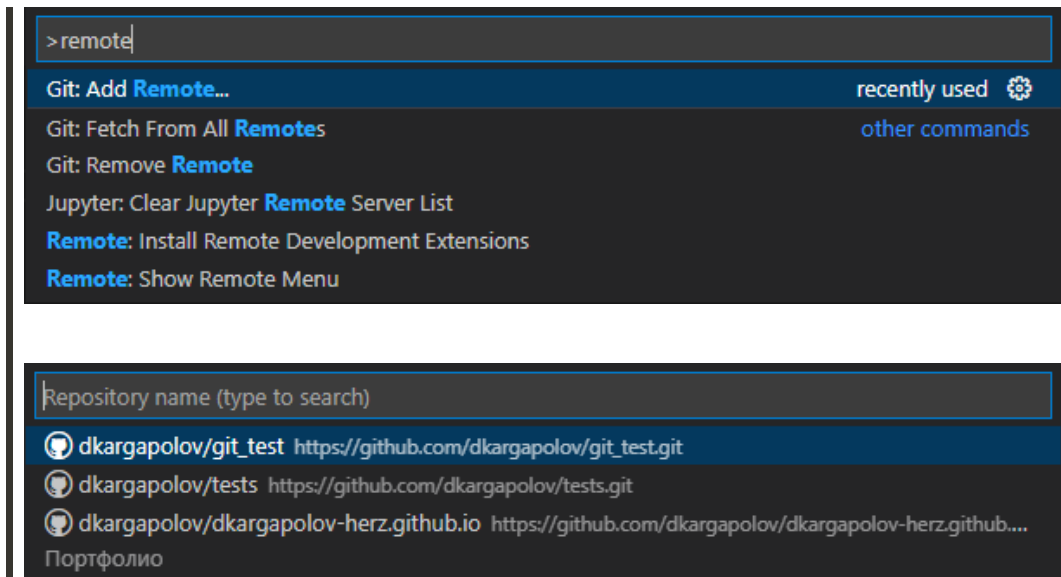


К сожалению сама программа не может визуально заменить git log, однако есть плагин git history, который справляется с этой задачей на ура!




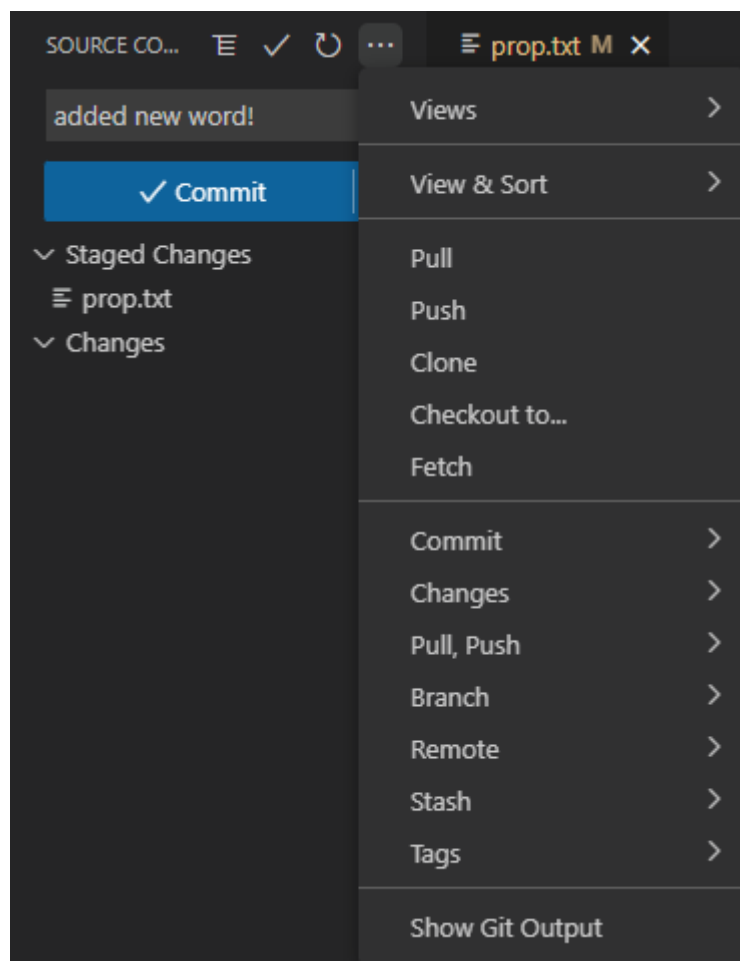
☐ git remote add origin \*repository link\*

Именно интерфейса для данной команды не предусмотрено, однако через комбинацию клавиш **Ctrl+Shift+P** можно вызвать данную команду, которая, помимо вставляемой ссылки, может подсказать готовые репозитории.



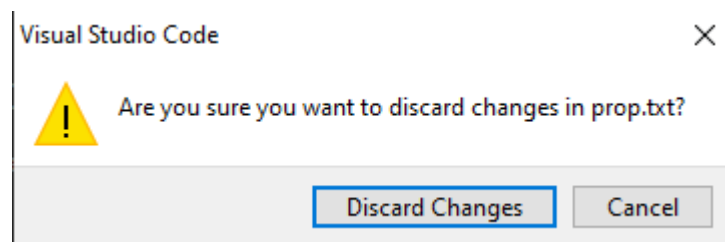
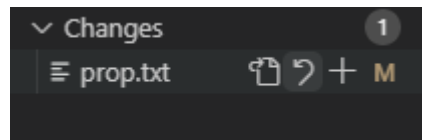
☐ `git push -u *origin master*` ( выгрузка в удалённый репозиторий )

Осуществление многих операций происходит через  В том числе и push и pull.



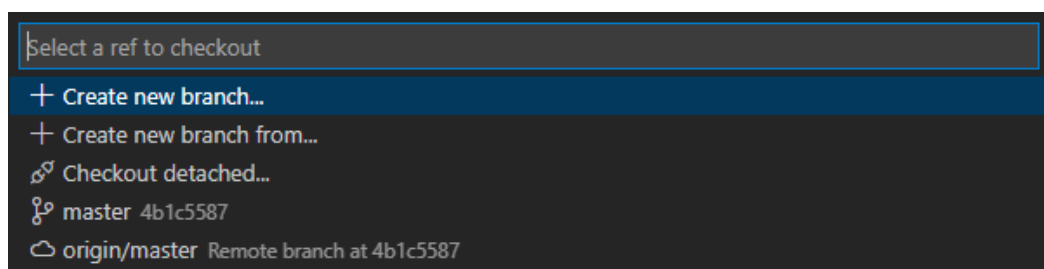
- ☐ git checkout — readme.txt (убрать последнее изменение)

Для удаления последних изменений также присутствует отдельная кнопка (загнутая стрелочка)



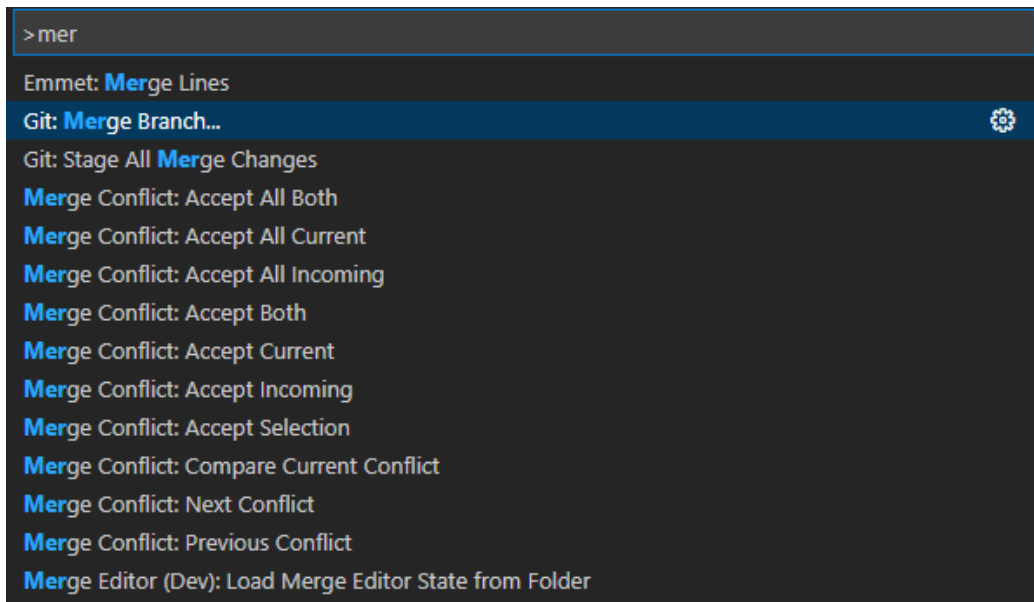
- ☐ git branch "название ветки" (создать новую ветку)

По нажатии на панель с текущей веткой, о которой уже говорилось ранее, выскочит панель с предложением создать новую ветку или переключиться на другую:

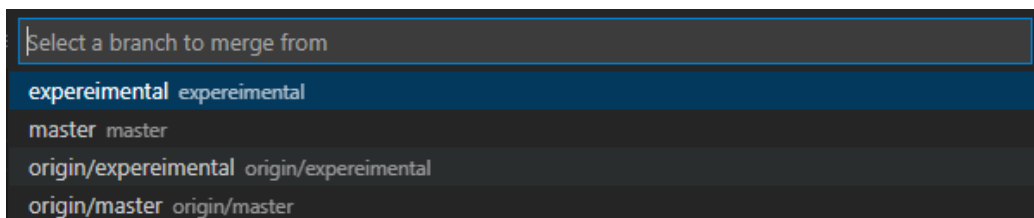


- ☐ git merge "название ветки, с которой мы хотим слить текущую ветку"

Для упрощённого слияние веток можно воспользоваться уже упомянутой комбинацией `Ctrl+Shift+P`



При нажатии на нужную команду, нам дадут выбор из существующих веток, которые мы хотим слить в одну, нажав на одну из них мы получим слияние с текущей веткой...



☐ `git branch -d` “название ветки, которую мы хотим удалить”

Воспользовавшись уже известной нам комбинацией клавиш, мы набираем `branch delete` и программа подсказывает нужную команду, далее она выдаст список веток, которые мы можем удалить:

