Ниже поэтапный процесс создания папки, файла, создание различных веток проекта их сохранений и слияние различных веток.

1. Создаём папку на рабочем столе

2. Заходим в Visiual Studio Cod

3. Открываем нашу папку

4. Создаём в ней файл

5. Открываем терминал в VSC, Вид - Терминал

6. git init вводим в терминале (создаёт в нашей папке репозиторий, в которой будет происходить отслеживание изменений.

7. git add file name (git начинает отслеживать изменения в файле) git commit -m “Initial commit” (сохранение с комментарием, Initial commit - типовой комментарий для первого сохранения.

8. Пишем в нашем файле дерево инструкции, сохраняем изменения Ctrl + S

9. git add file name git status (смотрим статус файла, проверяем, что в нём появились изменения это будет обозначено как midifiled: file name (цвет шрифта зеленый) git commit -m “Добавили структуру файла”

10. git branch (выводит на экран список веток, branch - ветка, \*master - выбираем за чистовик, \* будет обозначена ветка, на которой мы сейчас находимся

11. git branch text\_formatting (создание новой ветки, где text\_formatting - имя нашей ветки)

- git branch (покажет уже 2 ветки, фото во вложении)

![2022-07-25.png](https://s3-us-west-2.amazonaws.com/secure.notion-static.com/93900554-b7f3-4297-9b87-b3281ddbbceb/2022-07-25.png)

Если мы вызываем команду git log в ветке text\_formatting, то мы видим все сохранения в обратном порядке вплоть до первого, если же git log вызываем в ветке master, то ветка text\_formatting уже не отобразится, мы не увидим что было дальше или в соседних ветках

Clear (вводим в терминале, чтобы очистить терминал от введенных выше команд для нашего удобства

1. git checkout (имя ветки) text\_formatting (переходим на нужную нам ветку)

2. Работаем с файлом, заполняем заголовок Выделение текста, Сохраняем изменения Ctrl + S git add file name git commit -m “Добавили выделение текста” git log (мы видим все сохранения, которые были сделаны, а так же показано, где мы располагаемся)

3. git checkout master (переходим на чистовик) git log (теперь мы видим только 2 сохранения, так как сменили ветку

4. Хотим создать ещё 1 ветку git branch (проверяем, что мы находимся в чистовике) git branch (имя ветки) Lists (создаём новую ветку) git branch (проверяем, что новая ветка появилась) git chekcout Lists (переходим на вновь созданную ветку)

5. Работаем с файлом Сохраняем Ctrl + S git log (проверяем, что появились изменения) git add file name git commit -m “Добавили нумерованные и ненумерованные списки) git log (смотрим, что покажет)

- Появилась еще 1 ветка (во вложении)

![2022-07-25 (1).png](https://s3-us-west-2.amazonaws.com/secure.notion-static.com/bdd50fd3-e62f-4673-9459-2f3810993485/2022-07-25\_(1).png)

6. Предположим, что теперь нас полностью устраивает ветка text\_formatting и мы хотим залить её в чистовик git branch (проверяем в какой ветке находимся) git status (проверяем, что все сохранено) git checkout master (возвращаемся в чистовик) git merge (имя ветки) text\_formatting (этой командой мы добавляем одну ветку в другую)

7. Если ветка text\_formatting нам больше не нужна, мы можем её удалить git branch -d (d - сокращение от deleat, используем именно маленькую букву d)

8. Мы хотим добавить фотографию в наш текст, для этого мы сначала добавляем фотографию в нашу папку. Не принято какие-то большие файлы или изображения добавлять в git, их обычно игнорируют и работают с текстовой составляющей, а всё остальное хранят в отдельных файлах и папках. Если мы хотим, чтобы git не спрашивал нас постоянно про нашу фотографию: Мы создаём новый файл .git ignore В первой строке этого файла пишем имя файла или изображения, которое нам надо игнорировать, далее git add (имя файла) .git ignore git commit -m “добавили gitignore файл” И всё, git игнорирует наш файл

9. Рассмотрим более сложную ситуацию, когда в разных ветках один и тот же текст написан по разному. Слияние пройдет, но пройдет с конфликтом и отображаться будет 2 варианта текста внутри файла с указанием какой текст с какой ветки пришел В VSC есть возможность: Принять текущую версию Accept Current Cgange Ту версию, которая к нам пришла Accept Incoming Change Оставить оба варианта Accept Both Change Либо их сравнить Cempare Change После того как мы решили какой вариант мы выбираем, необходимо сохранить то что получилось командой: git log - - graph после этой команды сохраняем как обычно git add file name git commit -m “comment”