**Перед началом работы:**Установить Git и VScode

Настройка Git

git config --global [user.name](http://user.name/) – имя пользователя

git config --global [user.email](http://user.email/) - электронная почта

git config --global color.ui auto - не обязательная команда включающая подсветку команд

git config --global core.editor - установка текстового редактора по умолчанию когда хочется своим редактором пользоваться

1. Запускаем VScode и открываем терминал в VScode
2. Для проверки работоспособности вводим команду git –version. Если все ОК выведет в терминал версию
3. Создали папку (где угодно), например, на рабочем столе и даем ей имя например git и открыли эту папку с помощью VScode.
4. git init - Для того что б git начал отслеживать папку, еу нужно инициализировать (активировать). После введения этой команды в папке появится скрытая папка .git это значит что папка отслеживается если ее удалить папка станет обычной
5. git status - Что б узнать что происходит с этой папкой. Сообщит что мы находимся в ветке master (главная), нет commit т.е. файлы не зафиксированы, отслеживать нечего
6. hello world.md - Создаем файл можно через интерфейс WSCode (расширение может быть любым в данном случае это расширение языка Markdown .md, что б синтаксис языка работал). Записываем в него текст, Что б на Markdown сделать обзац необходимо писать через строку иначе это будит одна строка, сохраняем ctrl+s
7. git statys - Проверяем статус снова. Теперь git говорит что есть файл но он его не отслеживает.
8. git add hello world.md – добавляем наш файл к отслеживанию. Теперь git отслеживает файл но мы еще не фиксировали его состояния (Не делали как бы сохранения к которому можно будит вернуться если что то пойдет не так)
9. git commit -m "Создали новый файл" - Что б зафиксировать состаяния файла и указываем какой то комментарий в кавычках что б знать в каком состоянии мы решили запомнить файл. После добавления commit терминал сообщит что создан новый coomit и какие изменения в нем произошли присвоен ему какой то номер. Изменим файл, например что нить в нем напишем. И сохраним ctrl+s. У нас есть измененный файл и есть файл до изменений благодаря git. Проверим теперь его состояние git status: Сообщит в терминале что ветка основная, изменения не зафиксированы.
10. Добавим к отслеживанию и зафексируем изменения: git add hello world.md, git commit -m "добавили в него текст". Теперь у нас 2 состояния одного и того же файла и мы можем между ними переходить. Что б переходить между состояниями надо знать к какому состоянию мы можем перейти. git log - Выведит в консоль внутренний номер commit. Имя и email автора, дату изменения, введенный комментарий. И так по всем зафиксированным изменениям. Что б вернуться к воду команд ввести Q
11. git checkout №…… - Для перехода к commit №........, достаточно ввести 5-6 первых символов номера. Произойдет переход к той версии номер которой будит введен
12. git checkout master – как мы в последствии узнаем эта камонда нас переводит на нужную ветку master – имя ветки по умолчанию значит основная, своим веткам мы даем свои названия.
13. Если файл переименовать git скажет что файл был удален и покажет новый не добавленный файл который можно добавить (add) и зафиксировать (commit)
14. git diff Если нужно проверить есть ли у не сохраненного файла изменения. Показывает изменения в файлах

22. git add. добавляет сразу все файлы

git –version

git init

git status

git add hello world.md

git commit –m "Пояснение к фиксации"

git log

git checkout master

git diff

git add.