Git commit- создание коммита (снимок всех файлов) может иметь предков

Git branch NAMEBRANCHES – создание ветки

Git checkout -b NAMEBRANCHES – создание ветки и моментальный переход на нее

Git merge – слияние веток

Git rebase – тоже слияние веток, но более чисто ( делает все в одну линию, оставляет копию коммита )

Git checkout ХЭШКОММИТА – отделяет голову к хэшу комита ( названию коммита )

Git log – находит хэш коммита

Git checkout main^ -- перенесет голову на один коммит назад ( на предка )

Git checkout HEAD~<num>--переходит назад на то количество предков, сколько указано в <num>

Git branch -f main HEAD~<num> -- прикрепляет ветку к коммиту ( можно указать его хэш)

Git reset HEAD~<num> – отменяет изменения перенося ссылку назад ( работает на локальных ветках)

Git revert HEAD ( на удаленных ветках ) – содержит изменения полностью противоположные тем, что были в коммите

Git cherry-pick <commit1> <commit2><….>-- копирует коммиты которые мы выбрали, не сливая ветки ( очень полезная штука )

Git rebase -I HEAD~<num>-- открывается окно и делает все, что там будет сказано

Git commit - - amend – вносит изменения

Git tag <name> <hash> -- создание тэга ( который мы инициализируем ) на определенном коммите

Git describe <ref>– показывает как далеко текущее состояния от тега

Можно совмещать тильду и ^ для более быстрого перехода по дереву

Git clone – создаст локальную копию удаленного репозитория

Git fetch—извлекает данные из удаленного репозитория

Git pull – упрощает скачивание и сливание

Git fakeTeamwork—создает несколько коммитов

Git push—добавляет ваши наработки в удаленный репозиторий

Git pull –rebase/git pull and git push – решает проблему, когда на удаленный проект был добавлен более новый коммит

Pull request

Git branch -u o/main foo—указывает ветке отслеживать удаленную ветку

Git push origin main – принудительно пушит указанные ветки

Git push origin foo:main из одного в другую