|  |  |  |
| --- | --- | --- |
| git add **directory** | Отметить все файлы в **directory** для стейджинга | Directory = . чтобы отметить все файлы в текущей папке и подпапках:  git add . |
| git remote add origin **URL** | Создать ремоут для локального репозитория | URL- адрес удаленного репозитория |
| git remote set-url origin **URL** | Изменить адрес ремоута для локального репозитория | URL- адрес удаленного репозитория |
| Git commit –m “**message**” | Зафиксировать (закоммитить) изменения на текущей ветке | **Message-** сообщение коммита |
| git push **remote** **main** **feature2** | Загрузить ветки **main** и **feature2** на ремоут с названием **url** | Веток можно указать сколько угодно  Вместо **remote** может быть адрес (https://github.com/name /repo.git)  По умолчанию выгрузка идет на origin |
| git push --set-upstream **url** | Указать для текущей ветки апстрим (**url**) и отправить изменения | Пример **url** - https://github.com/name /repo.git |
| git checkout -b **branch1** | Создать новую ветку **branch1** от текущей ветки и забрать туда несохраненные изменения |  |
| git checkout **branch2** | Переключиться на существующую ветку **branch2** с текущей ветки и забрать туда несохраненные изменения |  |
| git merge **branch** | Сделать слияние (merge) ветки **branch** в текущую ветку | Используйте no fast-forward чтобы сохранить всю историю коммитов:  git merge --no-ff |
| git rebase **main** | Сделать перемещение  (rebase) текущей ветки на ветку **main** |  |
| При конфликте во время ребейза  Accept incoming change - предпочитать изменения из текущей ветки  accept current change - предпочитать изменения из другой ветки (**main** в примере выше)  решить конфликты, сохранить файлы -> застейджить изменения (git add) -> продолжить ребейз  **git rebase --continue** - продолжить ребейз  **git rebase --abort** отменить ребейз  **git status** – узнать текущий статус | | |
| Git diff develop.. feature2 | Посмотреть разницу (дифф) между ветками develop и feature2 |  |
| Git reset --hard HEAD~1 | Откатить текущую ветку на 1 коммит назад | HEAD~2 – откатить на 2 коммита и т.д.  --hard – удалить все изменения  --soft – оставить изменения в редакторе |
| git reset HEAD **directory** | Отменить изменения всех файлов в директории **directory** (откат до состояния последнего коммита) | .точка вместо directory чтобы откатить всю папку  Для отката пока коммит еще не был сделан |
| git branch -d **testing** | Удалить ветку **testing** локально | Перед удалением переключитесь на другую ветку  Если ветка не смержена до конца, выполняйте  git branch -D **testing** |
| git push upstream --delete **hotfix** | Удалить ветку **hotfix** на ремоуте upstream | Вместо названия ремоута (upstream) можно указать url |
| git tag -a 1.0 -m ”message” | Поставить текущей ветке тэг 1.0 с сообщением “message” |  |
| git clean -fd | Удалить незастейдженные изменения в текущей папке | До того как сделали git add |