https://github.com/kontur-courses/git

Git Rules Sheet

R1. Fetch Any Time

Всегда можно получить коммиты любого репозитория через fetch

- `git remote -v` — вывести список удаленных репозиториев с их адресами
- `git remote add <name> <url>` — добавить удаленный репозиторий с URL и дать ему указанное имя
- `git fetch = git fetch origin` — получить содержимое основного удаленного репозитория
- `git fetch --all` — получить содержимое всех удаленных репозиториев из списка

R2. Will Push Force Be With You

Изменить удаленный репозиторий — это push

- `git push <remote> <local_branch>:<remote_branch>` — добавить изменения из локальной ветки `<local_branch>` и переместить ветку `<remote_branch>` удаленного репозитория
- `git push = git push origin HEAD` — добавить изменения из текущей локальной ветки и переместить соответствующую ветку удаленного репозитория
- `git push -f` — выполнить `push`, даже если удаленная ветка находится в другом поддереве
- `git push <remote> -d <branch>` — удалить ветку в удаленном репозитории

R3. Upstream Mapping

Связь локальных и удаленных веток устанавливается явно

- `git branch -vv` — вывести список локальных веток с указанием привязанных к ним upstream-веток
- `git branch -u <upstream> [<branchname>]` — задать upstream-ветку для указанной или текущей ветки
- `git push -u origin HEAD` — создать удаленную ветку, соответствующую локальной и установить между ними upstream-связь, затем добавить изменения из локальной ветки в удаленный репозиторий
- `git checkout <remote_branchname>` — создать локальную ветку, соответствующую удаленной и установить между ними upstream-связь, затем переместить HEAD на нее
- `git pull = git pull origin` — получить содержимое основного удаленного репозитория и влить изменения из удаленной ветки в соответствующую локальную ветку
- `git pull --ff-only` — получить содержимое, а затем влить, если возможен fast-forward merge
- `git pull --rebase` — получить содержимое и выполнить rebase локальной ветки на удаленную ветку
- `git config --global push.default simple` — задать simple-режим действий с upstream-связями при push. Это режим по умолчанию в Git 2.0 и выше

S1. Everything Is Local

Все работает локально

- `git init` — создать пустой репозиторий
- `git clone <url>` — склонировать репозиторий в новую директорию

S2. Tree Of Commits

Хранится последовательность состояний некоторой директории

- `git add .` — добавить все измененные файлы в индекс
- `git commit -m <msg>` — записать изменения из индекс в репозиторий
- `gitex commit` — открыть интерфейс коммита из Git Extensions
- `git status -sb` — вывести состояние директории и индекса кратко с указанием текущей ветки
- `git show <commit>` — показать содержимое коммита
- `git log --oneline` — вывести список коммитов, связанных с HEAD, в кратком виде
- `git log --oneline --graph` — вывести историю коммитов, связанных с HEAD, в виде дерева
- `git log --oneline --graph --all` — вывести историю всех коммитов в виде дерева
- `gitk` — открыть графическое представление репозитория
- `gitex` — открыть Git Extensions для текущей папки

S3. Refer To Branch

Используются именованные ссылки

- `git tag` — вывести список тегов
- `git tag <tagname>` — создать тег
- `git branch` — вывести список локальных веток
- `git branch -av` - вывести список локальных и удаленных веток
- `git branch <branchname>` — создать ветку
- `git branch -d <branchname>` — удалить ветку
- `git checkout <commit>` — переместить HEAD на коммит
- `git checkout <branch>` — переместить HEAD на ветку
- `git checkout -m <branch>` — переместить HEAD, объединить текущие изменения с состоянием ветки
- `git checkout -b <new_branch>` — создать новую ветку и перейти на нее

A1. Merge Them All

Два состояния можно объединить через merge, mergetool и commit

- `git merge <commit>` — объединить текущую ветку с другой
- `git mergetool` — разрешить имеющиеся конфликты

A2. Immutable History

Нельзя переписать историю — можно создать новую

- `git commit --ammend` — заменить последний коммит ветки на отредактированный с дополнительными изменениями
- `git rebase <upstream>` — применить все коммиты от общего родителя до текущего к `<upstream>`
- `git rebase -i <upstream>` — применить заново все коммиты, указав действие с каждым коммитом
- `git cherry-pick <commit>` — применить указанный коммит к HEAD

A3. Reset The Difference

Хранятся файлы, разница вычисляется на лету

- `git reset --hard <commit>` — переместить текущую ветку на `<commit>`, задать индекс и директорию согласно коммиту, устранив всю разницу
- `git reset --mixed <commit>` — переместить текущую ветку на `<commit>`, задать индекс согласно коммиту, оставить разницу между исходным и новым состоянием в директории
- `git reset --soft <commit>` — переместить текущую ветку на `<commit>`, не задавать индекс и директорию согласно коммиту, а оставить разницу между исходным и новым состоянием в индексе и директории
- `git reset --hard HEAD~1` — отменить последний коммит
- `git stash` — сохранить все модифицированные файлы в виде набора изменений
- `git stash pop` — восстановить последний сохраненный набор изменений и удалить его из списка
- `git stash list` — показать список сохраненных наборов изменений
- `git revert <commit>` — создать коммит, отменяющий изменения из коммита
- `git diff <from_commit> [<to_commit>]` — вывести разницу между двумя коммитами
- `git diff --name-status <from_commit> [<to_commit>]` — список измененных файлов
- `git difftool <from_commit> [<to_commit>]` - вывести разницу с помощью difftool из настроек

A4. Hide The Garbage

Видно только то, на что есть ссылки

- `git gc` — удалить ненужные файлы и оптимизировать локальный репозиторий
- `git clean` — удалить неотслеживаемые файлы из директории
- `git reflog show <ref>` — показать лог действий со ссылкой
- `git reflog = git reflog show HEAD` — показать лог действий с HEAD

H1. Helpful And Configurable

Гибкая настройка под любой процесс

- `git help` — список команд
- `git <command> -h` — помощь по команде в терминале
- `git <command> --help` — документация по команде в браузере
- `git config -e --system` — редактировать настройки системы
- `git config -e --global` — редактировать настройки пользователя
- `git config -e` — редактировать настройки репозитория
- `git config --global user.name "<name>"` — задать имя пользователя
- `git config --global user.email "<email>"` — задать почту пользователя

Aliases

- `git config --global alias.x "!start GitExtensions.exe"`
- `git config --global alias.ci "!start GitExtensions.exe commit"`
- `git config --global alias.st "status -sb"`
- `git config --global alias.graph "log --oneline --graph --all"`
- `git config --global alias.co "checkout"`
- `git config --global alias.pushup "push -u origin HEAD"`
- `git config --global alias.puff "pull --ff-only"`
- `git config --global alias.pure "pull --rebase"`
- `git config --global alias.undo "reset --hard HEAD~1"`