Spis treści

[Где тренировать работу с git 2](#_Toc171028777)

[Сайты 2](#_Toc171028778)

[Видео 2](#_Toc171028779)

[Oсновные команды git 2](#_Toc171028780)

[add 2](#_Toc171028781)

[commit 2](#_Toc171028782)

[branch 2](#_Toc171028783)

[switch/checkout 2](#_Toc171028784)

[checkout -b 2](#_Toc171028785)

[merge 3](#_Toc171028786)

[rebase 3](#_Toc171028787)

[gitk 3](#_Toc171028788)

[log 3](#_Toc171028789)

[branch -f 3](#_Toc171028790)

[reset 3](#_Toc171028791)

[revert 3](#_Toc171028792)

[Работа с удальонным репозитрием 4](#_Toc171028793)

[clone 4](#_Toc171028794)

[fetch 4](#_Toc171028795)

[pull 4](#_Toc171028796)

[push 4](#_Toc171028797)

[branch -r 4](#_Toc171028798)

[Что делать если код не совпадает когда кто то обновил удальоный репозиторий 5](#_Toc171028799)

[как создать удальонную ветку 5](#_Toc171028800)

[как привязатся к удальонной ветке 5](#_Toc171028801)

[как удалить удаленную ветку 5](#_Toc171028802)

# Где тренировать работу с git

## Сайты

**Основы работоы с git**

<https://learngitbranching.js.org>

**Полигон для проверки команд**

<https://git-school.github.io/visualizing-git/>

## Видео

**Колаборацыя на github**

<https://www.youtube.com/watch?v=MnUd31TvBoU>

# Oсновные команды git

## add

сохраяняет измения в память  
git add .

## commit

создает новый комит

git commit -m ”<< название комита >>”

## branch

создает новую ветку

git branch << название ветки >>

## switch/checkout

пеключает на выбраную ветку

git switch/checkout << название ветки >>

checkout -b

создает новую ветку и переключается на нее (делает работу **branch** a потом **switch**)

git checkout -b << название ветки >>

## merge

делает слияние двух веток и создает новый комит с двух веток

git merge << название ветки >>

например если ты находишся на ветке **main** и хочеш сделать слияние с веткой **html-branch** нужно наисать команду:

git merge html-branch

## rebase

переноси комиты даной ветки на на конец выбраной вами ветки

git rebase master

## gitk

Показывает древо репозитория

> gitk

## log

Показывает все комиты в консоли

git log --oneline

## branch -f

Прикрипляет ветку к опередельонному комиту  
git branch -f << название текущей ветки >> HEAD/<< название комита >>

например: git branch -f html-branch HEAD~3

## reset

удаляет комиты и переносит ветку на старый комит

git reset HEAD/<< название комита >>

## revert

копирует комит и переносит его как HEAD ветки

git revert HEAD/<< название комита >>

# Работа с удальонным репозитрием

## clone

копирует репозиторий

git clone << сылка на репозиторий >>

## fetch

скачивает все изменения с удальонного репозитория

git fetch << удалённый репозиторий >> << целевая ветка >>

git fetch << удалённый репозиторий >> << источник >>:<< получатель >>

## pull

скачивает все изменения с удальонного репозитория а потом исполняет слияние веток **(git fetch; git merge; == git pull)**

git pull << удалённый репозиторий >> << целевая ветка >>

git pull << удалённый репозиторий >> << источник >>:<< получатель >>

git pull - - rebase (если хочется испльзовать ребейз а не мердж)

## push

высылает все изменения с локального репозиторя на удальонный

если получателя нет то ветка сама создастя удаленно

git push << удалённый репозиторий >> << целевая ветка >>

git push << удалённый репозиторий >> << источник >>:<< получатель >>

источник – может быть **branch(^~)** или **HEAD(^~)**

например:

git push origin main

git push origin master:main

## branch -r

позволяет увидеть удаленные ветки

git branch -r

## Что делать если код не совпадает когда кто то обновил удальоный репозиторий

1. git fetch; git rebase original/main; git push;
2. git fetch; git merge original/main; git push;
3. git pull; git push
4. git pull - - rebase; git push

## как создать удальонную ветку

git branch << название ветки >>

git push --set-upstream origin << название ветки >>

## как привязатся к удальонной ветке

1. git branch -b << название ветки >> origin/<< название ветки >>
2. git branch -u origin/<< название ветки >> << название ветки >>

Например:

1. git branch -b test origin/test
2. git branch -u origin/main test

## как удалить удаленную ветку

git push origin :main