



Введение

GitHub - это веб-сервис для хостинга и совместной работы над Git-репозиториями.

Для того, чтобы клонировать репозиторий к себе на локальное устройство:

```
git clone <url-repository>  
git clone <ssh-repository>
```

SSH - это протокол безопасного удалённого доступа. Можно создать и добавить на github свой ssh ключ с устройства, чтобы не входить в свой аккаунт постоянно.

```
ssh-keygen -t ed25519 -C "your_email@example.com"
```

Ключ генерируется в

```
/c/Users/sghdz/.ssh/id_ed25519)
```

Запуск ssh agent

```
eval "ssh-agent -s"
```

Добавления ключа в ssh-agent

```
ssh-add c:/Users/YOU/.ssh/id_ed25519
```

Далее необходимо добавить ключ на GitHub

Подтверждение привязки

```
ssh -T git@github.com
```

Работа в репозитории

Подключение

Для просмотра списка связанных удаленных репозиториев

```
git remote -v
```

Чтобы установить связь между локальным и удаленным репозиторием, необходимо добавить удаленный доступ

```
git remote add origin <url>
```

Origin - это короткое название url. Чтобы обращаться не url, а к названию origin
Чтобы поменять данное название

```
git remote rename <old-name> <new-name>
```

Чтобы удалить удаленный доступ

```
git remote remove <name>
```

Отправка данных

Команда для отправки данных

```
git push <url-name> <branch_name>  
git push origin main
```

При отправки данных создается удаленная ветка с таким же название, поэтому мы можем отправить данные с одной локальной ветки на другую удаленную ветку

```
git push <url-name> <local-branch_name>:<remote-branch_name>
```

Параметр `-u` (или `--set-upstream`) в команде `git push` связывает локальную ветку с удалённой — то есть задаёт `upstream`-ветку. Это упрощает дальнейшую работу с этой веткой.

```
git push -u <remote> <branch>
```

Посмотреть список удалённых веток

```
git branch -r
```

Получение данных

Ветки на удалённом репозитории выглядят так: `origin/main origin/dev`

Чтобы получить ветки которых нету в локальном репозитории

```
git fetch
git fetch --all
```

Чтобы создать локальную ветку из удалённой

```
git checkout -b my-branch origin/my-branch
```

`git fetch` — просто забрать изменения.

`git fetch` скачивает все изменения с удалённого репозитория, но не сливает их с текущей веткой.

```
git fetch origin
```

`git pull` — `fetch + merge` (или `rebase`) `git pull` сначала делает `fetch`, затем автоматически сливает (`merge`) или переносит (`rebase`) изменения из удалённой ветки в текущую локальную.

```
git pull origin main
```

Резюме по работе в команде в репозитории

Клонируешь репозиторий

```
git clone git@github.com:company/project.git  
cd project
```

Просмотр всех удаленных веток

```
git branch -r
```

Переходишь в основную ветку

```
git checkout main  
# или  
git checkout -b main origin/main
```

Создаешь ветку для своей работы

```
git checkout -b my-feature
```

Перед коммитами необходимо проверять обновления в **main**

```
git checkout main  
git pull origin main
```

Для обновления в своей ветке

```
git checkout my-feature  
git fetch origin  
git merge origin/main # или git rebase origin/main
```

Отправка своей ветки на удаленный репозиторий

```
git push -u origin my-feature
```

- Создание **Pull Request (PR)** на **github** из своей ветки в **main**.
- Появляются правки, просто пушишь свою ветку снова, **PR** автоматически обновится

После одобрения **PR**

```
git checkout main  
git pull origin main
```