

## Итоги занятия. Работа с удаленным репозиторием через GitHub

---

### Итоги занятия. Работа с удаленным репозиторием через GitHub

Давайте подведем итоги занятия “Работа с удаленным репозиторием через GitHub”. Отметим основные мысли, которые стоит вынести из занятия.

**GitHub** - самый популярный в мире сервис для публикации и хранения в интернете проектов, а также для командной разработки.

Для начала работы с сервисом необходимо пройти регистрацию на [сайте](#)

**SSH-ключ** - это уникальный набор символов, который позволяет связать локальный и удаленный репозитории, чтобы:

- из командной строки отправлять данные прямо в удалённый репозиторий
- проекты попадали в нужный профиль, а не в чужой

**Ветка в Git** представляет собой указатель на снимок изменений. Эта опция в Git, которая позволяет создавать «параллельную реальность» в проектах, когда:

- над проектом работает несколько человек
- каждый делает свою задачу
- чтобы не мешать друг другу, каждый работает в своей ветке
- когда задача готова, то всё сливается в основную ветку

### Ключевые действия и команды

- Чтобы сгенерировать SSH-ключ, нужно в терминале ввести команду **ssh keygen**, скопировать окончание пути ключа. Далее в терминале введите команду **cat ~ окончание пути** до файла, который вы только что скопировали. Скопируйте длинный набор символов. В GitHub нажмите **New SSH Key** и вставьте данный набор символов.
- Чтобы создать новый удаленный репозиторий, нажмите в GitHub кнопку **New Repository**
- Чтобы связать удаленный и локальный репозитории, выполните команду **git remote add origin ссылка на папку удалённого репозитория**
- Чтобы отправить что-то в удалённый репозиторий, нужно воспользоваться командой **git push -u origin main**

- Чтобы создать копию чужого проекта на GitHub, нужно нажать кнопку **Fork** на странице проекта
- Чтобы клонировать удалённый репозиторий на компьютер, нужно скопировать ссылку по кнопке **Code** и затем выполнить в терминале команду **git clone ссылка-на-репозиторий**. После клонирования не забыть перейти в новую папку командой **cd имя-папки**
- Чтобы создать новую ветку и переключиться на неё, нужна команда **git checkout -b имя-новой-ветки**
- Чтобы переключаться между ветками, выполняем команду **git checkout имя-ветки**
- Для слияния находимся на ветке, куда сливаем, и выполняем **git merge имя-сливаемой-ветки**
- При разрешении конфликта обязательно удаляем служебные символы, оставляем нужный вариант и коммитим изменения

После изучения теоретического материала рекомендуем приступить к практике. Выполните обязательное домашнее задание с проверкой от преподавателя. Вы научитесь создавать репозиторий на GitHub, связывать его с локальным репозиторием, публиковать изменения, делать форк и клонирование удаленного репозитория, а также решать возникшие конфликты при работе с ветками.

Желаем успехов в обучении!

