

Лекция: Введение в Git и GitHub

Что такое Git?

Git — это распределённая система контроля версий. Она позволяет:

1. Хранить разные версии проекта.
2. Отслеживать изменения в коде.
3. Быстро переключаться между версиями.
4. Работать в команде над одним проектом.
5. Разрабатывать новые функции параллельно (в отдельных ветках).

Git vs GitHub vs GitLab

Git:

- Устанавливается локально.
- Управляет версиями проекта на вашем компьютере.
- Не зависит от подключения к интернету.

GitHub:

- Онлайн-платформа для хранения Git-репозитория.
- Предоставляет веб-интерфейс, pull request'ы, issue-трекинг.
- Работает как "облачный сервер" для совместной разработки.

GitLab:

- Похож на GitHub, но может быть развернут на ваших серверах.
- Важен для компаний, которым нужен контроль над безопасностью и конфиденциальностью.

Зачем использовать GitHub

- Хранение кода в облаке. Репозиторий доступен с любого устройства.
- Резервное копирование.
- Совместная разработка с другими через pull requests.
- Отслеживание задач и ошибок.

Лекция: Введение в Git и GitHub

Установка Git и настройка

1. Установка Git: <https://git-scm.com/>
2. На Mac: `brew install git`
3. Настройка Git:
`git config --global user.name "Имя"`
`git config --global user.email "email@example.com"`
4. Проверка: `git config --list`
5. Генерация SSH-ключа:
`ssh-keygen -t ed25519 -C "email@example.com"`
`cat ~/.ssh/id_ed25519.pub` → вставить в GitHub (Settings → SSH and GPG Keys)

Создание проекта

1. Зарегистрироваться на <https://github.com/>
2. Создать новый репозиторий
3. В IDE создать проект
4. В терминале:
`git init`
`git add .`
`git commit -m "Первый коммит"`
`git branch -M main`
`git remote add origin git@github.com:username/repo.git`
`git push -u origin main`

Основные команды Git

- `git init` — инициализация репозитория
- `git status` — текущее состояние
- `git add .` — добавление файлов
- `git commit -m "msg"` — сохранение изменений
- `git push` — отправка на GitHub

Лекция: Введение в Git и GitHub

`git pull` — получение изменений

`git clone <url>` — клонирование репозитория

`git branch` — просмотр/создание веток

`git checkout <branch>` — переключение

`git merge <branch>` — слияние веток

Работа с репозиторием

1. Изменили код → `git status`
2. Добавили в индекс → `git add .`
3. Сделали коммит → `git commit -m "комментарий"`
4. Отправили на сервер → `git push`
5. Получили изменения → `git pull`

Важно

`git init` нужно выполнять для каждого нового локального проекта, если вы не клонируете его с GitHub.

Если используете `git clone` — `init` не требуется, репозиторий уже готов.