

Лекция 3

Углубляемся в контроль версий





Лекция 3. Видео

Интерактивный учебник-тренажёр по Git

В учебнике есть всё, что нужно знать о Git джуниор-программисту.

Репозиторий с примерами из лекции



[3. Углубляемся в контроль версий.pdf](#)



Таймкоды

1:30

Введение. Интерактивный учебник по Git.

9:25

Разграничение понятий Git и GitHub

11:29

Работа с удаленными репозиториями. Скачивание из текущего репозитория и слияние со своей версией

30:40

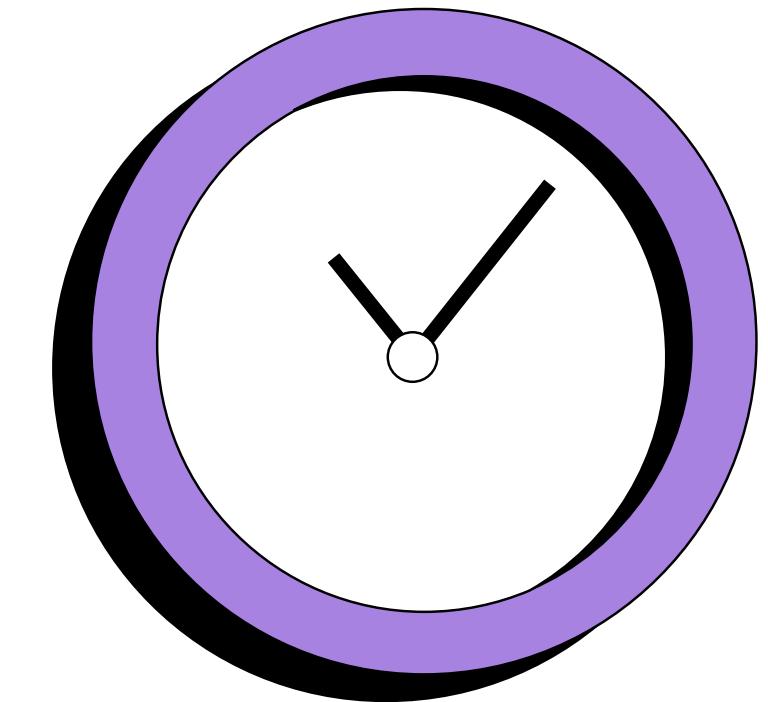
Демонстрация использования GitHub. Настройка совместной работы

43:22

Pull request

59:42

Краткое резюме лекции





Цель лекции:

- * Освоить работу с удаленными репозиториями, которые находятся не на локальной, а на удаленной машине, например, на сервере.



При запуске команды `git init` всё происходит только в локальном репозитории: в папке на компьютере пользователя, эту папку создавшего. Но для работы в команде программисты используют удаленные репозитории.



Разграничение понятий Git и GitHub



- * Освоить работу с удаленными репозиториями, которые находятся не на локальной, а на удаленной машине, например, на сервере.
- * git – одна из систем контроля версий
- * Способ организации и поддержания версионности
- * Самая популярная система контроля версий
- * Сервис компании Майкрософт для организации работы удаленных репозиториев
- * Самый популярный сервис Git
- * Много полезных функций
- * Огромный архив различного кода



Работа с удаленными репозиториями. Скачивание из текущего репозитория и слияние со своей версией

Освоить работу с удаленными репозиториями, которые находятся не на локальной, а на удаленной машине, например, на сервере.

Копировать внешний репозиторий на свой ПК можно командой `git clone`.

```
ilnarsafigullin@MacBook-Pro-Administrator: Lesson 2 % git clone https://github.com/ilnar-geekbrains/version_control_lection_3.git
```

↑
Адрес репозитория, который копируем на свой ПК

`git clone`



Команда **git clone** составная: она не только загружает все изменения, но и пытается слить все ветки на локальном компьютере и в удаленном репозитории.

git pull

* Эта команда позволяет скачать все из текущего репозитория и автоматически сделать merge с нашей версией

```
ilnarsafigullin@MacBook-Pro-Administrator Lesson 2 % git pull
```

git pull

git push

Отправить свою версию репозитория во внешний репозиторий поможет команда **git push**. При первом её использовании нужна авторизация.

* Эта команда позволяет отправить нашу версию репозитория на внешний репозиторий. ТРЕБУЕТ АВТОРИЗАЦИИ на внешнем репозитории.

```
ilnarsafigullin@MacBook-Pro-Administrator Lesson 2 % git push
```





Как настроить совместную работу

1. Создать аккаунт на GitHub.com
2. Создать локальный репозиторий
3. “Подружить” ваш локальный и удалённый репозитории.
GitHub при создании нового репозитория подскажет, как это можно сделать
4. Отправить (push) ваш локальный репозиторий в удалённый (на GitHub), при этом, возможно, вам нужно будет авторизоваться на удалённом репозитории
5. Провести изменения “с другого компьютера”
6. Выкачать (pull) актуальное состояние из удалённого репозитория



pull request

- команда для предложения изменений
- запрос на слияние изменений в репозиторий

В больших компаниях один ответственный за проект создает аккаунт. Другие пользователи дают команду pull request. Предлагать изменения на GitHub нужно в отдельной ветке. Сначала пользователь копирует репозиторий на свой компьютер, делает fork репозитория, затем клонирует версию на своём ПК, создаёт ветку с предлагаемыми изменениями, отправляет изменения командой push в свой аккаунт на GitHub и даёт команду pull request.



Как сделать pull request

- * Делаем **fork** (ответвление) репозитория
- * Делаем git clone **СВОЕЙ** версии репозитория
- * Создаем новую ветку и в НЕЕ вносим свои изменения
- * Фиксируем изменения (делаем коммиты)
- * Отправляем свою версию в свой GitHub
- * На сайте GitHub нажимаем кнопку **pull request**

