



HTML и CSS. Адаптивная вёрстка и автоматизация

Уровень 2, с 21 ноября 2022 по 30 января 2023

Меню курса

[Главная](#) / [1. Старт](#) /

1.10. Клонирование репозитория

 ~ 4 минуты

После того как вы сделали форк, нужно клонировать его себе на компьютер, чтобы вести работу с кодом локально. Тут нам и пригодится SSH.

Заходим на страницу форка, нажимаем на зелёную кнопку «Code» и копируем команду клонирования проекта по SSH.

HTTPS или SSH?

Выпадающее меню кнопки «Code» предлагает открыть репозиторий в приложении GitHub Desktop, скачать zip-архив с исходным кодом или клонировать репозиторий по протоколу SSH или HTTPS.

Что лучше — HTTPS или SSH? При использовании HTTPS вам при каждом клонировании приватного репозитория или для отправки изменений в любой репозиторий нужно будет ввести ваш логин и пароль для GitHub. **При каждой отправке изменений вводить пароль.** Можно настроить Git, чтобы он запоминал пароль на некоторое время, но тогда где-то на вашем компьютере будет храниться ваш логин и пароль от GitHub, который злоумышленник может легко украсть и с его помощью изменить или удалить ваш профиль на GitHub. А ещё иногда компьютер будет забывать пароль и придётся вводить его снова.

Если вы используете двухфакторную аутентификацию в GitHub, то пароль не подойдёт: для отправки изменений нужно будет сгенерировать специальный токен.

Если вы используете SSH-ключи, то на каждом компьютере достаточно сгенерировать ключ один раз и однажды добавить его в GitHub. Не нужно вво

никаких паролей при работе с репозиториями. Конечно, приватный ключ могут украсть, как и пароль. Но с приватным ключом не получится авторизоваться на GitHub, менять настройки или удалить профиль, можно только отправлять и получать изменения.

Если вы обнаружили, что один из ваших ключей скомпрометирован, просто удалите его из GitHub.

Если вы хотите, можете защитить приватный SSH-ключ паролем. Просто введите пароль при генерации ключа. Тогда ключом сможет пользоваться только тот, кто знает пароль. Если при этом использовать SSH-агент, то пароль нужно будет вводить только один раз — при входе в консоль. Но это тема для другого рассказа.

В общем, использовать протокол SSH и удобнее, и безопаснее. Но есть один случай, когда использовать SSH не получится — если у вас на работе или в университете такие настройки сети, что SSH просто не работает. Такое редко, но бывает. Тогда придётся использовать HTTPS.

Открываем терминал, переходим в директорию, где хотим сохранить папку с проектом, вводим команду `git clone` и вставляем скопированную команду. Полностью команда будет выглядеть примерно так:

```
git clone git@github.com:your-nickname/your-project.git
```

Если вы правильно настроили SSH-ключи, Git начнёт процесс копирования репозитория на ваш компьютер. Если вы видите ошибку, в которой написано `Error: Permission denied (publickey)`, скорее всего, вы ошиблись где-то при выполнении инструкции по настройке SSH-ключа. Вернитесь в главу про настройку SSH и попробуйте повторить процесс настройки.

Кстати, если вы хотите, чтобы название папки с проектом у вас на компьютере отличалось от имени репозитория, можете дополнить команду клонирования, добавив в конце другое название:

```
git clone git@github.com:_your-nickname_/_your-project_.git folder_name
```

Теперь, на вашем компьютере, в папке `your_project` или в той, название которой вы указали самостоятельно, находится полная копия репозитория с GitHub.

Чтобы начать работу с проектом, надо оказаться в его директории. Для этого уже известную нам команду `cd`, после которой указываем название проекта на е

компьютере: `cd your-project`.

Прочитали главу?

Нажмите кнопку «Готово», чтобы сохранить прогресс.

Готово

⚠ Если вы обнаружили ошибку или неработающую ссылку, выделите ее и нажмите Ctrl + Enter

Поиск по материалам

Git

[Все материалы](#)

В самом начале



- ☐ [Пройдите опрос](#)
- ☐ [Укажите персональные данные](#)
- ☐ [Изучите регламент](#)
- ☐ [Прочитайте FAQ](#)
- ☐ [Добавьте свой Гитхаб](#)
- ☐ [Выберите наставника](#)
- ☐ [Создайте проект](#)

Мой наставник



[Выбрать наставника](#)

Работа с наставником

У вас осталось **10** из 10 консультаций.

[История](#)



Практикум

Тренажёры

Подписка

Для команд и компаний

Учебник по PHP

Курсы

HTML и CSS. Профессиональная вёрстка сайтов

HTML и CSS. Адаптивная вёрстка и автоматизация

JavaScript. Профессиональная разработка веб-интерфейсов

JavaScript. Архитектура клиентских приложений

React. Разработка сложных клиентских приложений

Node.js. Профессиональная разработка REST API

Node.js и Nest.js. Микросервисная архитектура

TypeScript. Теория типов

Алгоритмы и структуры данных

Паттерны проектирования

Webpack

Vue.js 3. Разработка клиентских приложений

Git и GitHub

Анимация для фронтендеров

Блог

С чего начать

Шпаргалки для разработчиков

Отчеты о курсах

Информация

Об Академии

О центре карьеры

Соглашение

Конфиденциальность

Сведения об образовательной организации

Лицензия № 4696

Профессии

Фронтенд-разработчик

JavaScript-разработчик

Фулстек-разработчик

Услуги

Работа наставником

Для учителей

Стать автором

Остальное

Написать нам

Мероприятия

Форум

