




HTML и CSS. Адаптивная вёрстка и автоматизация

Уровень 2, с 21 ноября 2022 по 30 января 2023

Меню курса

[Главная](#) / [1. Старт](#) /

1.4. Система контроля версий Git

 ~ 5 минут

Для начала определим, что такое система контроля версий.

Так называют программу, которая позволяет хранить разные версии одного и того же документа, легко переключаться между ранними и поздними вариантами, вносить и отслеживать изменения.

Системы контроля версий используют, например, команды разработчиков, так как они часто трудятся над одним проектом, а значит, сразу несколько человек могут изменять один файл одновременно. Система контроля версий помогает таким командам избежать путаницы: она позволяет хранить историю изменений проекта и при необходимости даёт возможность вернуться к предыдущей версии.

Одна из самых популярных систем контроля версий называется Git. Работу Git, как и любой другой системы контроля версий, можно сравнить с процессом сохранения и загрузки в компьютерных играх:

- если впереди ждёт тяжёлое сражение, то перед этим лучше заранее сохраниться;
- чтобы это сделать, нужно выполнить специальную команду;
- после этого сохранение попадает в специальную папку и содержит состояние игры;
- теперь при необходимости всегда есть возможность вернуться к предыдущей версии игры.

Папка, содержащая данные игры, могла бы выглядеть так:

```
SomeGame/  
| - saves  
| | - save001.sav  
| | - save002.sav
```

```
| | ...
| | папка с сохранениями
|
| - game.exe
| ...файлы игры
```

В папке `saves` хранится история всех сохранений игры. Git сохраняет код вашего проекта по такому же принципу: сохранения попадают в специальную скрытую папку.

Основные понятия

Репозиторий

Проект, в котором была инициализирована система Git, называется репозиторием. Репозиторий хранит все рабочие файлы и историю их изменений.

Рабочая область и хранилище

```
barbershop/
| - .git
| | - bea0f8e
| | - d516600
| |   Хранилище
|
| - css
| - index.html
|   Рабочая область
```

Корневая папка проекта — это рабочая область. В ней находятся все файлы и папки, необходимые для его работы.

Хранилище — это содержимое скрытой папки `.git`, которая добавляется в проект при инициализации. В этой папке хранятся все версии рабочей области и служебная информация. Этим версиям система автоматически даёт название, состоящее из букв и цифр. В примере выше — это `bea0f8e` и `d516600`.

Коммит

Точно так же, как и в игре, в системе контроля версий Git можно сохранить текущее состояние проекта. Для этого есть специальная команда — `commit`. Она делает так, что новая версия проекта сохраняется и добавляется в хранилище. Каждый коммит хранит список изменений, по которым можно восстановить полное состояние рабочей области, её папок и файлов проекта.

В итоге проект работает так:

1. Репозиторий хранит все версии проекта. В случае передачи этого проекта другому человеку, он увидит всё, что с ним происходило до этого.
2. Ничего не теряется и не удаляется бесследно. При удалении файла в новой версии добавляется запись о том, что файл был удалён.
3. Всегда можно вернуться к любой из версий проекта, загрузив её из хранилища в рабочую область.

Окружение Git

Плюсом Git является то, что вокруг него создано множество сервисов, которые позволяют сделать работу с ним удобнее. Расскажем о тех, что будут вам полезны в начале работы.

GitHub

GitHub — это веб-сервис, который основан на системе Git. Это такая социальная сеть для разработчиков, которая помогает удобно вести коллективную разработку IT-проектов. Здесь можно публиковать и редактировать свой код, комментировать чужие наработки, следить за новостями других пользователей. Именно в GitHub работаем мы, команда Академии, и студенты курсов.

Терминал

Отличие Git от других программ — отсутствие графической версии. Поэтому работа с Git ведётся через командную строку. В разных операционных системах свои программы для взаимодействия с Git. В мире разработки такие программы называют «терминал» или «консоль». А работает это так: мы вводим команду и получаем реакцию машины: сообщение об ошибке, запрос на подтверждение информации, результат выполненных действий.

GUI (Graphical User Interface — Графический Интерфейс Пользователя)

Облегчить работу с Git и GitHub могут специальные программы. Такие программы в удобной форме показывают изменения в коде, список коммитов и обладают другими удобными возможностями. Обычно в подобных программах есть возможность выполнять стандартные Git команды, просто нажав на кнопку.

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

Профессии

Фронтенд-разработчик

JavaScript-разработчик

Фулстек-разработчик

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

Услуги

Работа наставником

Для учителей

Стать автором

Остальное

Написать нам

Мероприятия

Форум

Соглашение

Конфиденциальность

Сведения об образовательной организации

Лицензия № 4696

