

4. Лабораторная работа №1. Система управления версиями GIT

[Дать ответ на задание](#)[Получить оценку](#)

Цель: изучить основные команды для работы с системой контроля версий GIT и получить навыки работы с хостингом репозитория.

Результат выполнения работы: письменный отчет и репозиторий размещенный на одном из хостингов (Github/Gitlab/Bitbucket).

Содержание и порядок выполнения лабораторной работы

Подготовка к выполнению лабораторной работы

1. Установить GIT. Скачать установщик можно на [официальном сайте](#). Во время установки будет задаваться большое количество вопросов по его настройке. Если вы не знаете, что делает та или иная настройка, то оставляйте как предлагает установщик.
2. Зарегистрироваться на одном из сервисов на ваш выбор:
 - [Github](#)
 - [Gitlab](#)
 - [Bitbucket](#)
3. Изучить:
 1. Основные команды терминала:
 - cd
 - ls
 - mkdir
 - pwd
 - cp
 - rm
 2. Основные команды GIT:
 - git config
 - git init
 - git add
 - git commit
 - git log
 - git status
 - git remote
 - git push
 - git pull
 - git clone
 - git branch
 - git checkout
 - git merge

Выполнение работы

1. Запустите терминал (cmd, powershell, terminal).
2. С помощью терминала и команды **cd** перейти в папку одной из ваших лабораторных работ по блоку "Основы HTML/CSS/JS" (где находится index.html).
3. Убедитесь в том, что вы находитесь внутри папки лабораторной работы. Список файлов после выполнения команды **ls** должен быть вам знаком.
4. Произведите конфигурацию GIT через команды **git config --global user.name "Your name"** и **git config --global user.email "email@example.com"**.
5. Используя команду **git init --initial-branch=main** инициализируйте репозиторий в папке вашего проекта.
6. Добавьте файлы вашего проекта в индекс репозитория - **git add**. Можно добавить все файлы разом **git add -A**.
7. Создайте ваш первый коммит **git commit -m "ваше описание коммита"**.
8. Убедитесь, что коммит создан - **git log**.
9. Создайте новую ветку в репозитории - **git branch ваше-имя-ветки**. Имя ветки выберите в соответствии с тем, как дальше вы будете изменять код: исправите ошибку, подберете лучшее имя класса или оптимизируете CSS.
10. Посмотрите весь список веток - **git branch**, сохраните скриншот.
11. Переключитесь на новую ветку **git checkout ваше-имя-ветки**.
12. Внесите какие-то изменения в код вашей работы.
13. Сохраните файлы, добавьте изменения в систему контроля версий **git add имя-измененного-файла**, и создайте коммит с этими изменениями **git commit -m "ваше описание коммита"**.
14. Повторите предыдущие пункты 1-2 раза.
15. Переключитесь на ветку main **git checkout main**. Обратите внимание, что изменения, которые были добавлены в ветке, отсутствуют и вы вернись к состоянию файлов до ее создания.
16. Влейте (объедините) вашу ветку в основную **git merge ваше-имя-ветки**.

17. Проверьте историю изменений `git log`, сохраните скриншот.
18. Посмотрите древо вашего репозитория `git log --graph --full-history --all`, сохраните скриншот.
19. На сервисе, где вы зарегистрировались, создайте новый проект, сделайте его публичным, чтобы преподаватель мог зайти в него.
И когда будете создавать проект, **обязательно** снимите галочку с "создать README.md".
20. На странице созданного проекта найдите кнопку "Clone", из нее скопируйте URL из поля "Clone with HTTPS". Он должен начинаться с https и заканчиваться .git.
21. Добавьте ссылку к вашему локальному репозиторию до удаленного - `git remote add origin https://ваш-url.git`.
22. Теперь вы готовы отправить все изменения в удаленный репозиторий - `git push -u origin main`.

Защита

Для защиты необходимо приложить **отчет по лабораторной работе**, который содержит:

1. Титульный лист
2. Список использованных вами команд и их кратким описанием.
3. Скриншоты выполнения команд:

1. git branch

2. git log

3. git log --graph
4. Ссылку на ваш проект на одном из сервисов.

Источники информации

1. [Команды терминала для новичков](#)
2. [Онлайн тренажер](#)
3. [Официальная документация](#)
4. [Как писать сообщения коммитов](#)
5. [Пошаговое руководство](#)
6. [Введение от Github \(EN\)](#)
7. [Визуализация команд Git](#)
8. [Курс на Stepik](#)
9. ["Инструмент для совместной работы, с нуля и до регламента в команде — Сергей Сергеев. Лекция от Яндекс](#)
10. [Git happens! 6 типичных ошибок Git и как их исправить](#)
11. [Репозиторий готовых команд](#)
12. [Про GitFlow на Habr](#)

Резюме оценивания

Изолированные группы

Все участники

Скрыто от студентов	Нет
Участники	254
Ответы	44
Требуют оценки	5

◀ Презентация к лекции 01.12.2021 Инструмент. средства (Даньшина М.В.)

Перейти на...

4. Лабораторная работа №2. Работа с хостингом ▶