Первые шаги в Git и GitHub

Урок 1. Работа в командной строке в Git Bash

Работать с Git можно через командную строку либо графический интерфейс какой-либо IDE, например, VS Code.

В данном руководстве будет использоваться способ через консоль.

Для этого нужно установить специальный консольный инструмент для Windows, который называется **Git Bash**.

Есть несколько способов установки. Мы рекомендуем пакет **Git for Windows**. Он установит не только Bash, но и сам Git. Вот что нужно сделать:

1. Перейдите [на эту страницу официального сайта Git](https://git-scm.com/download/win).
2. Скачайте одну из двух версий из категории **Standalone Installer** (англ. «автономный установщик»). Узнать тип вашей системы Windows можно в настройках.
3. Запустите программу установки. Обратите внимание, куда будет установлен Git. Обычно это директория C:\Program Files\Git.
4. Проверьте, что в списке устанавливаемых программ стоит галочка напротив пункта **Git Bash Here** — это позволит открывать консоль с Git в любой папке.
5. Далее установщик предложит много опций. Для нашего курса достаточно оставить все настройки по умолчанию. Несколько раз нажмите **Next** (англ. «далее»), пока не начнётся процесс установки.
6. После окончания установки нажмите **Finish** (англ. «завершить»).
7. Теперь на вашем компьютере есть не только командная строка, но и Git.

### Первый запуск Git Bash

Запустите программу Git Bash. Сделать это можно двумя способами. Можно ввести название программы в окно поиска на панели задач (через Пуск).

А можно открыть директорию, в которую был установлен Git. Обычно это директория C:\Program Files\Git\bin. Перейдите в bin и запустите файл bash.exe.

Если вы видите консоль, значит, установка прошла успешно.

# Знакомство с командной строкой

Символ доллара ($) означает, что программа ждёт ваших команд.

В командной строке вы всегда находитесь в какой-то папке — просто этого не видно. Узнать, где вы сейчас, поможет команда pwd (от англ. ***p***rint ***w***orking ***d***irectory — «показать рабочую папку»). Она выводит путь к текущей директории.

Введите pwd в командную строку и нажмите клавишу Enter для выполнения команды.

*Основные команды:*

$ pwd выводит путь к текущей директории

$ cd ~ перейти к домашней директории

Домашняя директория может быть разной для разных пользователей одного и того же компьютера.

$ cd .. перейти к родительской директории

$ cd имя\_папки сменить директорию

Обратите внимание: если в названии папки есть пробелы, при вводе нужно использовать кавычки.

$ cd . перейти к текущей директории

Чтобы обратиться к текущей директории, можно использовать одну точку. Но это нужно довольно редко — например, для запуска скриптов и программ, которые принимают папку в качестве параметра.

Также cd позволяет перемещаться сразу через несколько директорий. Для этого нужно разделить их названия знаком /.

$ ls вывести содержимое директории

$ ls -a вывести содержимое директории со скрытыми файлами

В списке отобразятся все скрытые файлы, которые начинаются с символа . (например, файлы конфигурации). В том числе два особых файла . и .., которые обозначают текущую и родительскую директории.

Как и другие команды, ls может работать с символами домашней директории, родительской директории, либо указывая путь.

$ touch my-new-file.txt # создание файла my-new-file.txt

$ mkdir new-dir # создание директории new-dir

Можно создать целую структуру директорий одной командой с помощью флага -p.

$ mkdir -p dir1/dir11/dir111

# создали папку dir111 в папке dir11, которая находится в папке dir1

По умолчанию touch и mkdir создают файлы и папки в текущей рабочей директории. Например, если вы находитесь в директории abs, команда touch file.txt создаст файл именно там: abs/file.txt.

Также можно использовать обе команды вместе с символом домашней директории (~) или родительской директории (..). Например, команда mkdir ~/my-git-projects создаст папку my-git-projects внутри домашней директории.

А команда touch ../../file.txt создаст файл file.txt на две папки выше по иерархии. Допустим, если вы находитесь в директории projects/git/hello, команда touch ../../file.txt создаст файл по такому пути: projects/file.txt.

$ cp что\_копируем куда\_копируем

$ cp index.html src/ # скопировали index.html в папку src

Но можно указать сразу несколько файлов.

$ cp что\_копируем что\_копируем что\_копируем куда\_копируем

$ cp index.html style.css script.js src/ # скопировали три файла (index.html, style.css и script.js) в папку src

$ mv что\_перемещаем куда\_перемещаем

$ mv table.csv ./very-important-files

# сначала указываем имя файла, который хотим переместить, потом путь — куда перемещаем

$ cat имя\_файла чтение файла, вывести (распечатать) содержимое файла на экране

Команда cat работает только с текстовыми файлами. Вывести этой командой файл другого типа (например, изображение) не получится.

$ rm example.txt # удалили файл example.txt из текущей папки

Удалить папку можно командой rmdir (от англ. **r***e***m***ove* **dir***ectory —* «удалить директорию»). Не забудьте указать имя папки.

$ rmdir имя\_папки # команда удалит папку из текущей директории, если папка пуста

Если в папке, которую вы пытаетесь стереть, есть какие-то файлы, то командная строка не удалит её и выведет сообщение о том, что папка не пуста (англ. Directory not empty).

$ rm -r images # удалили папку images со всем её содержимым из текущей директории

Будьте осторожны: удаление объектов командами rm и rmdir необратимо — в этом случае файлы и папки не попадают в корзину и исчезают навсегда.

Команды в терминале необязательно вбивать и выполнять по очереди. Их можно указывать не по одной, а сразу списком. Для этого их нужно разделить двумя амперсандами (&&).

Создаём папку second-project, переходим в папку second-project и создаём в ней два файла: index.html и style.css:

$ mkdir second-project && cd second-project && touch index.html style.css

### *Вызывайте команды из буфера*

Чтобы обратиться к последней введённой команде, нажмите на клавиатуре стрелку вверх (↑). Если нажать ещё раз, появится предпоследняя команда; ещё раз — предпредпоследняя; и так далее. Чтобы вернуться — например, от предпоследней команды к последней, — нажмите стрелку вниз (↓).

### *Используйте автозаполнение*

Необязательно заучивать все команды наизусть. Если нужно найти какую-нибудь из них, достаточно вспомнить, с каких букв она начинается. Можно набрать их в командной строке и дважды нажать клавишу Tab. Терминал покажет список всех команд, которые начинаются с этих символов.

Tab автоматически дописывает не только команды, но и пути. Начните печатать имя папки или файла (они должны быть в той же директории) и нажмите Tab. Терминал заполнит имя автоматически.

Если этого не происходит, значит, есть несколько файлов или папок, которые начинаются так же. Нажмите Tab ещё раз, и вы увидите их список. Терминал не знает, как ему дозаполнить такой ввод и что именно выбрать, поэтому показывает все варианты, чтобы вы могли уточнить запрос.

А ещё можно почти мгновенно перемещаться в ключевые папки. Допустим, вы хотите увидеть содержимое **корневой директории** (англ. *root directory*). Это верхняя в иерархии папка, в которой хранится всё, что есть на вашем жёстком диске.

$ cd c:/ # переместились в корневую директорию