# Git

Git - децентрализованное хранилище, система управления версиями.

Репозиторий - хранилище чего-либо (в данном случае, кода).

[GitHub](https://github.com/) - площадка для работы с кодом, имплементация git.

Имплементация - программная или аппаратная реализация какого-либо протокола, алгоритма, технологии.  
  
Не путайте git и GitHub!  
  
На курсе мы будем использовать [GitHub](https://github.com/) для работы с open-source проектами (с открытым кодом) и [GitLab](https://gitlab.com/) для внутренних проектов, где мы будем учиться взаимодействовать между собой.  
  
Задание: Завести аккаунты на обоих перечисленных площадках, если ещё этого не сделали.  
  
Перед началом работы с git, ознакомимся с командами терминала, использованными в лекции:

cd путь (change directory) - команда смены текущей директории.

mkdir название (make directory) - команда создания папки с определенным названием.

ls - команда отображения списка файлов находящихся в текущей директории.

git - команда работы с git

## Как загрузить свой первый проект в git

Первым делом нам предстоит сгенерировать ssh-ключ для нашего ПК, чтобы не приходилось каждый раз вводить логин и пароль при работе с удалённым репозиторием.

Для всех устройств (mac, ubuntu, wsl) команда генерации одна:

ssh-keygen -t rsa -b 4096 -C "your\_email@example.com"

"your\_email@example.com" - ваш e-mail (пишите в кавычках!).  
  
После генерации ключа у вас в терминале будет несколько вопросов, разберем каждый из них.

Сначала вам предлагают выбрать путь сохранения вашего ssh-ключа и его имя. Самый простой вариант - ничего не трогать и просто нажать Enter.

> Enter a file in which to save the key (/Users/you/.ssh/id\_rsa): [Press enter]

Далее вы можете создать ключевую фразу, для дополнительной защиты вашего ключа. Самый простой вариант - не создавать ключевую фразу и просто нажать Enter.

> Enter passphrase (empty for no passphrase): [Type a passphrase]

Если вы всё же ввели ключевую фразу, вам придется вводить её каждый раз при работе с удалённым репозиторием.  
  
После выполнения действий выше, нам предстоит скопировать наш ssh-ключ и вставить его в своём профиле GitHub.

Для начала нам нужно отобразить содержимое файла с ключом.

Если вы ничего не меняли и просто нижимали Enter, то ваш ключ откроется командой:

cat ~/.ssh/id\_rsa.pub

В консоли отобразится полное содержимое файла, которое нужно будет выделить и скопировать (Command + C или Ctrl + C).

Скопированный ключ нужно добавить в список своих ключей по ссылке: [github.com/settings/keys](https://github.com/settings/keys)

Нажимаете кнопку New SSH key и вводите:

Title: любое имя для вашего ключа

Key: вставляете свой ssh-ключ

После чего нажимаете Add SSH key.

Теперь мы можем начинать работать с git.  
  
Для инициализации своего репозитория, нам нужно создать папку проекта, перейти в неё с помощью терминала и ввести следующую команду:

git init

После применения данной команды, в папке вашего проекта будет создана папка .git, с данными обо всех файлах вашего проекта.

Данная папка будет скрытой в mac и linux (ubuntu), так как точка перед названиеем папки/файла делает их скрытыми.

Папку .git можно как-либо редактировать, но лучше этого не делать, дабы не столкнуться с вероятными проблемами.  
  
Обратите внимание! Если у вас установлен oh-my-zsh, то у вас будет отображаться слово master перед курсором терминала.

master - это главная ветка вашего проекта.

Про ветки проектов мы поговрим позднее.  
  
Для демонстрации работы git, создадим в папке нашего проекта файл readme.md, любым доступным нам способом.

После создания файла, нам нужно ввести в терминале команду:

git add .

Данная команда добавляет все файлы из папки проекта в наш репозиторий.

add - добавляет файл в проект.

. (точка) - означает все файлы находящиеся в папке проекта.

Вместо точки мы можем написать конкретное имя файла, например, readme.md, чтобы добавить в репозиторий конкретный файл.  
  
Для просмотра состояния нашего репозитория используется команда:

git status

В данный моменты мы можем увидеть строку

new file: readme.md

Данная строка означает, что у нас есть новые файлы готовые к коммиту.  
  
Коммит (commit) - снимок состояния вашего проекта.

Каждый раз, когда мы хотим отправить в хранилище файлы, нужно делать коммит и писать лаконичный комментарий к внесенным изменениям.

git commit -m 'added file readme.md'

commit - создаёт коммит текущих файлов проекта.

-m - позволяет писать комментарий к коммиту.

'added file readme.md' - комментарий к коммиту.

Можно объединить две команды (add и commit) в одну:

git commit -am 'added file readme.md again'

-am - добавление всех файлов и комментария в коммит

Не стоит злоупотреблять ключом -am, но знать его нужно.  
  
Всегда пишите комментарии к коммитам, даже если вам кажется, что именения были незначительными!  
  
После коммита, стоит проверять состояние проекта уже знакомой нам командой git status.

Если мы видим в ответ сообщение:

nothing to commit, working tree clean

значит мы всё сделали правильно и можем продолжать работать над проектом.  
  
На текущий момент, мы создали локальный репозиторий и отправили в него коммит.

Далее мы научимся создавать удалённые репозитории, для совместной работой над проектами.  
  
Для примера, создадим новый приватный репозиторий на своём GitHub по инструкции в лекции.

После создания удалённого репозитория на GitHub, нам нужно выполнить в терминале две команды, которые даются при создании репозитория.

Пример:

git remote add origin git@github.com:ВАШЕ\_ИМЯ/НАЗВАНИЕ\_РЕПОЗИТОРИЯ.git

git push -u origin master

remote add origin - подключение удалённого репозитория, задавая ему название origin.

push -u origin master - загрузка файлов в master-ветку верпозитория origin.

После выполнения данных команд, можно обновить страницу удалённого репозитория и увидеть, что наш файл благополучно отправился.

В будещем, для выгрузки измененных файлов после коммита, достаточно использовать команду

git push

## Как клонировать проект из git

Предположим, что нам предстоит работать над каким-то проектом. Первым делом нам нужно склонировать себе репозиторий данного проекта.

В каждом репозитории есть кнопка Clone or download, нажав на которую мы можем выбрать работу через SSH или HTTPS.

Нажимаем Use SSH и копируем оттуда строку следующего вида:

git@github.com:ИМЯ/НАЗВАНИЕ\_РЕПОЗИТОРИЯ.git

Данную строку мы должны будем использовать в терминале:

git clone git@github.com:ИМЯ/НАЗВАНИЕ\_РЕПОЗИТОРИЯ.git

clone - команда клонирования удалённого репозитория к себе на ПК.

После применения команды git clone, репозиторий будет скопирован в папку с названием данного репозитория, внутри директории из которой была запущена команда.

## Как получать обновленные файлы проекта из git

Для теста, предварительно внесем изменения в наш файл readme.md через сайт.

Теперь, если внесем изменения в файл readme.md, который находится в нашем локальном репозитории (на нашем ПК), мы не сможем загрузить его командой git push в удалённый репозиторий.

Всё дело в том, что удалённый файл readme.md уже был кем-то изменён и мы должны предварительно загрузить его изменения к себе в локальный репозиторий, после чего уже сможем отправить обновлённый нами файл.

Для получения данных с удалённого репозитория, мы используем команду:

git pull

Однако, при попытке загрузить файлы, мы столкнёмся с новой проблемой - конфликт. То есть изменения в наших файлах будут конфликтовать между собой.  
  
При столкновении с конфликтом, мы обязаны его решить.  
  
Чтобы решить конфликтную ситуацию, мы откроем наш проект уже знакомой нам командой:

code .

И постараемся устранить конфликт.   
  
После устанения конфликтов, мы добавляем файлы в проект, делаем коммит и отправляем файлы на удалённый репозиторий:

git add .

git commit -m 'resolve conflicts'

git push

## Что мы можем делать в GitHub

На GitHub мы можем посмотреть историю коммитов проекта.

Для этого мы заходим в наш репозиторий и нажимаем ссылку n commits, где n - число коммитов.

Каждый коммит имеет свой уникальный хеш.  
  
Отслеживать свою активность на git.

Таблица Contribution отображает количество ваших коммитов за год.  
  
И всегда помните мудрость от Будды:

Боль работы с гитом неизбежна, страдания - по выбору.

### Известные проблемы и их решения

Проблема:  
Git постоянно спрашивает логин и пароль и выдаёт ошибку The authenticity of host 'github.com (числа)' can't be established.  
Решение:  
Примените в терминале команду:

ssh-keyscan github.com >> ~/.ssh/known\_hosts