# **GIT**

**GitHub** — крупнейший[[7]](https://ru.wikipedia.org/wiki/GitHub#cite_note-7) [веб-сервис](https://ru.wikipedia.org/wiki/%D0%92%D0%B5%D0%B1-%D1%81%D0%B5%D1%80%D0%B2%D0%B8%D1%81) для [хостинга](https://ru.wikipedia.org/wiki/%D0%A5%D0%BE%D1%81%D1%82%D0%B8%D0%BD%D0%B3) [IT-проектов](https://ru.wikipedia.org/wiki/IT) и их совместной разработки.

Веб-сервис основан на системе контроля версий [Git](https://ru.wikipedia.org/wiki/Git) и разработан на [Ruby on Rails](https://ru.wikipedia.org/wiki/Ruby_on_Rails)[[8]](https://ru.wikipedia.org/wiki/GitHub#cite_note-8) и [Erlang](https://ru.wikipedia.org/wiki/Erlang) компанией GitHub, Inc (ранее Logical Awesome)[[9]](https://ru.wikipedia.org/wiki/GitHub#cite_note-9). Сервис бесплатен для проектов с [открытым исходным кодом](https://ru.wikipedia.org/wiki/%D0%9E%D1%82%D0%BA%D1%80%D1%8B%D1%82%D0%BE%D0%B5_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%BD%D0%BE%D0%B5_%D0%BE%D0%B1%D0%B5%D1%81%D0%BF%D0%B5%D1%87%D0%B5%D0%BD%D0%B8%D0%B5) и (с 2019 года) небольших частных проектов, предоставляя им все возможности (включая [SSL](https://ru.wikipedia.org/wiki/SSL)[[10]](https://ru.wikipedia.org/wiki/GitHub#cite_note-10)), а для крупных корпоративных проектов предлагаются различные платные тарифные планы[[11]](https://ru.wikipedia.org/wiki/GitHub#cite_note-11).

Репозито́рий — место, где хранятся и поддерживаются какие-либо данные. Чаще всего данные в репозитории хранятся в виде файлов, доступных для дальнейшего распространения по сети. Существуют репозитории для хранения программ, написанных на одном языке или предназначенных для одной платформы.

# **Git или Гит** — система контроля и управления версиями файлов.

# **GitHub или Гитхаб** — веб-сервис для размещения репозиториев и совместной разработки проектов.

# **Репозиторий Git** — каталог файловой системы, в котором находятся: файлы конфигурации, файлы журналов операций, выполняемых над репозиторием, индекс расположения файлов и хранилище, содержащее сами контролируемые файлы.

# **Локальный репозиторий** — репозиторий, расположенный на локальном компьютере разработчика в каталоге. Именно в нём происходит разработка и фиксация изменений, которые отправляются на удалённый репозиторий.

# **Удалённый репозиторий** — не тот, который удалили, а репозиторий, находящийся на удалённом сервере. Это общий репозиторий, в который приходят все изменения и из которого забираются все обновления.

# **Форк (Fork)** — копия репозитория. Его также можно рассматривать как внешнюю ветку для текущего репозитория. Копия вашего открытого репозитория на Гитхабе может быть сделана любым пользователем, после чего он может прислать изменения в ваш репозиторий через пулреквест.

# **Обновиться из апстрима** — обновить свою локальную версию форка до последней версии основного репозитория, от которого сделан форк.

# **Обновиться из ориджина** — обновить свою локальную версию репозитория до последней удалённой версии этого репозитория.

# **Клонирование (Clone)** — скачивание репозитория с удалённого сервера на локальный компьютер в определённый каталог для дальнейшей работы с этим каталогом как с репозиторием.

# **Ветка (Branch)** — это параллельная версия репозитория. Она включена в этот репозиторий, но не влияет на главную версию, тем самым позволяя свободно работать в параллельной. Когда вы внесли нужные изменения, то вы можете объединить их с главной версией.

# **Мастер или Main (Master)** — главная или основная ветка репозитория.

# **Коммит (Commit)** — фиксация изменений или запись изменений в репозиторий. Коммит происходит на локальной машине. Не путать с комментарием.

# **Пул (Pull)** — получение последних изменений с удалённого сервера репозитория.

# **Пуш (Push)** — отправка всех неотправленных коммитов на удалённый сервер репозитория.

# **Пулреквест (Pull Request)** — запрос на слияние форка репозитория с основным репозиторием. Пулреквест может быть принят или отклонён вами, как владельцем репозитория.

# **Мёрдж (Merge)** — слияние изменений из какой-либо ветки репозитория с любой веткой этого же репозитория. Чаще всего слияние изменений из ветки репозитория с основной веткой репозитория.

# **Кодревью** — процесс проверки кода на соответствие определённым требованиям, задачам и внешнему виду.

**Snapshot** - снимок репозитория или его части.

# 

**делаем репозиторий в консоли**

(чтобы сделать репозиторий с командами, при создании репозитория НЕ НУЖНО указывать add README и gitignore)

Связь репозитория с папкой:

echo "# test123" >> README.md (записываем в файл данные)

git init (инициализируем гит)

git add README.md (добавляем файлы)

git commit -m "first commit" (пишем комментарий)

git branch -M main (изменяем название на мэйн)

git remote add origin git@github.com:iohichu/test123.git (добавляем к отслеживанию)

git push -u origin main (пушим в мейн)

чтобы удалить файл:

git rm \_имя файла\_

git commit -m "deleting file png"

git push -u origin main

**Создание репозитория в IDEA**

1. Подготовить папку с проектом
2. записать в нее файлы
3. создать проект в этой папке
4. открыть терминал в IDEA (alt+f12 / option +f12)
5. написать в терминале git init (инициализация папки)
6. Создать репозиторий с таким же названием, как и папка в github
7. зайти в него и в CODE поставить SSH и оттуда взять ссылку
8. Сделать необходимые изменения в проекте
9. Нажать push and commit (зеленая птичка сверху справа - commit , чуть правее зеленая стрелка - push)
10. ввести ссылку из github на проект
11. Обновить страницу в github

Запись репозитория из GitHub:

1. Создаем репозиторий на gitHub с любым названием
2. зайти в него и в CODE поставить SSH и оттуда взять ссылку
3. В ide нажать file>new>project from Version control
4. ввести ссылку
5. сделать изменения в проекте
6. Нажать push and commit (зеленая птичка сверху справа - commit , чуть правее зеленая стрелка - push)