**Git и Github для начинающих**

Git - это система контроля версий, позволяющая эффективно управлять историей исходного кода. Любые изменения которые ты вносишь в проект могут быть сохранены с помощью Git. Ты можешь вернуться к любым ранее сохраненным версиям. Без Git пришлось бы создавать копии проекта, что было бы проблемой при увеличении объема кода в приложении. На текущий момент Git — стандарт для разработки.

**Git и GitHub**

Для начинающих может быть непонятна разница между этими терминами.

**Git** — система контроля версий. Технология, которую можно скачать на компьютер.

**Github** — сервис, который позволяет работать с твоими Git проектами. Это называется репозитории. Помимо Github существуют другие сервисы (например Bitbucket, GitLab).

**Как установить Git**

Это бесплатная технология и скачать ее можно тут:

<https://git-scm.com/downloads>

Запускается через командную строку или терминал с команды git

**Основные команды**

**git init**

Позволяет проинициализировать репозиторий в текущей папке

**git remote**

**git remote add origin** [**https://github.com HYPERLINK "https://github.com/\*\*\*/\*\*\*.git"/\*\*\* HYPERLINK "https://github.com/\*\*\*/\*\*\*.git"/ HYPERLINK "https://github.com/\*\*\*/\*\*\*.git"\*\*\* HYPERLINK "https://github.com/\*\*\*/\*\*\*.git".git**](https://github.com/***/***.git)— привязать локальный с удалённым на сайте

**git remote remove origin** — удалить привязку локального и удалённого репы

**git remote** — проверить связь репозитория

**git status**

Показывает текущий статус

git add

Отслеживает изменения файлов

**git add index.html** — добавляет index.html

**git add .** — добавляет все файлы

git commit

Сохраняет изменения в коммит

**git commit -m 'commit message'** — создает коммит с сообщением

**git commit -am 'commit message'** — добавить файлы и закоммитить

git branch

Работа с ветками в репозитории

**git branch** — показывает список веток

**git branch branch-name** — создает новую ветку branch-name

**git branch -D branch-name** — удаляет ветку branch-name

git checkout

Переключается на другую ветку

**git checkout branch-name** — переключается на последний коммит в ветке branch-name

**git checkout -b branch-name** — создает и переключается на ветку branch-name

git merge

Совмещает текущую ветку с выбранной

**git merge branch-name** — совмещает текущую ветку с branch-name

git config

Конфигурация и параметры git

**git config --global user.name** — Показывает имя пользователя

**git config --global user.name 'new user'** — Изменяет имя пользователя

**git config --global user.email** — Показывает email пользователя

**git config --global user.email 'test@mail.ru'** — Изменяет email пользователя

**git push**

Заливает текущие локальные коммиты в удаленный репозиторий

**git push -u origin master** — отправить изменения в репозиторий на Github

**git pull**

Забирает изменения с удаленного репозитория в локальный

**git pull origin master** — получает изменения из мастер ветки например в dev ветку

**git clone**

Клонирует проект с удаленного репозитория

**git log** — посмотреть логи коммитов

**ls** — посмотреть файлы в репозитории

**git ls-files** — посмотреть треканые файлы