Git и его функции

Для начала работы необходимо подключить для проверки совместимости данных Git и сервера (GitHub) ваши данные. Данные в Git и на GitHub должны совпадать! Подключение можно сделать так:

1. git config --global user.name "Ваш\_логин" – подключит в логи ваш логин для подключения к удалённому серверу (GitHub).
2. git config --global user.email "Ваша\_почта" – подключит в логи вашу почту для подключения к удалённому серверу (GitHub).

Для того, чтобы закинуть репозиторий на удалённый сервер (GitHub) в первый раз нужно проделать эти операции:

1. git remote add origin https://github.com/Имя\_аккаунта/Название\_репозитория.git
2. git push -u origin master

Действия с коммитами и репозиториями можно выполнить через эти команды:

1. git init – создать репозиторий Git в текущей папке
2. git status – выведет изменения после последнего сохранения коммита.
3. git log – выведет существующие коммиты.
4. git commit -m “Название\_коммита” – создаст новый коммит с неопределённым названием.
5. git add (<Имя\_файла> или .) – добавит изменения, которые Git нашёл (либо для определённого файла, либо для всех – как с точкой).
6. git commit -a --amend – дозапишет изменения в последний коммит.
7. git reset HEAD Название\_удалённого\_файла – восстановит удалённый файл.
8. git reset --hard HEAD^ – удалит коммит и его изменения.
9. git reset --soft HEAD^ – отменит коммит (удалит коммит, оставив изменения)
10. git rm Название\_удалённого\_файла – подтвердит удаление файла.

В дальнейшем для соединения Git с сервером используются команды:

1. git push – переносит данные с Git на сервер
2. git pull – переносит данные с сервера на Git
3. git push –f origin имя\_ветки – переносит данные с Git на сервер, определяя неопределённую ветку (выполняется в случае, если команда 1 не работает, обычно в Git стандартная ветка – master).

Ветка – что–то нечто нового репозитория, но только внутри текущего репозитория... Ветки можно связывать, чтобы какие–то файлы из разных веток работали и контактировали между друг другом. Ветка позволяет продолжать работать, не заморачиваясь с создания нового репозитория, но только ВЫ можете связывать файлы разных веток. А без соединения веток, они не смогут контактировать, следовательно, можно не бояться сломать билд, соответственно, можно провести какую–нибудь фичу, в которой вы не уверены.  
Ветка – что–то вроде новой темы в тексте. Темы, то есть ветки объединены только текстом, то есть репозиторием. И без использования слов, соединяющих темы, их не соединить... Только ветки должны иметь отношение к репозиторию, а не просто валяться, как чужие. Вот какие существуют команды, для работы с ветками:

1. git branch название\_ветки – создаёт новую ветку с неопределённым названием
2. git branch – выводит список имеющихся в репозитории веток.
3. git checkout название\_существующей\_ветки – переключает вас на ветку, для отдельной работы с ней.
4. git branch –d название\_существующей\_ветки – удаляет существующую ветку