

GIT - Комманды

1. Перемещение по папкам

Прежде чем ты подготовишься к работе или отправишь изменения в репозиторий, надо будет попасть в папку с которой будут проводиться все операции:

```
cd [Полный путь до папки] [Пишем адрес до директории или перетягиваем папку в терминал]
cd .. [Выйти на директорию выше]
```

2. Скачать проект с github

Если тебя пригласили в проект, то для начала надо скачать с репозитория исходники (если они есть) и создать локальную копию, чтобы потом заливать изменения из своей локалки:

```
cd [Полный путь до папки] [Перемещаемся в корень гитхаба]
git clone [Ссылка SSH ключа] [Скачать репозиторий на компьютер, для дальнейшей работы]
```

3. История изменений

Просмотр изменений пригодится, если над проектом работаешь не в одиночку и кто-то тоже заливает изменения. Поэтому сначала нужно убедиться, что твоя локальная версия актуальна и никто ничего не пушил:

```
git log [Посмотреть лог последних изменений (Нужно ли мне обновляться и делать пулл реквест?)]
q [Выйти из просмотра лога]
git pull [Обновить локальные файлы, если в репо кто-то что-то вливал]
```

4. Пуш изменений в репозиторий

Ориентировочный порядок действий, чтобы залить изменения:

```
cd [полный путь до папки] [Перемещаемся в нужную папку]
git status [Проверяем статус изменяемых файлов: всё ли в списке или что-то забыл сделать?]
git add . [Добавить все изменения в репозиторий]
git commit -m '' '' [Написать комментарий к изменениям (писать нужно между « „)]
git push [Отправить изменения в репозиторий]
```

5. Работа с ветками

Ветки нужны для отдельных фиш, которые не должны мешать остальным. Это очень удобно и гибко:

```
git checkout -b [Имя твоей ветки] [Если необходимо создать новую ветку]
git branch [Если зашел в проект, но не знаешь в какой сейчас ты ветке]
git checkout [Имя ветки] [Перейти в желаемую ветку]
git checkout [Имя ветки] [Перейти в желаемую ветку]
git push --set-upstream origin [Имя ветки] [Запустить изменения в нужную ветку]
```

6. Отменить изменения

Бывает так, что пушить ничего не надо а ты уже добавил коммит. Тогда можно сделать как-то так:

```
git reset [Отменить добавление файлов, будто ты не делал „git add .“]
git reset --soft HEAD^ [Отменить последний коммит, но сохранить изменения]
```

Revision #2

Created 31 December 2022 22:42:59 by Zerr0

Updated 31 December 2022 22:46:29 by Zerr0