<https://online.sbis.ru/shared/disk/ddfd5d71-a391-4fce-a630-5df8f4fa21e8>

# Вопросы по Git

## Как можно найти на локальном компьютере склонированный репозиторий?

Куда склонировал репозиторий там и будет

## Ваша команда push отклонена git-ом. Почему такое произошло и ваши дальнейшие действия.

В моем случае из-за не соблюдения стилистики кода.

Также push сработает в том случае, если вы клонировали с сервера, на котором у вас есть права на запись, и если никто другой с тех пор не выполнял команду push. Если вы и кто-то ещё одновременно клонируете, затем он выполняет команду push, а после него выполнить команду push попытаетесь вы, то ваш push точно будет отклонён. Вам придётся сначала получить изменения и объединить их с вашими и только после этого вам будет позволено выполнить push.

## Для чего служат pre-push хуки на git.sbis.ru?

В

## Объясните, что означает origin в команде git push origin master? Расскажите об отличии удаленных веток от локальных. Какими командами можно обновить локальные ветки? В чем отличие этих команд?

**origin** – это удаленный репозиторий (откуда делали clone).

**git fetch origin -** извлекает все наработки, отправленные на удаленный сервер, но не сливает их с какими-либо вашими наработками и не модифицирует то, над чем вы работаете в данный момент. Вам необходимо вручную слить эти данные с вашими, когда вы будете готовы.

**git pull** - автоматически получит изменения из удалённой ветки master и сольет их со своей текущей веткой master

**git clone <repo>** - получить репо как он есть на github

## Какие состояния файла могут быть в git? С помощью каких команд можно изменить состояние файла?



**git add <file>** - добавить в индекс не отслеживаемый файл (сделать Staged), либо добавить в индекс измененный файл (сделать Staged).

**git reset HEAD <file> -** убрать файл из индекса

**git checkout -- <file>** - отменить изменения в файле до состояния последнего комита (если изменений еще нет в индексе)

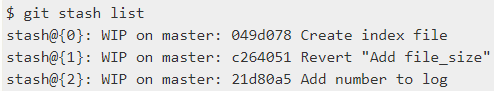
**git commit –** зафиксировать изменения в локальном репо.

**git commit –amend –** заменит предыдущий комит

## Расскажите про параметры skip, abort и continue у команд git.

## Что делает команда $ git stash? Как посмотреть список всех сохраненных состояний в стеке? Как применить последнее изменение назад? Как удалить изменения, «спрятанные» командой $ git stash? Как создать ветку и сразу же применить «спрятанные» изменения?

* Делает вот что [https://git-scm.com/book/ru/v2/%D0%98%D0%BD%D1%81%D1%82%D1%80%D1%83%D0%BC%D0%B5%D0%BD%D1%82%D1%8B-Git-%D0%9F%D1%80%D0%B8%D0%BF%D1%80%D1%8F%D1%82%D1%8B%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5-%D0%B8-%D0%BE%D1%87%D0%B8%D1%81%D1%82%D0%BA%D0%B0#r\_git\_stashing](https://git-scm.com/book/ru/v2/Инструменты-Git-Припрятывание-и-очистка" \l "r_git_stashing)
* **Чтобы посмотреть список припрятанных изменений**, вы можете использовать git stash list:



* Вы **можете применить только что припрятанные изменения**, используя команду, указанную в выводе исходной команды: git stash apply. Если вы хотите применить одно из предыдущих припрятанных изменений, вы можете сделать это, используя его имя, вот так: git stash apply stash@{2}. Если вы не укажете имя, то Git попытается восстановить самое последнее припрятанное изменение.
* Команда apply только пытается восстановить припрятанные наработки — при этом они останутся в хранилище. **Для того, чтобы удалить их,** вы можете выполнить git stash drop, указав имя удаляемых изменений:



* Вы можете выполнить команду git stash branch, **которая создаст для вас новую ветку, перейдёт на коммит, на котором вы были, когда прятали свои наработки, применит на нём эти наработки** и затем, если они применились успешно, удалит эти припрятанные изменения:

## Какой командой можно изменить описание, указанное при создании коммита? Что следует учесть при использовании этой функции?

git commit —amend -m "New commit message."