[https://docs.google.com/document/d/19k\_b\_KAqh8ZVrI1ec1NwH0zrdntefcxG-QF3HavUdfc/edit#](https://docs.google.com/document/d/19k_b_KAqh8ZVrI1ec1NwH0zrdntefcxG-QF3HavUdfc/edit%23)

<https://git-scm.com/docs/gittutorial>

<https://code.visualstudio.com/docs/editor/github>

<https://zencod.ru/articles/first-step-git/>

<https://learngitbranching.js.org/?locale=ru_RU>, ответы - <https://gist.github.com/CraftingGamerTom/d334740303b647339b00875d27d38d34>

# **1. Временно переключиться на другой коммит**

Если вам нужно просто переключиться на другой коммит, чтобы, например, посмотреть на его содержимое, достаточно команды [git checkout](https://ru.stackoverflow.com/tags/git-checkout/info):

git checkout aaaaaa

(wt)

(i)

A - B - C - D

↑ ↑

(HEAD) master

Сейчас репозиторий находится в состоянии «detached HEAD». Чтобы переключиться обратно, используйте имя ветки (например, master):

git checkout master

# **2. Переключиться на коммит и продолжить работу с него**

Если вы хотите продолжить работу с другого коммита, вам понадобится новая ветка. Можно переключиться и создать ее одной командой:

git checkout -b имя-новой-ветки aaaaaa

(wt)

(i)

A - B - C - D

↑ ↑

new master

(HEAD)

# **3. Удалить изменения в рабочей области и вернуть ее к состоянию как при последнем коммите.**

Начальное состояние:

(i) (wt)

A - B - C - D - ? - ?

↑

master

(HEAD)

## **3.1 Безопасно — с помощью кармана (stash)**

### **3.1.1 Только неиндексированные**

Можно ~~удалить~~ прикарманить только те изменения, которые еще не были индексированы (командой add):

git stash save --keep-index

Конечное состояние:

(wt)

(i)

A - B - C - D - ? ?

↑ ↑

master stash{0}

(HEAD)

### **3.1.2 Индексированные и нет**

Эта команда отменяет все индексированные и неиндексированные изменения в рабочей области, сохраняя их в карман (stash).

git stash save

Конечное состояние:

(wt)

(i)

A - B - C - D ?

↑ ↑

master stash{0}

(HEAD)

**Восстановление несохраненных изменений**: легко и просто.

git stash apply

Если stash совсем не нужен, его можно удалить.

# удалить последнюю запись кармана

git stash drop

[Подробнее про использование stash](https://ru.stackoverflow.com/a/153503/181472).

После этого восстановить изменения всё ещё можно, но сложнее: [How to recover a dropped stash in Git?](https://stackoverflow.com/q/89332/2790048)

## **3.2 Опасный способ**

**Осторожно!** Эта команда **безвозвратно удаляет несохраненные текущие изменения из рабочей области и из индекса** Если они вам все-таки нужны, воспользуйтесь [git stash](https://ru.stackoverflow.com/tags/git-stash/info).

**Восстановление несохраненных изменений**: неиндексированные потеряны полностью, но [вы можете восстановить то, что было проиндексировано](https://ru.stackoverflow.com/a/424384/181472).

Здесь мы будем использовать [git reset --hard](https://ru.stackoverflow.com/tags/git-reset/info)

Выполняем:

git reset --hard HEAD

Конечное состояние:

(wt)

(i)

A - B - C - D - х - х

↑

master

(HEAD)

# **4. Перейти к более раннему коммиту в текущей ветке и удалить из нее все последующие (неопубликованные)**

**Осторожно!** Эта команда переписывает историю Git-репозитория. **Если вы уже опубликовали ([git push](https://ru.stackoverflow.com/tags/git-push/info)) свои изменения, то этот способ использовать нельзя (см.** [**почему**](https://ru.stackoverflow.com/questions/429512/)**)**. Используйте вариант из пункта 5 ([git revert](https://ru.stackoverflow.com/tags/git-revert/info)).

## **4.1 При этом сохранить изменения в индекс репозитория:**

git reset --soft bbbbbb

После этого индекс репозитория будет содержать все изменения от cccccc до dddddd. Теперь вы можете сделать новый коммит (или несколько) на основе этих изменений.

(wt)

(i)

A - B - C - D

↑

master

(HEAD)

## **4.2 Сохранить изменения в рабочей области, но не в индексе.**

git reset bbbbbb

Эта команда просто перемещает указатель ветки, но не отражает изменения в индексе (он будет пустым).

(i) (wt)

A - B - C - D

↑

master

(HEAD)

## **4.3 Просто выбросить изменения.**

**Осторожно!** Эта команда **безвозвратно удаляет несохраненные текущие изменения**. Если удаляемые коммиты не принадлежат никакой другой ветке, то они тоже будут потеряны.

**Восстановление коммитов**: Используйте [git reflog](https://ru.stackoverflow.com/tags/git-reflog/info) и [этот вопрос](https://ru.stackoverflow.com/questions/232455/) чтобы найти и восстановить коммиты; иначе сборщик мусора удалит их безвозвратно через некоторое время.

**Восстановление несохраненных изменений**: неиндексированные потеряны полностью, но [вы можете восстановить то, что было проиндексировано](https://ru.stackoverflow.com/a/424384/181472).

Начальное состояние:

(i) (wt)

A - B - C - D - ? - ?

↑

master

(HEAD)

Выполняем:

git reset --hard bbbbbb

Конечное состояние:

(wt)

(i)

A - B - C - D - х - х

↑

master

(HEAD)

# **5. Отменить уже опубликованные коммиты с помощью новых коммитов**

Воспользуйтесь командой [git revert](https://ru.stackoverflow.com/tags/git-revert/info). Она создает новые коммиты, по одному на каждый отменяемый коммит. Таким образом, если нужно отменить все коммиты после aaaaaa:

# можно перечислить отменяемые коммиты

git revert bbbbbb cccccc dddddd

# можно задать диапазон от более раннего к более позднему (новому)

git revert bbbbbb..dddddd

# либо в относительных ссылках

git revert HEAD~2..HEAD

# можно отменить коммит слияния, указывая явным образом номер предка (в нашем примере таких нет):

git revert -m 1 abcdef

# после этого подтвердите изменения:

git commit -m'детальное описание, что и почему сделано'