# **Undoing Things with Git**

A cheat sheet presented by

## **TOWER** - the best Git client for Mac and Windows



### **Undoing Uncommitted Local Changes**

#### Use "checkout" and "reset" to discard local changes

To discard uncommitted changes in a single file only, use the "git checkout" command:

#### \$ git checkout HEAD <filename>

To restore your complete project / working copy to its last committed state, use "git reset":

#### \$ git reset --hard HEAD

Notice that discarding uncommitted local changes cannot be undone!

## Fixing the Very Last Commit

#### Use "--amend" to change the last commit

To fix a typo in your last commit's message or to add some forgotten changes, use the "--amend" flag to the "git commit" command:

#### \$ git add <forgotten-file.ext>

\$ git commit --amend -m "Correct commit message"

Please only use this on commits that have NOT been published to a remote server, yet!

## Restoring a Previous State of Your Project

#### Use "reset" to roll back to a historic revision

With the "git reset" command, you can roll back your complete project to an old revision:

#### \$ git reset --hard ac023f9a

Your current HEAD branch's history now ends at the chosen revision (and all commits that came afterwards aren't visible anymore).



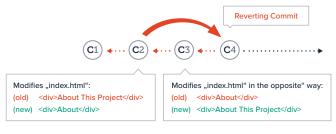
## Undoing / Reverting an Old Commit

#### Use "revert" on a bad commit

With the "git revert" command, you can undo the effects of a certain commit - while leaving any other revisions untouched.

#### \$ git revert ac023f9a

No commits will be deleted. Instead, "git revert" automatically creates a *new* commit that contains changes with the *opposite effect*.



## Restoring a Deleted Branch or Commit

The "reflog" can help you recover a deleted branch or commit

When you were too quick deleting a branch or rolling back to an old commit, have a look at "git reflog" - a journal that protocols every movement of your project's HEAD pointer.

#### \$ git reflog

0165bbb HEAD@{0}: commit: Fix bug #221 f07a9dc HEAD@{1}: checkout: moving from dev to master d536b6d HEAD@{2}: reset: moving to 5255a

With the "lost" commit's hash identifier, you can easily return to any previous state:

\$ git reset --hard d536b6d

#### Learn More

#### Free tutorials and guides on the web

Online book "Learning Version Control with Git" www.git-tower.com/learn/git/ebook/

Video Series on Git and Version Control https://www.git-tower.com/learn/git/videos/

