

### Undo changes in working directory

- Accidental delete
  - Git knows what the file used to look like
  - Git diff can show you what the differences are
  - "*git checkout -- file\_name*"

### Retrieve old versions

- Undo some changes we have made
- remember git log shows all commits
- *git show SHA*
  - Will show the difference between my current version and the commit whose SHA value I am referencing
- *Git checkout SHA -- file\_name*
  - It take the version of file\_name from the SHA reference and stage that in my current version
  - To double check what's different from my current version *git diff --staged*
- 

### Unstage files

- Make the staging tree match the git repository
- "*git reset HEAD file\_name(s)*"
- This command will unstage the change

### Revert a commit

- Made a mistake... we just want to go back in time and bring a commit to the front
- *git revert SHA*
- Gives motivation to make commits more common

### Remove untracked files

- Add some junk files to repository
- *git clean -n*
  - deletes untracked files that live in the working directory that don't match the staging or the repository
- *git clean -f*
  - Removes the files from *git clean -n*
- then what if we unstage what is in staging:
  - remember *git reset HEAD file\_name*
- Then *git clean -n* will tell you that which was in the staging directory will now be removed (since we brought it to the working directory)
- Then *git clean -f* removes that stuffs

### Amend commits

- Amending commits that are already in the repository
- Remember, every single time we commit git sends the code changes and the metadata through an algorithm to generate a SHA value. Remember, the SHA value from the previous commit is also part of this metadata (as it is the parent commit)
- As a beginner, you should probably only change the last commit made since it does not harm the chain
- Say you want to include a new change to your previous commit:
  - "*Git commit --amend -m 'new message'*"
  - This takes whatever was staged in the old commit and brings it back to staging along with the new staged items and then it will commit the new set of changes in staging