

# Introduction to Git

---



**Jeriel Ng (@jerielng)**

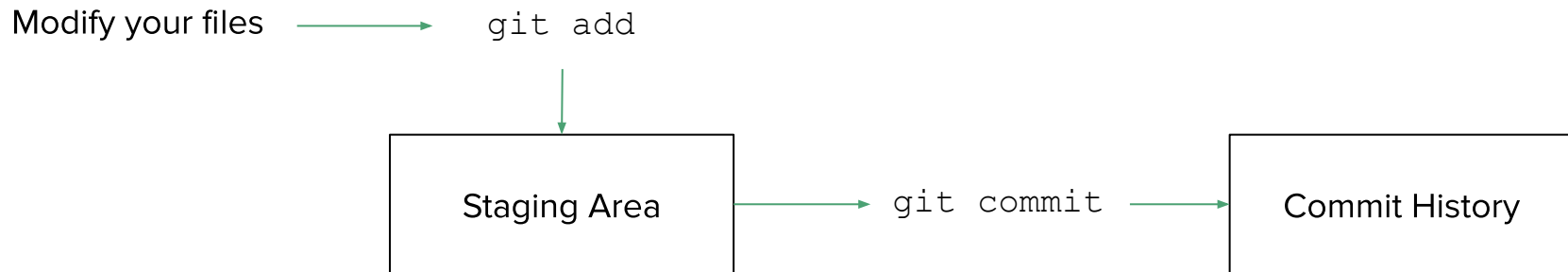
Mobile Software Developer at NCR  
Auburn University 2019

# What is Git?

- Version control system
- Developed by Linus Torvalds (Linux)
- Why is version control important?
  - Logged history of changes
  - Collaborative codebase



# Commit Process



## What is a commit?

A single point of time in your Git history that contains a log of changes

# Commit History: Reviewing Your Changes

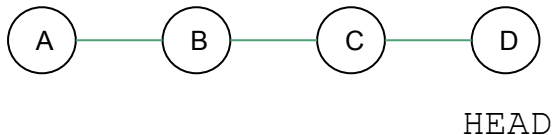
```
185 - public void createInputDialog() {
186 -     AlertDialog.Builder builder = new AlertDialog.Builder(getActivity());
187 -     builder.setMessage(getString(R.string.enter_workout_name));
188 -     final EditText editText = new EditText(getActivity());
189 -     editText.setInputType(InputType.TYPE_CLASS_TEXT);
190 -     builder.setView(editText);
191 -     builder.setPositiveButton(getString(R.string.generate_button_text),
199 + public void createInputDialog(String previousInput) {
200 +     mInputBuilder = new AlertDialog.Builder(getActivity());
201 +     mInputBuilder.setMessage(getString(R.string.enter_workout_name));
202 +     mInputField = new EditText(getActivity());
203 +     mInputField.setInputType(InputType.TYPE_CLASS_TEXT);
204 +     mInputBuilder.setView(mInputField);
205 +     if (!TextUtils.isEmpty(previousInput)) {
206 +         mInputField.setText(previousInput);
207 +     }
208 +     mInputBuilder.setPositiveButton(getString(R.string.generate_button_text),
192 209         new DialogInterface.OnClickListener() {
193 210             @Override
194 211             public void onClick(DialogInterface dialog, int which) {
195 212                 Intent detailIntent = new Intent(getActivity(), WorkoutDetailActivity.class);
196 213                 detailIntent.putExtra(getString(R.string.workout_name_extra),
197 -                     editText.getText().toString());
214 +                     mInputField.getText().toString());
```

Reviewing your diff:

- Additions - green lines
- Deletions - red lines
- Untouched - white lines

# Commit History: Undoing Your Changes

First off,  
what is HEAD?



**HEAD** is the most recent commit of the  
branch you're currently on

## Default

`git reset --hard`

Completely discard  
changes, whether  
staged or not

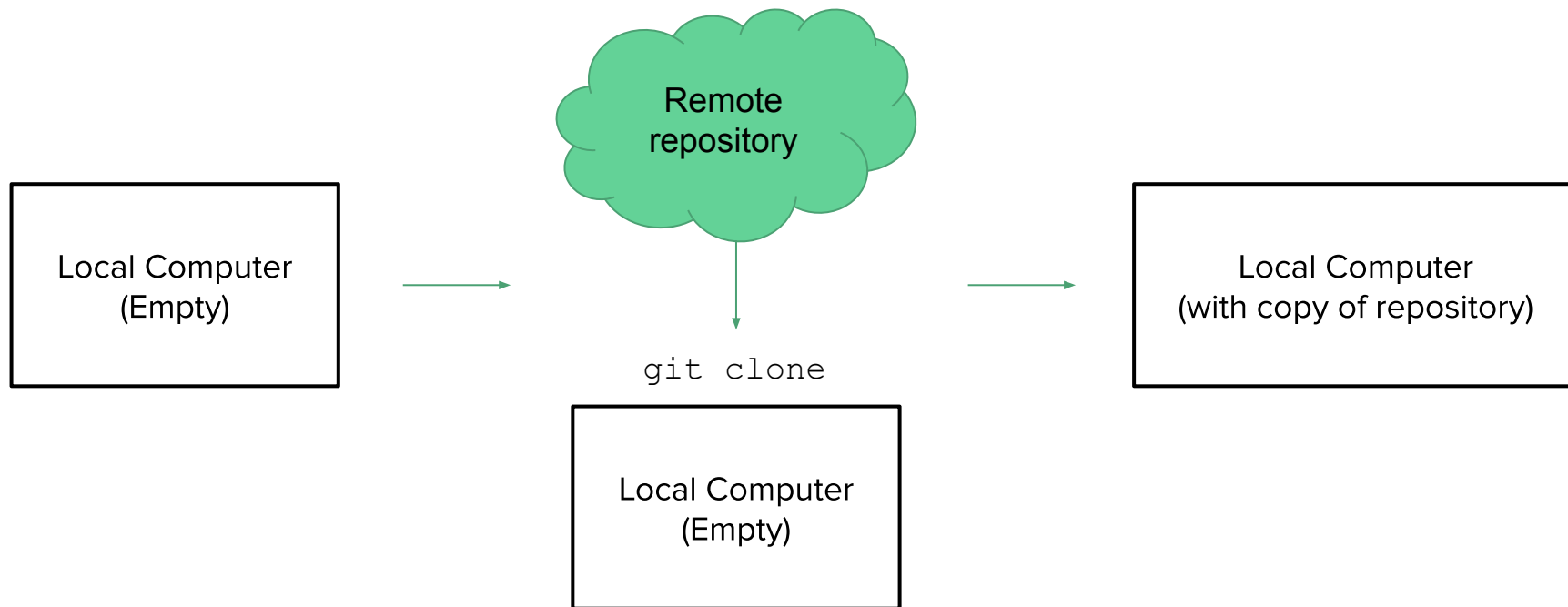
`git reset --mixed`

Removes files from  
staging area, but keeps  
changes

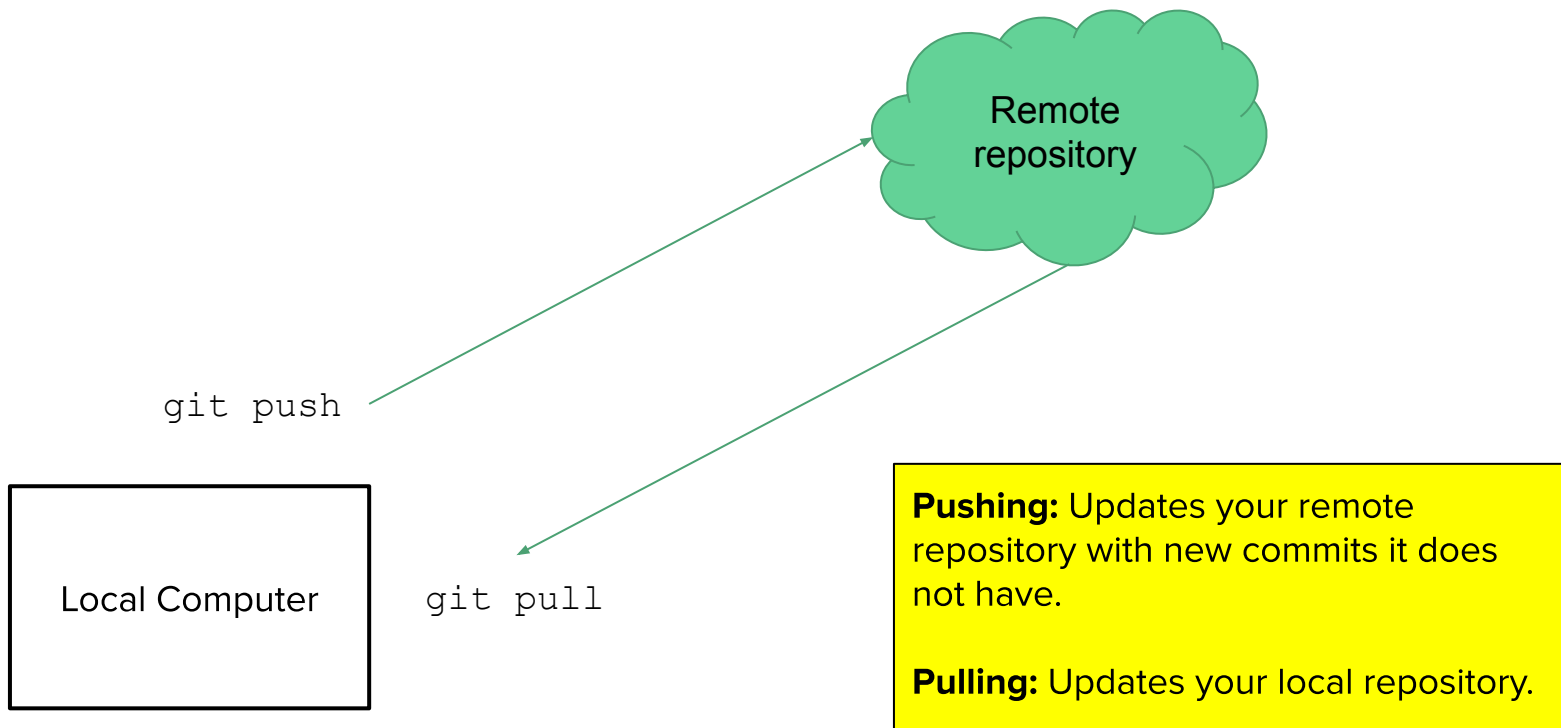
`git reset --soft`

Moves HEAD pointer,  
but keeps changes **in**  
the staging area

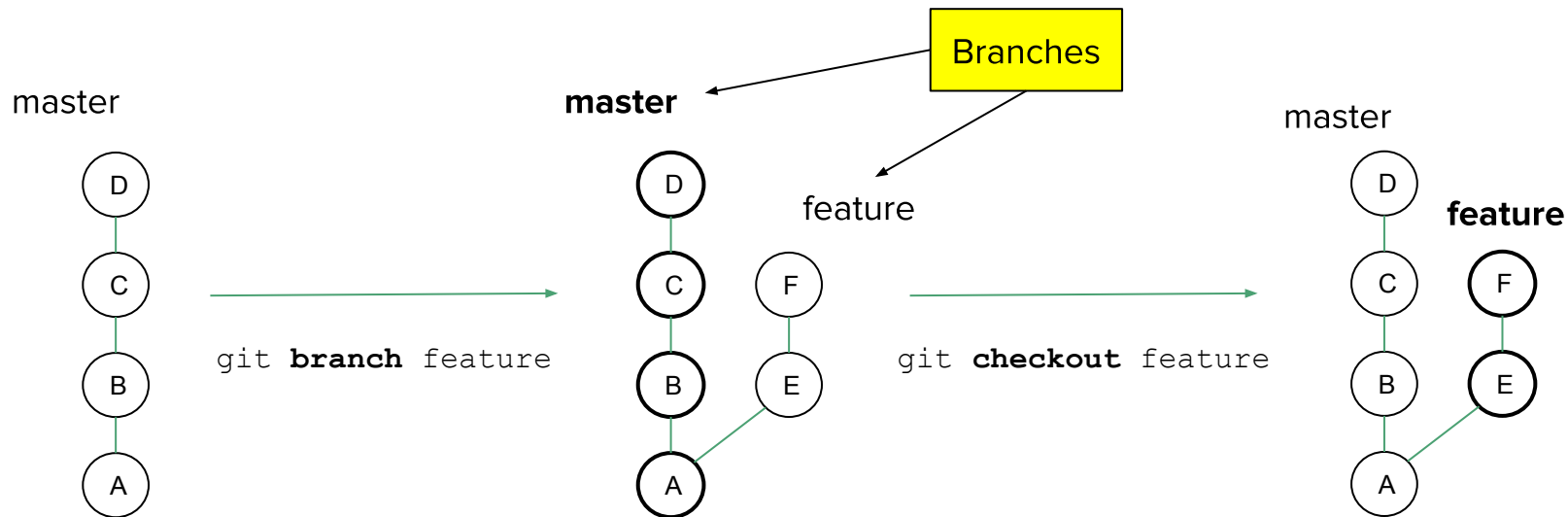
# Working with Remotes: Setup



# Working with Remotes: Updating



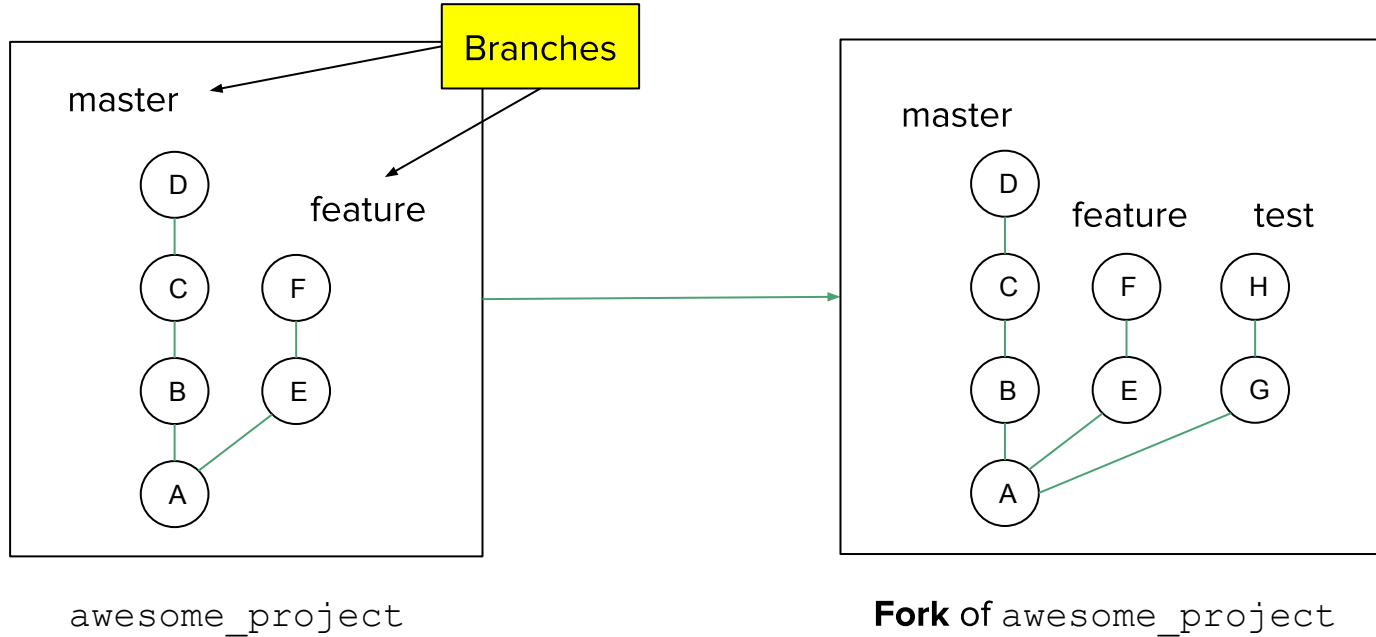
# Alternate Timelines: Branches & Forks



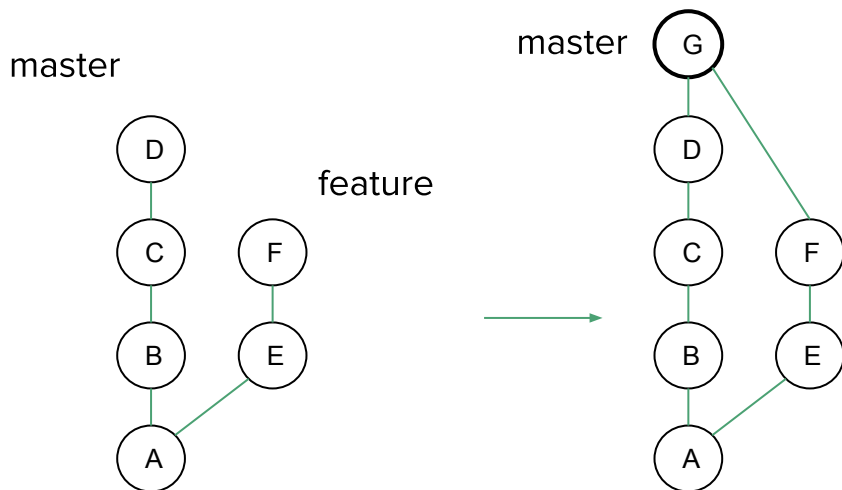
You can also accomplish this with a single line:  
`git checkout -b feature`



# Alternate Timelines: Branches & Forks



# Merging vs. Rebasing

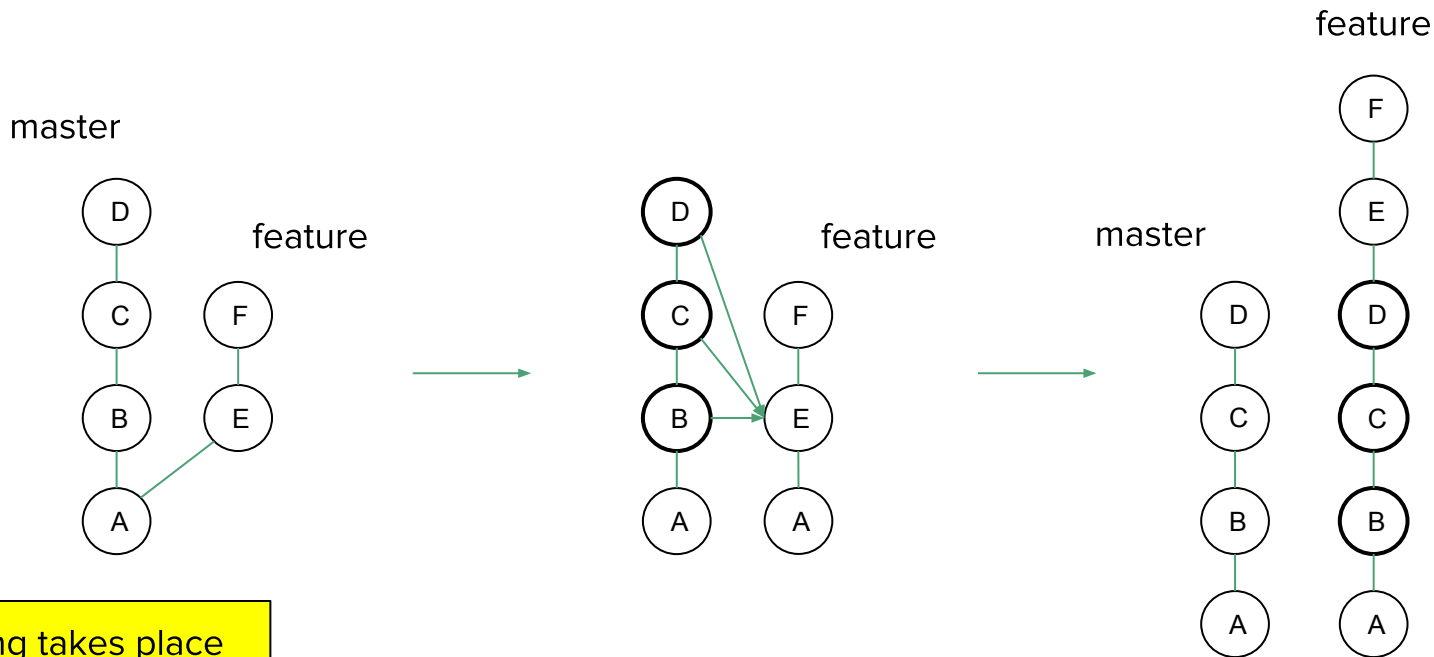


**G** is a newly created “merge commit” to combine the histories of both master and feature

`(master): git merge feature`

Notice merging takes place from the **master** branch, not the **feature** branch

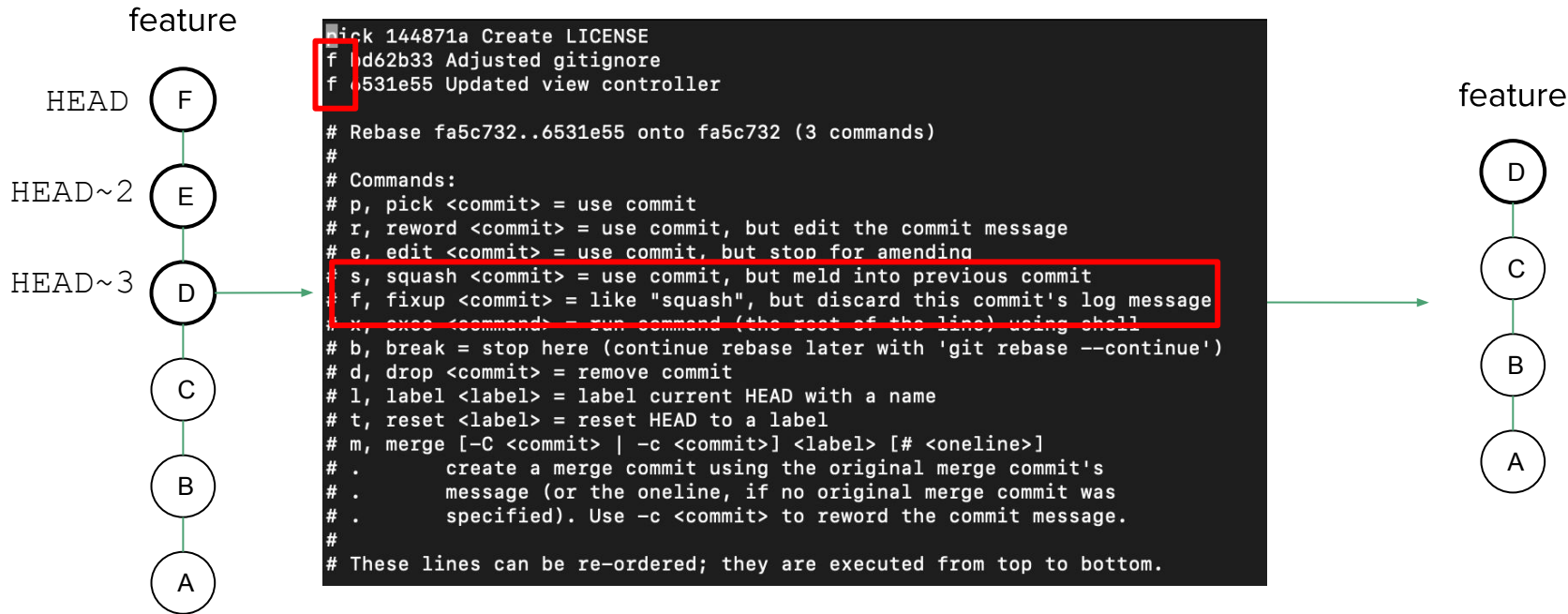
# Merging vs. **Rebasing**



Notice rebasing takes place from the **feature** branch, not the **master** branch

**(feature):** `git rebase master`

# Squashing Your Commits



(feature): `git rebase -i HEAD~3`

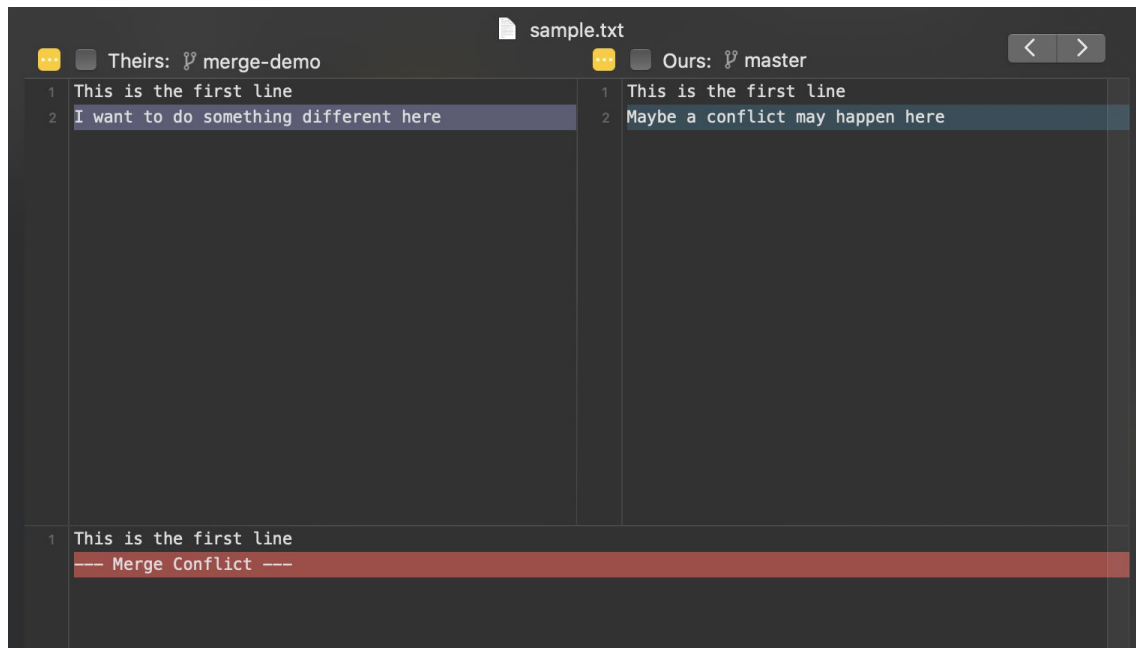
# Merge Conflicts & How to Resolve Them

```
This is the first line  
Maybe a conflict may happen here
```

Coder A's changes

```
This is the first line  
I want to do something different here
```

Coder B's changes



# Merge Conflicts & How to Resolve Them

- Best approach: **Use a text editor or GUI tool** (Especially for big files)
  - Visual Studio Code
  - GitKraken
  - Fork
  - SourceTree
- Have an understanding of **who** changed **what** and **why**
- Run your code afterwards to make sure it compiles

# What is GitHub? I've heard of that



GitLab



Bitbucket

Built on top of Git

- GUI-based platforms
- Features:
  - Repository Hosting
  - Pull Requests
  - CI/CD Integration

# Further Resources

- [Git Reference Manual](#)
- [Interactive Tutorials on Git](#)
- [GitHub Learning Lab](#)
- Desktop GUI tools for Git
  - [Fork](#)
  - [GitKraken](#)
  - [Sourcetree](#)



# Questions

---