Practical Git & GitHub Course

Getting to Known Core Concepts,
Best Practices and Project Collaboration

José Miguel Moreno
hello@josemmo.io

April 3, 2025

1.

Introduction to Git

Stuff you (probably) already know

What is Git?

- Distributed **Version Control System** (VCS)
- Intended **for text files**
- Created in 2005 by Linus Torvalds as an alternative to BitKeeper
- Currently, Linux has 36M+ lines of code

https://www.bitkeeper.org/
https://openhub.net/p/linux/analyses/latest/languages_summary

Filesystem

Stage

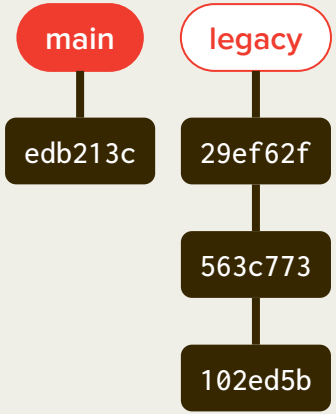
Local Repository

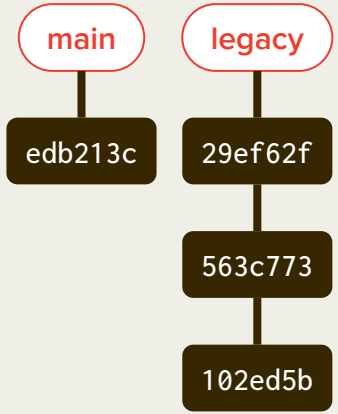
Remote Repository

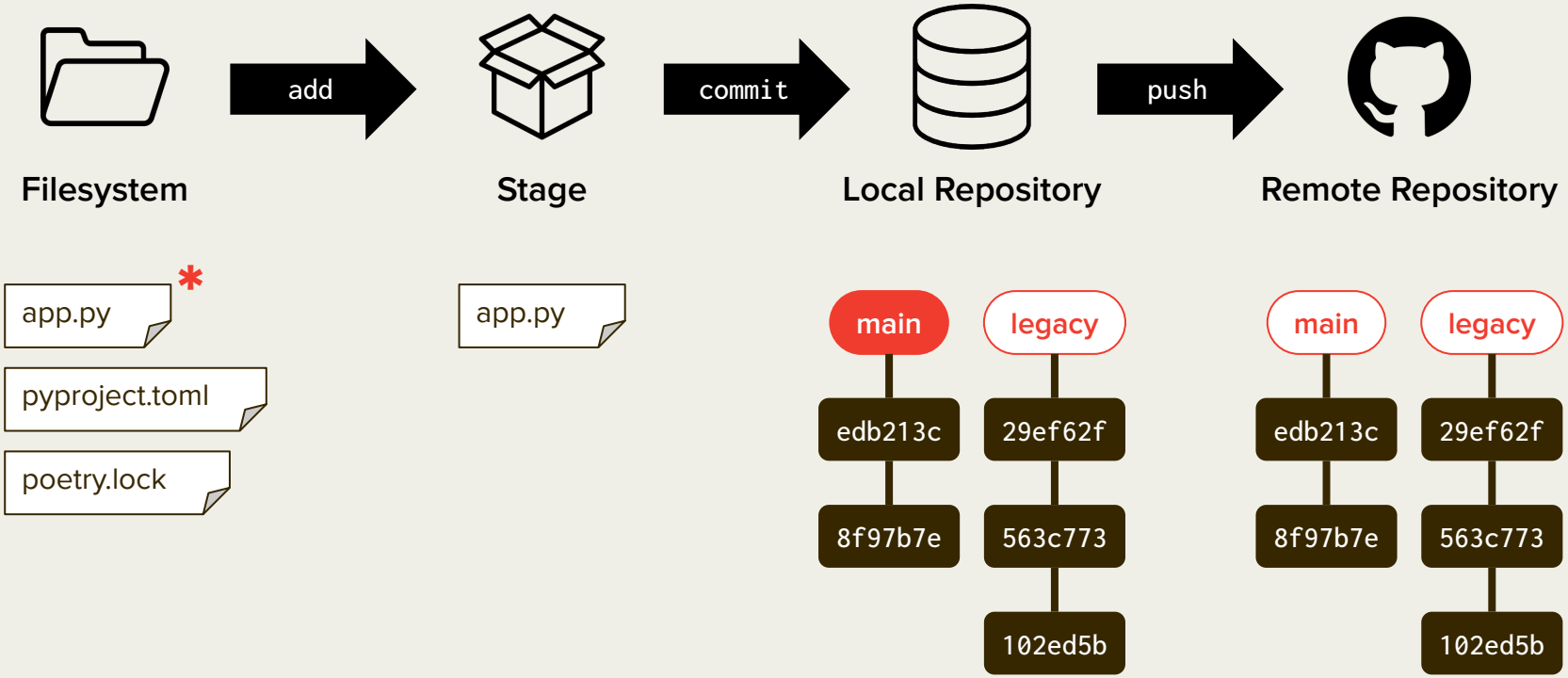
app.py

pyproject.toml

poetry.lock






add
commit
push
Filesystem
Stage
Local Repository
Remote Repository
app.py
*
pyproject.toml
poetry.lock
app.py
main
legacy
edb213c
29ef62f
8f97b7e
563c773
102ed5b
main
legacy
edb213c
29ef62f
8f97b7e
563c773
102ed5b

Pick any GUI, it's Free

desktop.github.com

www.gitkraken.com

sourcetreeapp.com

code.visualstudio.com

>_ Command Line Interface

Sh*t Happens, but You Can Fix It

- **Don’t be afraid** to try new stuff
- Almost all operations in Git are **reversible**
- You can always backup & restore the `.git` directory
- In case of an emergency
    1. Remain chill
    2. DON’T push to upstream
    3. Backup the `.git` directory

2.

Using Git when Working Alone

Just you and no one else to blame

Setting Up the Environment

- `.gitconfig`
- `.gitattributes`
- `.gitignore`
- `.editorconfig`

```
>_ git config --list --show-origin
```

```
>_ git config --global core.autocrlf "false"
```

You can find this file at:

- Windows: `%USERPROFILE%\.gitconfig`
- Linux and macOS: `~/.gitconfig`

Setting Up the Environment

- .gitconfig
- .gitattributes
- .gitignore
- .editorconfig

```
* text=auto eol=lf
```

Setting Up the Environment

.gitconfig

.gitattributes

.gitignore

.editorconfig

```
/.env
__pycache__
/dist
!/dist/index.html
```

Setting Up the Environment

- .gitconfig
- .gitattributes
- .gitignore
- .editorconfig

```
root = true

[*]
charset = utf-8
end_of_line = lf
indent_style = space
indent_size = 4
trim_trailing_whitespace = true
insert_final_newline = true

[*.{tex,bib,yml}]
indent_size = 2
```

https://editorconfig.org/

Commit Message Etiquette

- Start titles **with a verb**: Added, Fixed, Removed
- Use **bullet points** to summarize changes
- When in a multi-component repo, indicate **scope**
- Reference **related issues** or commits

NOTE: This is not a sacred dogma, be rational in your choice of when to follow or ignore it!

🪐 Fixed double spending bug

- Modified PaymentGateway#getPayment() method
- Created Locks util class
- Updated unit tests

> Closes issue #103

So You Screwed Up a Commit

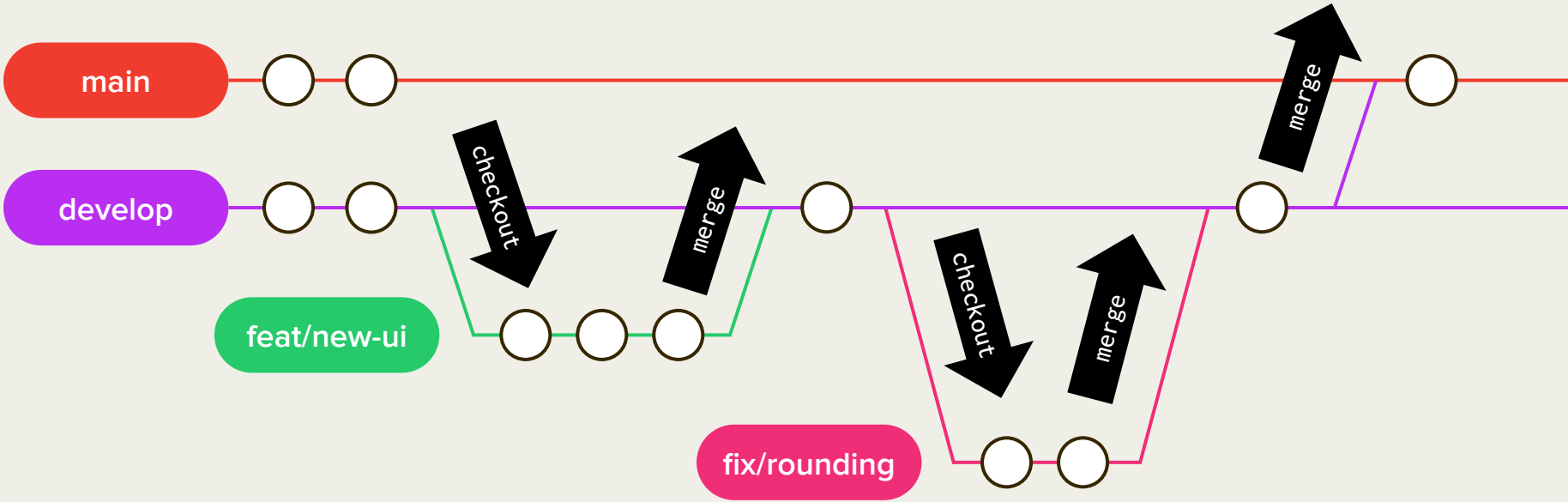
Amend

- **Modifies last commit** directly
- For when you haven't pushed to remote yet
- Use carefully, it's **destructive**
- Ideal for last second changes

Revert

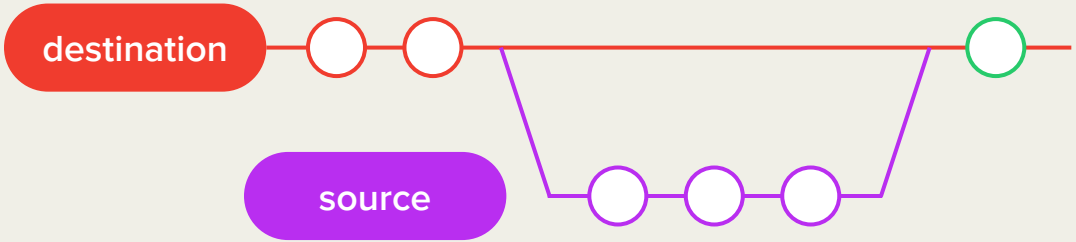
- Creates a **new commit**
- Works even after having pushed to remote
- Preserves history of changes
- Ideal for fixing regression bugs

Gitflow Workflow



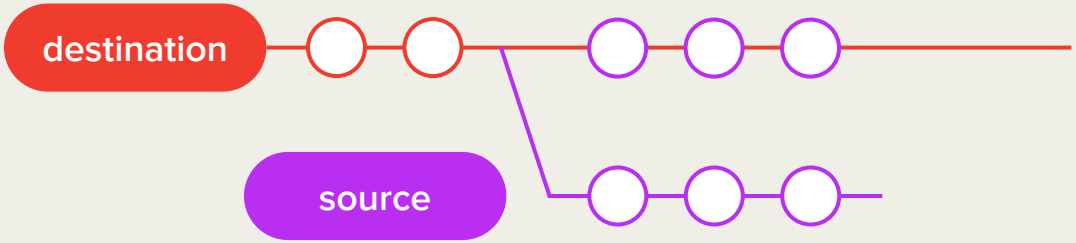
Merge Strategies:

Merge



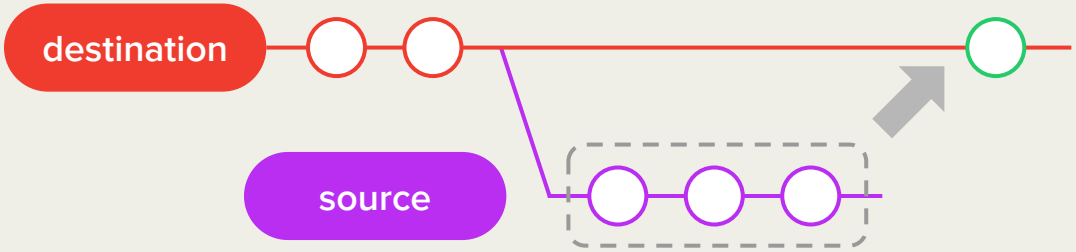
Merge Strategies:

Rebase



Merge Strategies:

Squash



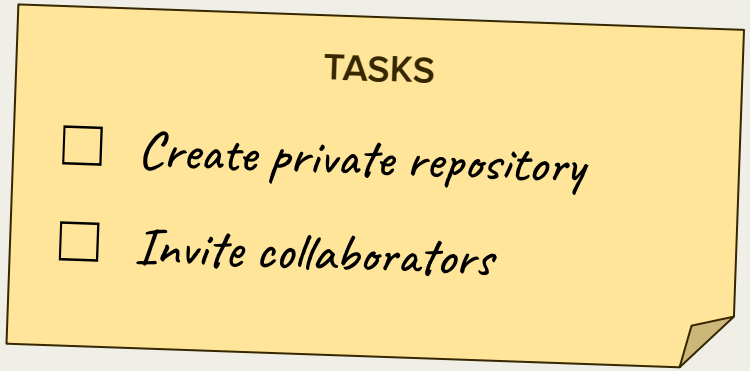
3.

Git for Team Collaboration

Sharing both code and headaches

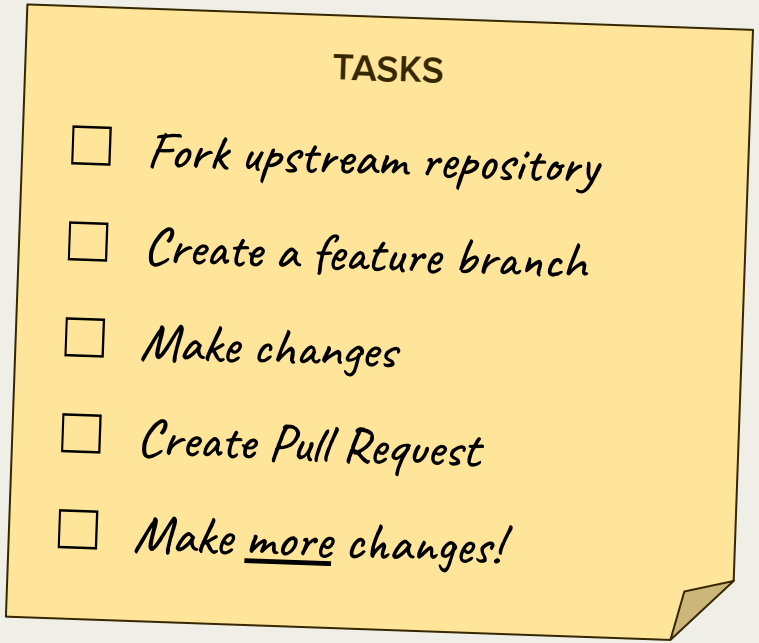
Private Collaboration

- For closed source projects (e.g., LaTeX manuscript)
- For open source projects (reduced set of people have more privileges)
- Only **granted collaborators** can contribute directly



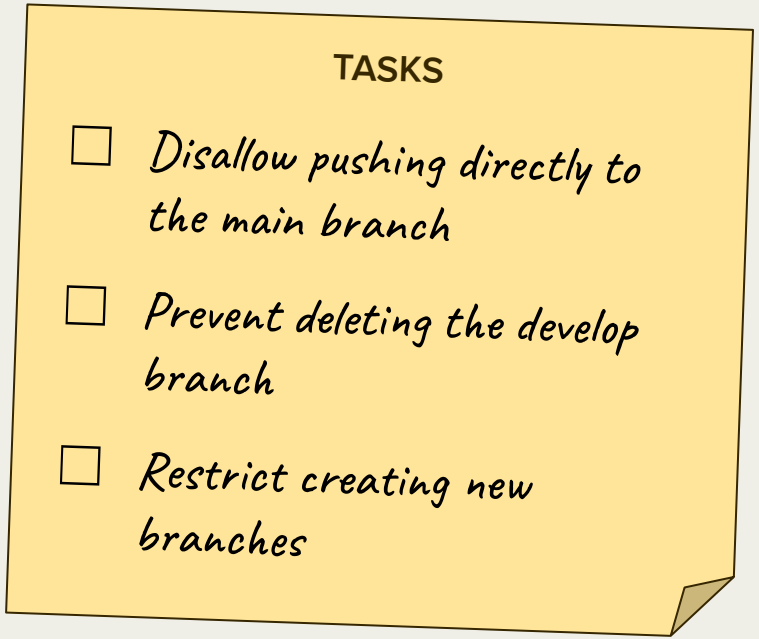
Forks and Pull Requests

- For letting **external contributors** propose changes
- Requires *forking* the upstream repository before *requesting a pull*
- PR needs to be **approved** by a collaborator



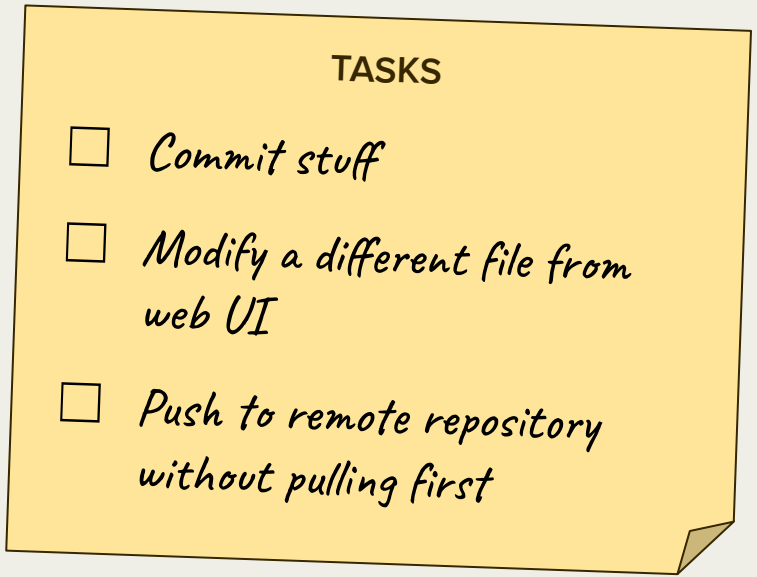
Protecting Branches

- People can mess up and end up breaking critical stuff
- It's good practice to **restrict changes** to important branches
- These rules are applied to *upstream* repositories, not forks



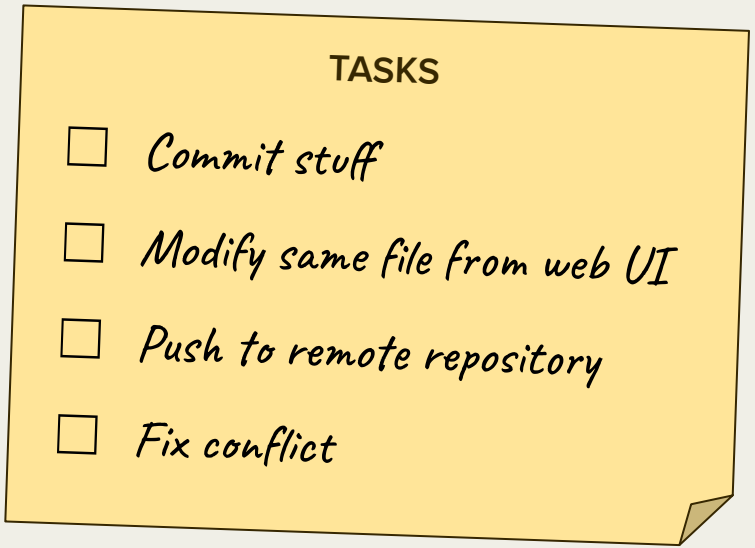
Solving Conflicts

- When several people push commits **at the same time**, we get a merge conflict
- Easy conflicts are solved automatically by Git



Solving Conflicts

- When several people push commits **at the same time**, we get a merge conflict
- Easy conflicts are solved automatically by Git
- More complex ones require **manual intervention**



That's It, But There's More

- GitHub CODEOWNERS
- GitHub Workflows
- GitHub Pages
- GitHub Code Search
- Tags and Releases
- Signed Commits
- CITATION.cff
- Dependabot