



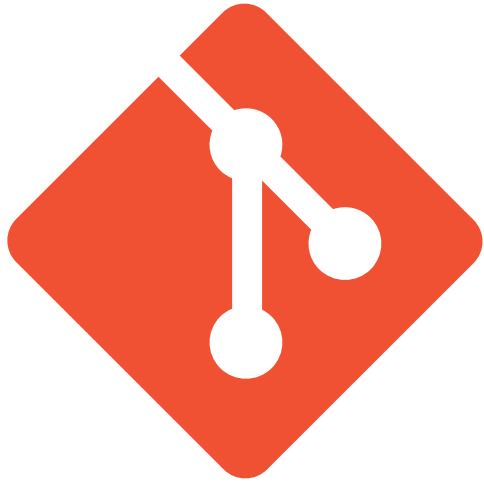
git



Parallel session

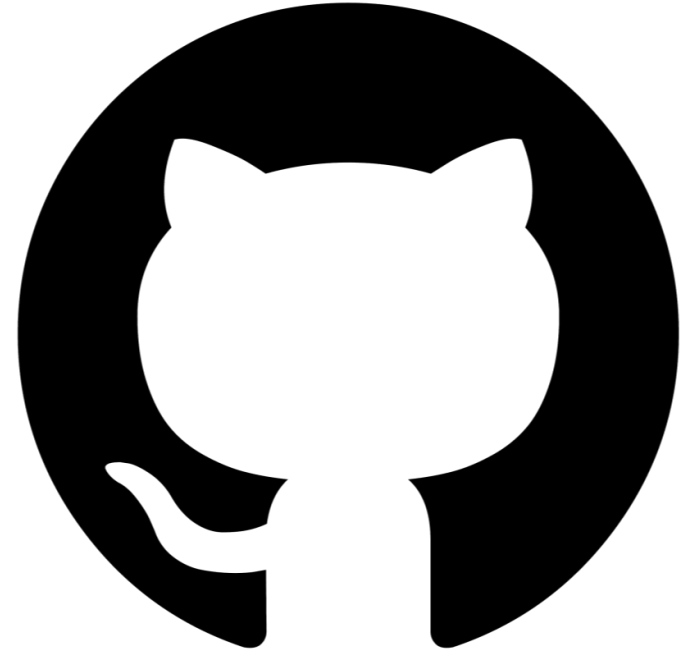
Alexander Garzón

What are we going to learn?



git

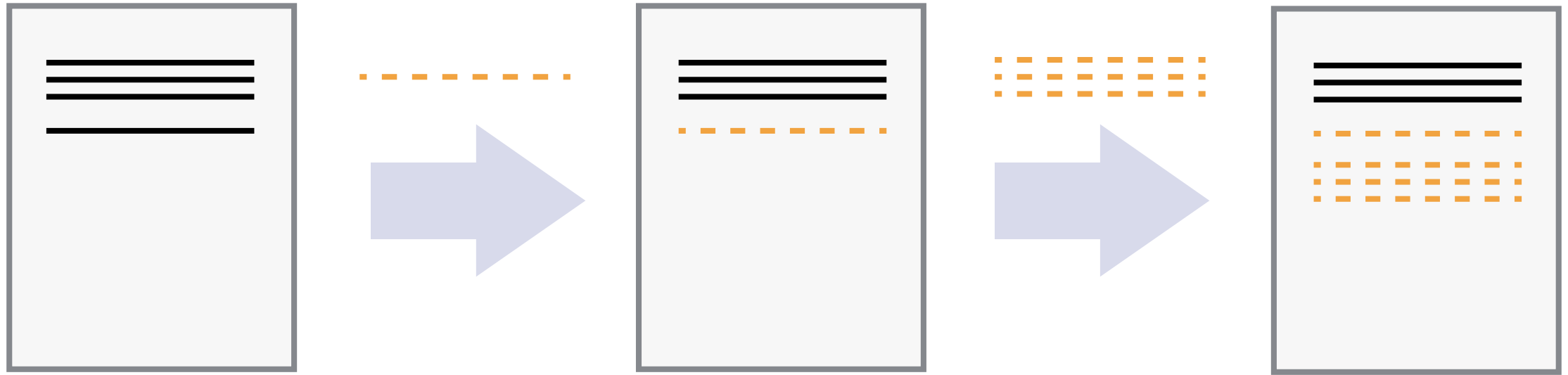
- Track documents' history
 - “Version control”



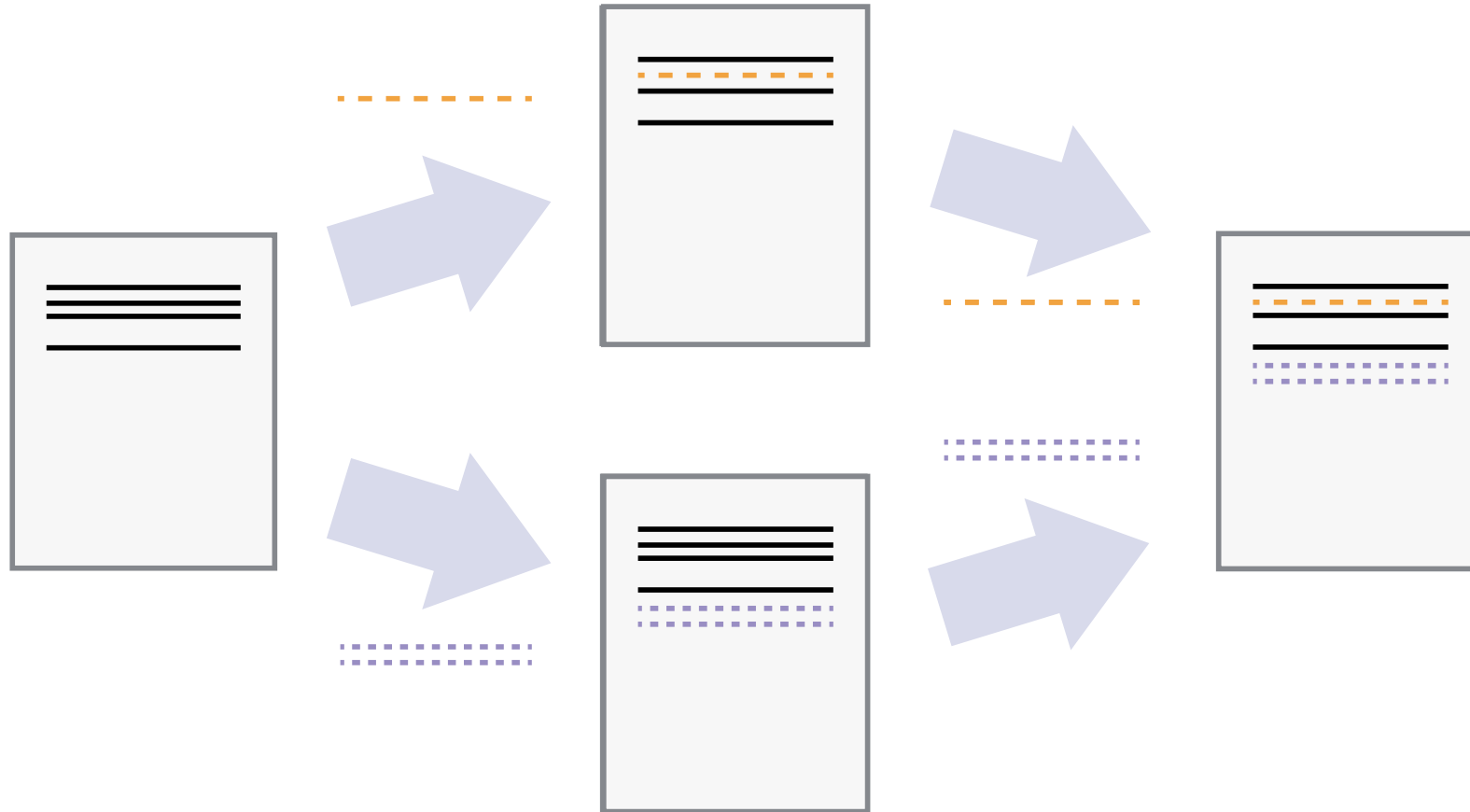
GitHub

- Collaboration

Changes



Versions and collaboration



What is Git?

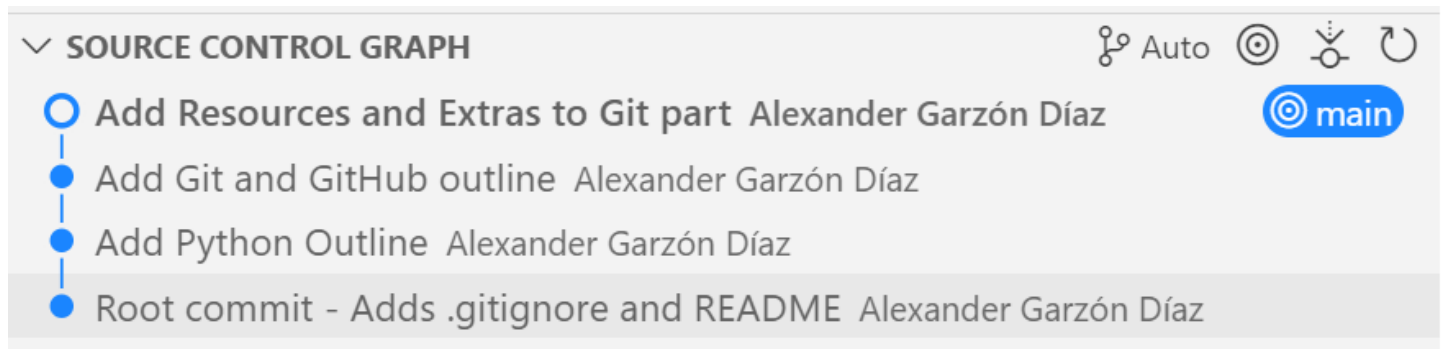
- **Version control system.**
- It creates a registry.
 - You can create records (called *Commits*) of changes in your files.
- Store and handle changes to your documents.
- You can keep track of previous versions and merge changes from different branches of development



Terminal

```
3d9db53 (HEAD -> main) Add Resources and Extras to Git part
07bd0bd Add Git and GitHub outline
7d68450 Add Python Outline
d052205 Root commit - Adds .gitignore and README
```

Graphical interface

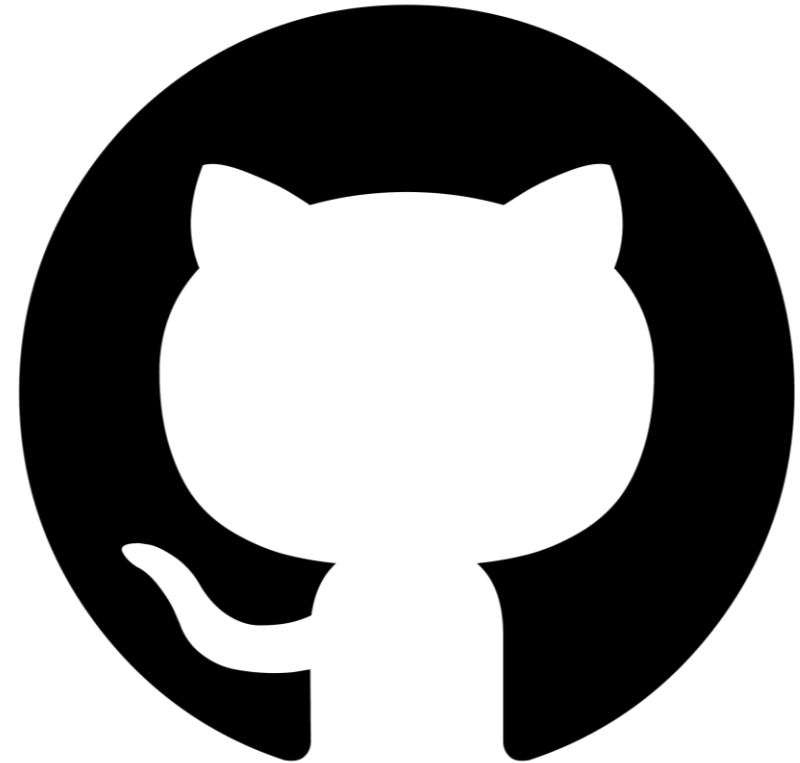


What is GitHub?

- Developer **platform**

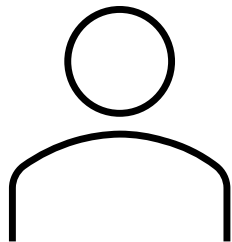
<https://github.com>

- It allows developers to
 - store,
 - share their code.
- Here, we store our project histories.

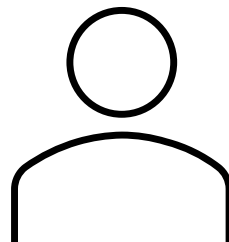


Today's agenda

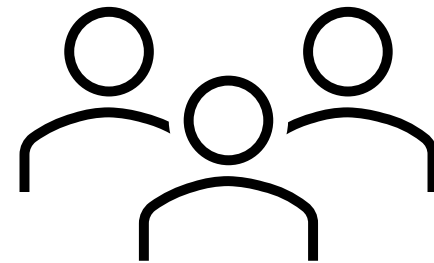
1. Set up and track changes



2. Branches and workflow

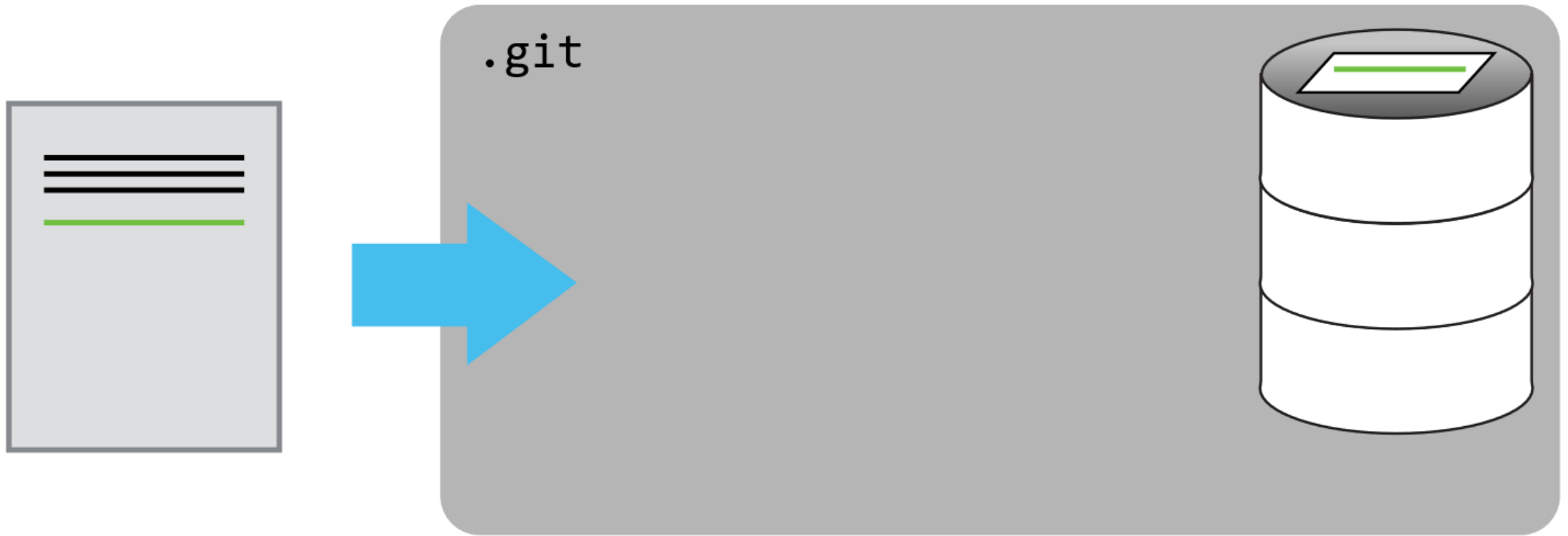


3. Online set up and collaboration



Mental Model!

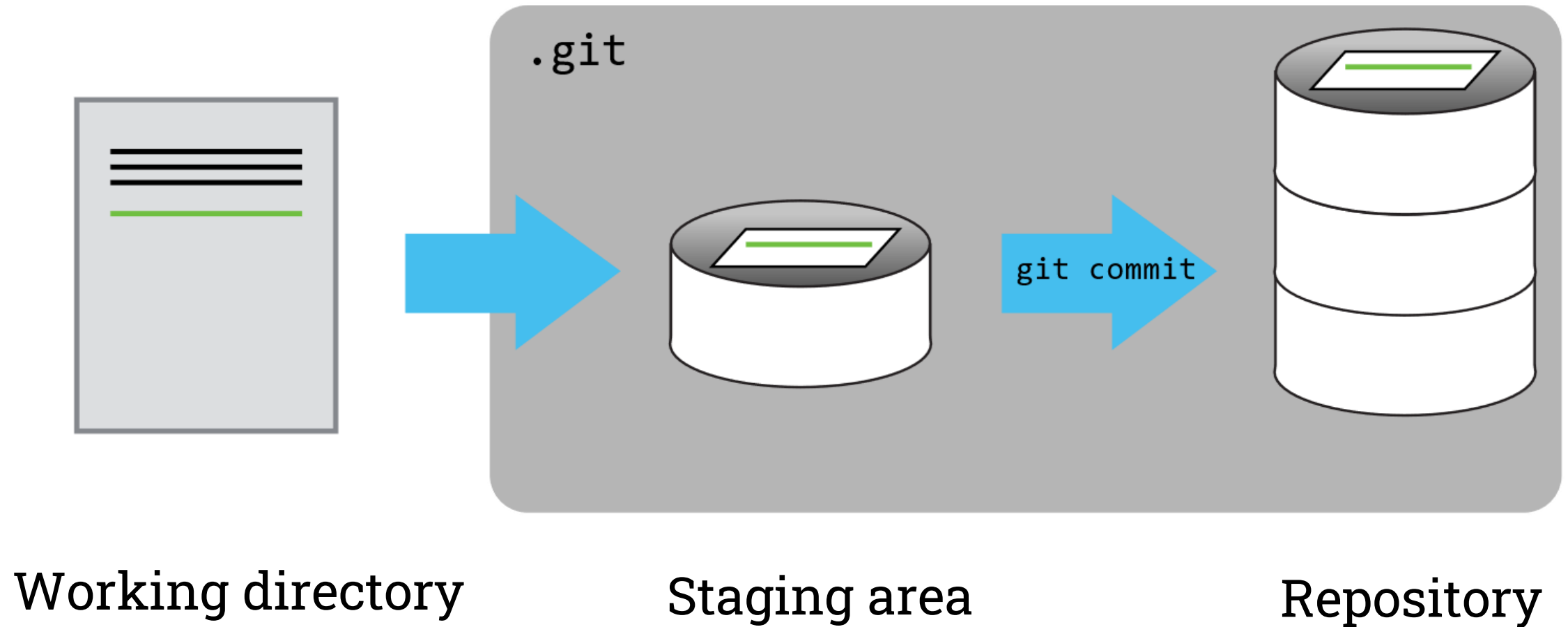
Concept



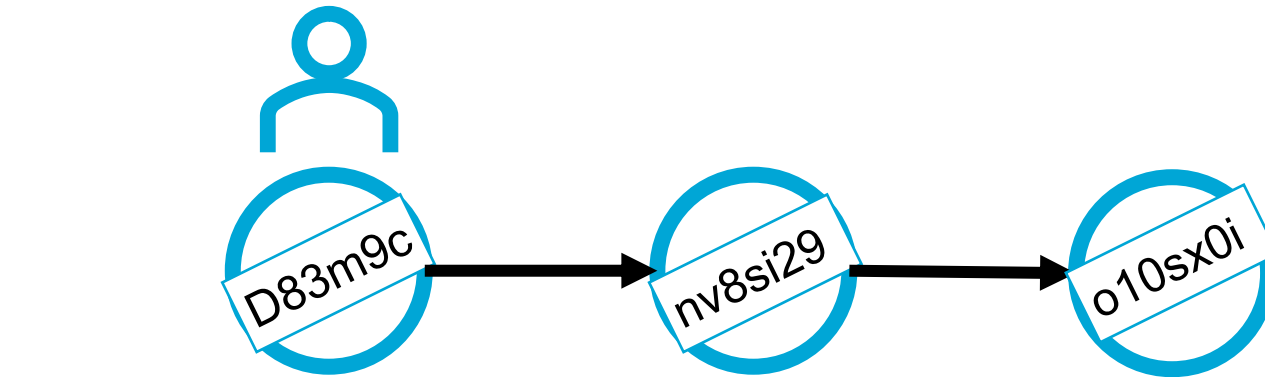
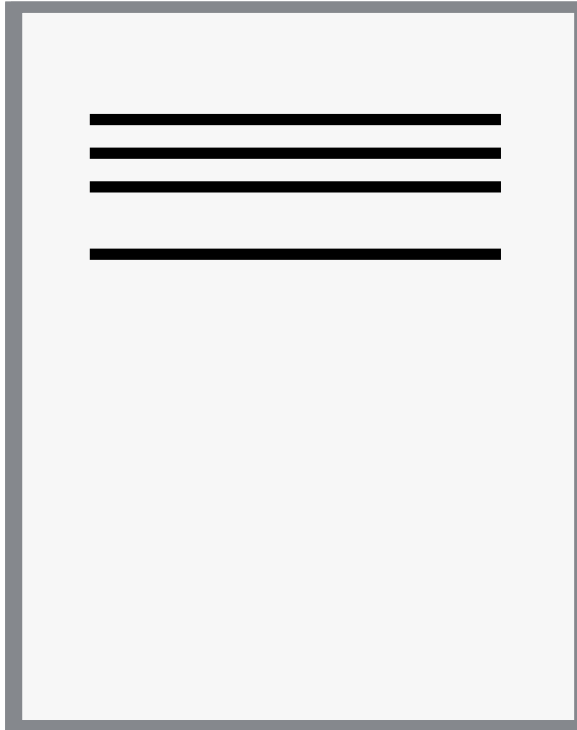
Working directory

Repository

Tracking Changes



Commits



Snapshot of the folder

Author

Email

Date

Message (an explanation)

Jargon alert!

commit pull
config HEAD
repository init cherry-pick
fetch merge
push
conflict ammend hash branch
git reset stash clone GitHub
ssh add log
diff checkout show
remote origin
fork
revert

Jargon alert!

Set up

git init
config

Tracking changes

add
commit

log
diff

History

branch
checkout
merge

HEAD
conflict
repository

Block 2

Remote repository

remote
clone

fetch
pull
push

GitHub ssh
origin

Block 3

show
ammend
revert

cherry-pick

stash

reset

fork

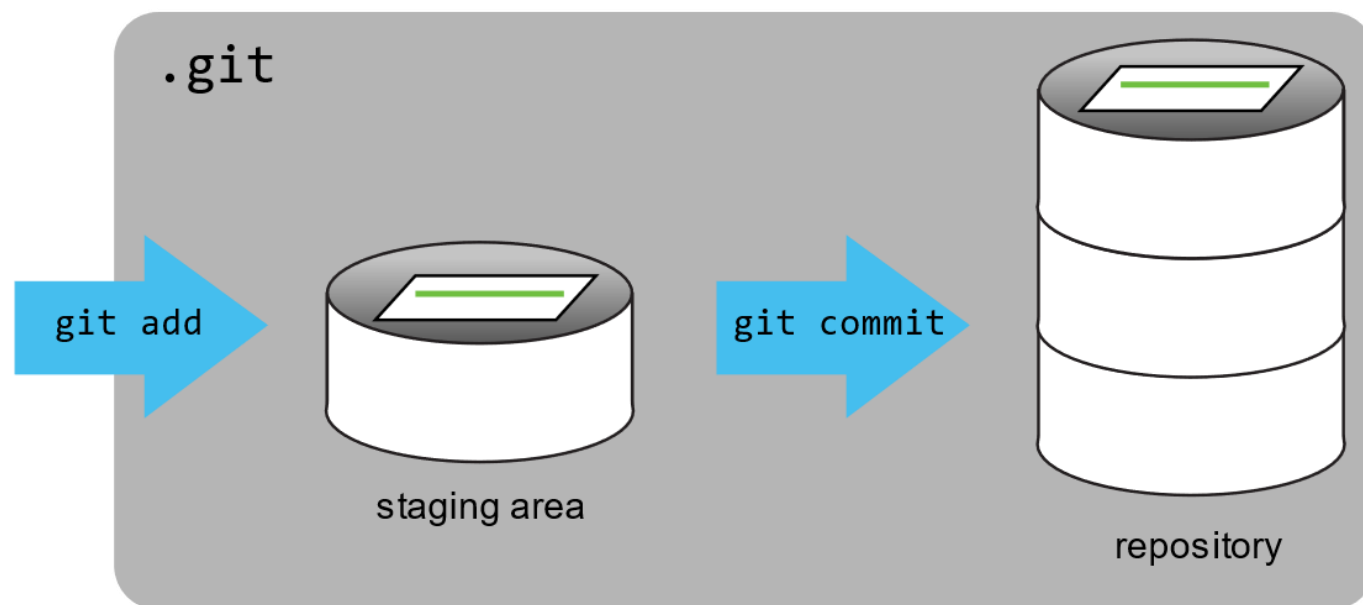
...

hash

Let's Git!

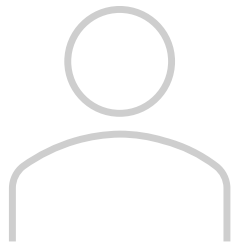
Key takeaways

- Git uses the pattern
 - `git <verb> --options`
- For setting up your name and email
 - `git config --global user.name "your name"`
 - `git config --global user.email your@email.com`
- To initialize a repository, we use `git init`
- We can commit our changes with
 - `git add`
 - `git commit`
- We can see our changes with
 - `git diff`
 - `git log`

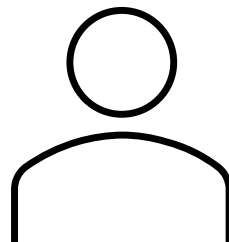


Today's agenda

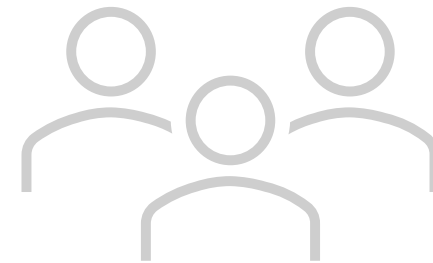
1. Set up and track changes



2. Branches and workflow

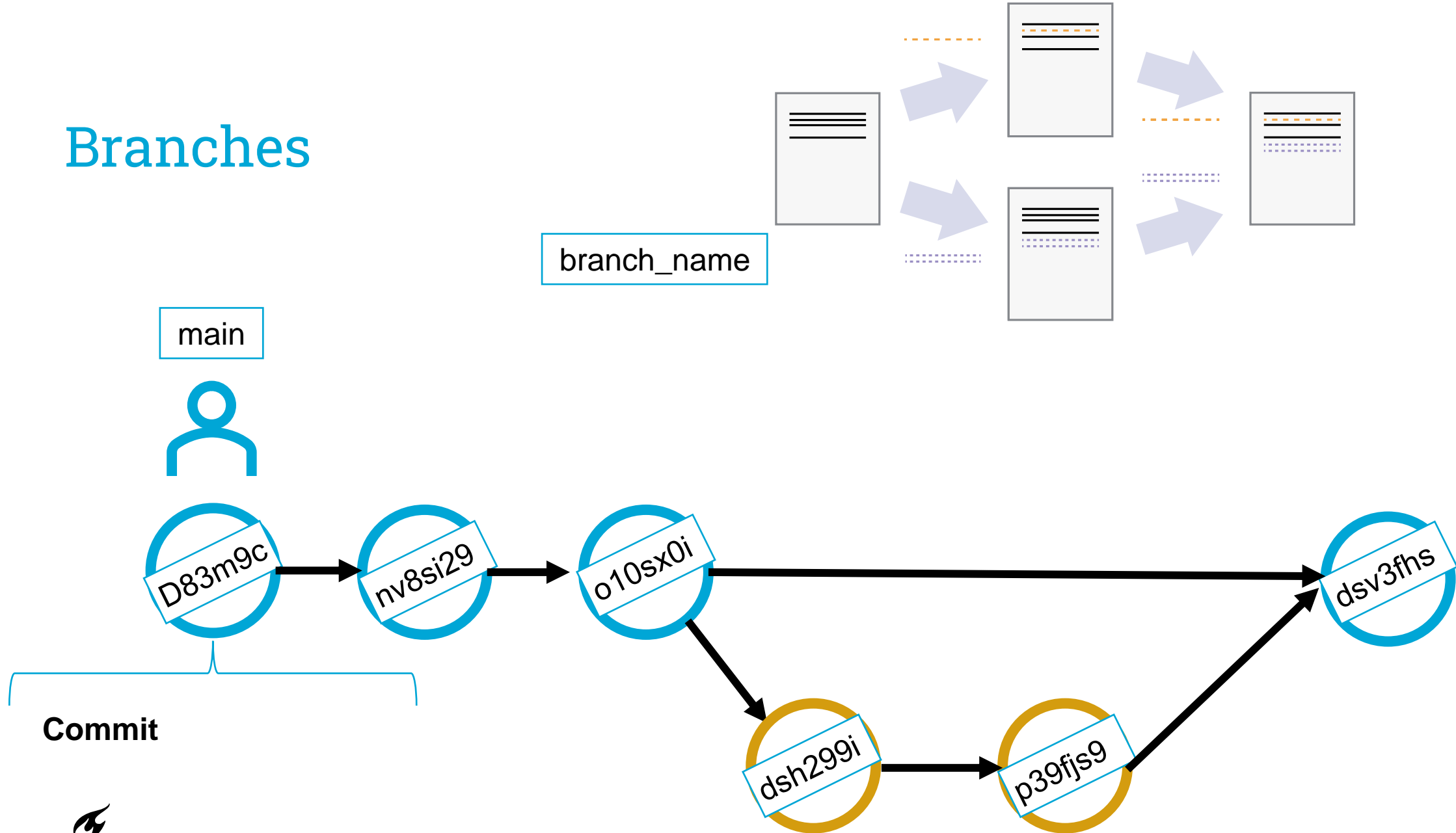


3. Online set up and collaboration

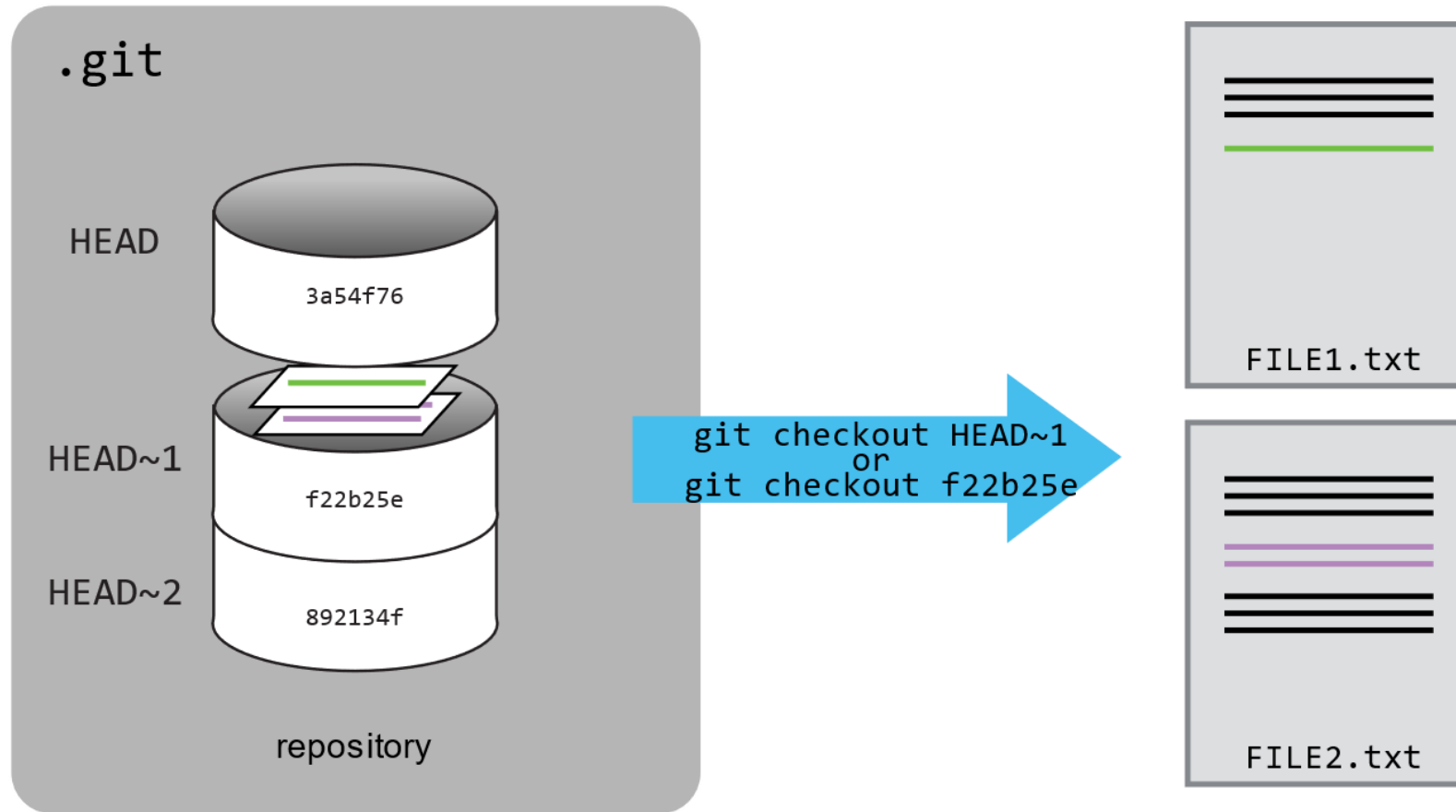


Mental Model!

Branches



git checkout

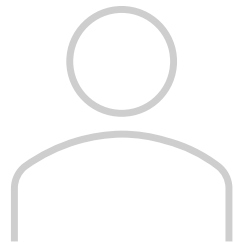


Key takeaways

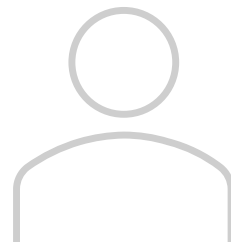
- git branch for creating parallel workflows
- git checkout to go to different branches
- git diff displays differences between commits

Today's agenda

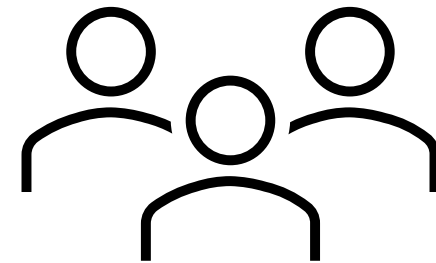
1. Set up and track changes



2. Branches and workflow



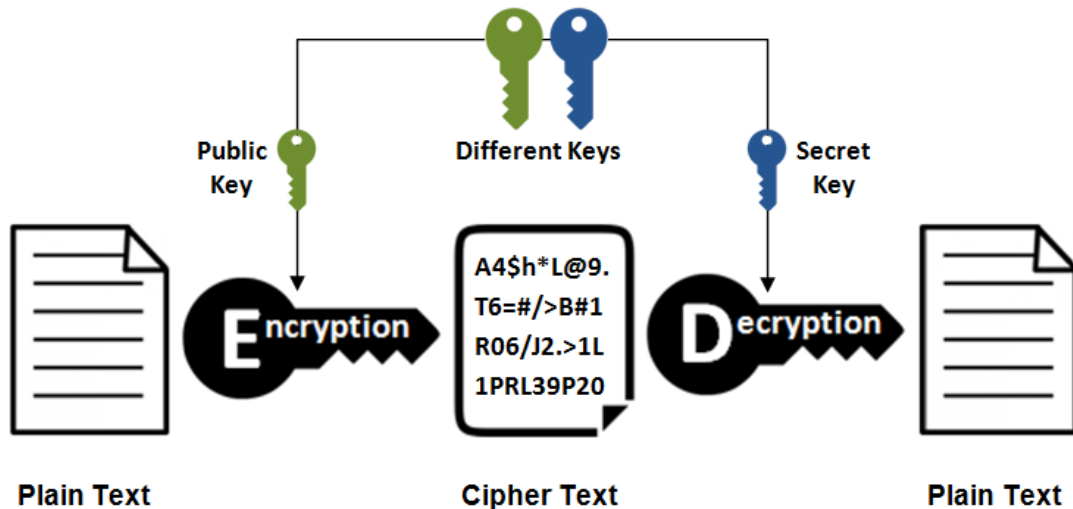
3. Online set up and collaboration



Preliminaries

SSH Keys

Asymmetric Encryption



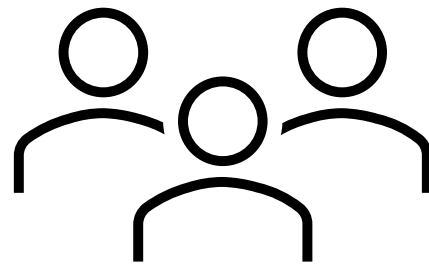
Steps

- Open Git Bash.
- `ssh-keygen -t ed25519 -C "your_email@example.com"`
 - > Enter passphrase (empty for no passphrase): [Type a passphrase]
 - > Enter same passphrase again: [Type passphrase again]
- `eval "$(ssh-agent -s)"`
- `ssh-add ~/.ssh/id_ed25519`

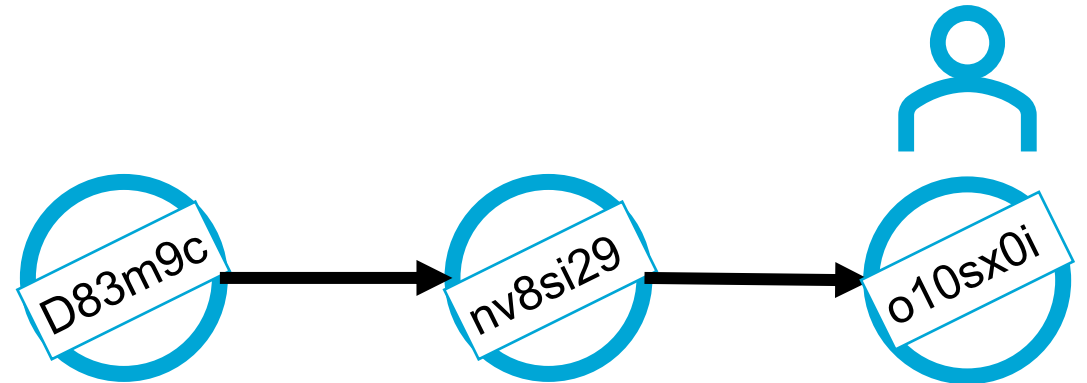
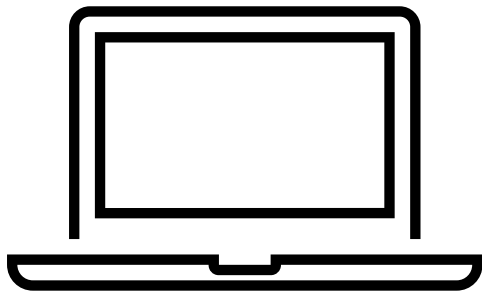
Best way, search for the documentation:

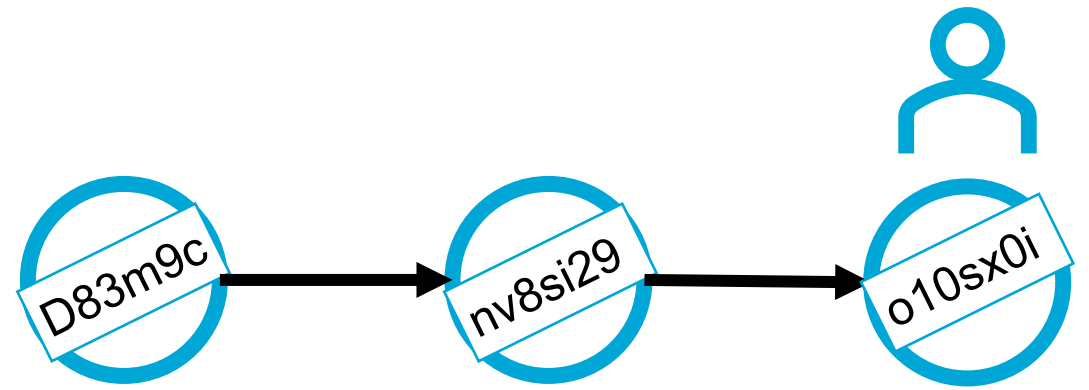
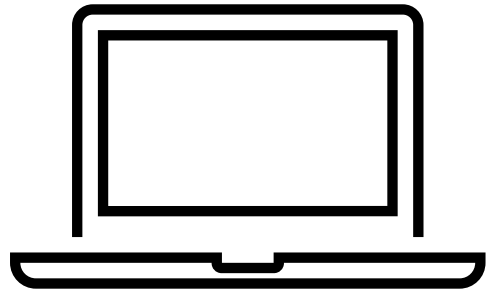
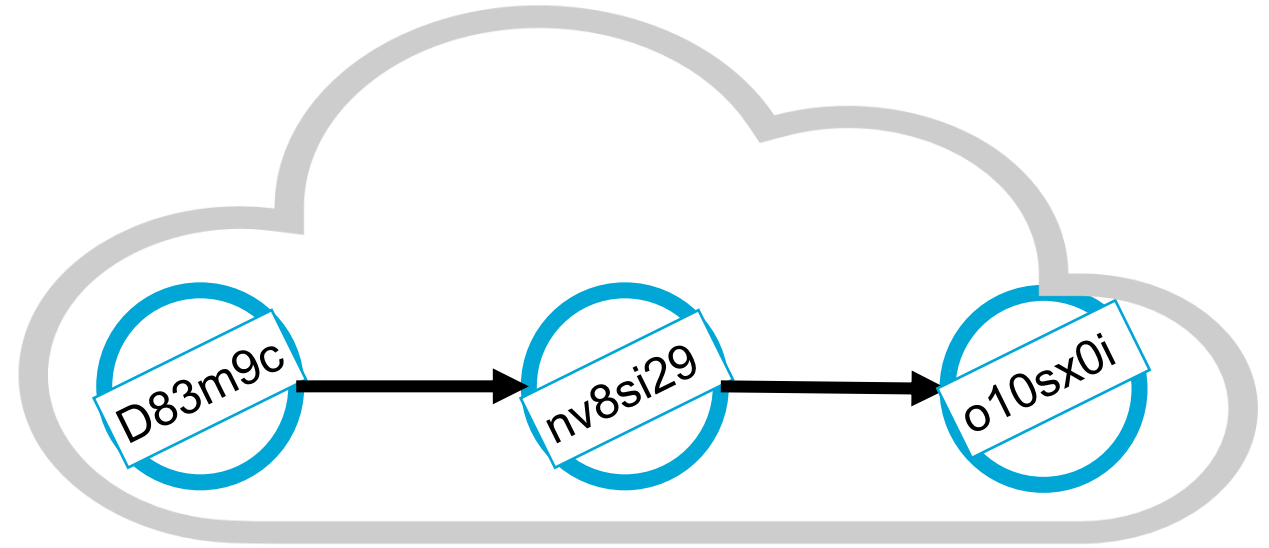
[Generating a new SSH key and adding it to the ssh-agent - GitHub Docs](#)

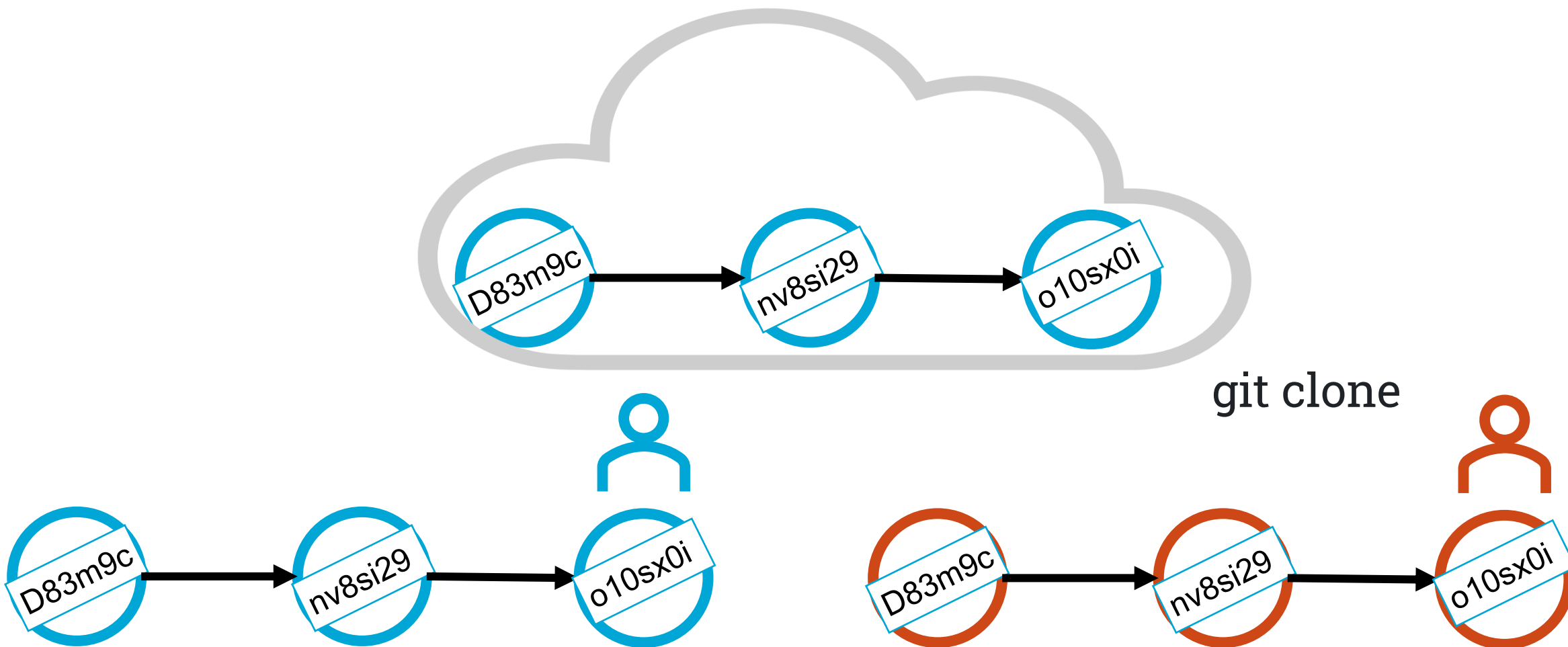
Let's learn about
GitHub!



Let's create a remote repository!





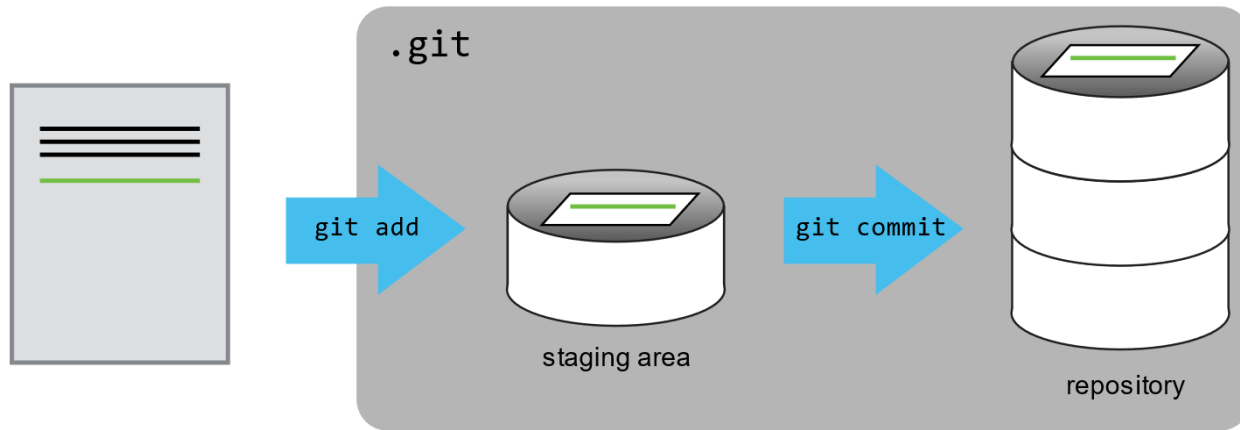


Key takeaways

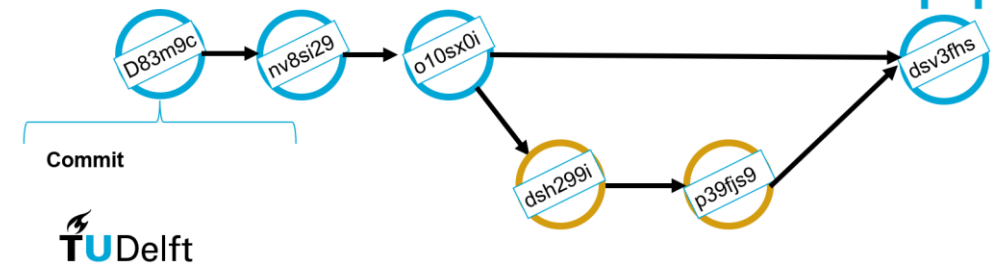
- A local Git repository can be connected to one or more remote repositories.
- Use the SSH protocol to connect to remote repositories.
- `git push` copies changes from a local repository to a remote repository.
- `git pull` copies changes from a remote repository to a local repository.
- `git clone` copies a remote repository to create a local repository with a remote called `origin` automatically set up.

Wrap up!

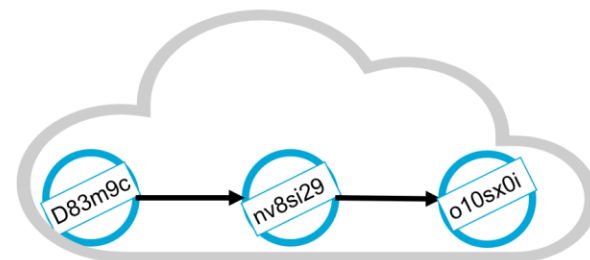
`git <verb> <--options>`



Branches



TU Delft



git clone

