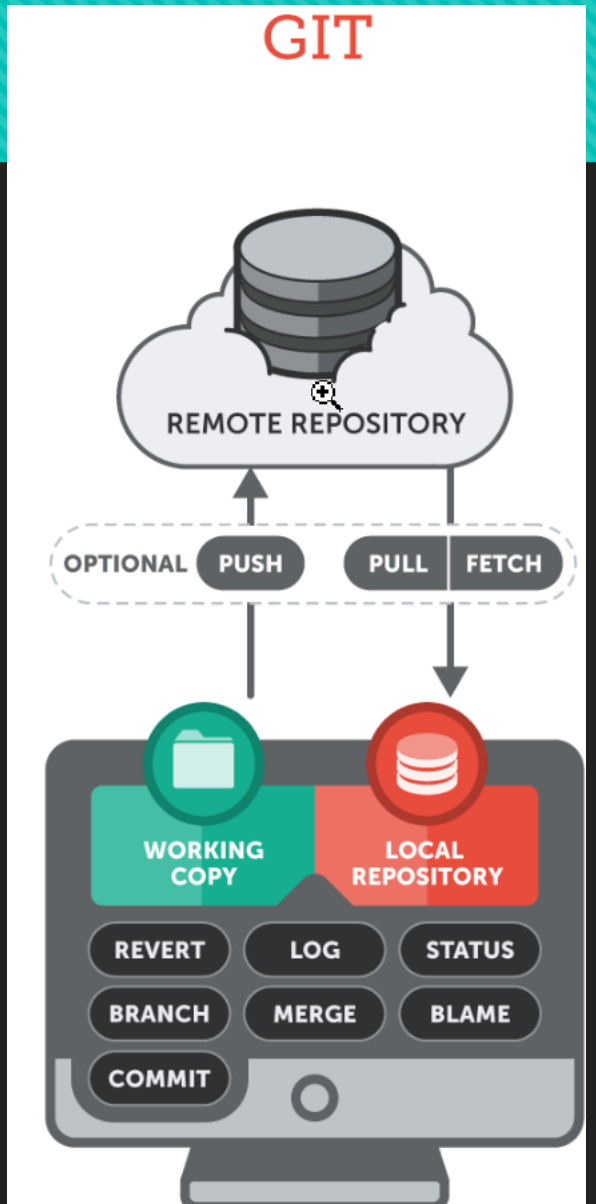
Introduction to Git

- What is Git?
  - Version control system
  - Essential for tracking changes in source code during software development
- Why use it?
  - Code management
  - Collaborative development



Git Configuration

- Configuring user information

```
git config --global user.name "Your Name"
git config --global user.email "your.email@example.com"
```

Setting Up a Git Repository

- Creating a central repository (remote repository)
  - On GitHub, GitLab, Azure Devops, etc …
- Cloning the repository

```
git clone [repository_url]
```

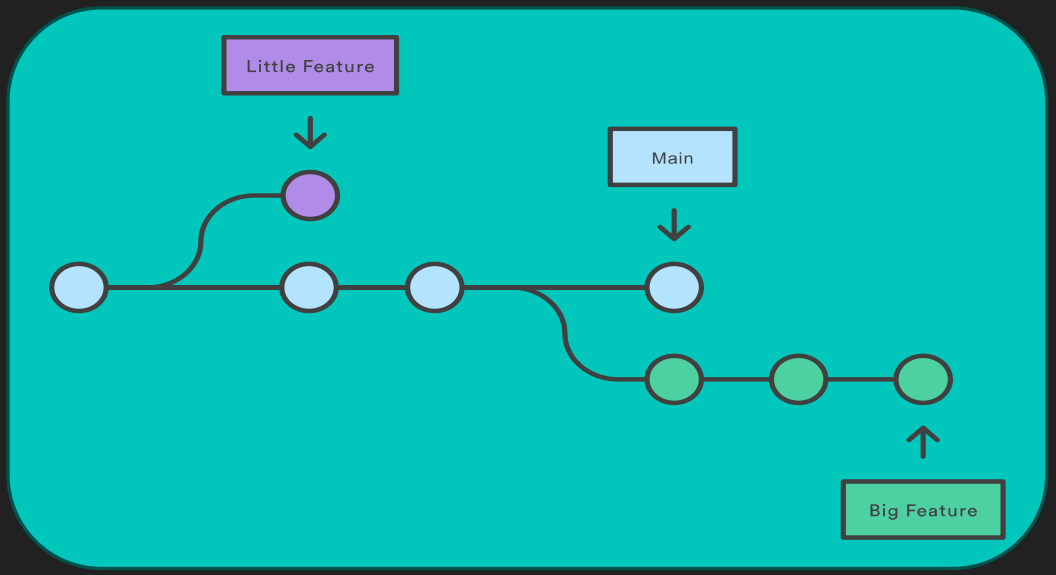
Working with Branches

- Creating and using branches
  - Branches are essential in collaboration

```
git checkout -b [branch_name]
```

- Check the present branch

```
git branch
```



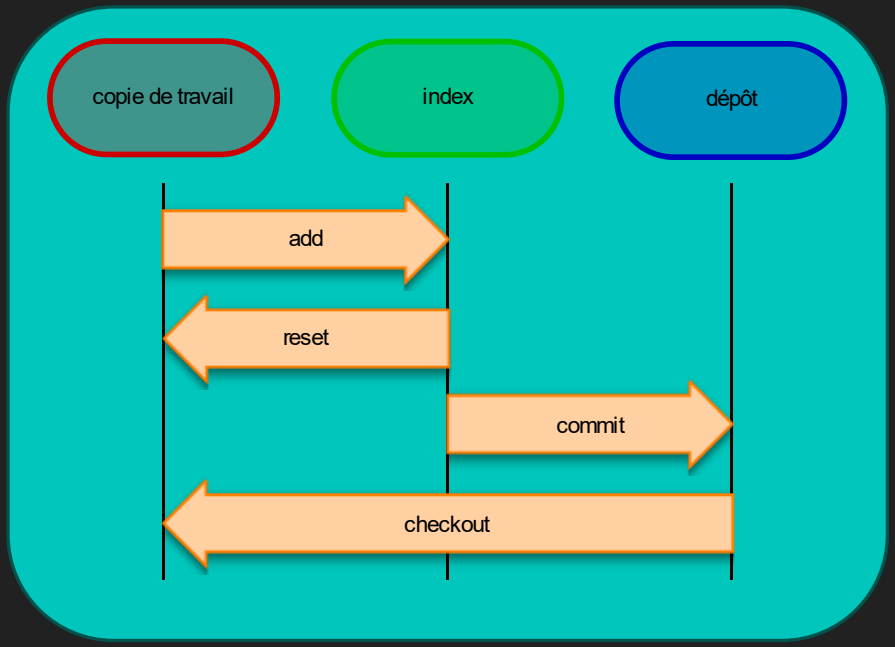
The Development Lifecycle with Git

- Local development

```
git status
```

- Staging and committing changes

```
git add .
git commit -m "commit message"
```



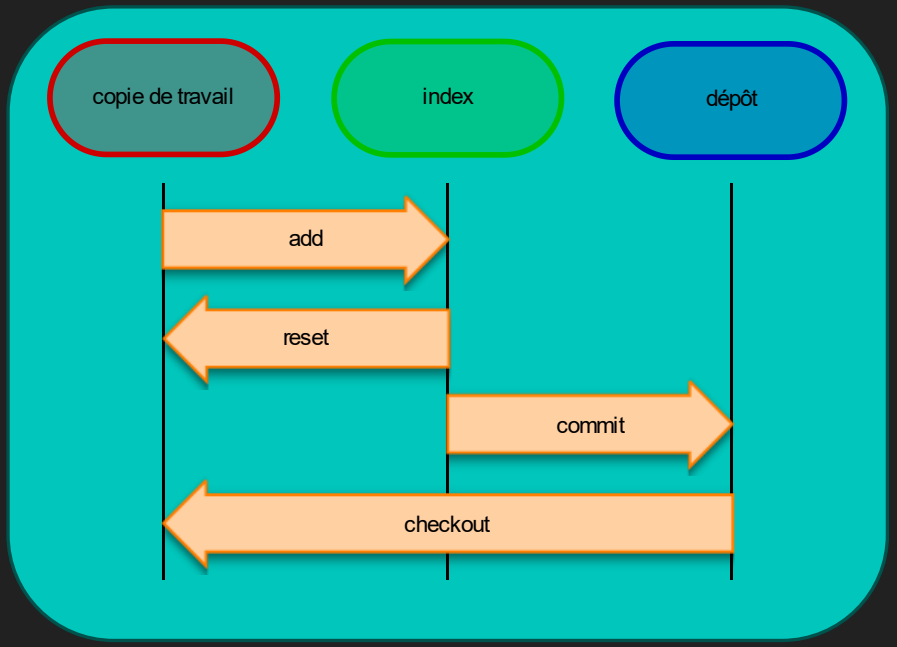
The Development Lifecycle with Git

- Before pushing your changes, make sure you have the latest version of the code.

```
git checkout main
git pull origin main
git checkout [branchs_name]
git rebase main
```

- Push your changes
  - Synchronizing with the central repository

```
git push origin [branch_name]
```



.gitignore

- Ignore all these files

*.csv

file/

- Ignore all except these files

/*

!.gitignore

!/templates/

Code Review Process

- Pull request/Merge request process
- Peer code review
- Creating a pull request on the platform like GitHub, GitLab, etc.

Merging Changes

- Before merge your branch, make sure you have the latest version of the code.

```
git checkout [branchs_name]
git pull origin main
```

- Merging pull requests

```
git checkout main
git merge [branchs_name]
```

- Conflict resolution

Modifications and `'git add'` & `'git commit'`

- After resolving conflicts and merging

```
git push origin main
```

Delete Branch

- Delete local

```
git branch -d [branchs_name]
```

- Delete distance

```
git push origin --delete [branchs_name]
```

Best Practices

- Never push directly to the main branch
- Rebase cautiously, never on shared branches
- Constant communication with the team

Pushing to Multiple Repositories

- Configuring additional remote repositories

```
git remote add [remote_name] [repository_url]
```

- Pushing changes to different repositories

```
git push [remote_name] [branch_name]
```

Managing large files

- Install Git Large File Storage (LFS)

```
git lfs install
```

- Initialize files that should be considered "Large"

```
git lfs track "*.csv"
```

Recap Push process

```
git add . OR git add fileName

git commit –m "your message"

git push
```

links

- Anglais
  - https://git-scm.com/doc
  - https://git-lfs.com/
  - https://code.visualstudio.com/docs/sourcecontrol/intro-to-git
- Français
  - https://openclassrooms.com/fr/courses/7688581-devenez-un-expert-de-git-et-github