



Git & Github

Présentation Introductive

Table of contents

01

Qu'est ce que
Git?

02

Qu'est ce que
VCS?

03

Terminologie

04

Commandes
Git

05

Création d'un
repos

06

Git sur Machine
Locale

1. Qu'est-ce que Git?



R: Free Open source Version Control System (VCS)

système de contrôle de versions



2. Qu'est-ce que VCS?



R: La gestion des modifications apportées aux documents, aux programmes informatiques, aux grands sites Web et à d'autres collections d'informations.

3. Terminologie

- Directory → Folder, Dossier.
 - Terminal, Command Line, cmd → Interface pour les commandes.
 - cd → Change Directory, Changer de dossier.
- Code Editor → Word Processor pour écrire du Code.
 - Repository → Projet, le dossier ou tout le projet est conservé.
 - Github → website pour hébergé les repos en ligne.

Exemple

Github

Repository

Directories

The screenshot shows a GitHub repository page. The repository name is "Digital-Twin-Architecture-of-an-IoT-System-for-System-Resilience-and-Prevention" and it is marked as "Private". The page shows the repository's main branch, "main", with 1 branch and 0 tags. A list of files and directories is displayed, including "Data", "ModelSelection", "Plots/datasets", "src", ".gitignore", and "README.md". The "Data" directory is highlighted with a blue box. The repository has 81 commits, with the most recent commit by "meriemsmt" dated Jul 4. The right sidebar shows the "About" section with no description, website, or topics provided, and the "Releases" section with no releases published.

File/Directory	Commit Message	Commit Date
Data	Final commit	2 months ago
ModelSelection	final code	3 months ago
Plots/datasets	final code	3 months ago
src	Final commit	2 months ago
.gitignore	saving PTData	4 months ago
README.md	Initial commit	6 months ago

Exemple

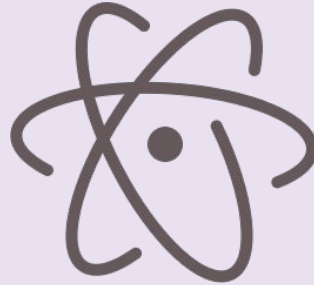
Code Editors



VS Code



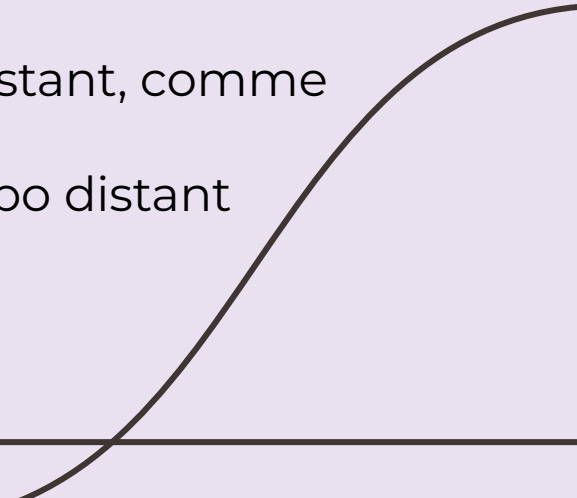
Sublime



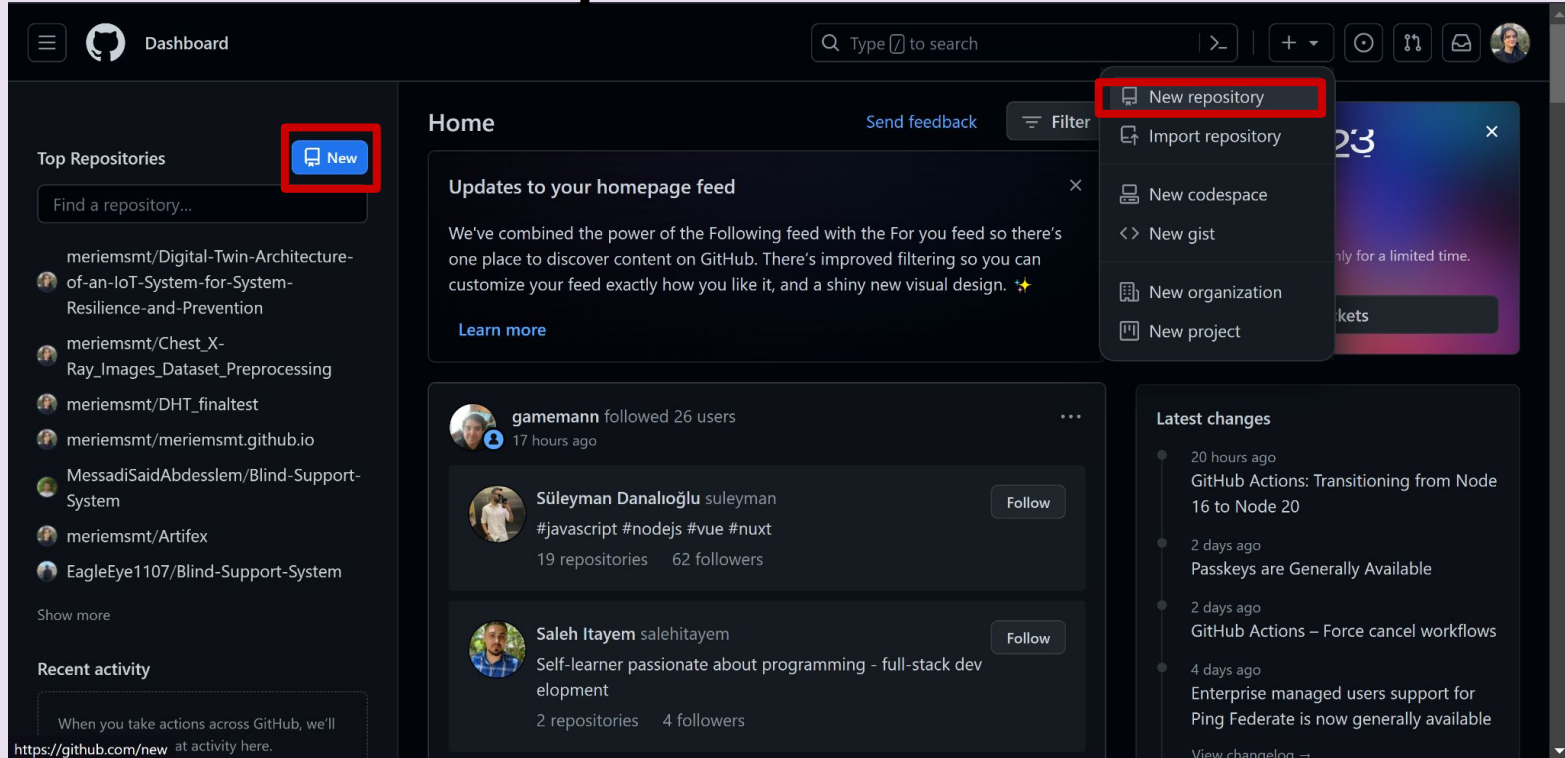
Atom

Etc.

4. Commandes Git

- **clone** → Apportez un repo hébergé quelque part comme **Github** dans un dossier sur votre machine local.
 - **add** → Tracker vos fichiers et vos modifications sur **Git**.
 - **commit** → Enregistrer vos modifications sur **Git**.
 - **push** → Télécharger les commits Git sur un repo distant, comme **Github**.
 - **pull** → Télécharger les modifications à partir du repo distant
Sur votre machine local.
- 

5. Création d'un repo



The screenshot displays the GitHub Dashboard interface. On the left sidebar, under 'Top Repositories', a blue 'New' button is highlighted with a red rectangle. In the main content area, the 'Home' section is visible. A dropdown menu is open, showing options for creating new content: 'New repository' (highlighted with a red rectangle), 'Import repository', 'New codespace', 'New gist', 'New organization', and 'New project'. The 'Latest changes' section on the right lists recent updates, including 'GitHub Actions: Transitioning from Node 16 to Node 20' and 'Enterprise managed users support for Ping Federate is now generally available'.

Top Repositories

Find a repository...

- meriemsmt/Digital-Twin-Architecture-of-an-IoT-System-for-System-Resilience-and-Prevention
- meriemsmt/Chest_X-Ray_Images_Dataset_Preprocessing
- meriemsmt/DHT_finaltest
- meriemsmt/meriemsmt.github.io
- MessadiSaidAbdesslem/Blind-Support-System
- meriemsmt/Artifex
- EagleEye1107/Blind-Support-System

Show more

Recent activity

When you take actions across GitHub, we'll
<https://github.com/new> at activity here.

Home

Send feedback Filter

Updates to your homepage feed

We've combined the power of the Following feed with the For you feed so there's one place to discover content on GitHub. There's improved filtering so you can customize your feed exactly how you like it, and a shiny new visual design. ✨

[Learn more](#)

Latest changes

- 20 hours ago
GitHub Actions: Transitioning from Node 16 to Node 20
- 2 days ago
Passkeys are Generally Available
- 2 days ago
GitHub Actions – Force cancel workflows
- 4 days ago
Enterprise managed users support for Ping Federate is now generally available

[View changelog](#)

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)

Required fields are marked with an asterisk (*).

Owner *  meriemsmt / Repository name *

☒ demo-repo is available.

Great repository names are short and memorable. Need inspiration? How about [psychic-guide](#) ?

Description (optional)

- ☐  **Public**
Anyone on the internet can see this repository. You choose who can commit.
- ☒  **Private**
You choose who can see and commit to this repository.

Initialize this repository with:

- ☒ **Add a README file**
This is where you can write a long description for your project. [Learn more about READMEs.](#)

README File:

README.md

demo-repo 

repo demonstration - IUT Lumière Lyon

← Files **demo-repo** / README.md in main Cancel changes **Commit changes...**

Edit Preview Spaces 2 Soft wrap

```
1 # demo-repo
2 Modification repo demonstration - IUT Lumière Lyon
3
4
```

Commit changes

Commit message

Update README.md

Extended description

Add an optional extended description..

☒ Commit directly to the main branch

☐ Create a new branch for this commit and start a pull request

[Learn more about pull requests](#)

Cancel Commit changes

meriemsmt Update README.md 2dfdb90 1 minute ago 2 commits

README.md Update README.md 1 minute ago

Update README.md

meriemsmt committed 5 minutes ago

Verified 2dfdb90 <> Tip

Initial commit

meriemsmt committed 14 minutes ago

Verified 6ec3b3e <> Tip

3 README.md

...	...	@@ -1,2 +1,3 @@
1	1	# demo-repo
2		- repo demonstration - IUT Lumière Lyon
	2	+ Modification repo demonstration - IUT Lumière Lyon
	3	+

Comment faire cela directement sur votre machine local?

6. Git sur Machine Locale

1. Vérifiez si vous avez Git sur votre machine:

- Ouvrir un Terminal: **git --version**

```
C:\Users\msmat>git --version  
git version 2.37.3.windows.1
```

Si la version ne s'affiche pas, vous n'avez pas Git.
Téléchargez la dernière version de Git:



<https://gitforwindows.org/>



<https://www.atlassian.com/git/tutorials/install-git#mac-os-x>



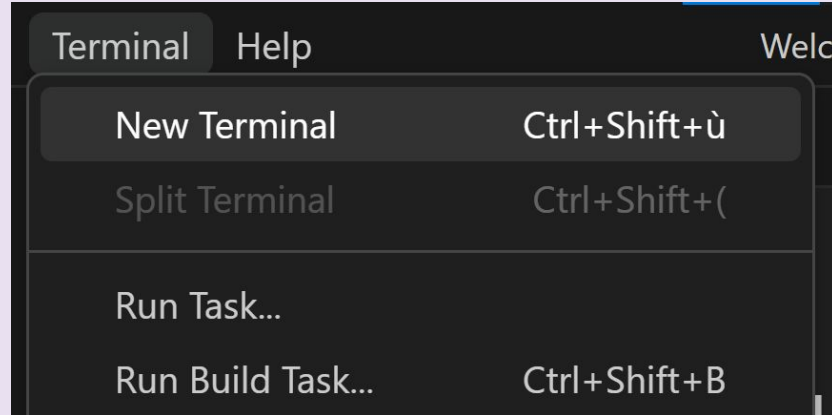
```
$ sudo apt-get update  
$ sudo apt-get install git
```

2. Configurez votre nom d'utilisateur et e-mail Git :

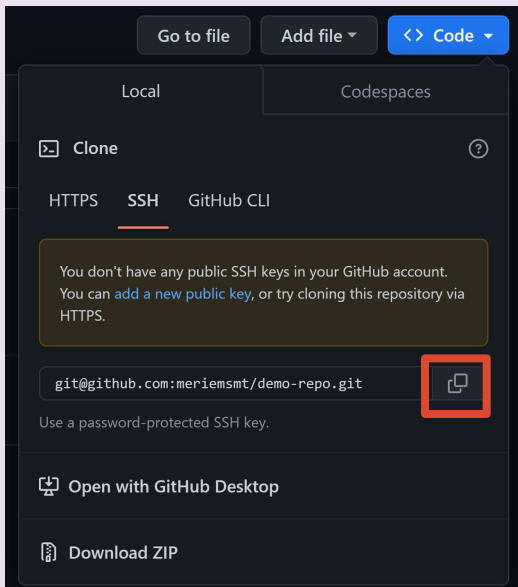
```
git config --global user.name "Meriem Smati"
```

```
git config --global user.email "meriem.smati@univ-lyon2.fr"
```

3. Ouvrez un dossier vide sur votre code editor (Exemple: VS Code)
4. Ouvrir un terminal à partir du code Editor.

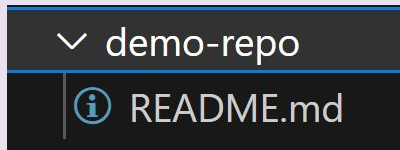


Clone le Repo sur notre Machine Locale



git clone 'lien git'

```
PS C:\Users\msmat\Desktop\Enseignement\Git> git clone https://github.com/meriemsmt/demo-repo.git
Cloning into 'demo-repo'...
remote: Enumerating objects: 6, done.
remote: Counting objects: 100% (6/6), done.
remote: Compressing objects: 100% (4/4), done.
remote: Total 6 (delta 0), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (6/6), done.
PS C:\Users\msmat\Desktop\Enseignement\Git>
```



Faire des modifications

 README.md M X

demo-repo >  README.md >  # demo-repo >  ## Sous-titre

1 # demo-repo

2 Modification repo demonstration - IUT Lumière Lyon

3

4 ## Sous-titre

5 Modification du fichier à partir de VS Code

6

✓ demo-repo

 index.html

U

 README.md

M

- Vérifier les modifications avec **git status**

```
PS C:\Users\msmat\Desktop\Enseignement\Git\demo-repo> git status
On branch main
Your branch is up to date with 'origin/main'.

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   README.md

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        index.html
```

- Informer Git des modifications **git add .** / **git add 'fichier'**

```
PS C:\Users\msmat\Desktop\Enseignement\Git\demo-repo> git add .
PS C:\Users\msmat\Desktop\Enseignement\Git\demo-repo> git status
On branch main
Your branch is up to date with 'origin/main'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    modified:   README.md
    new file:   index.html

PS C:\Users\msmat\Desktop\Enseignement\Git\demo-repo> █
```

- Enregistrer les modifications localement **git commit -m "message"**

```
PS C:\Users\msmat\Desktop\Enseignement\Git\demo-repo> git commit -m "L'ajout d'un nouveau fichier Index"
[main 6ace997] L'ajout d'un nouveau fichier Index
 2 files changed, 14 insertions(+)
 create mode 100644 index.html
```

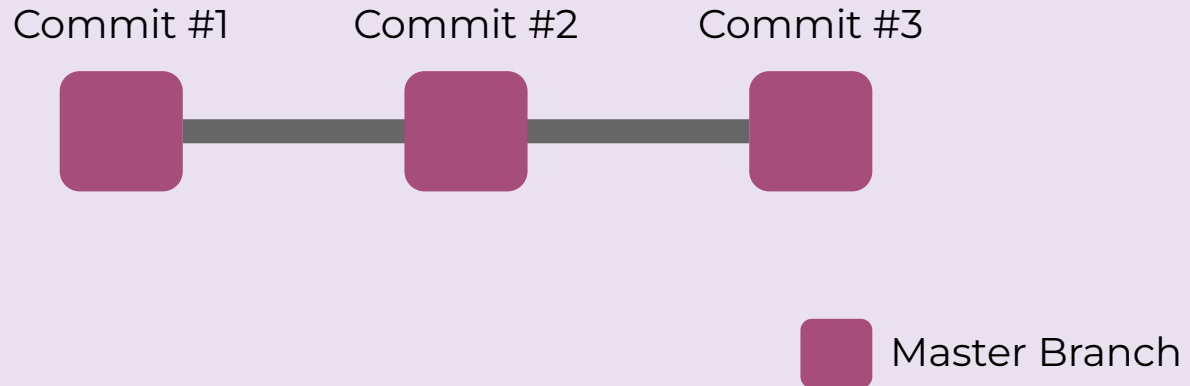
- Mettre en ligne sur GitHub **git push**

```
PS C:\Users\msmat\Desktop\Enseignement\Git\demo-repo> git push
Enumerating objects: 6, done.
Counting objects: 100% (6/6), done.
Delta compression using up to 8 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (4/4), 568 bytes | 189.00 KiB/s, done.
Total 4 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/meriemsmt/demo-repo.git
 2dfdb90..6ace997  main -> main
```

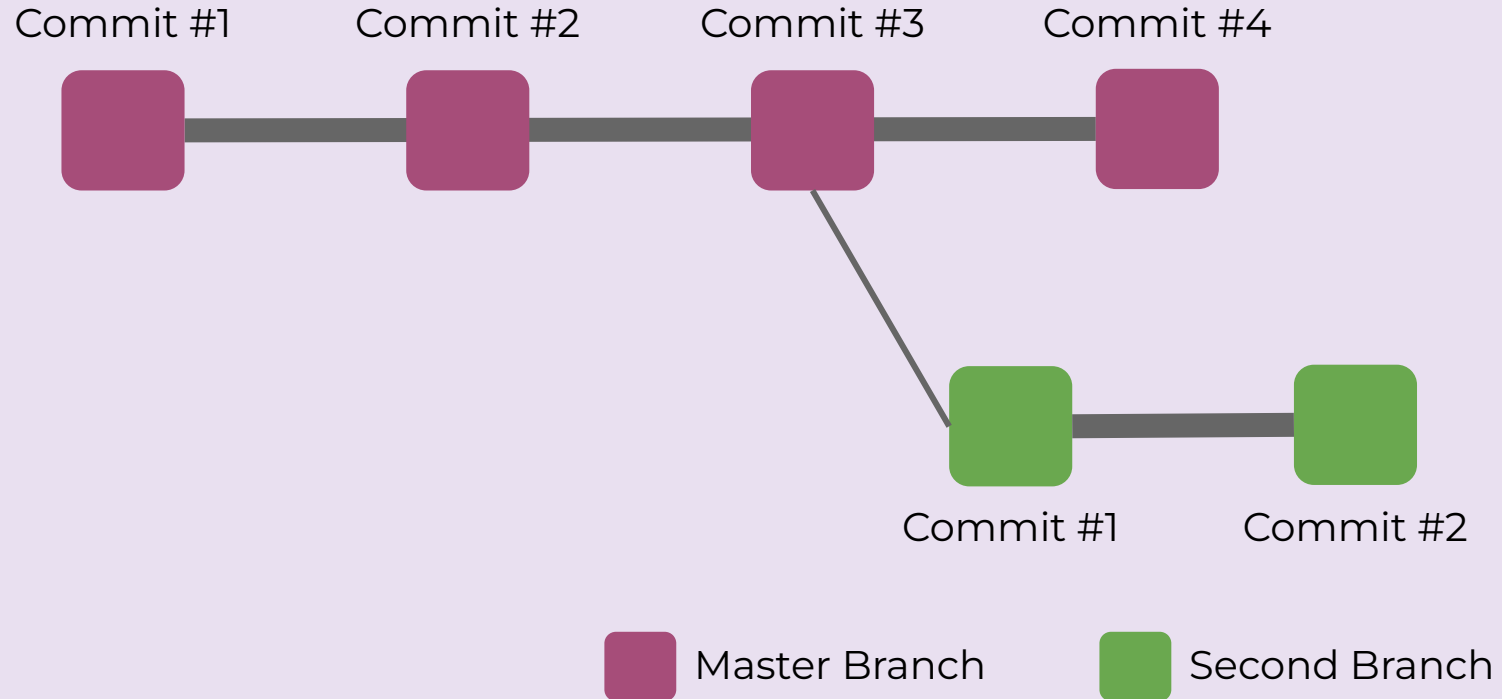
- Récupérer des modifications de Github **git pull**

6. Git Branching

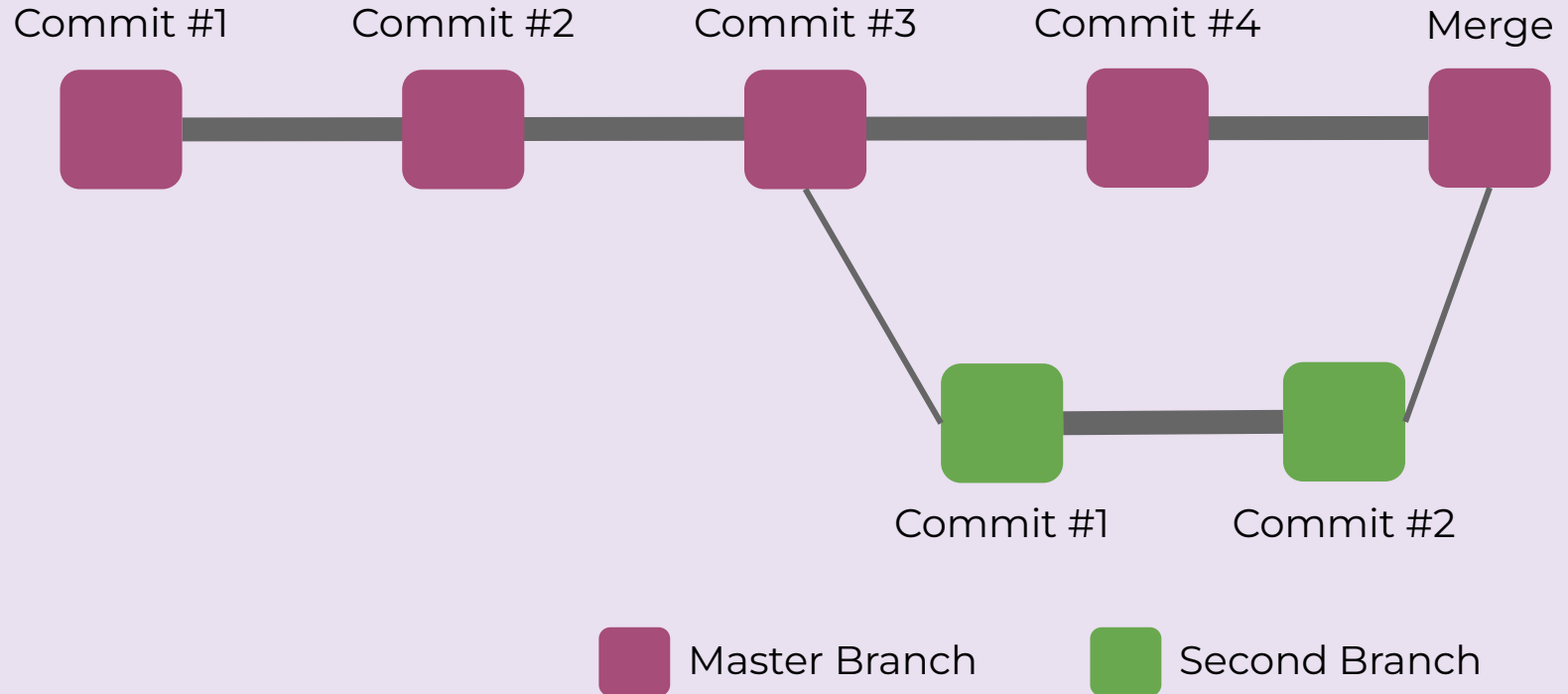
- Au début, on est sur le **Master Branch (Main Branch / Default Branch)**.



- Et si on créait une deuxième branche?



- Et si on créait une deuxième branche?



Example

- Voir combien de branches sont existantes: **git branch**

```
PS C:\Users\msmat\Desktop\Enseignement\Git\demo-repo> git branch
* main
```

- Créer une Branche nommée 'second': **git checkout -b second**

```
PS C:\Users\msmat\Desktop\Enseignement\Git\demo-repo> git checkout -b second
Switched to a new branch 'second'
```

- Revenir sur la Branche main: **git checkout main**

```
PS C:\Users\msmat\Desktop\Enseignement\Git\demo-repo> git checkout main
Switched to branch 'main'
Your branch is up to date with 'origin/main'.
```

- Accomplir des modifications sur cette branche et enregistrer:

```
# demo-repo
```

```
Modification repo demonstration - IUT Lumière Lyon
```

```
## Sous-titre
```

```
Modification du fichier à partir de VS Code
```

```
## Sous titre 2
```

```
Deuxième modif
```

```
PS C:\Users\msmat\Desktop\Enseignement\Git\demo-repo> git add .
```

```
PS C:\Users\msmat\Desktop\Enseignement\Git\demo-repo> git commit -m "modification sur branche second"
```

```
[main 52bd856] modification sur branche second
```

```
1 file changed, 3 insertions(+)
```

- Revenir à la branche main:

```
1  # demo-repo
2  Modification repo demonstration - IUT Lumière Lyon
3
4  ## Sous-titre
5  Modification du fichier à partir de VS Code
6
7
```

Les modifications apparaissent uniquement sur la branche correspondante

- Combiner les deux branches:

Pour voir la différence entre les deux branches: **git diff 'nom branch'**

Switched to branch 'second'

```
PS C:\Users\msmat\Desktop\Enseignement\Git\demo-repo> git diff main
```

```
diff --git a/README.md b/README.md
```

```
index 56c135d..6b46cd3 100644
```

```
--- a/README.md
```

```
+++ b/README.md
```

```
@@ -4,6 +4,3 @@ Modification repo demonstration - IUT Lumière Lyon
```

```
## Sous-titre
```

```
Modification du fichier à partir de VS Code
```

```
-## Sous titre 2
```

```
-
```

```
-Deuxième modif
```

- Combiner les deux branches:

Pour voir la différence entre les deux branches: **git diff 'nom branch'**

Switched to branch 'second'

```
PS C:\Users\msmat\Desktop\Enseignement\Git\demo-repo> git diff main
```

```
diff --git a/README.md b/README.md
```

```
index 56c135d..6b46cd3 100644
```

```
--- a/README.md
```

```
+++ b/README.md
```

```
@@ -4,6 +4,3 @@ Modification repo demonstration - IUT Lumière Lyon
```

```
## Sous-titre
```

```
Modification du fichier à partir de VS Code
```

```
-## Sous titre 2
```

```
-
```

```
-Deuxième modif
```

- Combiner les deux branches - **deux méthodes**

Méthode 1: **git merge** 'nom branch'

```
PS C:\Users\msmat\Desktop\Enseignement\Git\demo-repo> git merge main
Updating 6ace997..52bd856
Fast-forward
 README.md | 3 +++
 1 file changed, 3 insertions(+)
```

Méthode 2: Push & Pull

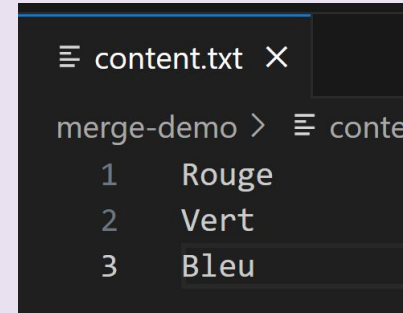
- Maintenant que le Merge est fait, on peut supprimer la Branche Second: **git branch -d 'nom branch'**

```
PS C:\Users\msmat\Desktop\Enseignement\Git\demo-repo> git branch -d second  
Deleted branch second (was aaba724).
```

7. Merge Conflicts

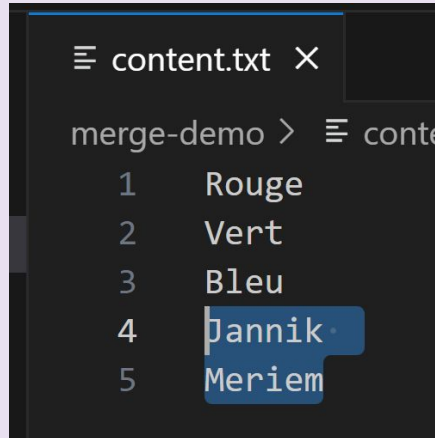
Simulation d'un conflit: On va créer deux branches “**my-branch**”, “**his-branch**” et faire un changement sur les deux et voir où le conflit apparaît et comment le résoudre.

```
PS C:\Users\msmat\Desktop\Enseignement\Git\demo-repo> git branch
* main
PS C:\Users\msmat\Desktop\Enseignement\Git\demo-repo> git branch my-branch
PS C:\Users\msmat\Desktop\Enseignement\Git\demo-repo> git branch his-branch
PS C:\Users\msmat\Desktop\Enseignement\Git\demo-repo> git checkout
Your branch is up to date with 'origin/main'.
```



- On va faire des changements:

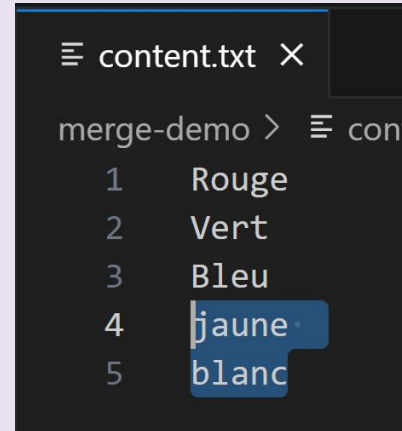
his-branch



```
≡ content.txt X
merge-demo > ≡ conte
1 Rouge
2 Vert
3 Bleu
4 Jannik
5 Meriem
```

```
git checkout his-branch
git add .
git commit -m "his changes"
```

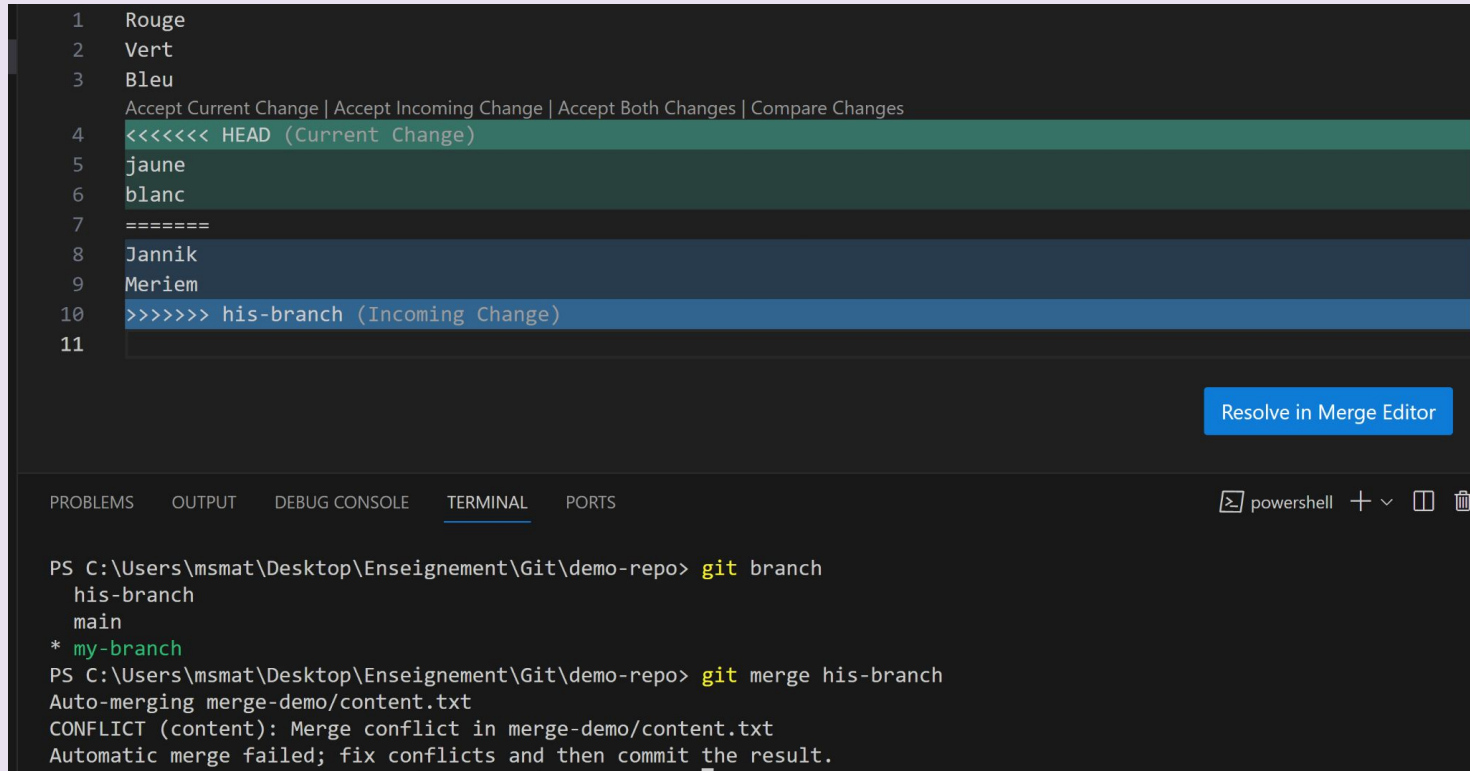
my-branch



```
≡ content.txt X
merge-demo > ≡ cont
1 Rouge
2 Vert
3 Bleu
4 jaune
5 blanc
```

```
git checkout my-branch
git add .
git commit -m "my changes"
```

- Maintenant, on va faire notre tentative de Merge:
 - Rapporter les modifications de “his-branch” dans “my-branch”



```
1 Rouge
2 Vert
3 Bleu
4 Accept Current Change | Accept Incoming Change | Accept Both Changes | Compare Changes
5 <<<<<< HEAD (Current Change)
6 jaune
7 blanc
8 =====
9 Jannik
10 Meriem
11 >>>>>> his-branch (Incoming Change)
```

Resolve in Merge Editor

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS C:\Users\msmat\Desktop\Enseignement\Git\demo-repo> git branch
his-branch
main
* my-branch
PS C:\Users\msmat\Desktop\Enseignement\Git\demo-repo> git merge his-branch
Auto-merging merge-demo/content.txt
CONFLICT (content): Merge conflict in merge-demo/content.txt
Automatic merge failed; fix conflicts and then commit the result.
```



Solution 1: Annuler le Merge

`git merge --abort`

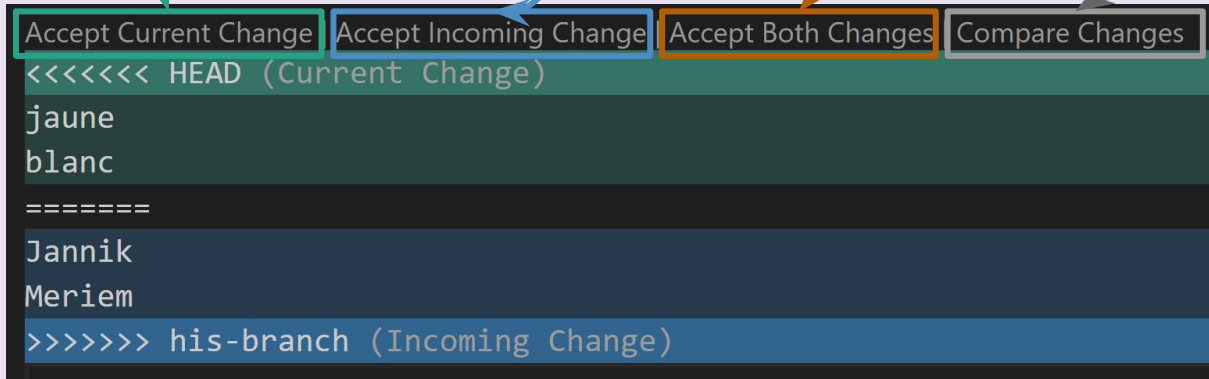
Solution 2: Résoudre le conflit

Accepter les
changements de la
branche courante
(my-branch)

Accepter les
changements de
l'autre branche

Accepter les deux
changements

Comparer les
changements



```
Accept Current Change Accept Incoming Change Accept Both Changes Compare Changes
<<<<<< HEAD (Current Change)
jaune
blanc
=====
Jannik
Meriem
>>>>>> his-branch (Incoming Change)
```