



git

// Medhat Riad

CHECK-IN

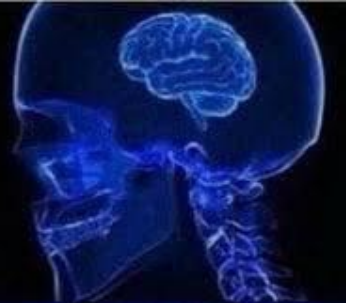
Scale 1:85,000,000 at 0°
Miller Cylindrical Projection

Coordinated Universal Time (UTC)
(formerly
Greenwich Mean Time (GMT))

French Southern and Antarctic Lands
(FRA)

THE FEAR OF GIT



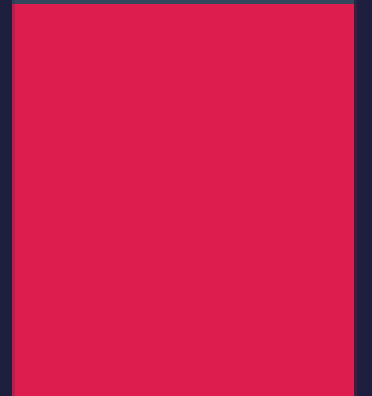
 git	
Name ■ v1.0 ■ v2.0 ■ v2.1 ■ v2.2	
■ project ■ project-revised ■ project-final ■ project-final-for-real	
Name ■ project ■ projectt ■ projecttttttt ■ aaaaaaaaaaaaaaaaaaaaaa...	

VERSION CONTROL

- Snapshots of your code over time
- Allows different members to work on different parts of the code simultaneously
- Rewind capabilities



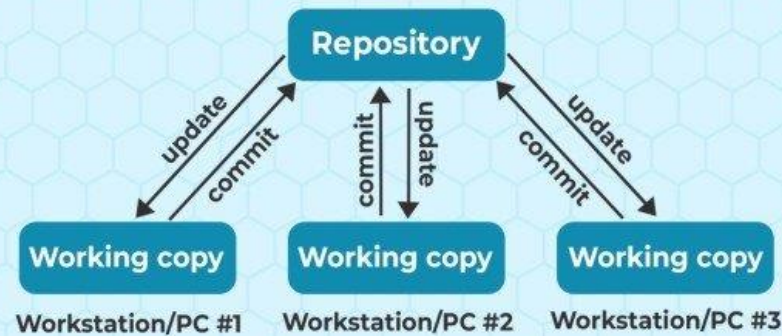
00:23.62



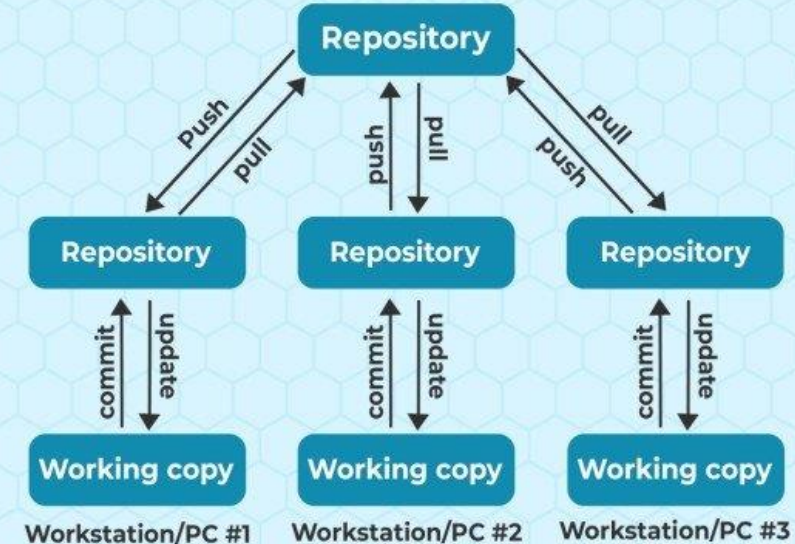


VERSION CONTROL

Centralized version



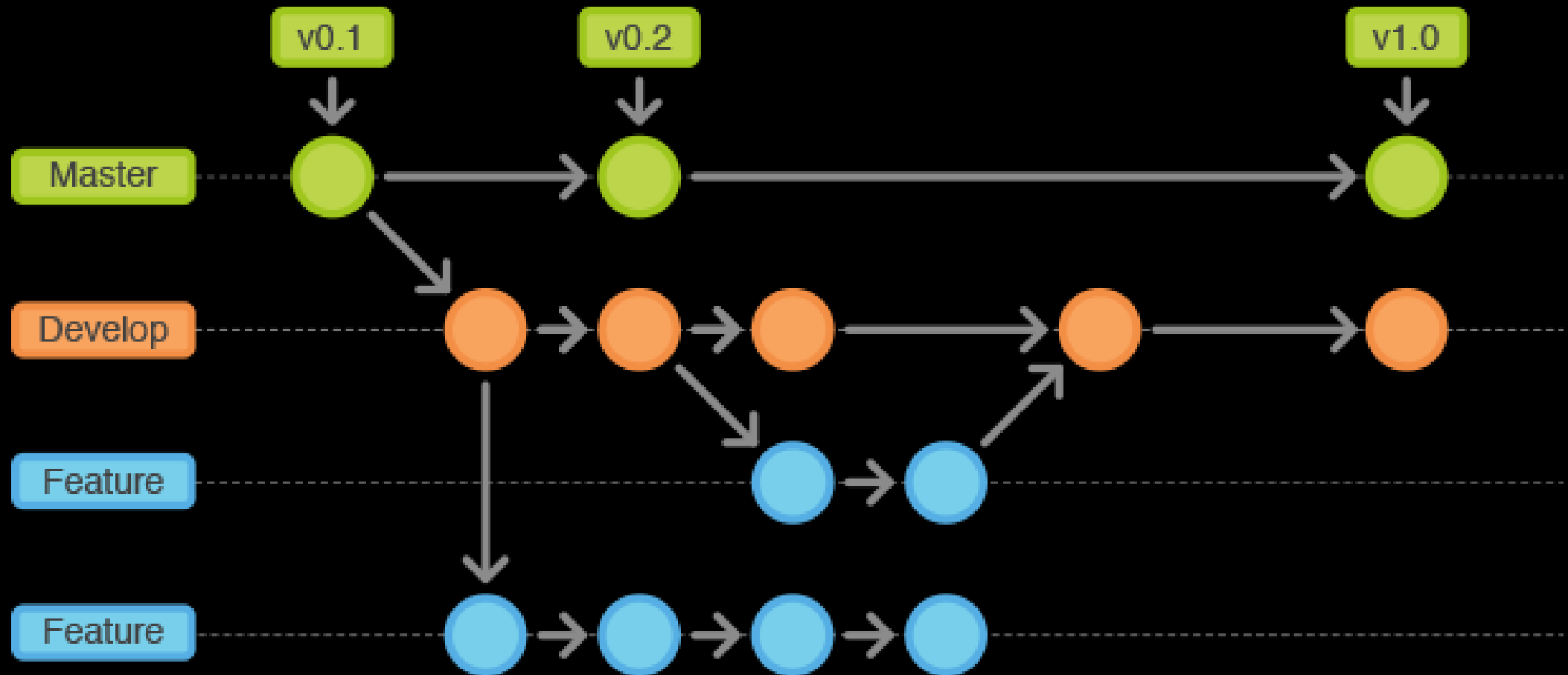
Distributed version

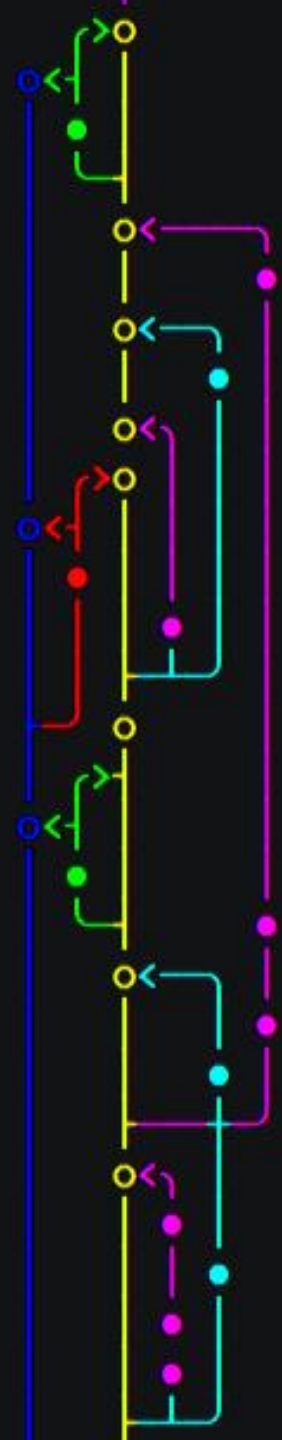


CENTRALISED VS DISTRIBUTED VERSION CONTROL SYSTEM



BRANCHES AND COMMITS





(develop) Merge branch 'master' into develop
(master) Merge branch 'develop' into master
increment version number
Merge branch 'feature/feature-1' into develop
finish work on feature 1
Merge branch 'feature/feature-2' into develop
start work on feature 2
Merge branch 'feature/feature-3' into develop
Merge branch 'feature/feature-4' into develop
Merge branch 'hotfix/fix-bug' into develop
fixing a critical bug
start work on feature 5
Merge branch 'feature/feature-5' into develop
Merge branch 'release/0.1.1' into master
Merge branch 'release/0.1.1' into develop
increment version number
more work on feature 3
Merge branch 'feature/feature-3' into develop
start work on feature 3
more work on feature 2
Merge branch 'feature/feature-2' into develop
even more work on feature 1
start work on feature 1
more work on feature 1
start work on feature 1

* c97feac increment version number
* e9898eb Merge branch 'feature-1' into develop
* 459d238 finish work on feature 1
* adfe0a0 more work on feature 1
* 5fd6701 start work on feature 2
* e5d230e Merge branch 'feature-2' into develop
* b88040c start work on feature 3
* d7a33d6 Merge branch 'feature-3' into develop
* 3ec6862 start work on feature 4
* 567694f Merge branch 'feature-4' into develop
* 20106f7 fixing a critical bug
* 70b9d1e Merge branch 'hotfix/fix-bug' into develop
* 6371196 Merge branch 'develop' into hotfix/fix-bug
* eae015c increment version number
* f1696fd Merge branch 'hotfix/fix-bug' into master
* 2003897 more work on feature 1
* 42b39dd start work on feature 2
* fb9203d Merge branch 'feature-2' into develop
* 78a0042 even more work on feature 1
* 687e6a4 more work on feature 1
* 9a8b8f0 start work on feature 1

DEMO

master Merge branch 'release/0.1.1' into master
increment version number
Merge branch 'feature/feature-1' into develop
finish work on feature 3
Merge branch 'feature/feature-2' into develop
start work on feature 5
Merge branch 'feature/feature-5' into develop
Merge branch 'hotfix/fix-bug' into develop
Merge branch 'hotfix/fix-bug' into master
fixing a critical bug
start work on feature 4
Merge branch 'release/0.1.1' into master
Merge branch 'release/0.1.1' into develop
increment version number
more work on feature 3
Merge branch 'feature/feature-3' into develop
start work on feature 3
more work on feature 2
Merge branch 'feature/feature-2' into develop
even more work on feature 1
start work on feature 2
more work on feature 1
start work on feature 1



develop Merge branch 'master' into develop
master Merge branch 'develop' into master
increment version number
Merge branch 'feature/feature-1' into develop
finish work on feature 1
Merge branch 'feature/feature-2' into develop
start work on feature 2
Merge branch 'feature/feature-3' into develop
Merge branch 'hotfix/fix-bug' into develop
Merge branch 'hotfix/fix-bug' into master
fixing a critical bug
start work on feature 5
Merge branch 'feature/feature-5' into develop
Merge branch 'release/0.1.1' into master
Merge branch 'release/0.1.1' into develop
increment version number
more work on feature 3
Merge branch 'feature/feature-3' into develop
start work on feature 3
more work on feature 2
Merge branch 'feature/feature-2' into develop
even more work on feature 1
start work on feature 1
more work on feature 1
start work on feature 1

GETTING STARTED

- Create a new repository

```
> git init
```

- Check status

```
> git status
```

- Add file to be committed

```
> git add <file>
```

- Commit added files

```
> git commit -m 'message'
```

- Create a new branch

```
> git branch <branch-name>
```

- Clone an existing repository

```
> git clone <url>
```

- Check log

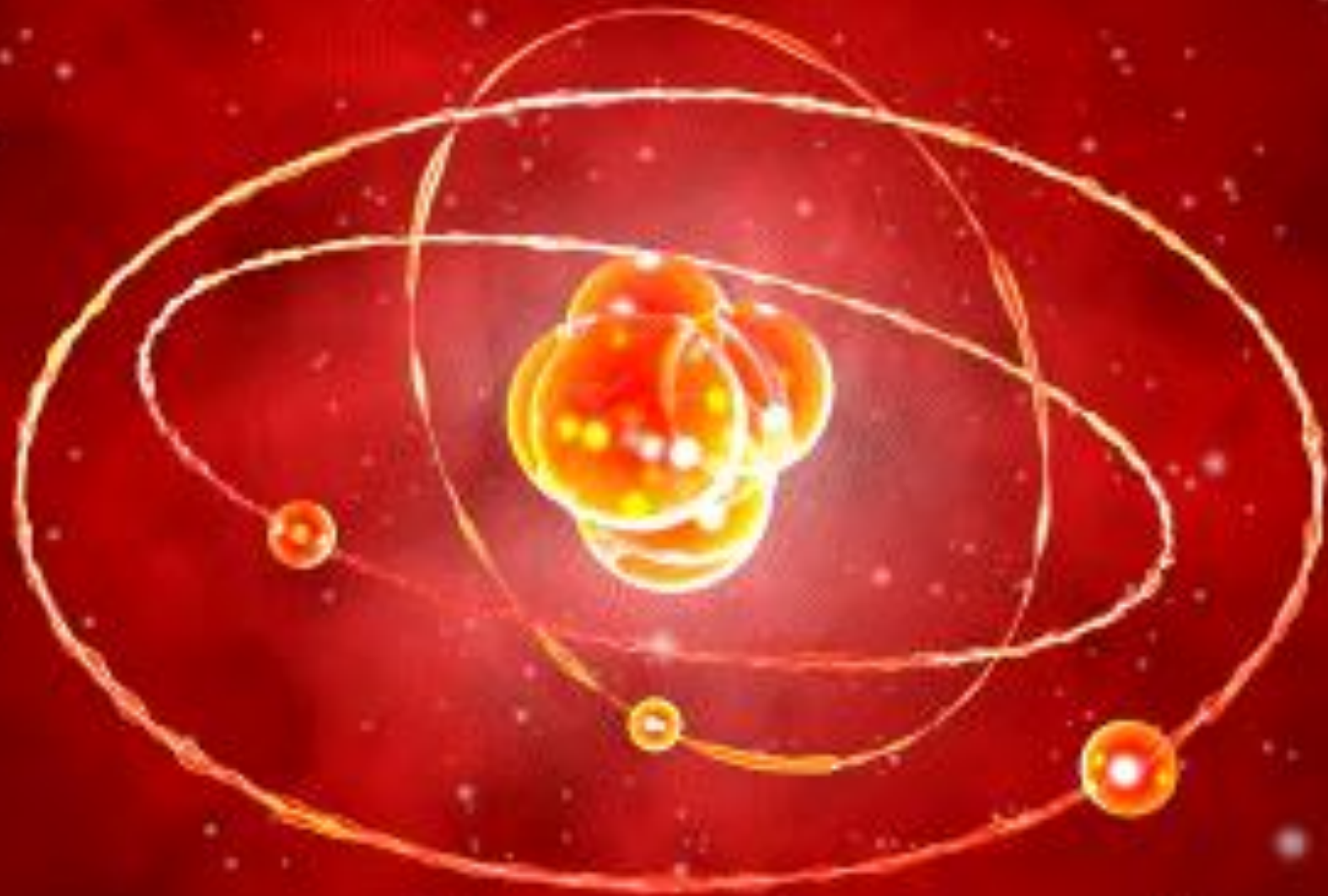
```
> git log
```

- Add all files to be committed

```
> git add .
```

- Switch to branch

```
> git checkout <branch-name>
```



ATOMIC COMMITS

USE MEANINGFUL COMMIT MESSAGES

	COMMENT	DATE
○	CREATED MAIN LOOP & TIMING CONTROL	14 HOURS AGO
○	ENABLED CONFIG FILE PARSING	9 HOURS AGO
○	MISC BUGFIXES	5 HOURS AGO
○	CODE ADDITIONS/EDITS	4 HOURS AGO
○	MORE CODE	4 HOURS AGO
○	HERE HAVE CODE	4 HOURS AGO
○	AAAAAAAAAA	3 HOURS AGO
○	ADKFJSLKDFJSDKLFJ	3 HOURS AGO
○	MY HANDS ARE TYPING WORDS	2 HOURS AGO
○	HAAAAAAAAAANDS	2 HOURS AGO

AS A PROJECT DRAGS ON, MY GIT COMMIT MESSAGES GET LESS AND LESS INFORMATIVE.

CONFLICT RESOLUTION

- Merging two branches may cause a conflict
- A conflict means **Git couldn't choose for you; it's not an error, it's a question.**
- You resolve by **editing**, not by commands alone.
- After resolving, you **add** the file (mark resolved) and **commit** to complete the merge.

A wide-angle photograph of a flat, open landscape, likely a field or plain. The foreground is filled with dry, yellowish-brown grass. In the middle ground, there are several small, dark, bushy shrubs scattered across the horizon. The sky is filled with large, white, fluffy clouds, and the overall lighting is bright, suggesting a sunny day. The word "DEMO" is overlaid in large, white, bold, sans-serif capital letters in the center of the image.

DEMO

MERGING AND CONFLICTS

- Merge branch into current branch

```
> git merge <branch>
```

- Abort failed merge (conflicts)

```
> git merge --abort
```

- Continue merge after resolving conflicts

```
> git merge --continue
```

- Add resolved file to be committed

```
> git add <file>
```

- Commit added files

```
> git commit
```

RESOLVING CONFLICTS IN VSCODE

```
213 |
    | Accept Current Change | Accept Incoming Change | Accept Both Changes | Compare Changes
214 | <<<<<<< HEAD (Current Change)
    | ~~~~~
215 |     readFileAndGenerating(file, split);
216 | =====
    | ~~~~~
217 |     readFileAndGenerate(input, split);
218 | >>>>>>> master (Incoming Change)
    | ~~~~~
219 |
220 |     i18nFileGenerate(output, options || null);
221 |
```

J

WORKFLOW EXAMPLE – SOLO DEV

1. Switch to main and get latest

```
> git checkout main
> git pull
```
2. Create a branch for your feature

```
> git checkout -b feature/add-login
```
3. Make changes

```
> git add .
> git commit -m "Add login form"
```
4. Sync with remote repo

```
> git push origin feature/add-login
```
5. Merge when done

```
> git checkout main
> git pull
> git merge feature/add-login
> git push
```
6. Clean up

```
> git branch -d feature/add-login
> git push origin --delete feature/add-login
```

WORKFLOW EXAMPLE – TEAM NOOBS

1. Switch to main and get latest

```
> git checkout main  
> git pull
```

2. Create a branch for your feature

```
> git checkout -b feature/add-login
```

3. Make changes

```
> git add .  
> git commit -m "Add login form"
```

4. Stay up to date

```
> git fetch origin  
> git rebase origin/main
```

5. Merge when done

```
> git checkout main  
> git merge --ff-only feature/add-login  
> git push
```

6. Clean up

```
> git branch -d feature/add-login
```

WORKFLOW EXAMPLE – TEAM PROS

1. Switch to main and get latest

```
> git checkout main  
> git pull
```

2. Create a branch for your feature

```
> git checkout -b feature/add-login
```

3. Make changes

```
> git add .  
> git commit -m "Add login form"
```

4. Sync with remote repo

```
> git push origin feature/add-login
```

5. Open a Pull request (On GitHub/GitLab)

- **Base branch:** main
- **Compare branch:** feature/add-login
- Add a **title**, **description**, and any context.

6. Review & discuss

- Comment or suggest changes
- Testing
- Fix feedback and push more commits (they appear automatically in the PR)

7. If main has changed since you branched

```
> git fetch origin  
> git merge origin/main  
> git push
```

8. Merge

- **Squash merge** → one clean commit
- **Rebase & merge** → keeps linear history
- **Normal merge** → not recommended

9. Clean up

```
> git branch -d feature/add-login  
> git push origin --delete feature/add-login
```



QUESTIONS

FEEDBACK



RESOURCES

