

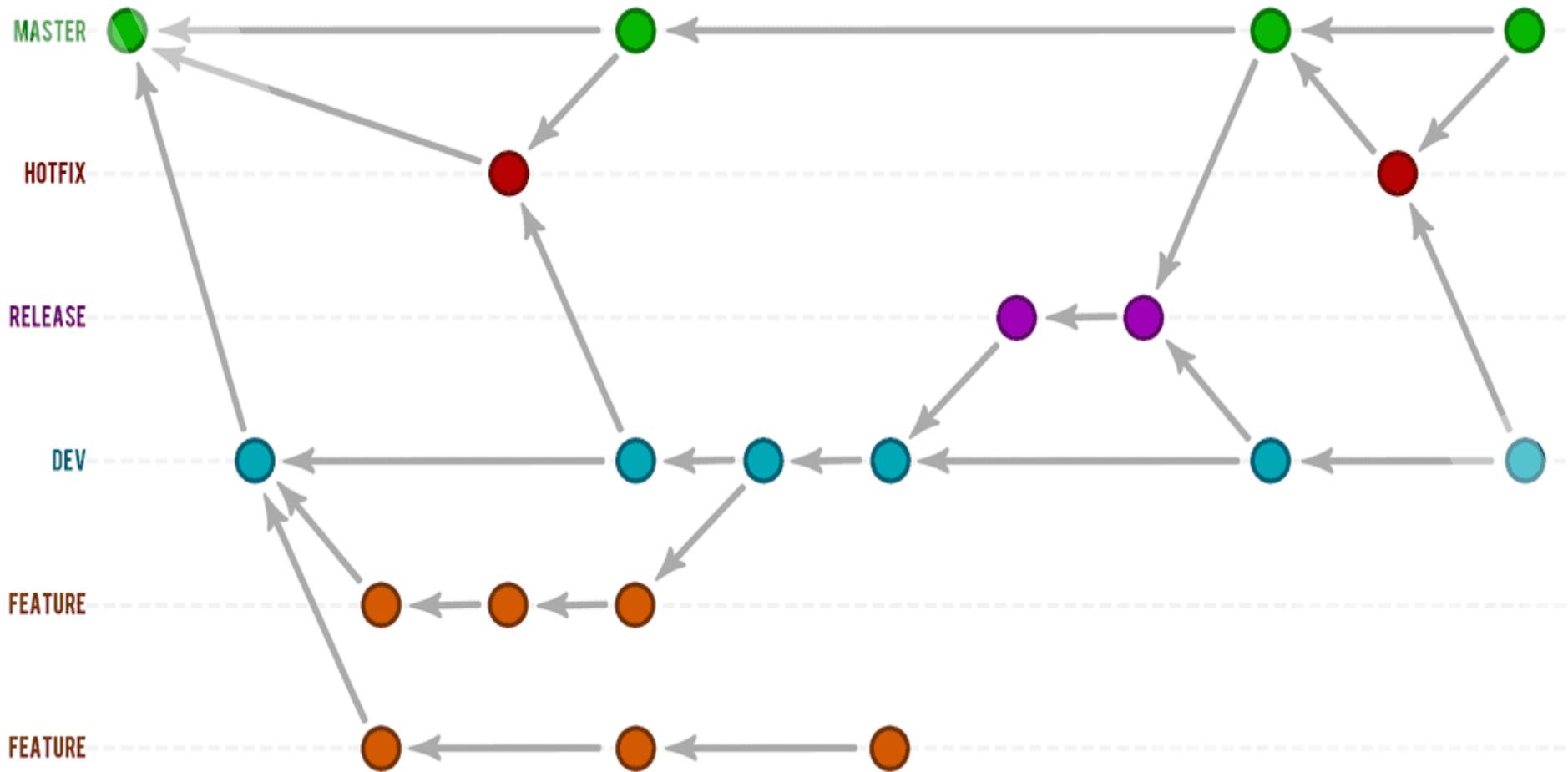
MARCELLO MISSIROLI
**Tecnologia
e Progettazione**
per il mondo *digitale*
e per il *web* II



000-000

5-0000000, 000000, 00000

0000. 00000000 000000000



1.

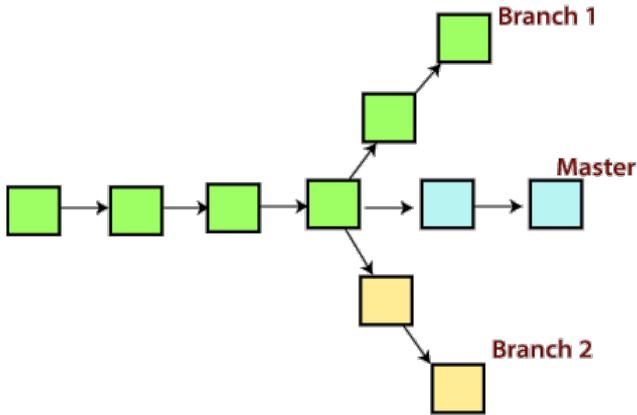
□□□□□□

□ 000000

000 000000 00 00000 0 000000, 000 000000000000 000 000000;
 0 000000 000000000 0 000000 000000000000 00000000000 000
 00000 00000000 000000000 0000 000000000 00000000000, 0000000
 000000000 “00000” 0 “00000000”, 00000000 0000000 0000000 00
 0000000. 0000000 00 0 0000000, 00 000 00 0000000000000 00
 0000000 000000000000, 0 0000 00 0000 00000000 00 000000000
 0000000000 000000 00000000000 00000000000.

Git

Git 是分布式版本控制系统，它允许开发者在本地计算机上保存完整的项目历史记录，而无需连接到中央服务器。Git 使用分布式架构，每个开发者都拥有完整的项目副本，这使得协作更加灵活和高效。Git 还支持分支管理，方便开发者进行并行开发和测试新功能。此外，Git 还提供了强大的版本回溯和合并功能，确保代码的稳定性和可追溯性。



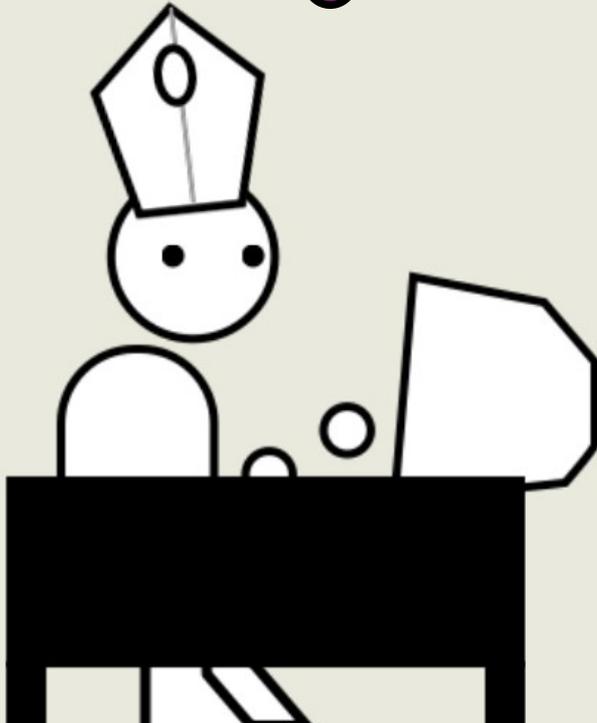
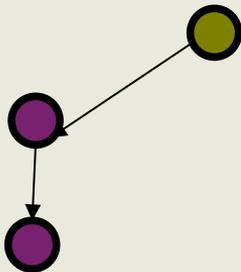
2.

□□□□□□ □□□□□□

□□□□□□□□

□□□□

“□□□□□□□□” □ □□□ □□ □□□□□□
□□□□□□□□. □ □□□□□□□ □□□□□□
□□□□□□□□ □□□□ □□ □□□□□□□□
□□□□□□□□



□ □□□ □□□□□□□□ -□ □□□□□□□□

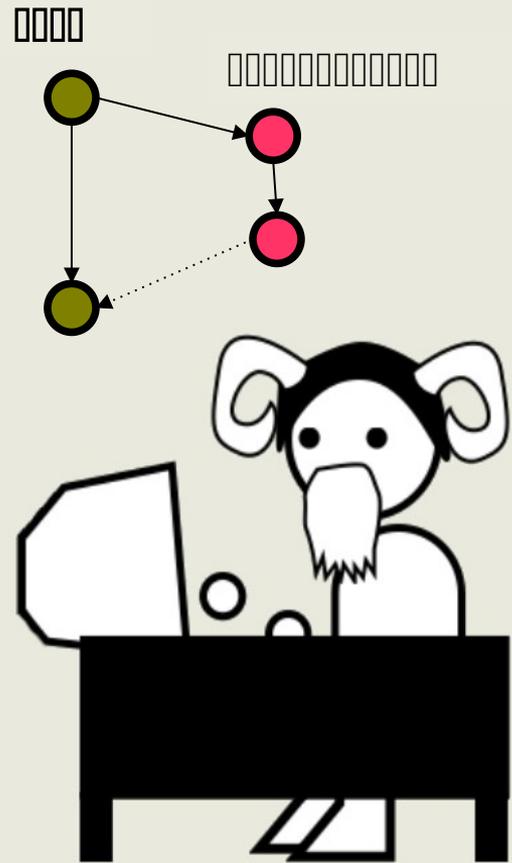
□ □□□ □□□□□□□□ -□□

“□□□□□□□□□□”

□□□□□

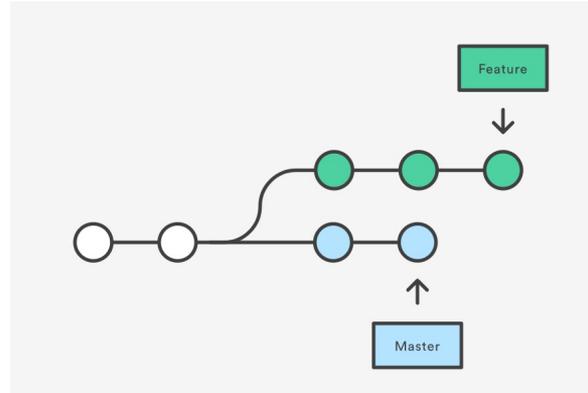
□□ □□□□□□ □□ □□□ □□□□□□□□□□□□ □□□ □□□□□ □□□□□□, □ □□□□□□□□ □□□□□□□□□□□ □ □□□□. □'□□□□□□□□□ □ □□□□□ □□□□□ (□□□□□□□□). □□□□ □□□ □□□□ □□□ □□□□ (□□□□ □ □□ □□□□□ □□ □□□□□), □□ □□□□□□□ □□□□□□□□□ □□□ □□□□□□□□□□.

```
0 000 00000000 0000
0 000 00000 000000000000
```



Git: 分支?

Git 分支功能 是 分支 功能
分支功能 是 分支功能 分支:
分支 是 分支。 是 分支功能
是 分支 分支功能 是 分支功能。



Git

Git 是什么？

Git 是一个分布式版本控制系统，它允许你跟踪和管理项目的源代码。它支持分支、合并、提交、推送和拉取等操作。Git 的核心理念是“本地优先”，即所有的操作都是在本地完成的，只有当你需要与他人协作时，才会与远程仓库进行交互。

Git 的架构非常灵活，你可以选择使用集中式仓库（如 GitHub、GitLab）或者分布式仓库（如 Bitbucket）。Git 的分支模型非常强大，可以让你轻松地管理不同的开发任务。Git 的提交历史是可读的，你可以随时查看项目的历史变化。

Git

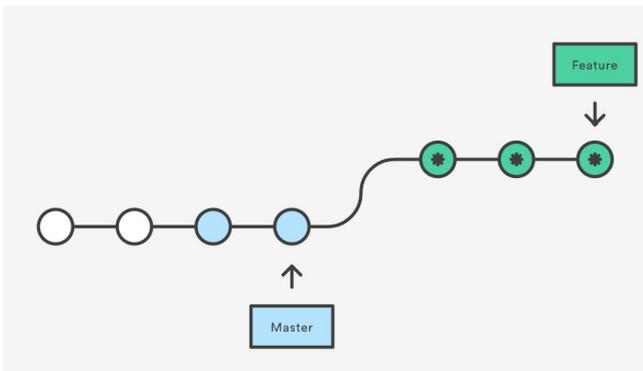
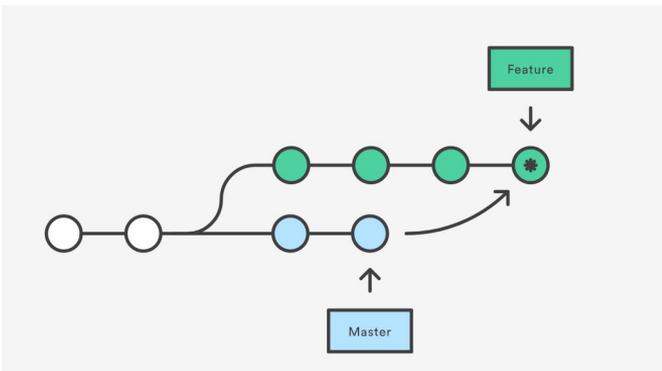
Git 是什么？

Git 是什么？

Git 是一个分布式版本控制系统，它允许你跟踪和管理项目的源代码。它支持分支、合并、提交、推送和拉取等操作。Git 的核心理念是“本地优先”，即所有的操作都是在本地完成的，只有当你需要与他人协作时，才会与远程仓库进行交互。

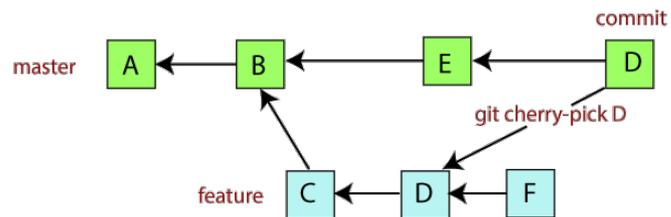
Git 的架构非常灵活，你可以选择使用集中式仓库（如 GitHub、GitLab）或者分布式仓库（如 Bitbucket）。Git 的分支模型非常强大，可以让你轻松地管理不同的开发任务。Git 的提交历史是可读的，你可以随时查看项目的历史变化。

Git 是什么？



Git 分支管理

Git 分支管理 (Git Branching Model) 是一种用于管理代码分支的策略。它允许开发者在开发新功能、修复bug或进行实验时，从主分支（通常是 master）创建新的分支。当开发完成或需要回滚时，可以轻松地将更改合并回主分支。这种策略有助于保持代码库的稳定性和可追溯性。





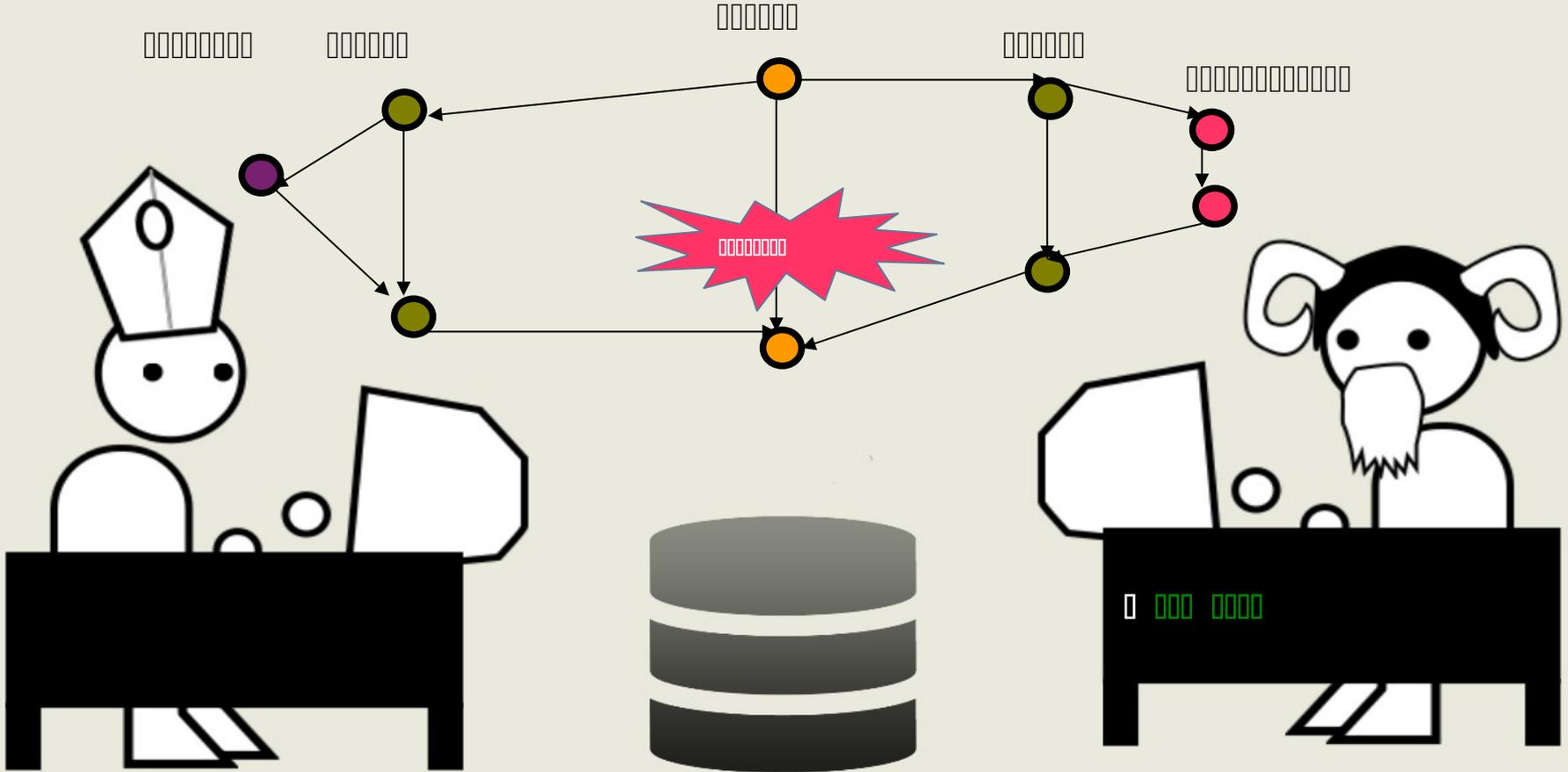
4.

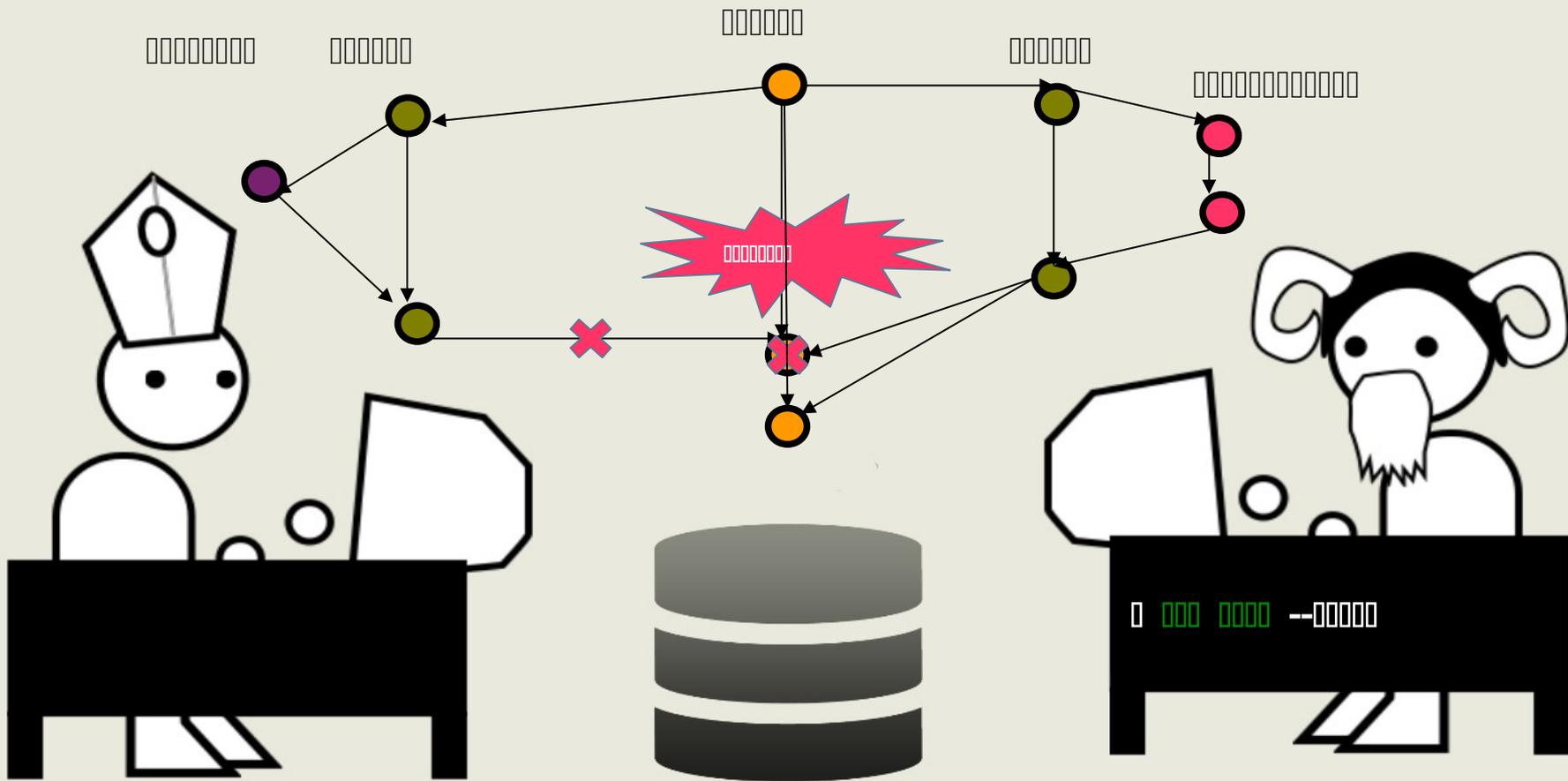
□□□□□□□□ □□□ □ □□□□□□□

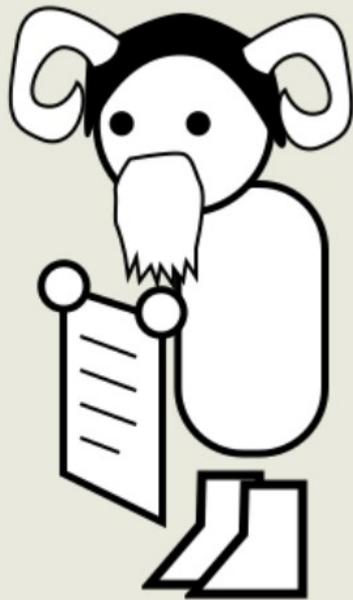
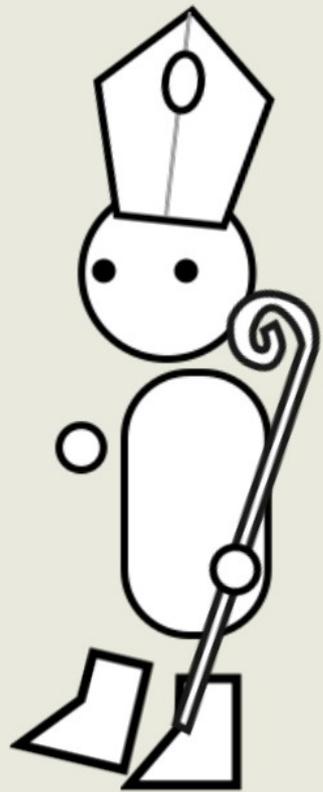


အိတ်အိတ်အိတ် အိတ် အိတ်အိတ်

အိတ်အိတ် အိတ်အိတ် အိတ်အိတ်, ဝ အိတ်အိတ်အိတ်အိတ် အိတ်အိတ် အိတ်အိတ် အိတ်အိတ် အိတ်အိတ် အိတ်အိတ်အိတ်အိတ် အိတ်အိတ်. အိတ်အိတ် အိတ် အိတ်အိတ် အိတ်အိတ် အိတ်အိတ်အိတ်အိတ် (အိတ်အိတ် အိတ် အိတ်အိတ်အိတ်, အိတ် အိတ်အိတ် အိတ်အိတ်)

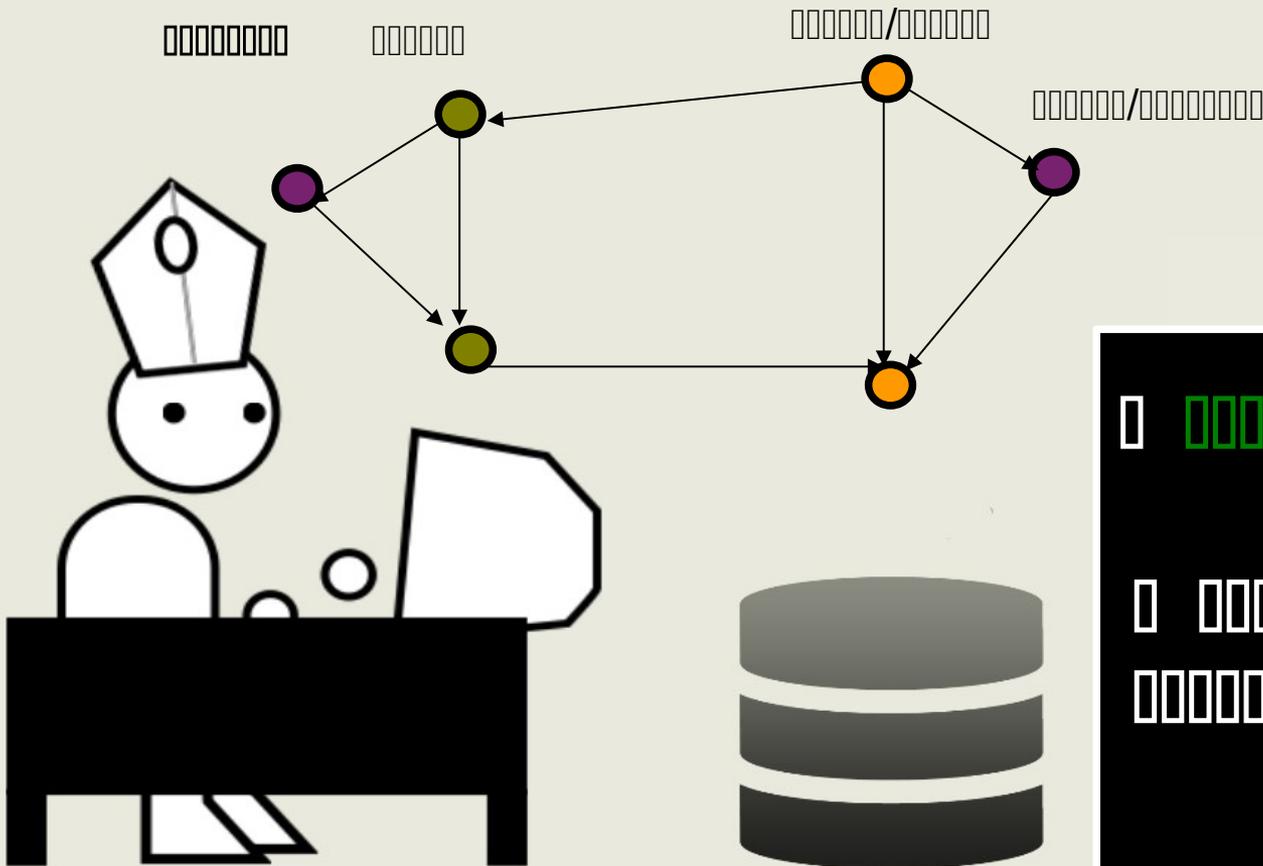






□□□□□□ □□□□□□

□□□□□□□□□□□□, □□ □□□□ □□□□□□ □□□□□□ □□ □□□□□□ □□□□□□ (□□□□□□□□□□□□). □□□□□□ □□□□□□ □□ □□□□ □□□□□□□□□□□□ □□□□□□ □□□□□□□□□□□□.



```
□ □□ □□□□□□□ □□□□□□□□  
□ □□ □□□□ -□ □□□□□□  
□□□□□□□□
```

□□□□□□□□□□

- » □□□□□□□□□□ □□ □□□□□□ □□□□□□ : □□□ □□□□□□ -□ [□□□□□]
- » □□□□□□□□□□ □□ □□□□□□ □□□□□□: □□□ □□□□ □□□□□□ --
□□□□□□ [□□□□□]
- » □□□□□□□□□□ □□ □□□□□□ □□□□□□: □□□ □□□□□□ -□ [□□□□□]
- » □□□ □□□ □□□ □□□□□ □□□□□□□□, “□□□□□”: □□□ □□□ --□□□□□□□□□□
--□□□□□□□□□□ --□□□□□□ □□□□□□

5.

□□□□□□ □□ □□□□□□

□□□□ □□□□ □□ □□□□□□□□ □□ □□□□□□

‘0 0000000000?

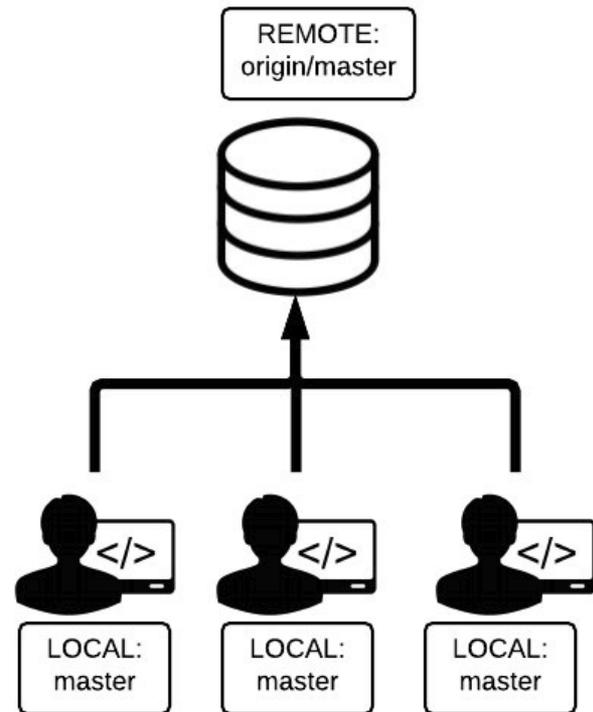
00000000 00000 000000 0
 000000000000 0000 00
 000000000. 000 0
 000000000000000000
 00000000000 0 00 000000
 0000 00000 00 0000000 00
 000000000 (0 000
 0000000000). 0000000000
 0000000.





Git 本地仓库与远程仓库

Git 本地仓库 (即 .git) 是 Git 本地仓库的
数据仓库 (“本地仓库”)。
本地仓库包含所有提交的数据以及提交历史记录。
本地仓库包含所有提交的数据以及提交历史记录。
本地仓库包含所有提交的数据以及提交历史记录。
本地仓库包含所有提交的数据以及提交历史记录。
本地仓库包含所有提交的数据以及提交历史记录。
本地仓库包含所有提交的数据以及提交历史记录。
本地仓库包含所有提交的数据以及提交历史记录。



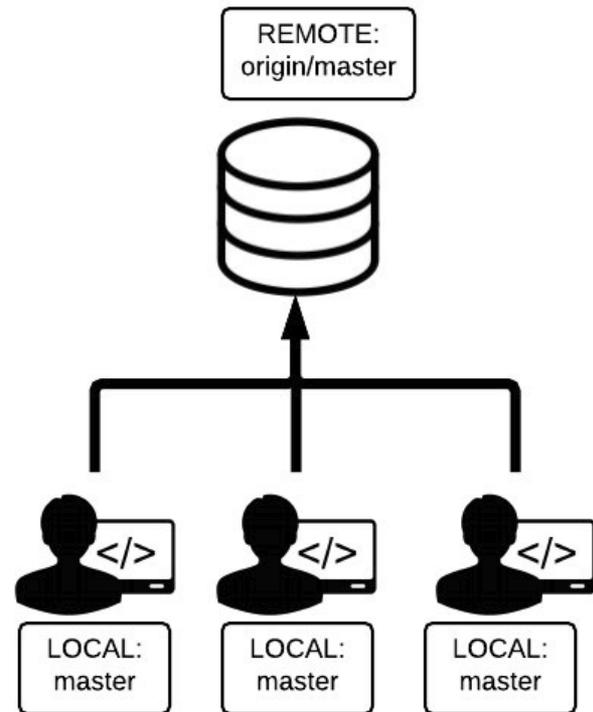
分布式数据库

分布式数据库

- » 分布式数据库 数据库
- » 分布式数据库 数据库 数据库

分布式数据库

- » 数据库 数据库

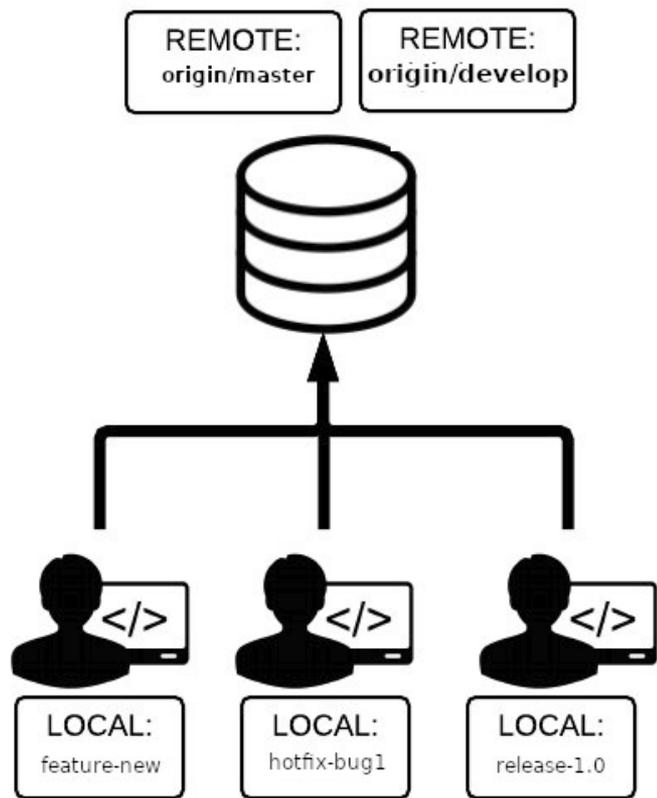




Git

Git 是分布式版本控制系统，由 Linus Torvalds 于 2005 年创建。它最初是为 Linux 内核开发而设计的，但现在广泛应用于各种软件开发项目。

- » 分布式版本控制 — 每个开发人员都有一个完整的项目副本
- » 分支管理 — 方便地管理开发中的新功能

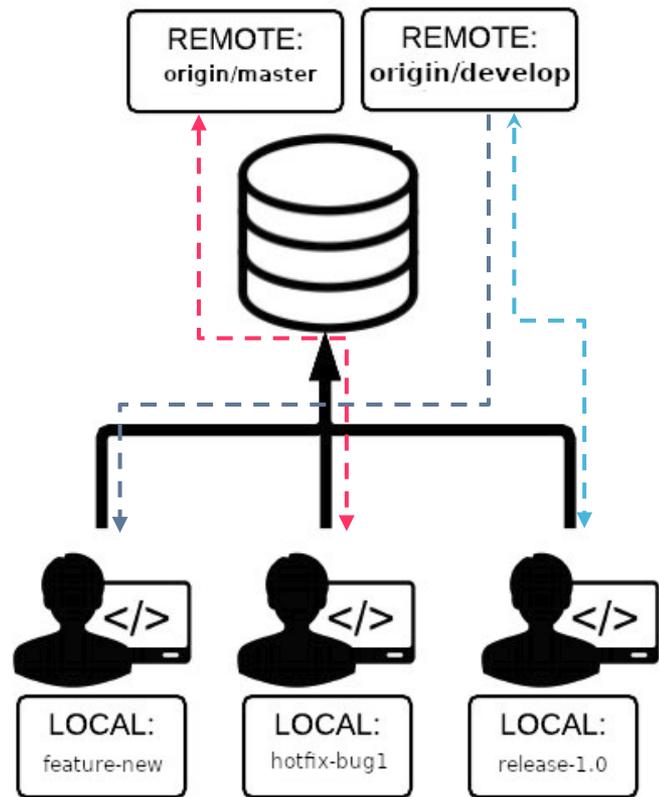


Git Branching Model (2)

Git Branching Model (2)의 주요 원칙과 구조를 설명합니다:

- » `origin/master` — 항상 배포 가능한 상태를 나타내는 유일한 브랜치입니다. 새로운 릴리스를 준비할 때 사용됩니다.
- » `origin/develop` — 새로운 기능을 개발하거나 버그를 수정할 때 사용되는 기본 브랜치입니다. 모든 개발 작업은 이 브랜치에서 시작됩니다.
- » `origin/feature-*` — 새로운 기능을 개발할 때 사용되는 브랜치입니다. 작업이 완료되면 `develop` 브랜치로 병합됩니다.

. [Git Branching Model](#) 문서 참조.



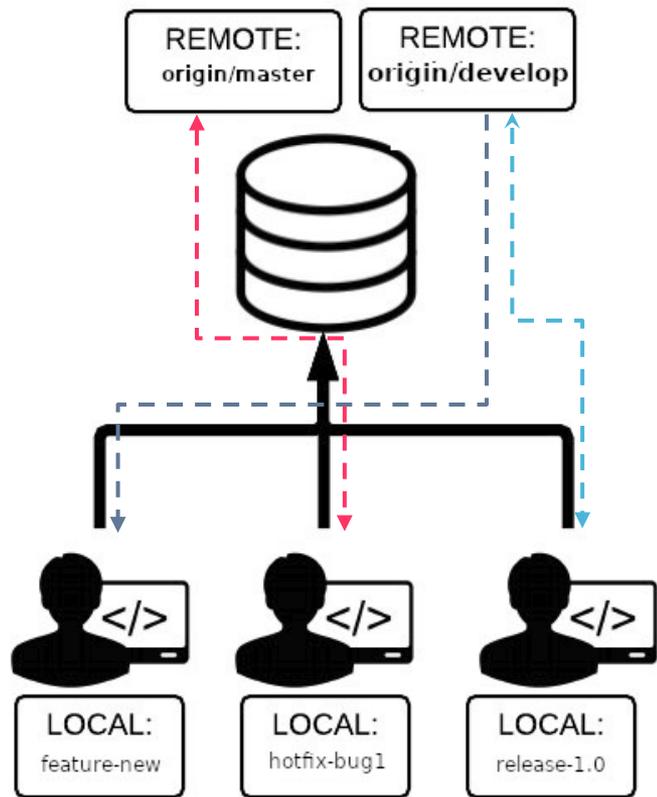
Git 分支管理

Git 分支管理

- » 分支管理是 Git 的一个重要特性
- » 分支管理可以帮助我们管理不同的开发任务
- » 分支管理可以帮助我们管理不同的版本

Git 分支管理

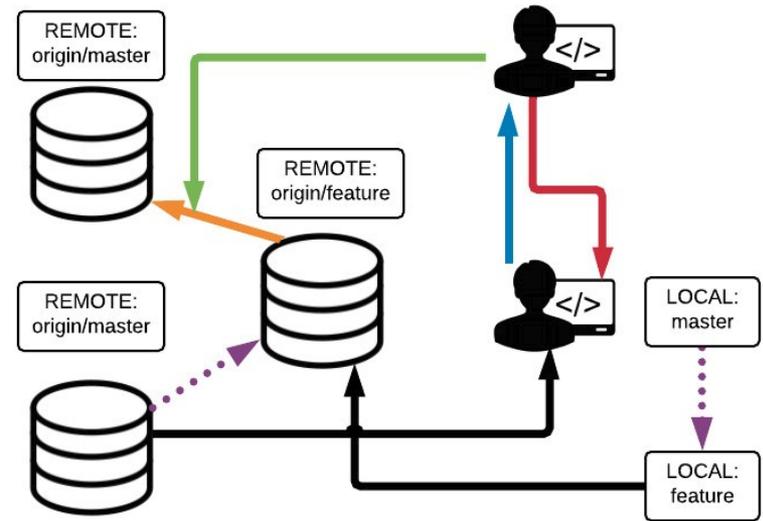
- » 分支管理可以帮助我们管理不同的开发任务
- » 分支管理可以帮助我们管理不同的版本
- » 分支管理可以帮助我们管理不同的版本
- » 分支管理可以帮助我们管理不同的版本





Git 本地分支 远程分支 本地仓库

Git 本地分支 本地仓库 本地仓库 本地仓库。
本地仓库 本地仓库 本地仓库 本地仓库，本地仓库
本地仓库 本地仓库；本地 本地仓库 本地仓库 本地仓库
本地仓库 本地仓库 本地仓库。本地 本地仓库 本地仓库
本地仓库 本地仓库 (本地仓库 本地仓库 本地仓库
本地仓库)。本地仓库 本地仓库 本地仓库 本地仓库
本地仓库 本地仓库 本地仓库 本地仓库。本地仓库 本地
本地仓库 本地仓库





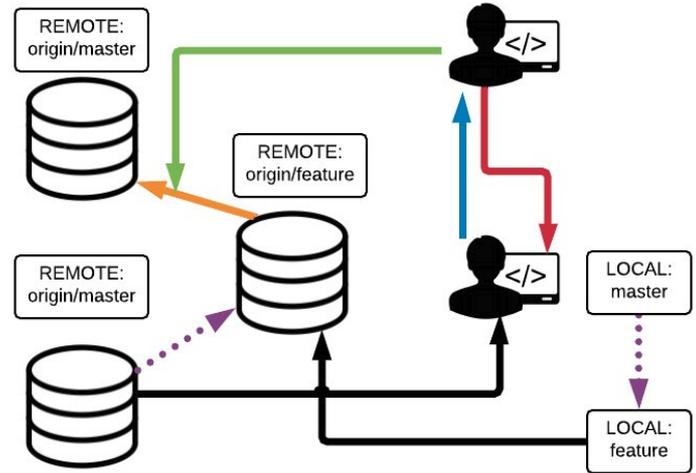
Git 分支管理

分支管理

- » 分支管理是 Git 中最重要的功能之一，它允许你在不影响主分支的情况下进行开发、测试新功能。
- » 分支管理可以让你在开发新功能时，随时回到主分支，而不需要担心破坏主分支的代码。

分支管理

- » 分支管理可以让你在开发新功能时，随时回到主分支，而不需要担心破坏主分支的代码。
- » 分支管理可以让你在开发新功能时，随时回到主分支，而不需要担心破坏主分支的代码。





□□□□□□□□ &

□□□□/□□□□□□

□□□□□□□□



□ 0000000?

0000000 0000000000 0 0000 0000 000 00000000000 00000000 00
0000 00000000 (00000000, 00000000000)/000000 00000000 (00000000).

0000 0000 00000000 00 00000000000 0 0000 00000000000 000000
00000000000, 00 000000 000 00000000000 00 0000000000 00 0000000000
00000000 00 000000 000000000000000. 00000000 000000 00000000000 00
00000000000 00000000 00000000000000, 00000000000 00000000 00 00000000000.



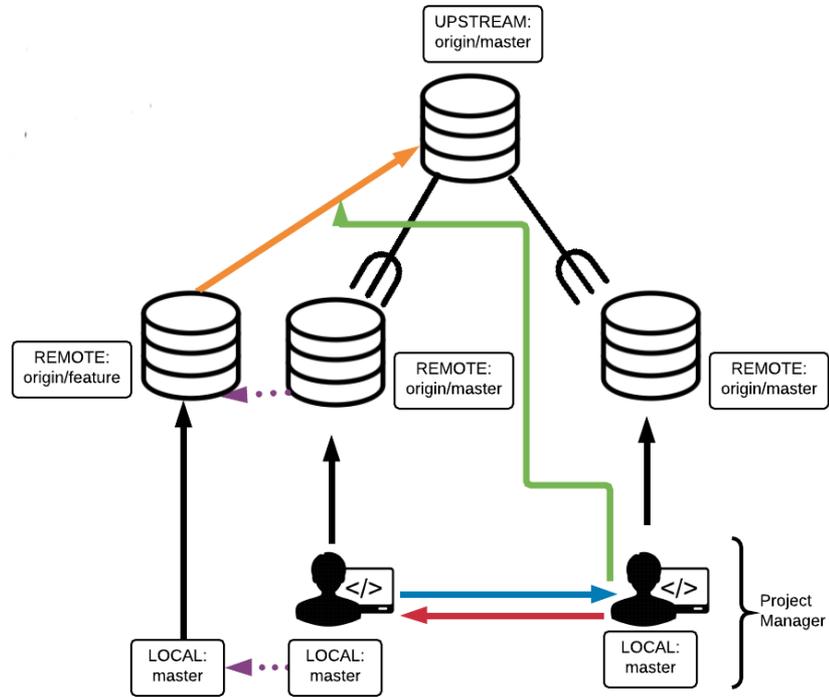
Git 工作流

Git 工作流 (Workflow) 是指团队在开发过程中遵循的一套规则和流程。

不同的团队可能会采用不同的工作流，但有一些常见的模式。了解这些模式可以帮助你更好地融入团队或为自己的团队制定规则。

本文将介绍几种常见的 Git 工作流，包括分支策略、提交规范以及合并流程。这些工作流旨在提高团队协作效率，减少冲突，并确保代码的质量和稳定性。

在开始之前，我们需要明确一些基本概念。Git 工作流的核心在于如何管理分支和提交。不同的分支通常用于不同的开发阶段，如开发、测试和生产。





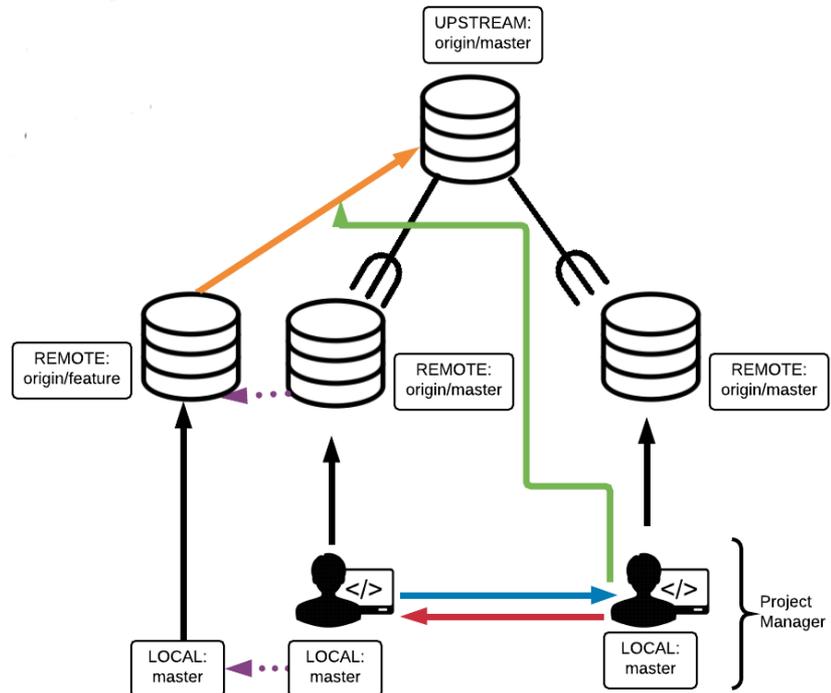
Git 專案管理

Git 專案管理

- » 專案管理 專案管理
- 專案管理 專案管理, 專案管理 專案管理
- 專案管理 專案管理
- 專案管理 專案管理
- 專案管理 專案管理

Git 專案管理

- » 專案管理 專案管理
- » 專案管理 專案管理
- 專案管理 專案管理
- 專案管理 專案管理



በግንባታ ላይ የሚደረግ ጥቅም

ጥቅም

ጥቅም ለግንባታ ላይ ሲደረግ ለግንባታ ላይ ሲደረግ

ጥቅም

ጥቅም ለግንባታ ላይ ሲደረግ ለግንባታ ላይ ሲደረግ

ጥቅም

ጥቅም ለግንባታ ላይ ሲደረግ ለግንባታ ላይ ሲደረግ

ጥቅም

ጥቅም ለግንባታ ላይ ሲደረግ ለግንባታ ላይ ሲደረግ

ጥቅም

ጥቅም ለግንባታ ላይ ሲደረግ ለግንባታ ላይ ሲደረግ

ጥቅም/ጥቅም ጥቅም

ጥቅም ለግንባታ ላይ ሲደረግ ለግንባታ ላይ ሲደረግ



6.

□□□□□□□□

□□□□□□□□□□□□□□□□





□□□□□□□□□□□□

□□□□ □□□□□□□□ □□□□ □□□□□□□□ □□□ □□□□□□
 □□□□□□□□□□□□. □□ □□□□□□□□, □□ □□□□□ □□□□ □□ □□□□□□ □□
 □□□□□□□□□□□□ □□□□□□□□□□ □□□□□□ □□ □□□□□□□□ □□□
 □□ □□□□□□□□□□ □□□ □□□□□□□□□□.

□□□ □□□□□□ □ □□□□□ □□□□□□ □□□□□□ □□ □□□□□□ □□□ □□ □□.



git-quick-stats

```

git quick-stats --help
git quick-stats --generate
git quick-stats --list
git quick-stats --suggest
git quick-stats --branch-tree-view
git quick-stats --all-branches
git quick-stats --all-contributors
git quick-stats --git-commits-per-author
git quick-stats --git-commits-per-date
git quick-stats --git-commits-per-month
git quick-stats --git-commits-per-weekday
git quick-stats --git-commits-per-hour
git quick-stats --git-commits-per-hour-by-author
git quick-stats --git-commits-per-timezone
git quick-stats --git-commits-per-timezone-by-author
git quick-stats --code-reviewers

```

```

Terminal — git-quick-stats

Generate:
 1) Contribution stats (by author)
 2) Contribution stats (by author) on a specific branch
 3) Git changelogs (last 10 days)
 4) Git changelogs by author
 5) My daily status
 6) Output daily stats by branch in CSV format
 7) Save git log output in JSON format

List:
 8) Branch tree view (last 10)
 9) All branches (sorted by most recent commit)
10) All contributors (sorted by name)
11) Git commits per author
12) Git commits per date
13) Git commits per month
14) Git commits per weekday
15) Git commits per hour
16) Git commits per hour by author
17) Git commits per timezone
18) Git commits per timezone by author

Suggest:
19) Code reviewers (based on git history)

Please enter a menu option or press Enter to exit.
>

```



■■■■■■■■■■■■■■ ■■■ ■■■■■■■■■■■■
 ■■■ ■■■■■ ■■■ ■■■■■■■ ■■
 ■■■■■■■ ■■■■■■■■ ■■■■■■■■■■■■■■.

Le seguenti informazioni storiche sui commit, per autore, sono state trovate nel repository.

Autore v	Commit	Inserimenti	Rimoziioni	% dei cambiamenti
 S1mone	11	163	12	53.19
 samuliga	5	149	5	46.81



Di seguito il numero di righe da ogni autore che sono sopravvissute a sono ancora intatte nella versione corrente.

Autore v	Righe	Stabilità	Età	% in commenti
 S1mone	175	107.4	0.1	16.00
 samuliga	120	80.5	0.0	12.50



La seguente timeline storica è stata ricavata dal repository.

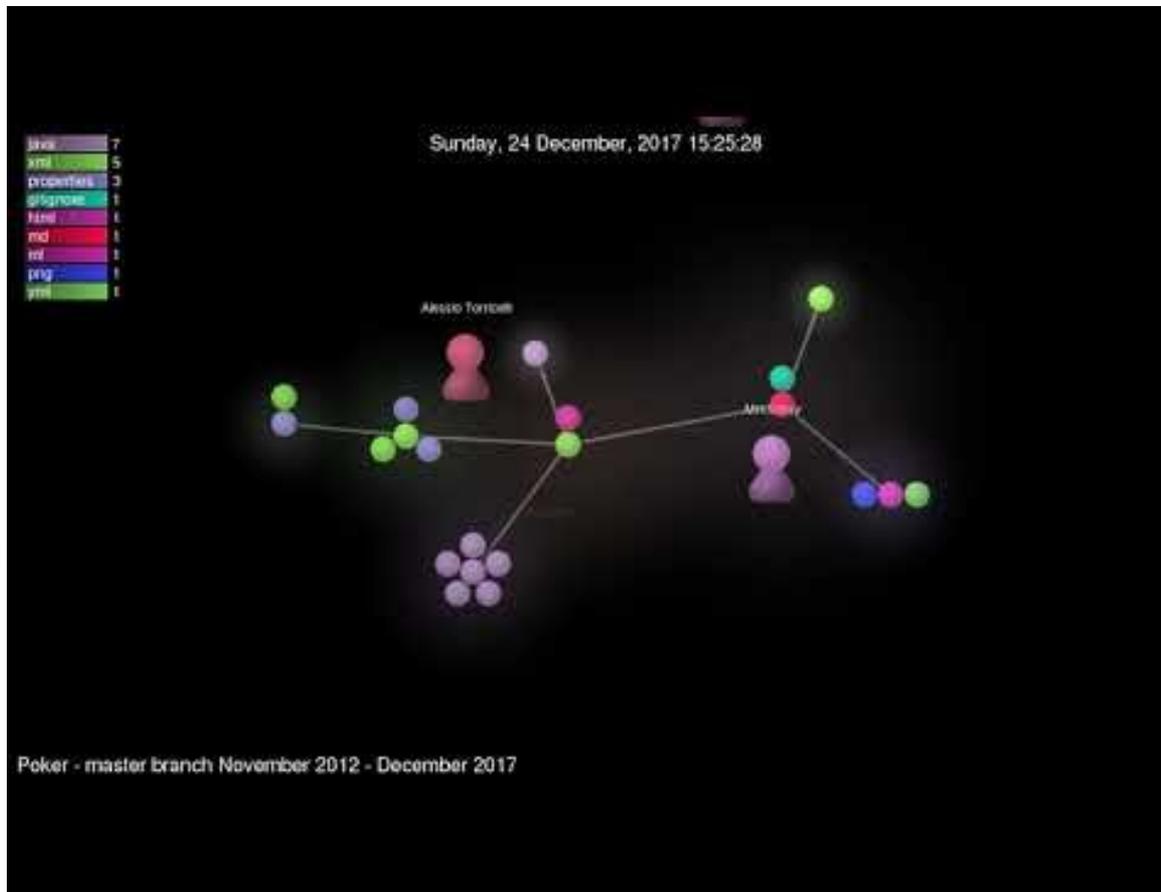
Autore	2021W12
 S1mone	
 samuliga	
Righe modificate:	329

Non sono state rilevate violazioni delle metriche nel repository.

Git

Git 是分布式版本控制系统，它可以在没有中央服务器的情况下跟踪和管理项目的变化。它允许用户将文件的变化保存到本地或远程的仓库中，并可以随时从仓库中恢复文件。Git 还支持分支、合并、标签等功能，使得团队协作和版本管理变得更加简单和高效。

Git 的分布式特性使得它非常适合用于团队协作和版本管理。每个开发人员都可以拥有自己的本地仓库，并可以随时从远程仓库中同步最新的代码。此外，Git 还支持分支和合并，使得开发人员可以在不影响生产环境的情况下进行开发和测试。Git 的标签功能可以用于标记特定的版本，方便回溯和发布。



物理學 科學 物理學 物理學 科學
物理學: 物理學, 物理學 物理學, 物理學
物理學物理學

-- 物理學物理學 (1804-1881)



“**በገንዘብ ላይ ትኩረት!**”

ገንዘብ ላይ ትኩረት ለማድረግ ለሁሉም ሰራተኞች ማሳሰቢያ ማድረግ አለብን። ለሰራተኞቹ ስራ ላይ ትኩረት ማድረግ ለሰራተኞች ስራ ላይ ትኩረት ማድረግ አለብን።

ሰራተኞች ስራ ላይ ትኩረት ማድረግ ለሰራተኞች ስራ ላይ ትኩረት ማድረግ አለብን። ሰራተኞች ስራ ላይ ትኩረት ማድረግ ለሰራተኞች ስራ ላይ ትኩረት ማድረግ አለብን። ሰራተኞች ስራ ላይ ትኩረት ማድረግ ለሰራተኞች ስራ ላይ ትኩረት ማድረግ አለብን።

ሰራተኞች ስራ ላይ ትኩረት ማድረግ ለሰራተኞች ስራ ላይ ትኩረት ማድረግ አለብን። ሰራተኞች ስራ ላይ ትኩረት ማድረግ ለሰራተኞች ስራ ላይ ትኩረት ማድረግ አለብን።

ሰራተኞች ስራ ላይ ትኩረት ማድረግ ለሰራተኞች ስራ ላይ ትኩረት ማድረግ አለብን።

HTML 2 (HTML)

HTML 2 HTML, HTML
 HTML HTML HTML HTML
 HTML HTML HTML HTML
 HTML HTML HTML HTML
 HTML HTML HTML HTML,
 HTML HTML HTML HTML HTML
 HTML (HTML HTML HTML HTML HTML
 HTML' HTML HTML).

HTML HTML - "HTML HTML HTML.

>

>

HTML-HTML-HTML: HTML

<HTML.HTML>

HTML-HTML-HTML: HTML.HTML

<HTML.HTML.HTML.HTML.HTML.HTML>"

HTML HTML

በግንባታው ላይ ያለው ስሜት (ግንባታው -ግንባታው)

በግንባታው ላይ ስሜት,
በግንባታው ላይ ስሜት ስሜት
በግንባታው ላይ ስሜት ስሜት

በግንባታው ላይ ስሜት.ግንባታው ስሜት.ግንባታው
በግንባታው ላይ -ግንባታው ስሜት

□□□□□□!

...□ □□□□□□ □□□□

□□□□□□□□...



□□□□□□

□□□□□□□□□□□□ □ :

- » □□□□□□□□□□□□ □□□□□□ □□ □□□□□□□□□□□□□□
- » □□□□□□□□□□□□ □□ □□□□□□□□
- » □□□□ □□□□□□ (□□□□.□□□□□□□□□□□□.□□□□)
- » □□□□ □□□□□□□□□□ (□□□□□□□□.□□□□/□□□□□□□□)
- » □□□□□□□□ □□□□

□□□□□□□□ □□□□□□□□□□ □□□□□□□□□□□□ □□□□

□□□□□□□□□□□□□□□□□□□□ □□-□□ 3.0