

Контроль версій в Git

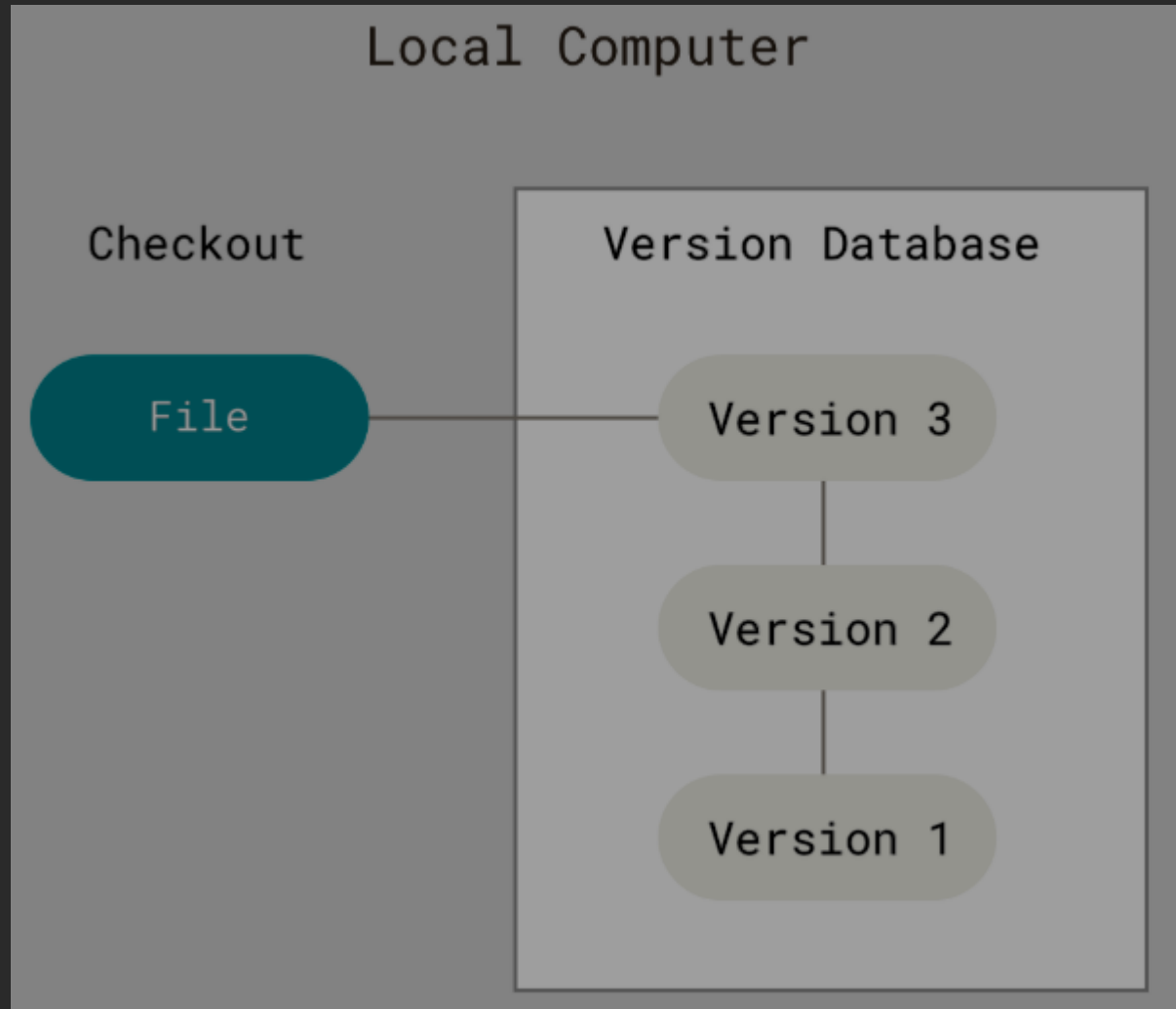


git

План:

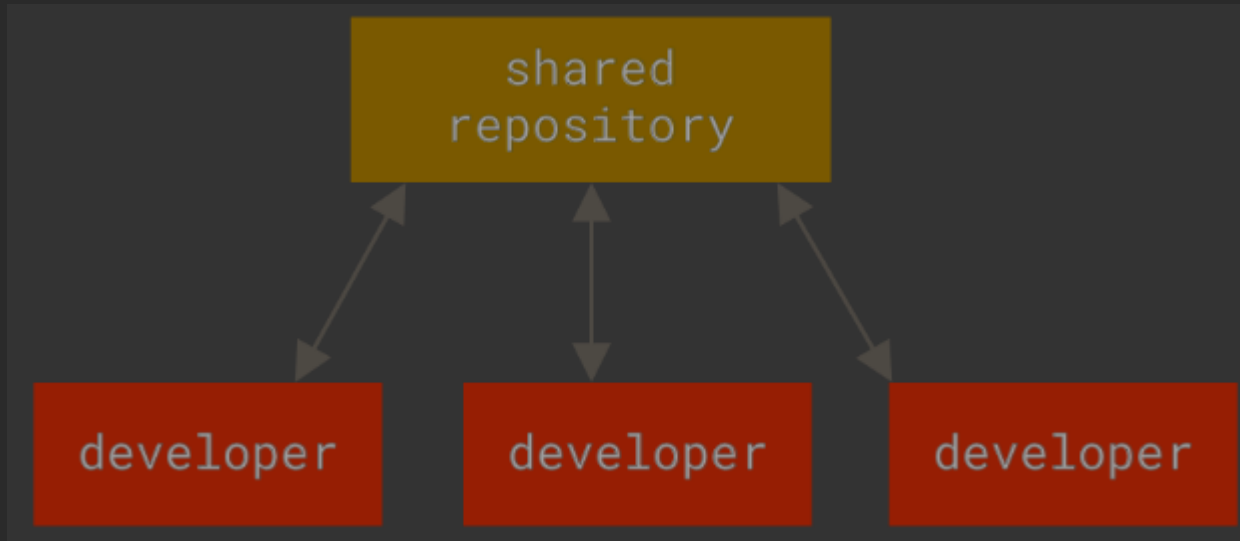
- Види систем для контролю версій;
- Основи GIT;
- Команди;
- Робота з гілками;
- Розгалуження.

Локальний контроль версій



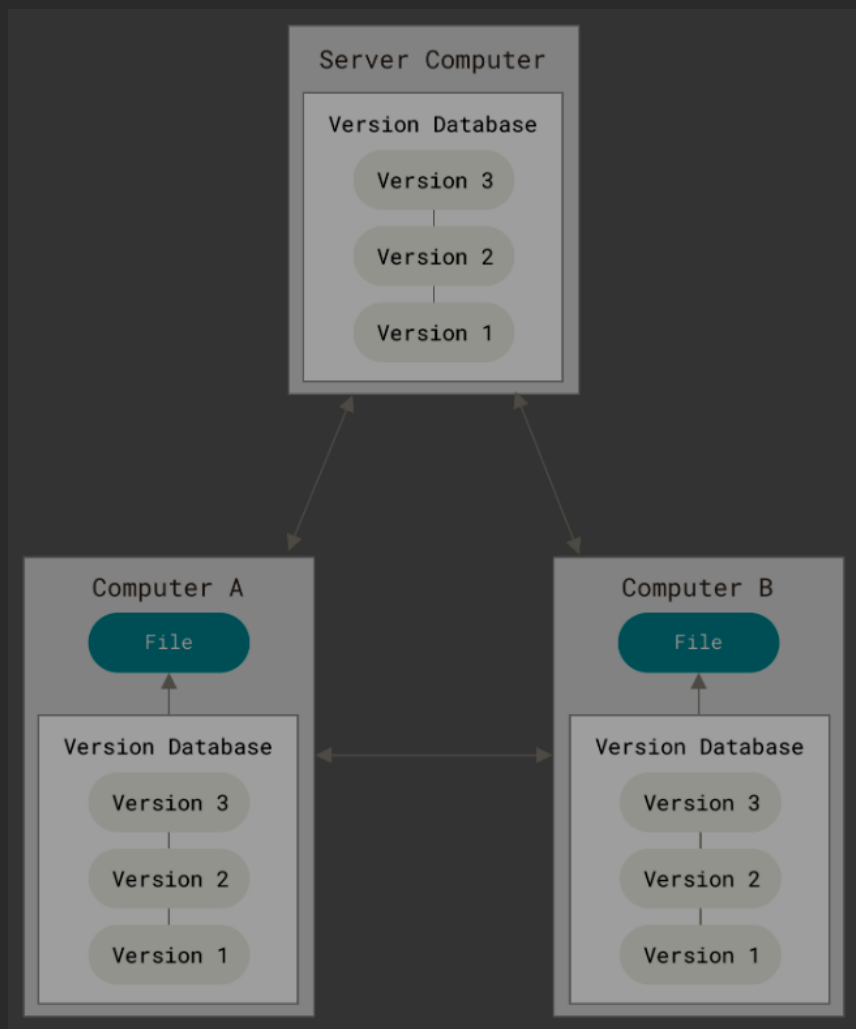
Одним із найпопулярніших інструментів системи контролю версій (VCS) була система під назвою RCS, яка й досі постачається з багатьма комп'ютерами. [RCS](#) працює, зберігаючи набори патчів (тобто відмінності між файлами) у спеціальному форматі на диску; потім вона може відтворити вигляд будь-якого файлу в будь-який момент часу, додавши всі патчі.

Централізовані системи контролю версій



Така схема пропонує багато переваг, особливо порівняно з локальними системами контролю версій (VCS). Наприклад, кожен певною мірою знає, що робить кожен інший у проекті. Адміністратори мають детальний контроль над тим, хто що може робити, і адмініструвати CVCS набагато легше, ніж мати справу з локальними базами даних на кожному клієнті.

Розподілені системи контролю версій



У Distributed Version Control Systems (таких як Git, Mercurial або Darcs) клієнти не просто переглядають останній знімок файлів, а скоріше повністю відображають репозиторій, включаючи його повну історію. Таким чином, якщо якийсь сервер вийде з ладу, а ці системи співпрацювали через цей сервер, будь-який з клієнтських репозиторіїв можна скопіювати назад на сервер для його відновлення. Кожен клон насправді є повною резервною копією всіх даних.

Відмінність між Git



Зберігання даних як змін до базової версії кожного файлу

Основна відмінність між Git та будь-якою іншою системою контролю версій (включно зі Subversion та її подібними системами) полягає в тому, як Git обробляє свої дані. Концептуально, більшість інших систем зберігають інформацію як список змін у файлах. Ці інші системи (CVS, Subversion, Perforce тощо) розглядають інформацію, яку вони зберігають, як набір файлів та змін, внесених до кожного файлу з часом (це зазвичай називають **дельта-системою** контролю версій).

Git



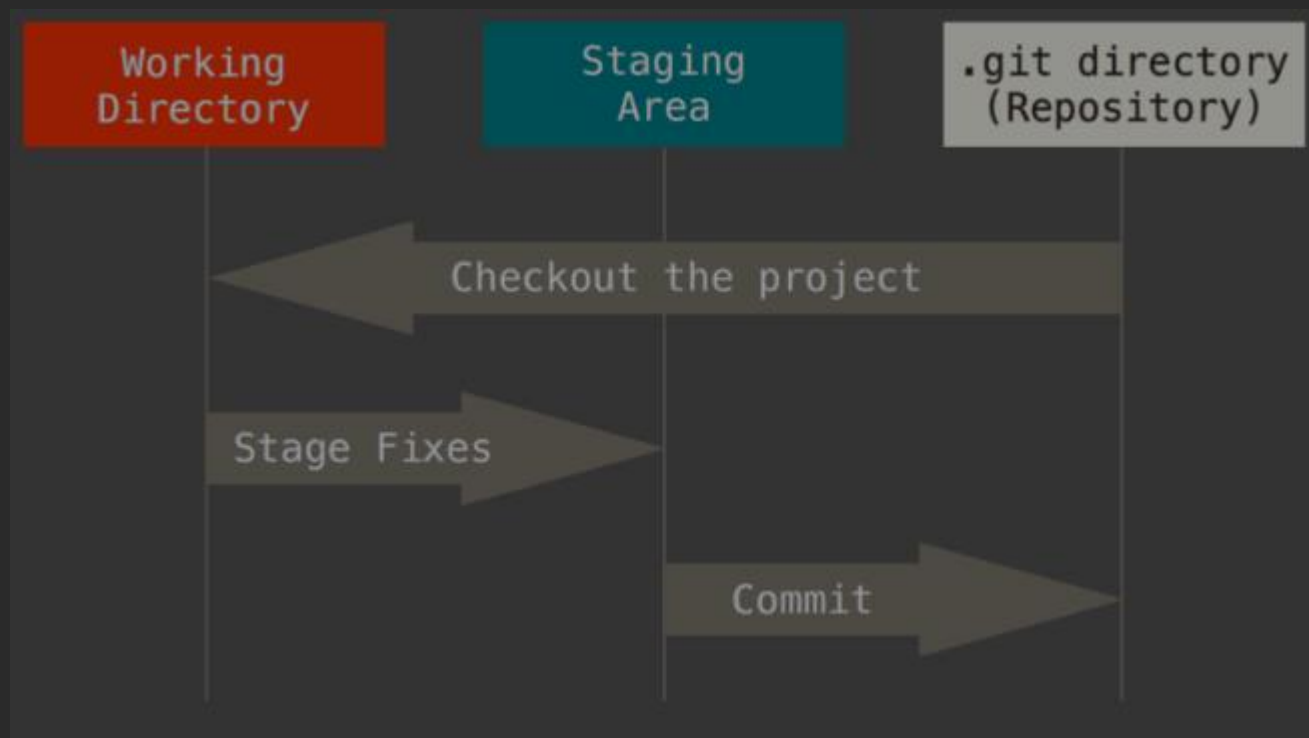
Git сприймає свої дані радше як серію знімків мініатюрної файлової системи. У Git щоразу, коли ви комітите або зберігаєте стан вашого проєкту, Git фактично робить знімок того, як виглядають усі ваші файли в цей момент, і зберігає посилання на цей знімок. Для ефективності, якщо файли не змінилися, Git не зберігає файл повторно, а лише посилання на попередній ідентичний файл, який він уже зберіг. Git сприймає свої дані радше як **потік знімків**.

Хеш SHA-1: Наріжний Камінь Git

Одним з найважливіших механізмів, який Git використовує для ідентифікації та контролю вмісту, є алгоритм хешування SHA-1 (Secure Hash Algorithm 1). Це криптографічна хеш-функція, яка генерує унікальний «відбиток» для будь-якого файлу або каталогу.

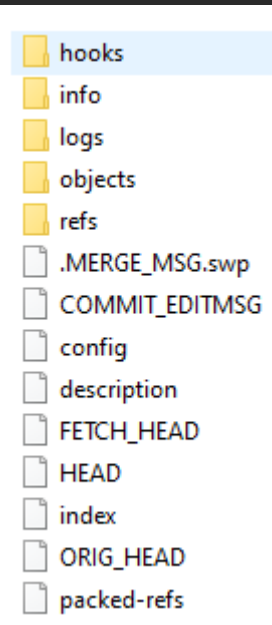
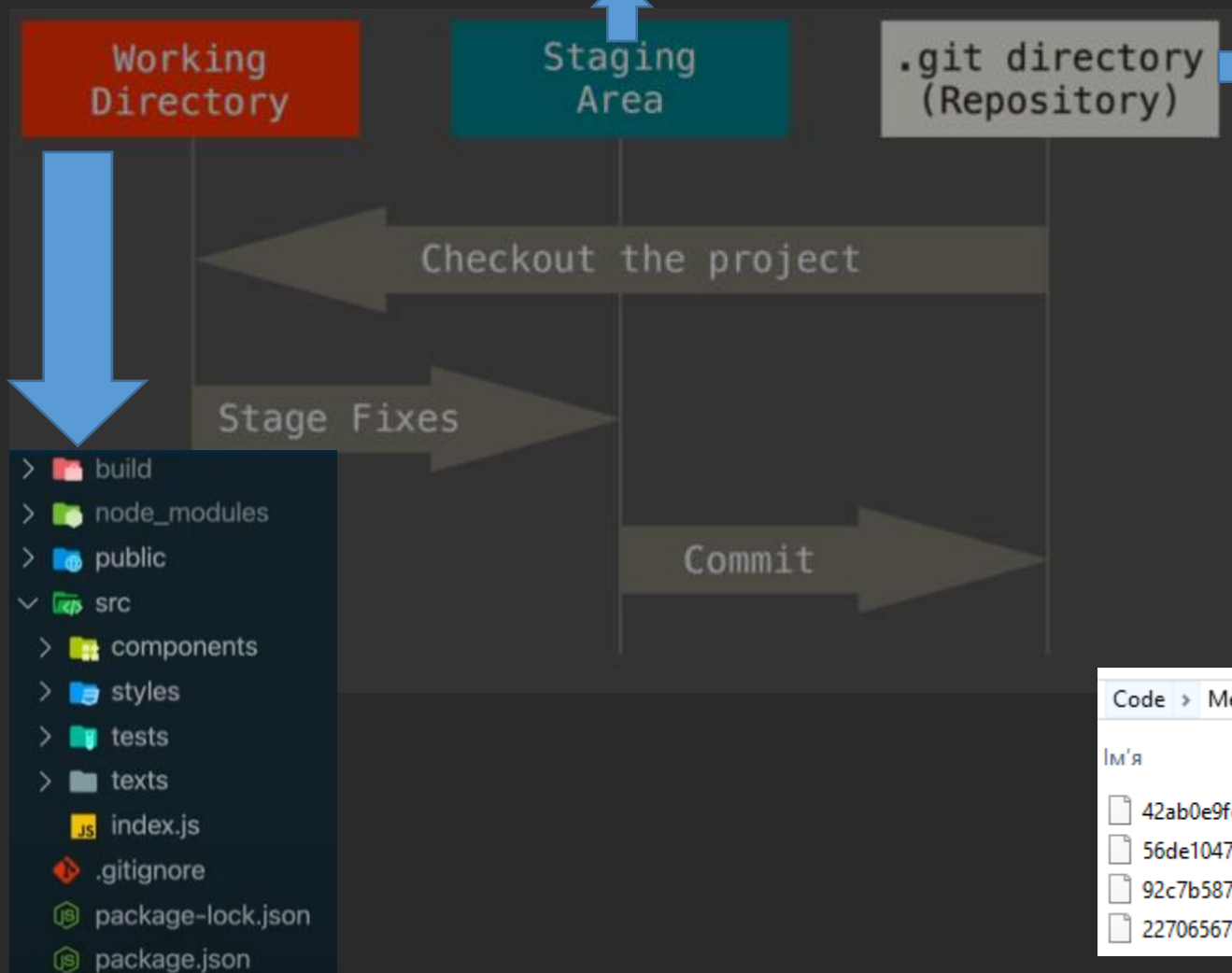
Цей «відбиток» являє собою рядок із 40 символів, що складається з шістнадцяткових символів (0–9 та a–f). Наприклад, типовий хеш SHA-1 виглядає так:

d3486ae9136e7856bc42212385ea797094475802



```
touch-typing > git status
On branch master
Your branch is up to date with 'origin/master'.

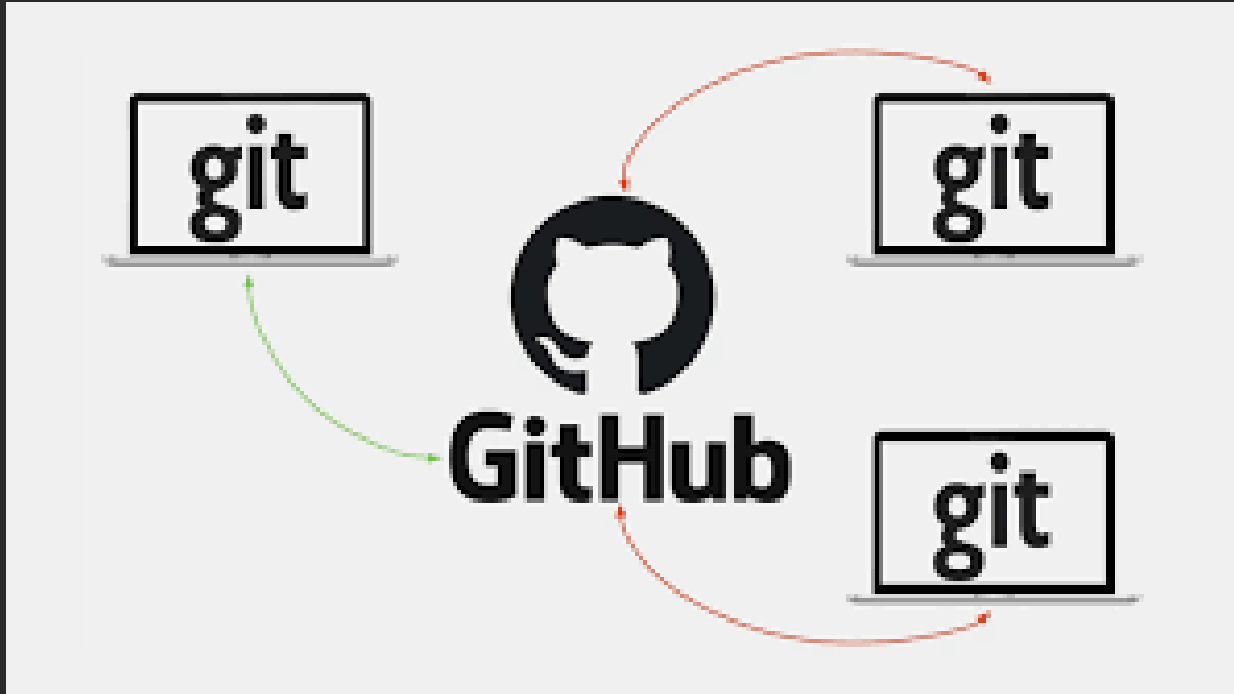
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    new file:   new-file.js
    modified:   src/index.js
```



Code > Meduza_BE > .git > objects >		
Ім'я	Дата змінення	Тип
00	17.02.2025 21:54	Папка файлів
0a	18.02.2025 16:13	Папка файлів
0b	24.01.2025 16:48	Папка файлів
0c	28.01.2025 21:32	Папка файлів
0d	10.02.2025 12:02	Папка файлів
0e	11.02.2025 11:08	Папка файлів

Code > Meduza_BE > .git > objects > 0a				
Ім'я	Дата змінення	Тип	Розмір	
42ab0e9fca1d767bb07739245c90ee03fdc...	05.02.2025 13:34	Файл	1 КБ	
56de1047abf489c2cce0da72ce71eed9488...	18.02.2025 16:13	Файл	1 КБ	
92c7b587032588f4cac7f28f87e20315817da6	30.01.2025 21:15	Файл	1 КБ	
227065676d4c2449ac5f3ff3051bdae6cc69d6	28.01.2025 21:32	Файл	1 КБ	

Git, GitHub, GitLab



GitHub та **GitLab** — це як соціальні мережі для вашої роботи, де ви можете зберігати свій проект в інтернеті та ділитися ним з іншими. Вони використовують систему **Git** для управління версіями ваших файлів, але також надають інструменти для співпраці, обговорення і спільної роботи над проектами.

Початок

```
$ git init
```

Додати до stage

```
$ git add
```

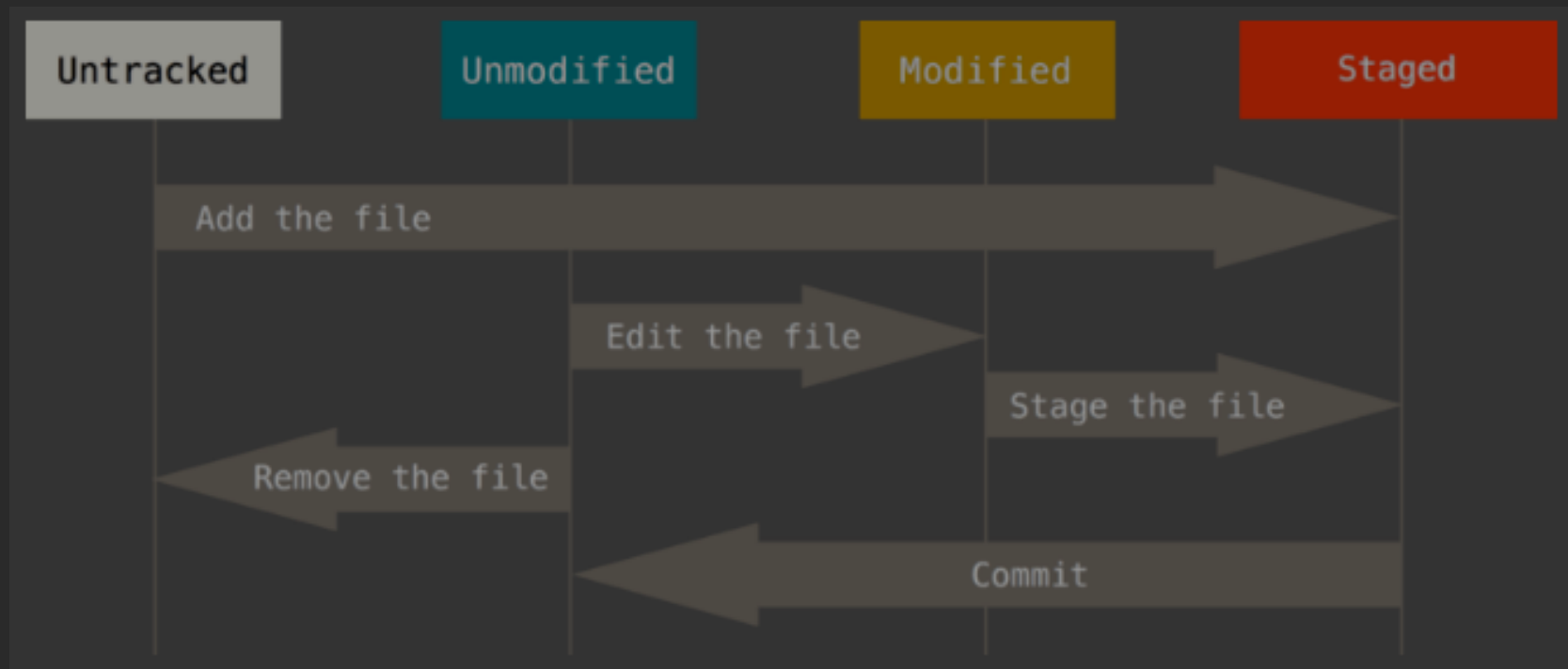
Закомітити зміни

```
$ git commit -m "Назва коміту"
```

Клонувати репозиторій

```
$ git clone https://github.com/test/test
```

Запис змін до репозиторія



Правила для вірців, які ви можете додати до файлу .gitignore:

- Порожні рядки та рядки, що починаються з #, ігноруються.
- Стандартні глоб вірці працюють, і будуть застосовані для всього робочого дерева рекурсивно.
- Ви можете почати вірець з прямої похилої риски (/) щоб уникнути рекурсії.
- Ви можете завершити вірець похилою рисою (/) щоб позначити директорію.
- Ви можете відкинути вірець, якщо почнете його зі знака оклику (!).

Індекс може бути дуже корисним для підготовки комітів саме до потрібного вам вигляду, іноді він може бути надто складним для ваших потреб. Якщо ви бажаєте обійти індекс, Git надає вам простий короткий шлях. Додавання опції -a до команди git commit, змушує Git автоматично додати кожен файл, що вже контролюється, до коміту, що дозволяє вам пропустити команди git add:

```
$ git commit -a -m "first commit"
```

Щоб видалити файл з Git, вам треба прибрати його з контрольованих файлів (вірніше, видалити його з вашого індексу) та створити коміт. Команда git rm це робить, а також видаляє файл з вашої робочої директорії, щоб наступного разу він не відображався неконтрольованим.

```
$ git rm file.py
```

Після того як створили декілька комітів, або якщо зробили клон репозиторія з існуючою історією комітів, можна дізнатись, що було та що відбувалось. Найбільш могутньою утилітою для цього є команда `git log`.

```
commit 34c3ea88154a7b0a11f9156638c4b4877ba131d4 (HEAD -> BE#12-feat-quiz-workflow)
Author: andrii910320 <128429619+andrii910320@users.noreply.github.com>
Date: Tue Feb 18 17:31:35 2025 +0200

    setUp upload files

commit 57b48866194cc24fc4a638e17398e5112ad09de5
Author: andrii910320 <128429619+andrii910320@users.noreply.github.com>
Date: Tue Feb 18 16:13:27 2025 +0200

    fix remark

commit a983f5525fd7fa6554b657042f520449ea988731
Author: andrii910320 <128429619+andrii910320@users.noreply.github.com>
Date: Mon Feb 17 21:54:08 2025 +0200

    implement workflow compled quiz

commit b27ee25e08e59ad2f879498ac99eeaf2005e1f2c
Merge: 7171e0a a8231e4
Author: Illia Puzdranovskyi <58227110+Ilia-puzdranovskuy@users.noreply.github.com>
Date: Mon Feb 17 13:26:19 2025 +0200

    Merge pull request #11 from andrii910320/BE#11/feat-add-quizzes

    Be#11/feat add quizzes

commit a8231e408419c05cd0d64e199b5ced02171479e0 (BE#11/feat-add-quizzes)
Author: andrii910320 <128429619+andrii910320@users.noreply.github.com>
Date: Sun Feb 16 22:23:28 2025 +0200

    implement quiz

commit 57b490cfee124e710cb41b996b4d444087b58115
Merge: bce3faf 2148f82
Author: andrii910320 <128429619+andrii910320@users.noreply.github.com>
Date: Sun Feb 16 20:33:10 2025 +0200

    Merge branch 'BE#10/feat-add-admin-role' into BE#11/feat-add-quizzes

commit 7171e0a8da969c20f64554215335a6a890a60763
Merge: cc44b6b 2148f82
```

Опція -p чи --patch, що показує різницю (вивід латки, англійською patch), привнесена при кожному коміті

```
$ git log -p
commit 34c3ea88154a7b0a11f9156638c4b4877ba131d4 (HEAD -> BE#12-feat-quiz-workflow)
Author: andrii910320 <128429619+andrii910320@users.noreply.github.com>
Date: Tue Feb 18 17:31:35 2025 +0200

    setUp upload files

diff --git a/package-lock.json b/package-lock.json
index 487924c..6ea2253 100644
--- a/package-lock.json
+++ b/package-lock.json
@@ -9,12 +9,13 @@
     "version": "0.0.1",
     "license": "UNLICENSED",
     "dependencies": {
-      "@nestjs/common": "^10.0.0",
+      "@nestjs/common": "^11.0.10",
+      "@nestjs/config": "^3.3.0",
+      "@nestjs/core": "^10.0.0",
+      "@nestjs/jwt": "^11.0.0",
+      "@nestjs/passport": "^11.0.5",
-      "@nestjs/platform-express": "^10.0.0",
+      "@nestjs/platform-express": "^10.4.15",
+      "@nestjs/serve-static": "^5.0.2",
+      "@nestjs/swagger": "^8.1.1",
+      "@nestjs/typeorm": "^11.0.0",
+      "bcrypt": "^5.1.1",
@@ -23,6 +24,7 @@
     "dayjs": "^1.11.13",
     "dotenv": "^16.4.7",
     "jwks-rsa": "^3.1.0",
+    "multer": "^1.4.5-lts.1",
     "passport": "^0.7.0",
     "passport-jwt": "^4.0.1",
     "pg": "^8.13.1",
@@ -37,6 +39,7 @@
     "@types/bcrypt": "^5.0.2",
     "@types/express": "^5.0.0",
     "@types/jest": "^29.5.2",
```

```
$ git log --stat
commit 34c3ea88154a7b0a11f9156638c4b4877ba131d4 (HEAD -> BE#12-feat-quiz-workflow)
Author: andrii910320 <128429619+andrii910320@users.noreply.github.com>
Date: Tue Feb 18 17:31:35 2025 +0200
```

setUp upload files

```
package-lock.json | 164 ++++++-----
package.json | 7 +-
src/app.module.ts | 6 ++
src/modules/user/dto/update-user.dto.ts | 21 +++-
src/modules/user/user.controller.ts | 7 +-
src/modules/user/user.module.ts | 19 +++-
src/modules/user/user.service.ts | 12 ++-
uploads/1739891920137-Mafioznic.jpg | Bin 0 -> 5559 bytes
8 files changed, 180 insertions(+), 56 deletions(-)
```

```
commit 57b48866194cc24fc4a638e17398e5112ad09de5
Author: andrii910320 <128429619+andrii910320@users.noreply.github.com>
Date: Tue Feb 18 16:13:27 2025 +0200
```

fix remark

```
...e_quiz.ts => 1739884850959-add_complete_quiz.ts} | 12 +++++-----
src/modules/company/service/company.service.ts | 14 ++++++
src/modules/quiz/entities/passed-quiz.entity.ts | 12 +++-----
src/modules/quiz/quiz.service.ts | 21 ++++++-----
4 files changed, 37 insertions(+), 22 deletions(-)
```

```
$ git log --pretty=oneline
34c3ea88154a7b0a11f9156638c4b4877ba131d4 (HEAD -> BE#12-feat-quiz-workflow) setUp upload files
57b48866194cc24fc4a638e17398e5112ad09de5 fix remark
a983f5525fd7fa6554b657042f520449ea988731 implement workflow compled quiz
b27ee25e08e59ad2f879498ac99eeaf2005e1f2c Merge pull request #11 from andrii910320/BE#11/feat-add-quizzes
a8231e408419c05cd0d64e199b5ced02171479e0 (BE#11/feat-add-quizzes) implement quiz
57b490cfee124e710cb41b996b4d444087b58115 Merge branch 'BE#10/feat-add-admin-role' into BE#11/feat-add-quizzes
7171e0a8da969c20f64554215335a6a890a60763 Merge pull request #10 from andrii910320/BE#10/feat-add-admin-role
2148f821e4ae13909fbf270ff64af0a94dc40777 (BE#10/feat-add-admin-role) adapt logic for new roles
e37d5408d2afa5250b441d02b830c94196da1e70 worker by company and save code
ed631216ae1dc27de372e3561a4ef2c2213b66e7 Update main.ts
11257494a2687b2849c48c2d5ca2915164e644db adapt code for new logic wokkers
47268f146f9b23e6d6249d6cea197be875c7bc91 auth
bce3faf661ed24c95e92f1393c3c6012137f87d4 Update main.ts
c980752059ea7477de5d14081fd6b0ecfe5afee4 save
04f5c18603f46039658b0af741e993a275dff4b4 Merge branch 'BE#10/feat-add-admin-role' into BE#11/feat-add-quizzes
6e32e236c07e09c8a1a54b586d94319c8450d21f save before merge
cbdeb825e30d188b8ecb624b76a7bf4a065dbb65 add admin role
cc44b6bd2fc3f63fb27b50aa1733ef6adc582806 Merge pull request #9 from andrii910320/BE#9/feat-company-actions
1aadfc681a29e3ef8a8c097df52ea87a00fabe90 (BE#9/feat-company-actions) add invite to copmany
021243c35f041d4e5da945b38e1d1cffe45df78b save progres
fea307acd6374423aaec1b98d85bd2e15a705947 Update main.ts
a358a4b50d2409233791db6bb3de3764d721b1fe Update main.ts
06a44e005451cae9e9525927962a2761becddfb4 Update main.ts
1539c039d1c6c1c24ebf0ee1ebe809584ec9b987 Update main.ts
1e565f9768905ff9a33a6212594c74e9a460cfcb Merge pull request #8 from andrii910320/BE8/feat-CRUD-for-company
31025b2c1b1ca4645c6f193e57db53d693e386ad (BE8/feat-CRUD-for-company) write decorators extract user fix notation
886998f8f0126f65b85e208a42994c807dbf2f90 Merge pull request #7 from andrii910320/BE7/feat-add-validation
```

```
$ git log --pretty=format:"%h - %an, %ar : %s"
34c3ea8 - andrii910320, 7 months ago : setUp upload files
57b4886 - andrii910320, 7 months ago : fix remark
a983f55 - andrii910320, 7 months ago : implement workflow compled quiz
b27ee25 - Illia Puzdranovskyi, 7 months ago : Merge pull request #11 from andrii910320/BE#11/feat-add-quizzes
a8231e4 - andrii910320, 7 months ago : implement quiz
57b490c - andrii910320, 7 months ago : Merge branch 'BE#10/feat-add-admin-role' into BE#11/feat-add-quizzes
7171e0a - Illia Puzdranovskyi, 7 months ago : Merge pull request #10 from andrii910320/BE#10/feat-add-admin-role
2148f82 - andrii910320, 7 months ago : adapt logic for new roles
e37d540 - andrii910320, 7 months ago : worker by company and save code
ed63121 - andrii910320, 7 months ago : Update main.ts
1125749 - andrii910320, 7 months ago : adapt code for new logic wokers
47268f1 - andrii910320, 7 months ago : auth
bce3faf - andrii910320, 7 months ago : Update main.ts
c980752 - andrii910320, 7 months ago : save
```

Одне з розповсюджених скасування відбувається, коли ви зробили коміт зарано, можливо забули додати деякі файли, або ви зіпсували повідомлення коміту. Якщо ви хочете переробити цей коміт, внести до нього додаткові зміни, що про них ви забули.

```
git commit --amend
```

```
$ git commit -m 'initial commit'  
$ git add forgotten_file  
$ git commit --amend
```

```
$ git add *
$ git status
On branch master
Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

    renamed:    README.md -> README
    modified:   CONTRIBUTING.md
```

```
$ git reset HEAD CONTRIBUTING.md
Unstaged changes after reset:
M       CONTRIBUTING.md
$ git status
On branch master
Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

    renamed:    README.md -> README

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

    modified:   CONTRIBUTING.md
```

Changes not staged for commit:

(use "git add <file>..." to update what will be committed)

(use "git checkout -- <file>..." to discard changes in working directory)

modified: CONTRIBUTING.md

```
$ git checkout -- CONTRIBUTING.md
```

```
$ git status
```

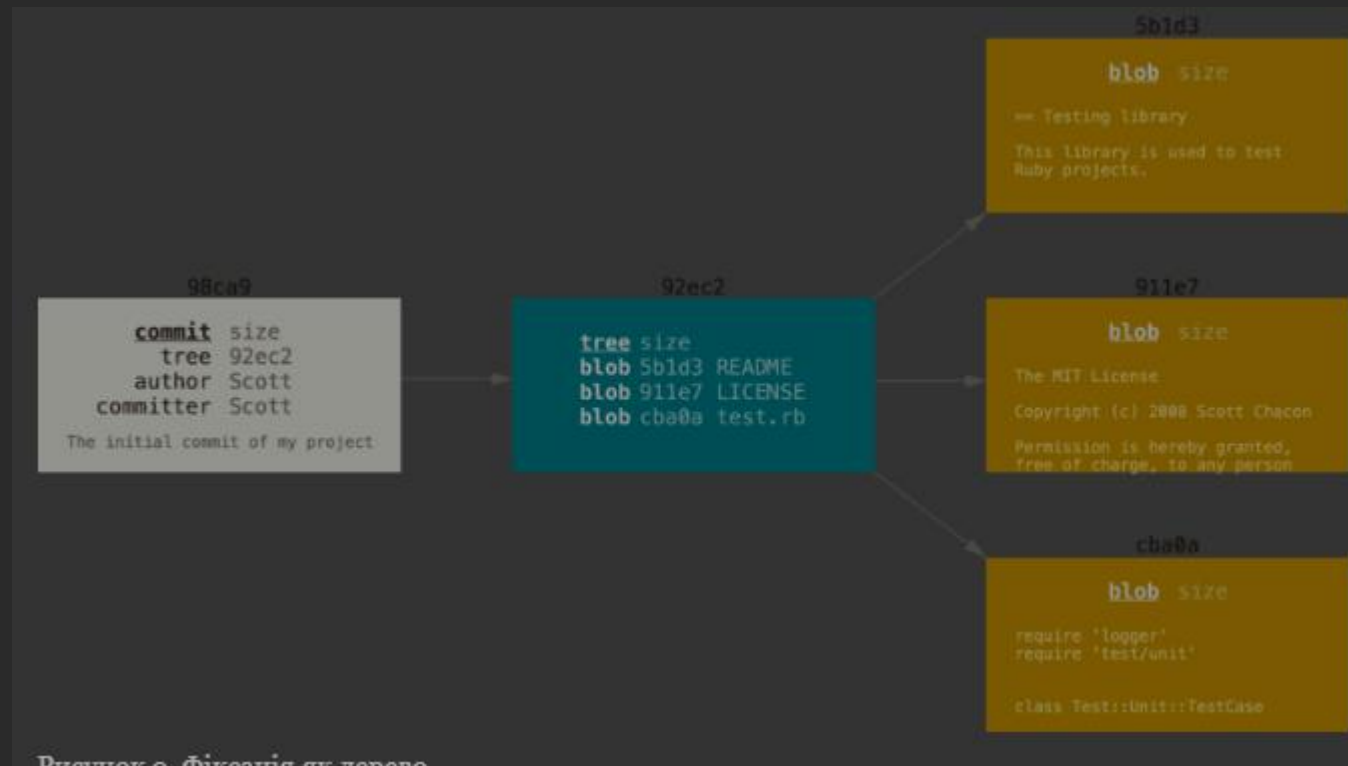
On branch master

Changes to be committed:

(use "git reset HEAD <file>..." to unstage)

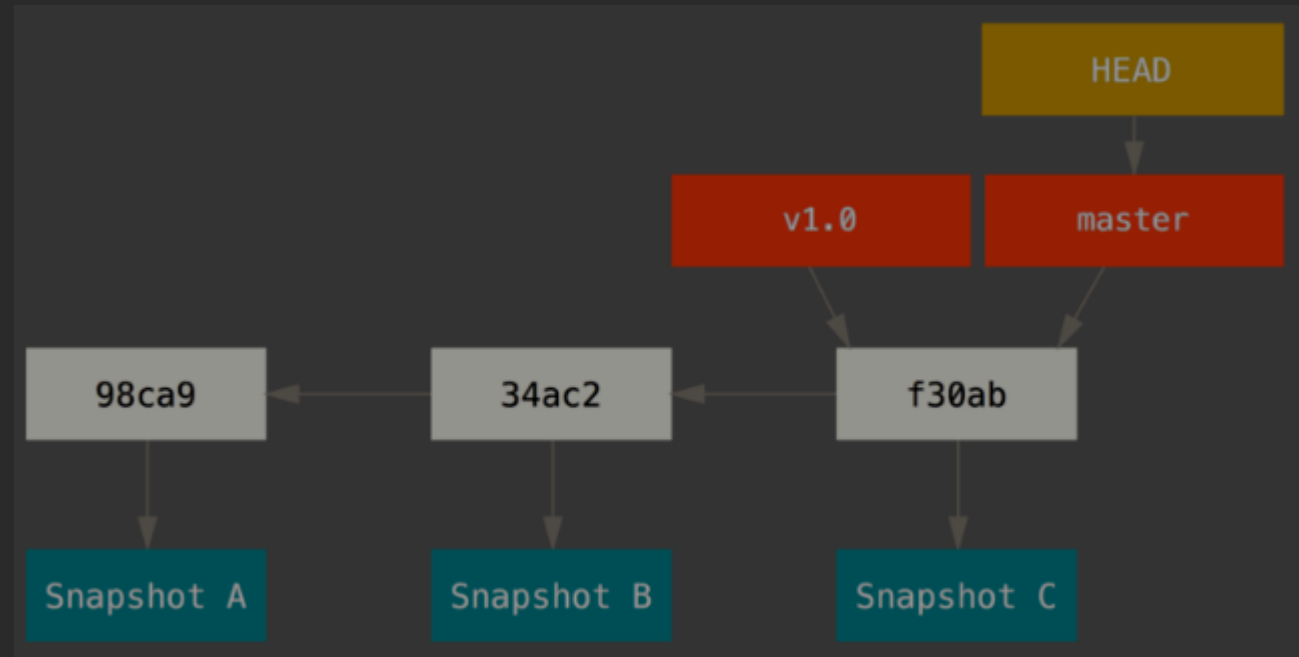
renamed: README.md -> README

```
$ git add README test.rb LICENSE
$ git commit -m 'The initial commit of my project'
```

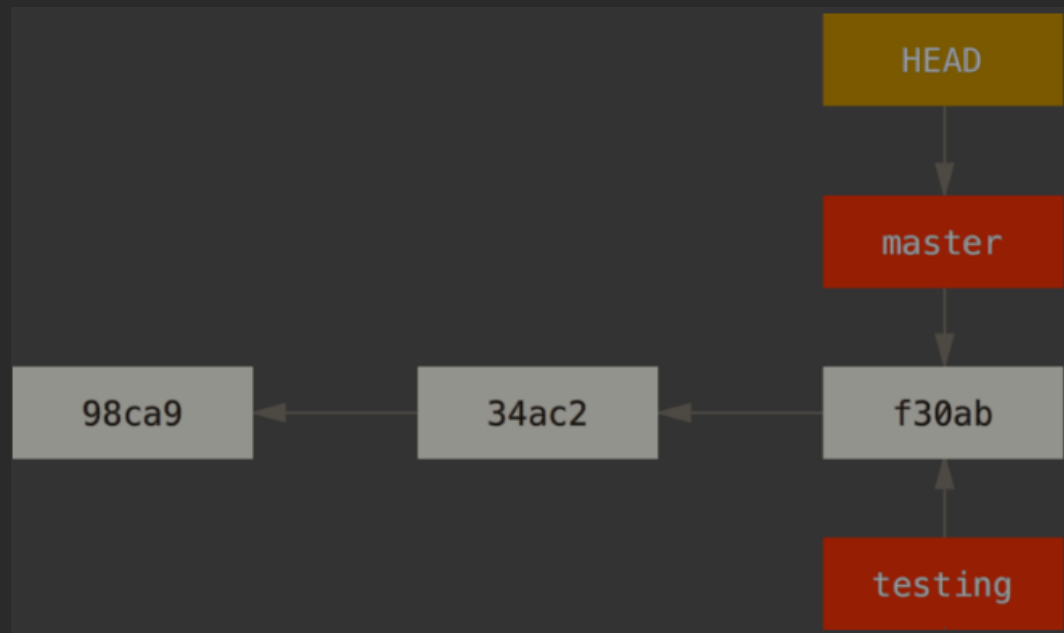


Якщо ви зробите якісь зміни та зафіксуєте знову, наступна фіксація буде зберігати вказівник на попередню.

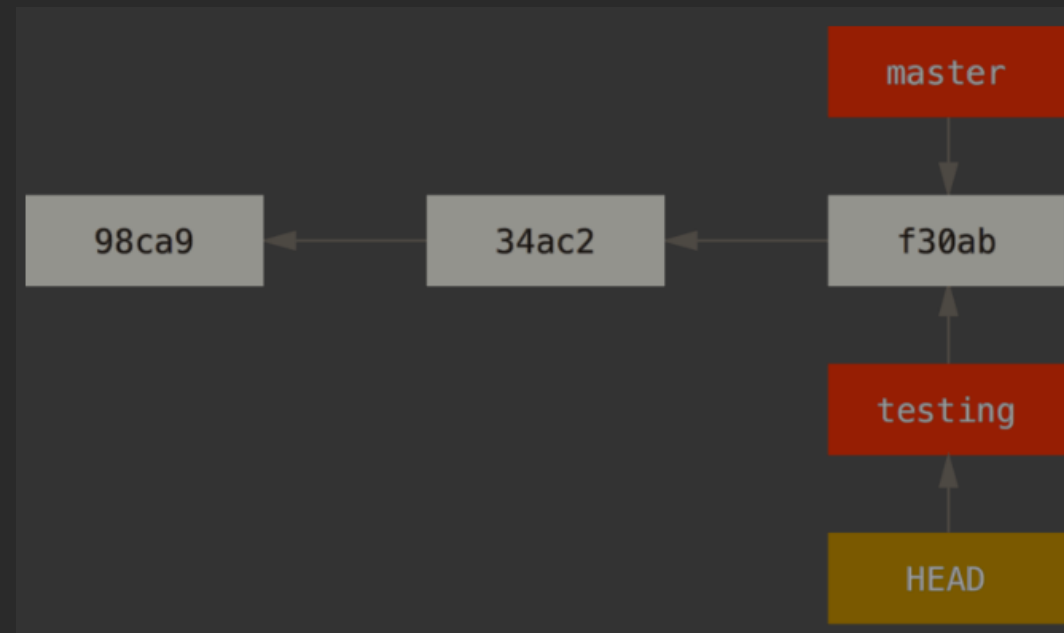




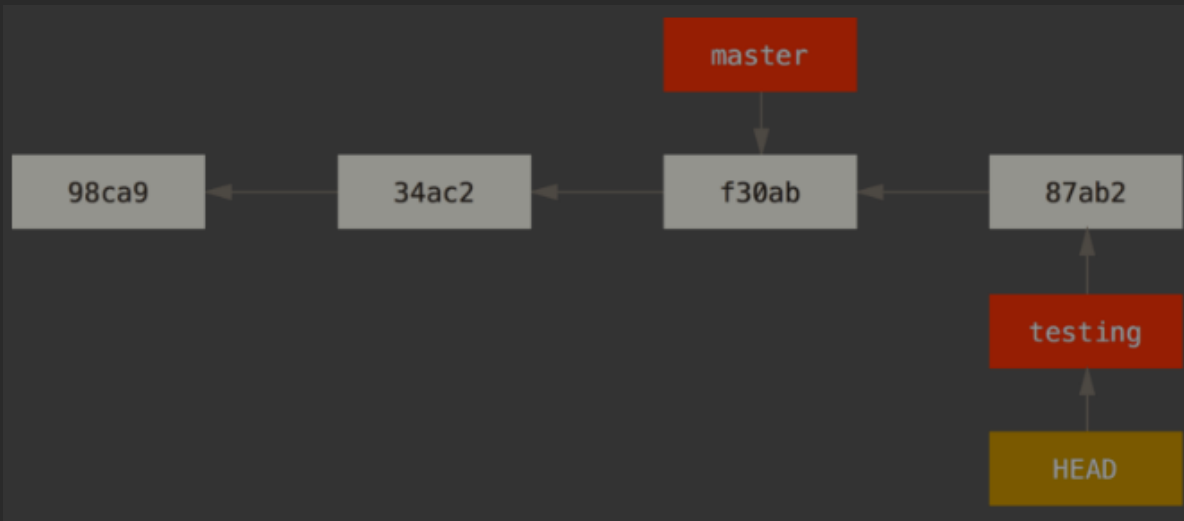
git branch testing



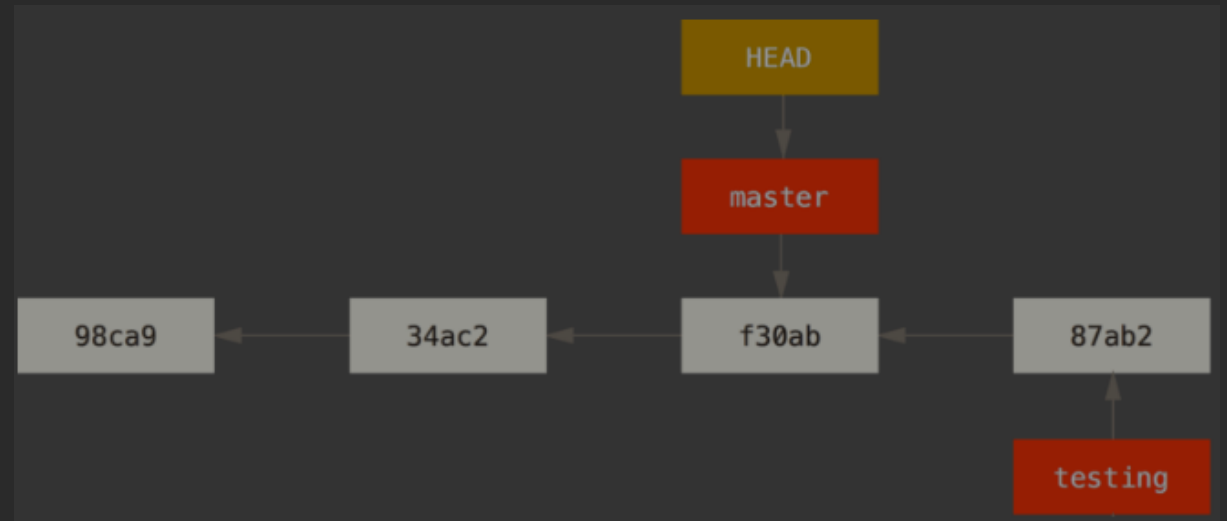
git branch testing



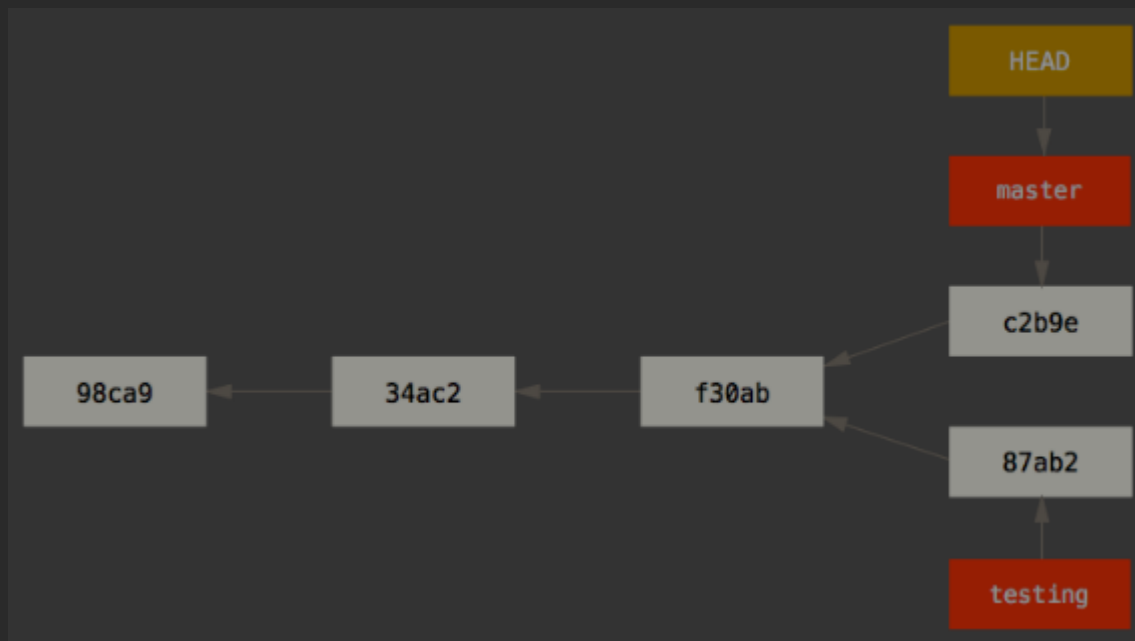
```
$ vim test.py  
$ git commit -a -m "Feature"
```

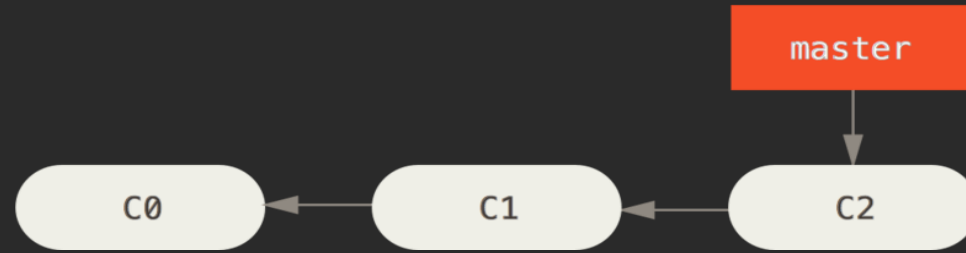


```
$ git checkout master
```



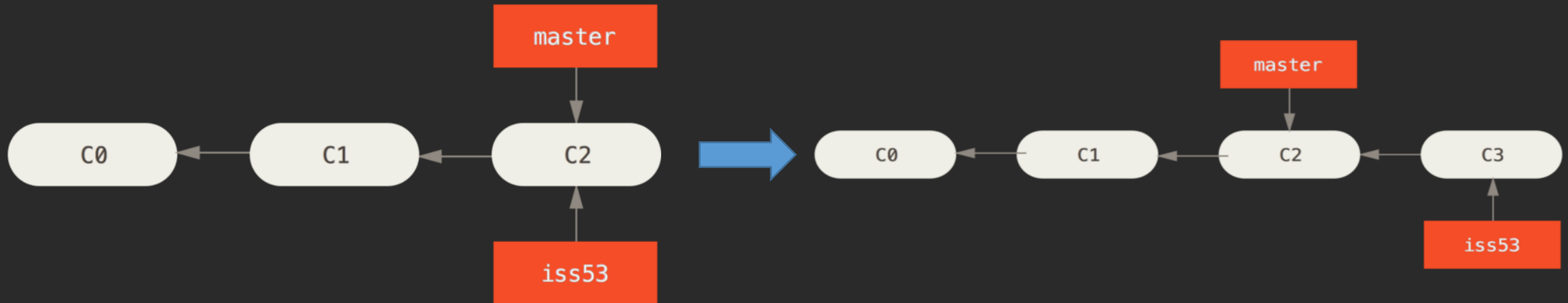
```
$ vim test.py  
$ git commit -a -m "Feature 2"
```



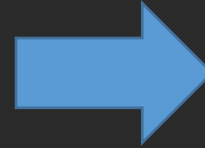


```
$ git branch iss53  
$ git checkout iss53
```

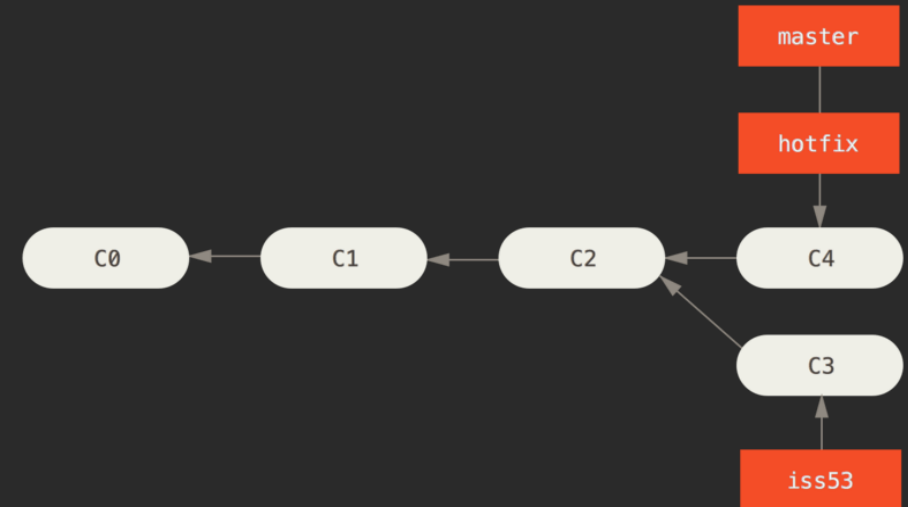
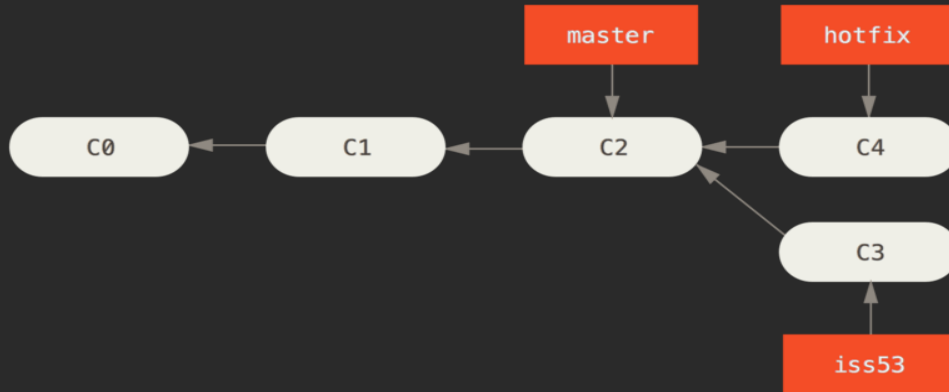
```
$ vim index.html  
$ git commit -a -m 'Feature [issue  
53]'
```



```
$ git checkout -b hotfix
Switched to a new branch 'hotfix'
$ vim index.html
$ git commit -a -m 'fixed the broken email address'
[hotfix 1fb7853] fixed the broken email address
1 file changed, 2 insertions(+)
```



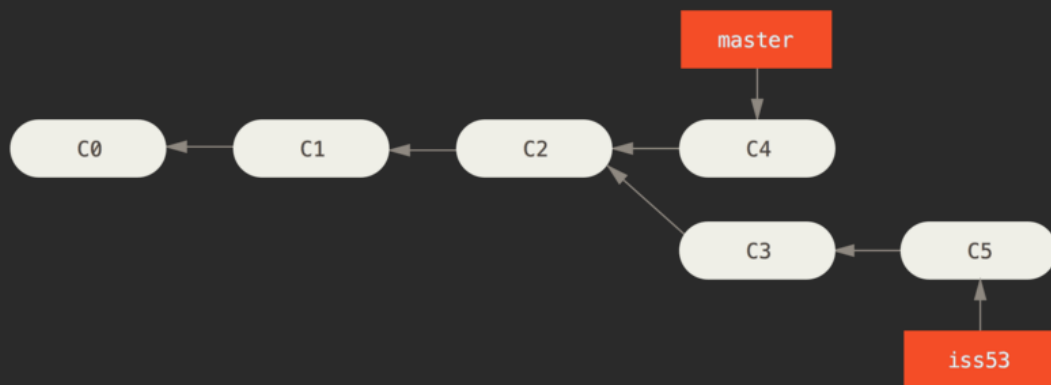
```
$ git checkout master
$ git merge hotfix
Updating f42c576..3a0874c
Fast-forward index.html | 2 ++
1 file changed, 2 insertions(+)
```



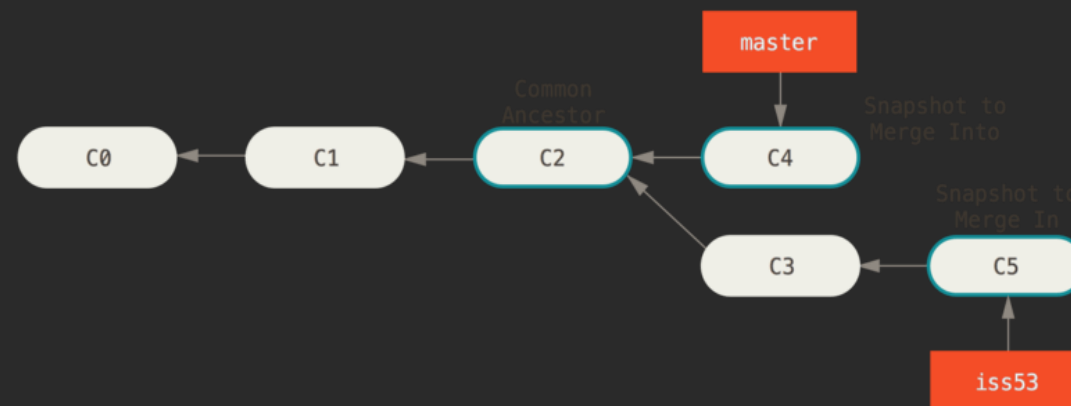


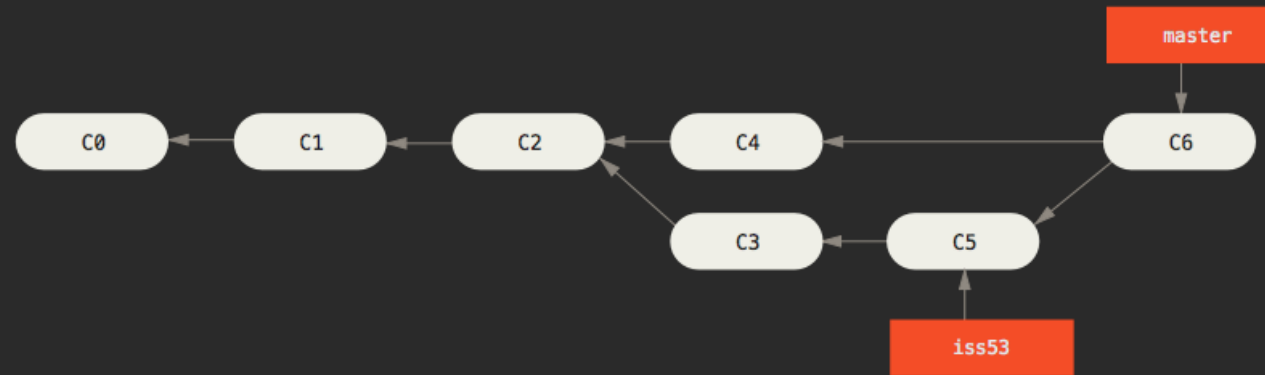
```
$ git branch -d hotfix  
Deleted branch hotfix (3a0874c).
```

```
$ git checkout iss53  
Switched to branch "iss53"  
$ vim index.html  
$ git commit -a -m 'finished the new footer [issue 53]'  
[iss53 ad82d7a] finished the new footer [issue 53]  
1 file changed, 1 insertion(+)
```



```
$ git checkout master  
Switched to branch 'master'  
$ git merge iss53  
Merge made by the 'recursive' strategy.  
index.html | 1 +  
1 file changed, 1 insertion(+)
```





```
$ git branch -d iss53
```

```
$ git merge iss53
Auto-merging index.html
CONFLICT (content): Merge conflict in index.html
Automatic merge failed; fix conflicts and then commit the result.
```

```
<<<<<< HEAD:index.html

<div id="footer">contact : email.support@github.com</div>

=====

<div id="footer">
  please contact us at support@github.com
</div>

>>>>>> iss53:index.html
```