



بعض أوامر GIT

التكليف الأول



رفيدة ياسين المقدمي
أ/ مالك المصنف

تشبيه Git كمسرحية: كيف نفهم ال Git ببساطة؟

GIT AS A THEATRE



The Theatre
(Git Repository)
The overall project



Backstage
(Working Directory)
The actual changes



Actors
(Modified Files)
The files you have worked on



The Filming
(git commit)
Recording a snapshot



The Filming
(git commit)
Recording a snapshot



The Movie Archive
(git log)
The history of commits

تشبيه Git بمسرح العروض

نتخيل إنك مدير مسرح، وتقوم بإعداد عروض مسرحية، وكل عرض هو تعديل في مشروعك

المسرح (Git Repository)

هو المجلد الكبير اللي فيه كل شيء يخص المسرحيات
يشبه مجلد (Git repo) اللي فيه كل الملفات وكل التاريخ

خلف الكواليس (Working Directory)

هنا يحصل كل شيء: كتابة السيناريو، تدريب الممثلين، تجهيز الملابس ...
المجلد الخفي الذي ينشأ في الجت وفيه تعديلاتك الحقيقية على الملفات قبل ما تضيفها إلى Git

خشبة المسرح (Staging Area)

المنصة اللي يصعد عليها الممثلين الجاهزين للأداء
المنطقة المؤقتة داخل Git
يتم فيها "تحضير" التعديلات قبل ما يتم حفظها بشكل نهائي في History

الممثلون (Modified Files)

الملفات المعدلة الملفات التي عملت عليها وأجريت فيها تغييرات.

عرض المسرحية على الخشبة (git add)

لما تستخدم git add كأنك تقول: يا ممثل، تعال اصعد المنصة واستعد للعرض
أضف هذا الملف (أو التعديلات عليه) إلى المنطقة المؤقتة (staging area)

التصوير الرسمي (git commit)

الآن تبدأ الكاميرا بتصوير العرض!

يعني تسجل العرض بالكامل وتحفظه في Repo history

"تم إنشاء snapshot داخل Git لحالة المشروع في لحظة معينة"

أرشف الفيديو (git log)

كل عرض تم تصويره يتم تخزينه في مكتبة المسرح، ونقدر نشوف سجل العروض السابقة.

يعني git log يرينا كل الـ commits اللي صارت، مع تاريخ جميع التعديلات

مثل أرشف الفيديوهات القديمة

مجلد المحفوظات (Repository History)

هو ألبوم العروض الكامل، فيه كل العروض اللي انعرضت، مرتبة حسب الوقت.

يعني سجل التعديلات Commits الكامل للمشروع.

عرض مستقل (Branch)

سلسلة من الـ Commits

كل فرع = عرض مسرحي مستقل

كل commit = مشهد من المسرحية

التنقل بين العروض المسرحية او بين مشاهدها (checkout)

ننتقل بين الفروع git checkout branch

ننتقل الى نسخة سابقة لملف معين او لتعديل معين git checkout commit

دمج عرضين في عرض واحد (merge)

دمج فرع مع فرع آخر

(Conflict)

تعارض يحدث عند الدمج

لما ممثلين من عرضين يتعارضون على نفس الدور

التطبيق العملي لأوامر Git

نفتح المسرح لأول مرة

```
Microsoft Windows [Version 10.0.26100.4061]
(c) Microsoft Corporation. All rights reserved.

C:\Users\PC>color 70
C:\Users\PC>cd C:\Users\PC\Desktop\Software_Engnieering_Rofaida\Repository_root

C:\Users\PC\Desktop\Software_Engnieering_Rofaida\Repository_root>git init
Initialized empty Git repository in C:/Users/PC/Desktop/Software_Engnieering_Rofaida/Repository_root/.git/
```

فتحنا مسرح جديد وفاضي، ما فيه عروض ولا مشاهد حتى الآن، بس جاهزين نبدأ

نُسجل معلومات المخرج

```
C:\Users\PC\Desktop\Software_Engnieering_Rofaida\Repository_root>git config --global user.name "Rofaida"

C:\Users\PC\Desktop\Software_Engnieering_Rofaida\Repository_root>git config --global user.email "rofaidaalmaqdami@gmail.com"

C:\Users\PC\Desktop\Software_Engnieering_Rofaida\Repository_root>cd C:\Users\PC\Desktop\Software_Engnieering_Rofaida\Repository_root\Lab_1
```

عرّفنا من هو مخرج العروض والمشاهد... لازم كل عرض يحمل توقيع المخرج

نخلي الجت يعرف من هو الشخص اللي يشتغل على المشروع عشان كل

Commit يتم توقيعه باسمه.

تجهيز النص المسرحي (الملفات الأساسية)

```
C:\Users\PC\Desktop\Software_Engineering_Rofaida\Repository_root\Lab_1>echo "Hello World"> myfile.txt
C:\Users\PC\Desktop\Software_Engineering_Rofaida\Repository_root\Lab_1>
```

كتبنا أول مشهد من المسرحية (الملفات الأساسية)

نتحقق من حالة المسرحية

```
C:\Users\PC\Desktop\Software_Engineering_Rofaida\Repository_root\Lab_1>git status
On branch master

No commits yet

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

nothing added to commit but untracked files present (use "git add" to track)
C:\Users\PC\Desktop\Software_Engineering_Rofaida\Repository_root\Lab_1>
```

نشوف قائمة الممثلين الجدد اليي للآن ما تدربوا على المسرح

نضيف المشهد إلى خشبة المسرح (منطقة الإعداد)

```
C:\Users\PC\Desktop\Software_Engineering_Rofaida\Repository_root\Lab_1>git add myfile.txt
```

جبنا المشهد (الملف) وخليناه على الخشبة الجاهزة للعروض

نُعرض المشهد رسمياً أمام الكاميرا

```
C:\Users\PC\Desktop\Software_Engineering_Rofaida\Repository_root\Lab_1>git commit -m "Scene One:The Opening Page"
[master (root-commit) ee86c6e] Scene One:The Opening Page
 1 file changed, 1 insertion(+)
 create mode 100644 Lab_1/myfile.txt
C:\Users\PC\Desktop\Software_Engineering_Rofaida\Repository_root\Lab_1>
```

صورنا المشهد وسجلناه في أرشيف المسرح

نشوف أرشيف العروض السابق

```
C:\Users\PC\Desktop\Software_Engnieering_Rofaida\Repository_root\Lab_1>git
log
commit ee86c6e816650b0b804df89c75955092a18cd49d (HEAD -> master)
Author: Rofaida <rofaidaalmaqdami@gmail.com>
Date: Thu Jul 24 22:58:22 2025 +0300

    Scene One:The Opening Page

C:\Users\PC\Desktop\Software_Engnieering_Rofaida\Repository_root\Lab_1>
```

نشوف شريط الفيديو لكل المشاهد السابقة

ننشئ ملف (صفحة ويب)

```
C:\Users\PC\Desktop\Software_Engnieering_Rofaida\Repository_root\Lab_1>echo
"Git is a tool for saving code changes\nit helps you go back,try new ideas
,and work with others easily"> index.html

C:\Users\PC\Desktop\Software_Engnieering_Rofaida\Repository_root\Lab_1>git
status
On branch master
Untracked files:
  (use "git add <file>..." to include in what will be committed)
    index.html

nothing added to commit but untracked files present (use "git add" to track
)

C:\Users\PC\Desktop\Software_Engnieering_Rofaida\Repository_root\Lab_1>
```

```
C:\Users\PC\Desktop\Software_Engnieering_Rofaida\Repository_root\Lab_1>git
add index.html

C:\Users\PC\Desktop\Software_Engnieering_Rofaida\Repository_root\Lab_1>git
commit -m"Second Scene:The Main Page"
[master 52809b2] Second Scene:The Main Page
1 file changed, 1 insertion(+)
create mode 100644 Lab_1/index.html

C:\Users\PC\Desktop\Software_Engnieering_Rofaida\Repository_root\Lab_1>
```

```
C:\Users\PC\Desktop\Software_Engnieering_Rofaida\Repository_root\Lab_1>git
log
commit 52809b24f2dcb4ddb3bfe19c85825b254f59ae2f (HEAD -> master)
Author: Rofaida <rofaidaalmaqdami@gmail.com>
Date: Thu Jul 24 23:15:28 2025 +0300

    Second Scene:The Main Page

commit ee86c6e816650b0b804df89c75955092a18cd49d
Author: Rofaida <rofaidaalmaqdami@gmail.com>
Date: Thu Jul 24 22:58:22 2025 +0300

    Scene One:The Opening Page

C:\Users\PC\Desktop\Software_Engnieering_Rofaida\Repository_root\Lab_1>
```

ننشئ فروع لعروض جانبية (dashboard, login...)

```
C:\Users\PC\Desktop\Software_Engnieering_Rofaida\Repository_root\Lab_1>git
branch login
```

بدأنا بروفة لمسرحية جانبية اسمها تسجيل الدخول

ننتقل إلى هذا الفرع

```
C:\Users\PC\Desktop\Software_Engnieering_Rofaida\Repository_root\Lab_1>git
switch login
Switched to branch 'login'
```

تركنا المسرح الرئيسي وانتقلنا لمسرح login وبدأنا نشتغل عليه

نعدل الملف (الفرع) ونضيفه الى Repository History

```
C:\Users\PC\Desktop\Software_Engineering_Rofaida\Repository_root\Lab_1>echo
"Login Form"> login.html

C:\Users\PC\Desktop\Software_Engineering_Rofaida\Repository_root\Lab_1>git
add login.html

C:\Users\PC\Desktop\Software_Engineering_Rofaida\Repository_root\Lab_1>git
commit -m"Login Scene:The Login Page"
[login 4f55ab6] Login Scene:The Login Page
1 file changed, 1 insertion(+)
create mode 100644 Lab_1/login.html

C:\Users\PC\Desktop\Software_Engineering_Rofaida\Repository_root\Lab_1>
```

كتبنا مشهد جديد، وعرضناه بالكاميرا وسجلناه في أرشيف Login فقط

دمج الفروع في العرض الرئيسي

نرجع للمسرح الرئيسي

```
C:\Users\PC\Desktop\Software_Engineering_Rofaida\Repository_root\Lab_1>git
switch master
Switched to branch 'master'
```

رجعنا للمسرحية الأساسية

ندمج عرض login في العرض الرئيسي

```
C:\Users\PC\Desktop\Software_Engineering_Rofaida\Repository_root\Lab_1>git
merge login
Updating 52809b2..4f55ab6
Fast-forward
 Lab_1/login.html | 1 +
1 file changed, 1 insertion(+)
create mode 100644 Lab_1/login.html

C:\Users\PC\Desktop\Software_Engineering_Rofaida\Repository_root\Lab_1>
```

دمجنا مشاهد login في النص المسرحي الأساسي

ما حصل أي تعارض...

حذف الفرع login لأنه لم يعد ضرورياً

```
C:\Users\PC\Desktop\Software_Engnieering_Rofaida\Repository_root\Lab_1>git status
On branch master
nothing to commit, working tree clean

C:\Users\PC\Desktop\Software_Engnieering_Rofaida\Repository_root\Lab_1>git branch -d login
Deleted branch login (was 4f55ab6).

C:\Users\PC\Desktop\Software_Engnieering_Rofaida\Repository_root\Lab_1>
```

إزالة "نسخة" من سيناريو المسرحية كانت تُستخدم للتجربة أو التطوير،

ولم تعد هناك حاجة لها

الرجوع الى مشهد قديم باستخدام معرف Commit

```
C:\Users\PC\Desktop\Software_Engnieering_Rofaida\Repository_root\Lab_1>git checkout ee86c6e816650b0b804df89c75955092a18cd49d
Note: switching to 'ee86c6e816650b0b804df89c75955092a18cd49d'.

You are in 'detached HEAD' state. You can look around, make experimental
changes and commit them, and you can discard any commits you make in this
state without impacting any branches by switching back to a branch.

If you want to create a new branch to retain commits you create, you may
do so (now or later) by using -c with the switch command. Example:

    git switch -c <new-branch-name>

Or undo this operation with:

    git switch -

Turn off this advice by setting config variable advice.detachedHead to false

HEAD is now at ee86c6e Scene One:The Opening Page

C:\Users\PC\Desktop\Software_Engnieering_Rofaida\Repository_root\Lab_1>
```

فتحنا شريط الفيديو ورجعنا نشوف عرض قديم (بدون تغييره)

عرض جميع الملفات التي تتبعها ال Git حالياً في المستودع Repository

```
C:\Users\PC\Desktop\Software_Engnieering_Rofaida\Repository_root\Lab_1>git switch master
Previous HEAD position was ee86c6e Scene One:The Opening Page
Switched to branch 'master'

C:\Users\PC\Desktop\Software_Engnieering_Rofaida\Repository_root\Lab_1>git ls-files
index.html
login.html
myfile.txt

C:\Users\PC\Desktop\Software_Engnieering_Rofaida\Repository_root\Lab_1>
```

عرض قائمة الممثلين المعتمدين رسمياً على المسرح والموجودون حالياً في نص المسرحية