


# توثيق مشروع Git و GitHub

مساق: هندسة البرمجيات

اسم الطالب: منور محمد حزام الحاج سيف  
إشراف الاستاذ: مالك المصنف

## المرحلة 1: إعداد بيئة العمل

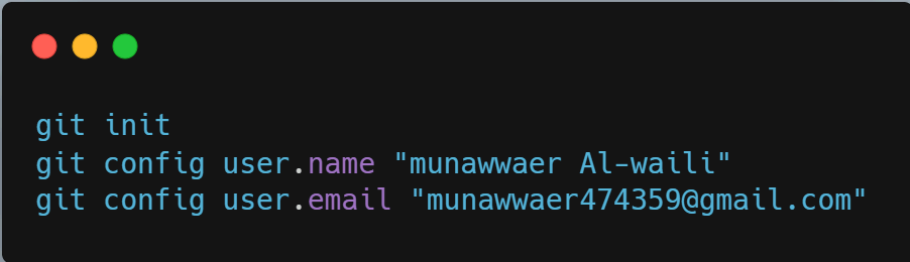
تم إنشاء هيكل المجلدات اللازم للمشروع على الجهاز المحلي للبدء في العمل.



```
C:\Users\USER>cd desktop/SoftwareEngineering  
C:\Users\USER\Desktop\SoftwareEngineering>mkdir Lab00  
C:\Users\USER\Desktop\SoftwareEngineering>cd Lab00  
C:\Users\USER\Desktop\SoftwareEngineering\Lab00>mkdir my-website  
C:\Users\USER\Desktop\SoftwareEngineering\Lab00>cd my-website
```

## المرحلة 2: تهيئة Git وإعداد المستخدم

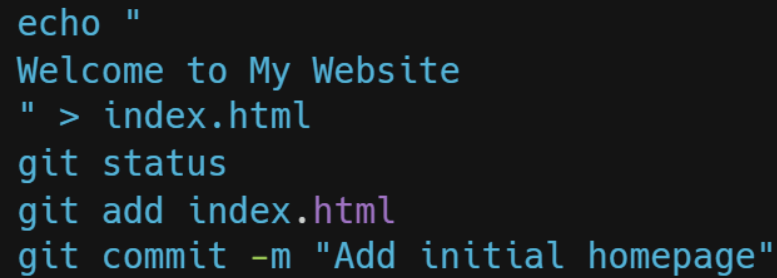
تم تحويل مجلد المشروع إلى مستودع Git باستخدام `git init`، ثم تم إعداد معلومات المستخدم (الاسم والإيميل) لتوثيق التغييرات القادمة.



```
git init
git config user.name "munawwaer Al-waili"
git config user.email "munawwaer474359@gmail.com"
```

## المرحلة 3: إنشاء أول ملف وال Commit الأول

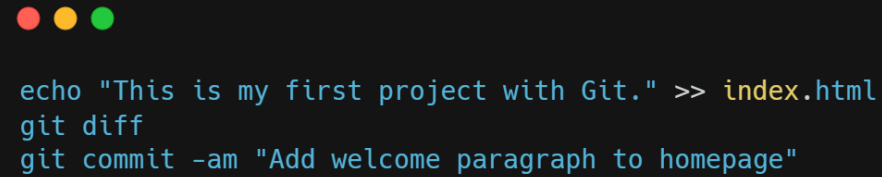
تم إنشاء ملف `index.html`، والتحقق من حالته (`git status`)، ثم إضافته لمنطقة التجهيز (`git add`)، وأخيرًا حفظه كأول commit في تاريخ المشروع.

A terminal window with a dark background and three colored window control buttons (red, yellow, green) in the top-left corner. It contains a series of Git commands and their output.

```
echo "  
Welcome to My Website  
" > index.html  
git status  
git add index.html  
git commit -m "Add initial homepage"
```

## المرحلة 4: تعديل ملف موجود

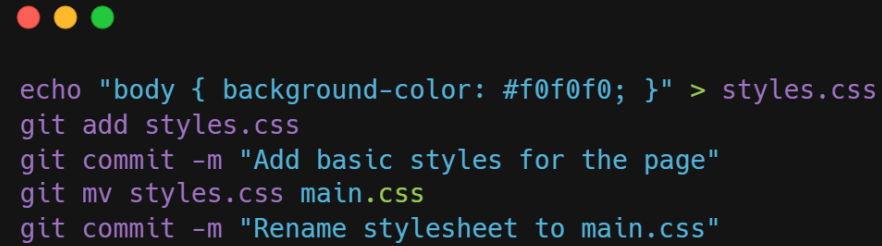
تم إجراء تعديل على ملف `index.html`، واستخدمنا `git diff` لمشاهدة الفروقات، ثم تم حفظ التغيير باستخدام الاختصار `git commit -am`.

A terminal window with a dark background and three colored window control buttons (red, yellow, green) in the top left corner. It contains three lines of text: 'echo "This is my first project with Git." >> index.html', 'git diff', and 'git commit -am "Add welcome paragraph to homepage"'.

```
echo "This is my first project with Git." >> index.html
git diff
git commit -am "Add welcome paragraph to homepage"
```

## المرحلة 5: إضافة ملف جديد وإعادة تسميته

تم إنشاء ملف `styles.css` وحفظه، ثم تم تغيير اسمه إلى `main.css` باستخدام الأمر المخصص `git mv` لحفظ عملية إعادة التسمية في تاريخ Git.

A terminal window with a dark background and three colored window control buttons (red, yellow, green) in the top-left corner. It contains a series of Git commands executed in a shell.

```
echo "body { background-color: #f0f0f0; }" > styles.css
git add styles.css
git commit -m "Add basic styles for the page"
git mv styles.css main.css
git commit -m "Rename stylesheet to main.css"
```

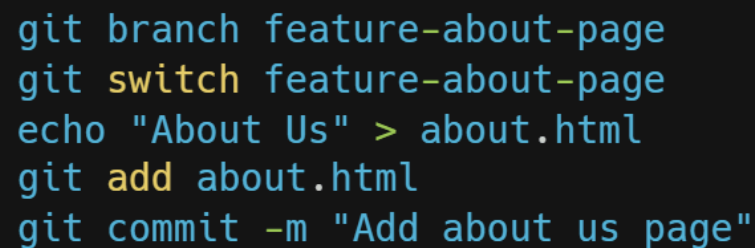
## المرحلة 6: مراجعة سجل التغييرات

تم استخدام `git log` لعرض السجل الكامل للتغييرات، و `git log --oneline --graph` لعرض نسخة مختصرة ومرئية من تاريخ المشروع.



## المرحلة 7: إدارة الفروع (Branching)


لإضافة ميزة جديدة، تم إنشاء فرع 'feature-about-page'. تم الانتقال إليه ('git switch'), وإضافة ملف 'about.html' وحفظه في هذا الفرع المعزول.



```
git branch feature-about-page
git switch feature-about-page
echo "About Us" > about.html
git add about.html
git commit -m "Add about us page"
```

## المرحلة 8: دمج الفروع وتنظيفها (Merging)

تم العودة للفرع الرئيسي `master`، ودمج التغييرات من فرع الميزة باستخدام `git merge`.  
بعد التأكد من نجاح الدمج، تم حذف الفرع باستخدام `git branch -d`.

A terminal window with a dark background and three colored window control buttons (red, yellow, green) in the top left corner. It contains three lines of text: 'git switch master', 'git merge feature-about-page', and 'git branch -d feature-about-page'.

```
git switch master
git merge feature-about-page
git branch -d feature-about-page
```

## المرحلة 9: التكامل مع GitHub ورفع المشروع

تم ربط المستودع المحلي بالبعيد على GitHub. تم تغيير اسم الفرع الرئيسي من `master` إلى `main`، ثم تم رفع المشروع لأول مرة بنجاح.



```
git remote add origin https://github.com/munawwaer/SoftwareEngineering_Munawwaer_Al-Hajj_Saif.git  
git branch -M main  
git push -u origin main
```

## المرحلة 10: إعادة هيكلة المشروع

تم إنشاء مجلد `Lecture\_01` ونقل جميع ملفات المشروع إليه باستخدام `git mv`. تم حفظ هذا التغيير الهيكلي ورفعته إلى GitHub.

```
mkdir Lecture_01  
git mv index.html Lecture_01/  
git mv about.html Lecture_01/  
git mv main.css Lecture_01/  
git commit -m "Organize project files into Lecture_01 folder"  
git push origin main
```

## المرحلة 11: فحص تفاصيل ال Commits


تم استخدام `git show` لفحص محتويات وتفصيل آخر (`HEAD`) وال `commit` الذي يسبقه (`HEAD~1`) بدقة.

```
git show HEAD  
git show HEAD~1
```



## المرحلة 12: التراجع عن تعديل محلي

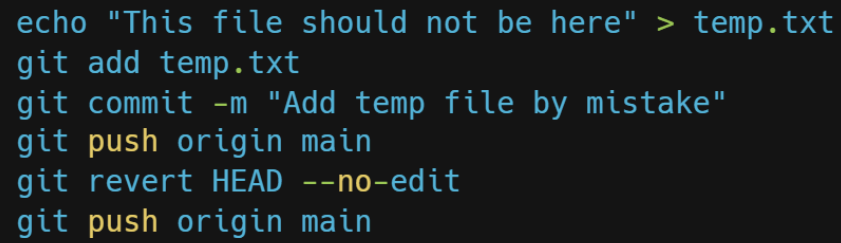
تمت محاكاة خطأ عبر تعديل ملف `index.html`. تم استخدام `git restore` للتراجع الفوري عن هذا التعديل قبل حفظه، وإعادة الملف لحالته النظيفة.



```
echo "!!!--THIS IS A MISTAKE!--!!!" >> Lecture_01/index.html
git status
git restore Lecture_01/index.html
git status
```

## المرحلة 13: التراجع عن خطأ تم رفعه (Revert)

تمت محاكاة خطأ كبير عبر إضافة ملف `temp.txt` ورفعته إلى GitHub. تم استخدام الأمر الآمن `git revert` لإنشاء commit جديد يلغي تأثير commit الخاطئ، ثم تم رفع هذا التراجع لتصحيح الخطأ في المستودع البعيد.



```
echo "This file should not be here" > temp.txt  
git add temp.txt  
git commit -m "Add temp file by mistake"  
git push origin main  
git revert HEAD --no-edit  
git push origin main
```

© 2025 - تقرير تم إعداده كجزء من متطلبات مساق هندسة البرمجيات.