

مقدمة إلى Git: أوامر أساسية في التحكم بالإصدارات

تكاليف المحاضرة الأولى

الطالب: محمد عبده ثابت محمد

المدرس: م/مالك المصنف

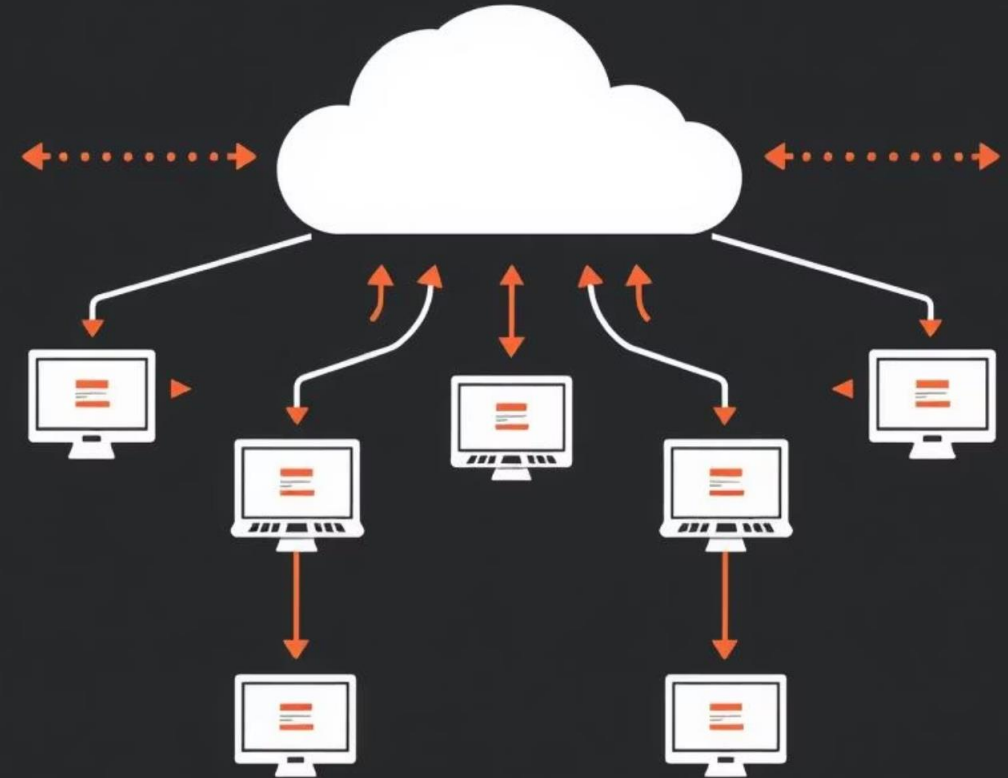


ما هو Git؟

Git هو نظام تحكم في الإصدارات الموزعة (DVCS) يسمح لك بتتبع التغييرات في ملفات مشروعك. تخيل أن لديك آلة زمنية لمشروعك البرمجي!

أهميته تكمن في قدرته على:

- تتبع كل تغيير يتم إجراؤه على الكود.
- التعاون بكفاءة مع فريق من المطورين.
- العودة بسهولة إلى أي إصدار سابق من المشروع.



الإعداد الأولي لمستودع Git



git init

لإنشاء مستودع Git جديد فارغ في المجلد الحالي .
هذه هي نقطة البداية لأي مشروع Git.



git config --global user.email

لتعيين بريدك الإلكتروني الذي سيظهر في سجل
التزاماتك.



git config --global user.name

لتعيين اسم المستخدم الخاص بك، والذي سيُربط
بالتزاماتك.



```
C:\Users\USER\Desktop\Software_Engineering_MohammedAbduThabet
λ git init
Initialized empty Git repository in C:/Users/USER/Desktop/Software_Engineering_MohammedAbduThabet/.git/

C:\Users\USER\Desktop\Software_Engineering_MohammedAbduThabet (master)
λ git config --global user.email
hmdramz41@gmail.com

C:\Users\USER\Desktop\Software_Engineering_MohammedAbduThabet (master)
λ git config --global user.name
sweileh
```

إضافة الملفات وحفظ التغييرات



git status

ما هي الملفات الجديدة،
المعدلة، أو المجهزة؟
لمعرفة حالة مجلد العمل



git add

لإضافة ملفات محددة أو كل التغييرات إلى منطقة
التجهيز، استعدادًا لحفظها.



git commit -m "رسالة"

مع Git لحفظ التغييرات المجهزة بشكل دائم في سجل
رسالة وصفية.



```
C:\Users\USER\Desktop\Software_Engineering_MohammedAbduThabet (master)
```

```
λ git status
```

```
On branch master
```

```
No commits yet
```

```
Untracked files:
```

```
(use "git add <file>..." to include in what will be committed)
```

```
file1.txt
```

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

```
C:\Users\USER\Desktop\Software_Engineering_MohammedAbduThabet (master)
```

```
λ git add file1.txt
```

```
C:\Users\USER\Desktop\Software_Engineering_MohammedAbduThabet (master)
```

```
λ git commit -m "the first file"
```

```
[master (root-commit) c60422f] the first file
```

```
1 file changed, 1 insertion(+)
```

```
create mode 100644 file1.txt
```

ملاحظات هامة حول هذا البروميت:

- التحديد الدقيق: هذا البروميت يحدد كل التفاصيل التي طلبتها. من أسماء الأشخاص إلى نسب الألوان واستخدام الأقواس وشاشات الكود.
- الوضوح لـ Gamma: كلما كان البروميت أكثر تفصيلاً ووضوحاً، زادت دقة Gamma في إنتاج العرض التقديمي كما تتخيله.
- المراجعة النهائية: تذكر دائماً أن تراجع العرض التقديمي بعد أن ينشأ باستخدام هذا البروميت، يجب أن تحصل على عرض تقديمي احترافي ومميز للغاية! بالتوفيق في محاضراتك.

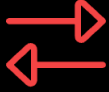


عرض السجل والتنقل



git log

يعرض سجل جميع الالتزامات (commits) في المستودع بترتيب زمني عكسي.



git checkout <commit_id>

للتبديل إلى حالة المستودع في نقطة زمنية محددة، أي عند التزام معين.



```
C:\Users\USER\Desktop\Software_Engineering_MohammedAbduThabet (master)
λ git log
commit e6da82bfa75931f1e39e7a2509c6b3912942101d (HEAD -> master)
Author: sweileh <hmdramz41@gmail.com>
Date: Sun Jul 20 19:23:09 2025 +0300
```

the second file

```
commit c60422f91bbca6b8a7245001f90fbd94abcb4b1
Author: sweileh <hmdramz41@gmail.com>
Date: Sun Jul 20 19:21:57 2025 +0300
```

the first file

```
C:\Users\USER\Desktop\Software_Engineering_MohammedAbduThabet (master)
λ git checkout c60422f91bbca6b8a7245001f90fbd94abcb4b1
Note: switching to 'c60422f91bbca6b8a7245001f90fbd94abcb4b1'.
```

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 c60422f the first file

إدارة الفروع في Git

الفروع تسمح لك بالعمل على ميزات جديدة أو إصلاحات دون التأثير على الخط الرئيسي للمشروع. إنها أساس سير العمل المتوازي في التطوير.

1

`git branch`

عرض الفروع.

2

`git branch <فرع_جديد>`

إنشاء فرع جديد.

3

`git checkout / switch <فرع>`

التبديل بين الفروع.

4

`git merge <فرع>`

دمج التغييرات من فرع إلى الفرع الحالي.

```
C:\Users\USER\Desktop\Software_Engineering_MohammedAbduThabet (second_branch)
λ git branch
  master
  new_branch
  other_branch
* second_branch

C:\Users\USER\Desktop\Software_Engineering_MohammedAbduThabet (second_branch)
λ git switch master
Switched to branch 'master'

C:\Users\USER\Desktop\Software_Engineering_MohammedAbduThabet (master)
λ git merge new_branch
Already up to date.
```

التعامل مع المستودعات البعيدة

المستودعات البعيدة (مثل GitHub) هي نسخ من مشروعك مخزنة على خادم آخر، مما يتيح التعاون ومشاركة الكود.

- `git remote add origin <URL>` لربط مستودعك المحلي بمستودع بعيد.
- `git pull origin <فرع>` لجلب التغييرات من المستودع البعيد ودمجها محليًا.
- `git push -u origin <فرع>` لدفع التغييرات المحلية إلى المستودع البعيد.
- `git clone <URL>` لإنشاء نسخة محلية كاملة من مستودع بعيد موجود.



```
C:\Users\USER\Desktop\Software_Engineering_MohammedAbduThabet (main)
λ git remote add origin https://github.com/swdesign2024/Software_Engineering_MohammedAbduThabet.git

C:\Users\USER\Desktop\Software_Engineering_MohammedAbduThabet (main)
λ git shortlog -s -n --all
3 sweileh

C:\Users\USER\Desktop\Software_Engineering_MohammedAbduThabet (main)
λ git remote -v
origin https://github.com/swdesign2024/Software_Engineering_MohammedAbduThabet.git (fetch)
origin https://github.com/swdesign2024/Software_Engineering_MohammedAbduThabet.git (push)
```



أوامر Git إضافية ومفيدة



git ls-files

لسرد الملفات التي يتتبعها Git.



git restore <file>

للتراجع عن التغييرات المحلية.



git rm <file>

لحذف ملف من مجلد العمل.



git clean -df

لإزالة الملفات والمجلدات غير المتتبعة.



git reset <file>

لإلغاء تجهيز ملف.



```
C:\Users\USER\Desktop\Software_Engineering_MohammedAbduThabet (master)
λ git ls-files
file1.txt
file2.txt
```

```
C:\Users\USER\Desktop\Software_Engineering_MohammedAbduThabet (master)
λ git clean -df
Removing file3.txt
```


نصائح وممارسات جيدة مع Git

الالتزام المنتظم والصغير:

احفظ عملك بتغييرات صغيرة ومتكررة لسهولة التتبع والعودة.

رسائل التزام واضحة وموجزة:

اكتب رسائل تصف بدقة ما تم تغييره في كل التزام.

العمل على فروع منفصلة:

استخدم فروعًا للميزات الجديدة أو إصلاحات الأخطاء لتجنب التأثير على الكود الرئيسي.

الجلب والدفع بانتظام:

حافظ على تزامن مستودعك المحلي مع البعيد لتجنب تعارضات الدمج.



النهاية!

Good luck