



Git & GitHub

المحاضرة الثالثة

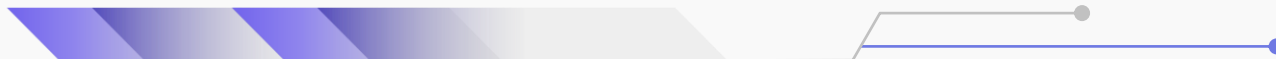
م. محمد السلوم





مدخل GitHub

Git أداة عمل موزع Distributed	جلب التعديلات المستودع البعيد الى المستودع المحلي pull
تسجيل حساب جديد في GitHub	انشاء مستودع بعيد محليا من مستودع محلي موجود
إنشاء مستودع في GitHub	انشاء الفروع Branches في GitHub
git clone	مخطط عمل GitHub
ارسال التعديلات المحلية الى المستودع البعيد push	خدمة GitHub Pages لاستضافة صفحات الويب
تعديل الملفات في GitHub مباشرة	الاشتقاق في GitHub





Git أداة عمل موزع Distributed

01

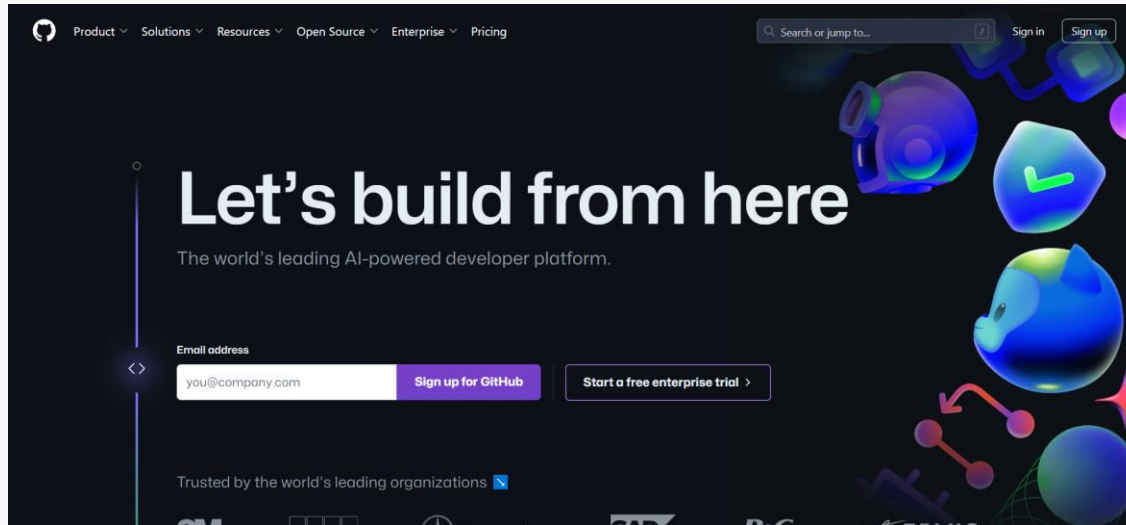
Git هو نظام تحكم بالإصدارات موزع (Distributed Version Control System) تم تصميمه في الأصل لتمكين أعضاء الفرق البرمجية من العمل بشكل مستقل تمامًا يوفر Git لكل عضو في الفريق نسخة محلية كاملة (clone) من المستودع الذي يتم العمل عليه هذه النسخة تحتوي على كل الملفات، الفروع، وتاريخ المشروع. عندما يحتاج الفريق إلى دمج المساهمات من أعضاء مختلفين يعملون على أجهزة متعددة، يمكنهم الاعتماد على مستودع بعيد (Remote) واحد أو أكثر، والذي يعتبر المصدر الموثوق للشفرة البرمجية (Source of Truth) في النهاية المستودع البعيد ما هو إلا نسخة مكررة clone من المستودع المحلي لتنظيم هذه العملية وتسهيلها، ظهرت مجموعة من المنصات التي تقدم خدمات إدارة المستودعات البعيدة، أشهرها GitHub و GitLab و BitBucket، في هذا الفصل، سنركز على GitHub، حيث سنتناول ميزاته وكيفية استخدامه بشكل مفصل.



تسجيل حساب جديد في GitHub

02

تسجيل الحساب في GitHub بسيط وسهل ومباشر و كل ما نحتاجه هو الانتقال إلى الموقع الرسمي
github.com ثم الضغط على زر **sign up** ثم نقوم بملء النموذج و تسجيل الحساب بشكل تقليدي،





إنشاء مستودع في GitHub

03

بعد تسجيل الحساب يمكننا إنشاء مستودع على GitHub مباشرة من خلال اختيار repository
و ملأ تفاصيل المستودع (اسمه و وصفه .. إلخ) كما تبين الصورة أدناه.

New repository

Search

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)

Required fields are marked with an asterisk (*).

Owner * Repository name *

mohamd1542 /

Great repository names are short and memorable. Need inspiration? How about [upgraded-octo-fortnight](#) ?

Description (optional)

☒ Public
Anyone on the internet can see this repository. You choose who can commit.

☐ Private
You choose who can see and commit to this repository.

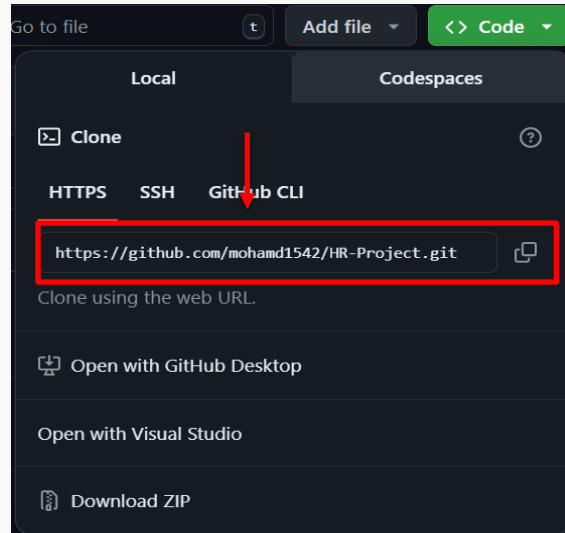
Initialize this repository with:

☐ Add a README file
This is where you can write a long description for your project. [Learn more about READMEs.](#)



إنشاء مستودع في GitHub

أهم خيار هو جعل المستودع خاصاً Private، أو عاماً Public، حيث أن جعل المستودع عاماً Public يعني أنه أصبح متاحاً لكل الناس للاطلاع عليه، و نسخه، و المساهمة فيه حتى إذا لم تضبط بعض إعدادات الحماية. خلال و الآن نضغط على زر **Create repository** لإتمام إنشاء المستودع البعيد على **GitHub**



و ثم سننتقل آلياً إلى صفحة المستودع الذي قمنا بإنشائه.

و لعل أهم ما في الصفحة في هذه المرحلة هو رابط

المستودع البعيد الذي يظهر في الصورة و الذي

سنستعمله بعد لمزامنة مستودعنا المحلي الذي

أعدناه سابقاً، مع المستودع البعيد الذي أنشأناه

للتو.

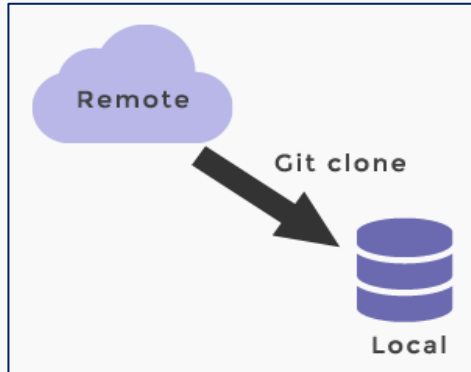


git clone

04

الأمر **git clone** هو أحد الأوامر الأساسية في Git، ويستخدم لإنشاء نسخة محلية من مستودع بعيد. عندما تقوم بتنفيذ هذا الأمر، فإنك تنسخ جميع ملفات المشروع، بالإضافة إلى كامل تاريخ الإصدارات (commits) والفروع (branches) الموجودة في المستودع البعيد.

```
git clone <repository-url>
```



وهذا يسمح لك بالعمل على نسخة محلية من المشروع بحيث يمكنك إجراء التعديلات والتفاعل مع المشروع دون الحاجة إلى الاتصال المستمر بالخادم البعيد.



git clone

04

```
git clone https://github.com/mohamd1542/HR-Project.git
```



MINGW64:/c/Users/MOHAMD-SALOUM/Desktop/project

```
MOHAMD-SALOUM@DESKTOP-K2UL453 MINGW64 ~/Desktop/project
$ git clone https://github.com/mohamd1542/HR-Project.git
Cloning into 'HR-Project'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused
from 0)
Receiving objects: 100% (3/3), done.
```




ارسال التعديلات المحلية الى المستودع البعيد push

يستخدم git في عمل دفع **push** للتعبير عن إرسال التعديلات المحلية إلى المستودع البعيد، وكثيراً ما يقال تدفع التعديلات "بمعنى أنّ نزامها مع المستودع البعيد من خلال الأمر **push** الآن مستودعنا البعيد **origin** و الموجود على GitHub لا يحتوي أية ملفات كونه أنشئ للتو، لذلك يمكننا مزامنة التعديلات الموجودة محلياً معه من خلال تنفيذ الأمر:

```
git push RemoteName BranchName
```

```
MOHAMD-SALOUM@DESKTOP-K2UL453 MINGW64 ~/Desktop/MyFolder/myproject (main)
$ git push origin main
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 12 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (4/4), 344 bytes | 344.00 KiB/s, done.
Total 4 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/mohamd1542/myproject.git
da61ed2..8ce4dcd main -> main
```

```
git push
```

اما في حال كنا في فرع الماستر فنستخدم الامر:



تعديل الملفات في GitHub مباشرة

06

The screenshot shows the GitHub web interface for a repository named 'myproject' by user 'mohamd1542'. The file 'index.html' is selected in the left sidebar. The main area displays the code for 'index.html' with a commit hash 'a4cea73' and a commit message 'newChang'. The code is as follows:

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <title>Document</title>
7 </head>
8 <body>
9   <div>Welcome in our Session</div>
10  <h1>Hello</h1>
11 </body>
12 </html>
```

In the top right corner of the code viewer, there is a toolbar with icons for 'Raw', 'Download', 'Edit' (pencil icon), and 'Add' (plus icon). The 'Edit' icon is highlighted with a red box, and a red arrow points to it from the right side of the screen.



تعديل الملفات في GitHub مباشرة

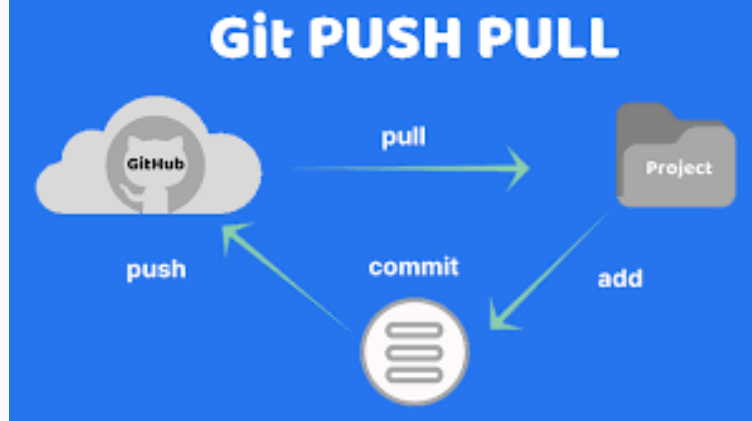
06

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <title>Document</title>
7 </head>
8 <body>
9   <div>Welcome in our Session</div>
10   <h1>Hello</h1>
11   <h2>Hello</h2>
12 </body>
13 </html>
14
```



جلب التعديلات المستودع البعيد الى المستودع المحلي pull

يستخدم في git على استخدام الفعل سحب pull للتعبير عن طلب التعديلات البعيدة و جلبها إلى المستودع المحلي، و كثيراً ما يقال تنسحب التعديلات بمعنى أنّ نزامن المستودع المحلي مع التحديثات التي دفعت على المستودع البعيد من قبل زملاء العمل





جلب التعديلات المستودع البعيد الى المستودع المحلي pull

لتجربة ذلك فلنقم بجلب التعديل الذي أجريناه على المستودع البعيد (مستودع GitHub) في الفقرة السابقة إلى المستودع المحلي و ذلك من خلال تنفيذ الأمر `git pull` كما توضح الصورة أدناه.

```
MOHAMD-SALOUM@DESKTOP-K2UL453 MINGW64 ~/Desktop/MyFolder/myproject (main)
$ git pull
remote: Enumerating objects: 8, done.
remote: Counting objects: 100% (8/8), done.
remote: Compressing objects: 100% (6/6), done.
remote: Total 6 (delta 4), reused 0 (delta 0), pack-reused 0 (from 0)
Unpacking objects: 100% (6/6), 1.84 KiB | 41.00 KiB/s, done.
From https://github.com/mohamd1542/myproject
  4bb805a..833cbf7  main      -> origin/main
Updating 4bb805a..833cbf7
Fast-forward
 index.html | 4 +++-
 1 file changed, 3 insertions(+), 1 deletion(-)
```

أحياناً ، إذا كان ثمة تعديلات على المستودع البعيد لم يتم سحبها بعد، فلن يسمح لك git بالقيام بعمل دفع push وسيطلب منك سحبها أولاً، وقد تحدث أثناء ذلك تعارضات تتطلب الحل



انشاء مستودع بعيد محليا من مستودع محلي موجود

ولنفرض ان لدينا مستودع محلي ونريد رفعه على github لمزامنة المستودع المحلي الذي لدينا مع المستودع البعيد الذي أنشأناه على GitHub نحتاج إلى تعريفه ضمن المستودعات البعيدة محلياً، ويتم

```
git remote add remote-name github-repo-url
```

ذلك من خلال الأمر:

حيث يعبر remote-name عن الاسم الذي سنسمي به المستودع البعيد (عادة يسمى origin)، بينما يعبر github-repo url عن رابط المستودع البعيد الذي قمنا بانشاءه، و لإضافة مستودعنا المحلي إلى البعيد يصبح الأمر:

```
MOHAMD-SALOUM@DESKTOP-K2UL453 MINGW64 ~/Desktop/test (master)
$ git remote add origin https://github.com/mohamd1542/Course.git
```



انشاء مستودع بعيد محليا من مستودع محلي موجود

بعد ذلك لدينا العملية التالية :

```
git push -u origin main
```

حيث انه نفس امر **push** العادي الذي يرفع التعديلات الى المستودع البعيد الخاص بنا ولكن **-u** حيث ان **-u** تعبر على انه تمكنتك من عمل **pull** في البداية ومن ثم عمل **push** ولكن لماذا نقوم بذلك لانه بفرض كان هناك اكثر من شخص بالفريق فمممكن ان يكونو قد قامو برفع بعض التعديلات ومن غير المنطقي ان تقوم بالرفع بشكل مباشر لانه من الممكن ان يحدث تعارضات كثيرة وبالتالي هي مسؤولة عن جلب اخر نسخة من المشروع ومن ثم حل التعارض في حال كان موجود ومن ثم الرفع



انشاء مستودع بعيد محليا من مستودع محلي موجود

كما هو موضح بالصورة التالية:

```
MOHAMD-SALOUM@DESKTOP-K2UL453 MINGW64 ~/Desktop/test (master)
$ git push -u origin master
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Writing objects: 100% (3/3), 207 bytes | 207.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/mohamd1542/Course.git
 * [new branch]      master -> master
branch 'master' set up to track 'origin/master'.
```

وبذلك نكون قد رفعنا المستودع المحلي الخاص بنا الى مستودع بعيد

ويمكننا استعراض المستودعات البعيدة من خلال الأمر:

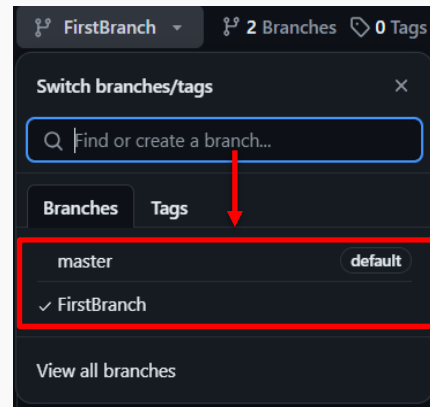
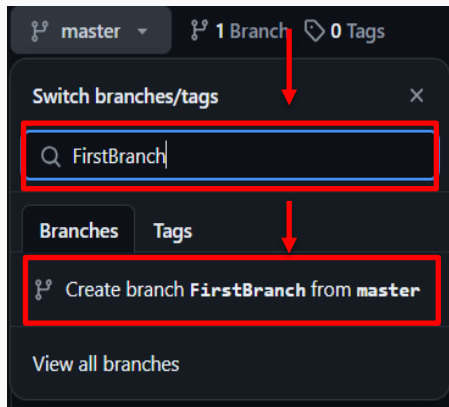
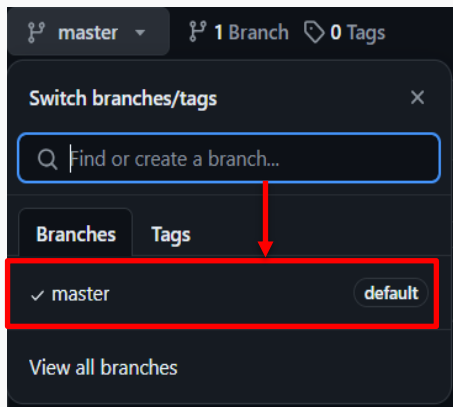
```
git remote -v
```



انشاء الفروع Branches في GitHub

09

نستطيع إنشاء فرع جديد للمشروع الذي نعمل عليه في منصة GitHub بالدخول إلى المستودع البعيد الخاص بنا و النقر على زر الفرع الرئيسي master ثم كتابة اسم الفرع الجديد مع الحرص على كتابة اسم يصف الفرع و الهدف منه، ثم ننقر على زر Create Branch





لأن git مرنة بحيث لا تفرض على مستخدميها أية قيود في ما يتعلق بكيفية استخدامها و توظيفها ضمن فريق العمل، قامت بعض الشركات بعمل معايير standards لاستخدام git لعل أشهرها و مخططات عمل GitHub أو GitHub Workflow والذي يركّز على توظيف إمكانيات git GitHub كافة للعمل بشكل معياري في فرق العمل بحيث لا يحتاج المساهمون الجدد أو الموظفون الجدد إلى كثير من الشرح للبدء بالمساهمة الفعلية في المشاريع. يركز أسلوب العمل هذا على الاستخدام المكثف للمفروع بحيث يمكن الفرق من اختبار التعديلات المزايا الجديدة بحرية و نشر البرمجيات deployment بانتظام و فكرة عمل هذا المخطط كما يلي :



- إنشاء فرع جديد : ينشئ عضو الفريق فرعاً جديداً بهدف عدم التأثير على الفروع الأخرى المستقرة، عادة تكون الفروع المستقرة بأسماء master الذي يستعمل للبيئة الإنتاجية production و develop الذي يستعمل لبيئات التطوير، أو اختبارات ضمان الجودة QA أو تجارب العملاء المبكرة staging
- إيداع المساهمات: يقوم عضو الفريق بإيداع المساهمات الخاصة بمهمته ضمن الفرع الذي أنشأه في الخطوة السابقة.
- فتح طلب سحب: Pull request في GitHub طلبات السحب Pull Request معلقة تتطلب مراجعة المساهمات من قبل واحد أو أكثر من أعضاء الفريق و عملية دمج هي Merge



- الموافقة عليها قبل أن يتم دمجها ، و في بعض البيئات تسمى طلب دمج Merge-Request والمقصود بها هنا أنّ عضو الفريق انتهى من المساهمات في فرعها و الآن يطلب من فريقه الموافقة على دمجها مع فروع المشروع المستقرة بهدف نشرها.
- مرحلة المراجعة Review Code تابعة للمرحلة السابقة حيث يقوم أعضاء الفريق الآخرون بمراجعة المساهمات التي طلب زميلهم دمجها، و عادة ما تحدث نقاشات مفيدة وإيجابية هنا لتحسين جودة المساهمات.
- مرحلة الدمج والنشر: يتم دمج الفرع المطلوب و نشره Deploy بشكل آلي أو يدوي.



خدمة GitHub Pages لاستضافة صفحات الويب

11

تتيح منصة GitHub خدمة استضافة صفحات الويب الموجودة في مستودع بعيد مباشرة





تتيح منصة GitHub لمستخدميها اشتقاق مستودعات جديدة خاصة أو عامة من مستودعات المستخدمين الآخرين العامة، وذلك بواسطة الأمر fork الذي يسمح لك بالبدء بمستودع جديد بناءً على ما تم بالفعل في مستودع مشروع آخر سواء كان المشروع الآخر مملوكاً لك أو عاماً. على أن fork نفسها ليست أحد الأوامر في نظام git، بل هي خاصية توفرها منصة GitHub والمنصات المشابهة ولاشتقاق مستودع ما يمكنك بكل بساطة الدخول إلى المستودع المطلوب ثم الضغط على زر Fork الذي توضحه الصورة التالية:



الاشتقاق في GitHub

12

hiusrapi Private

main 2 Branches 0 Tags

Go to file Add file <> Code About

mohamd1542 Merge pull request #13 from abdalmohaymenesmaeel/IdintiyBranch 4f1e5b0 · 5 days ago 34 Commits

src	Update End Point => Get List Months For Employee	5 days ago
.editorconfig	first intial	last month
.gitignore	first intial	last month
AARProject.nuspec	first intial	last month
CODE_OF_CONDUCT.md	first intial	last month
HRProject.sln	first intial	last month
LICENSE	first intial	last month
README.md	first intial	last month
global.json	first intial	last month

README Code of conduct MIT license

Releases: No releases published. Packages: No packages published.

Contributors 2

Create a new fork

A fork is a copy of a repository. Forking a repository allows you to freely experiment with changes without affecting the original project.

Required fields are marked with an asterisk (*).

Owner * mohamd1542 Repository name * hiusrapi
hiusrapi is available.

By default, forks are named the same as their upstream repository. You can customize the name to distinguish it further.

Description (optional)

☒ Copy the main branch only

Contribute back to abdalmohaymenesmaeel/hiusrapi by adding your own branch. [Learn more.](#)

You are creating a fork in your personal account.

Create fork



والحمد لله رب العالمين