



git + GitHub

اعداد :

م. أحمد حاج محمد سلامة

الجامعة الدولية للعلوم والنهضة



# ماذا سوف نتعلم في هذا المقرر

## مدخل GitHub

Git أداة عمل موزع Distributed

تسجيل حساب جديد في GitHub

إنشاء مستودع في GitHub

git clone

تعريف المستودع بعيد محليا

ارسال التعديلات المحلية الى المستودع البعيد push

تعديل الملفات في GitHub مباشرة

جلب التعديلات المستودع البعيد الى المستودع المحلي pull

## مدخل GitHub

انشاء الفروع Branches في GitHub

مخطط عمل GitHub

خدمة GitHub Pages لاستضافة صفحات

الويب

الاشتقاق في GitHub



# اساسيات Git



git أداة عمل موزَّع Distributed صممت git في الأصل لتتيح لأعضاء فرق العمل المساهمة بشكل مستقل تماماً بحيث يملك كل عضو من أعضاء الفريق نسخته الخاصة clone من المستودع الخاص بالمشروع الذي يتم العمل عليه، ثمَّ عند الحاجة لدمج المساهمات القادمة من مساهمين مختلفين يعملون على أكثر من جهاز يمكن الاعتماد على مستودع بعيد Remote واحد أو أكثر يكون بمثابة المصدر الموثوق للشفيرة اصطلاحاً Source of truth أنه مع في نهاية المطاف ما هو إلا نسخة Clone من المستودع نفسه.

لتنظيم هذه العملية و تسهيلها ظهرت مجموعة من المنصات أشهرها GitHub و BitBucket و Gitlab و سنقوم في هذا الفصل بالحديث عن GitHub بشكل مفصل.

git أداة عمل موزَّع Distributed

تسجيل حساب جديد في GitHub

إنشاء مستودع في GitHub

تعريف المستودع بعيد محلياً

ارسال التعديلات المحلية الى

المستودع البعيد push

تعديل الملفات في GitHub مباشرة

جلب التعديلات المستودع البعيد الى

المستودع المحلي pull



# اساسيات Git



تسجيل الحساب في GitHub بسيط و مباشر و كل ما تحتاجه هو الانتقال إلى الموقع [github.com](https://github.com) ثم الضغط على زر sign up ثم تقوم بملء النموذج و تسجيل الحساب بشكل تقليدي، و ربما تكون التوصية الوحيدة التي أنصح بها هنا هي تسجيل الحساب بنفس عنوان البريد الذي استخدمناه في تهيئة git في الفصل الأول.



git أداة عمل موزع Distributed

تسجيل حساب جديد في GitHub

إنشاء مستودع في GitHub

تعريف المستودع بعيد محليا

ارسال التعديلات المحلية الى

المستودع البعيد push

تعديل الملفات في GitHub مباشرة

جلب التعديلات المستودع البعيد الى

المستودع المحلي pull



# اساسيات Git



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.

**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 \*

ahmadSalameh1995 /

Great repository names are short and memorable. Need inspiration? How about [psychic-waffle](#) ?

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.](#)

[Add .gitignore](#)

git أداة عمل موزع Distributed  
تسجيل حساب جديد في GitHub  
إنشاء مستودع في GitHub  
تعريف المستودع بعيد محليا  
ارسال التعديلات المحلية الى  
المستودع البعيد push  
تعديل الملفات في GitHub مباشرة  
جلب التعديلات المستودع البعيد الى  
المستودع المحلي pull

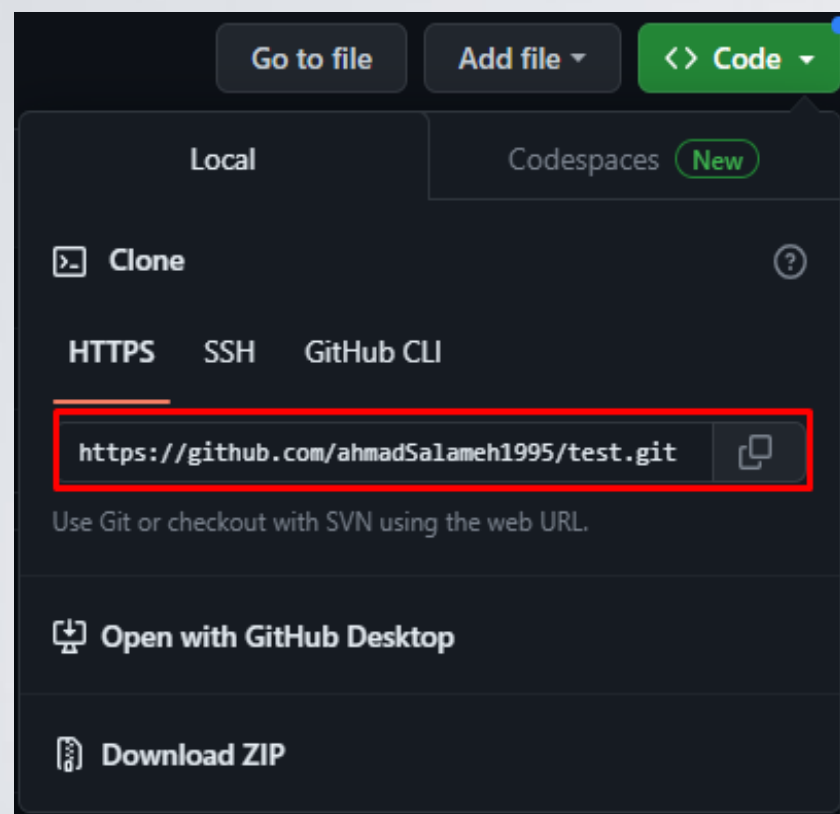




# اساسيات Git



أهم خيار ربّما في هذه الشاشة هو خيار جعل المستودع خاصاً Private، أو عاماً Public، حيث أن جعل المستودع عاماً Public يعني أنّه أصبح متاحاً لكل الناس للاطلاع عليه، و نسخه، و المساهمة فيه حتّى إذا لم تضبط بعض إعدادات الحماية.



خلال و الآن نضغط على زر Create repository لإتمام إنشاء المستودع البعيد على GitHub زمن لا يذكر 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 أداة عمل موزع Distributed

تسجيل حساب جديد في GitHub

إنشاء مستودع في GitHub

تعريف المستودع بعيد محلياً

ارسال التعديلات المحلية الى

المستودع البعيد push

تعديل الملفات في GitHub مباشرة

جلب التعديلات المستودع البعيد الى

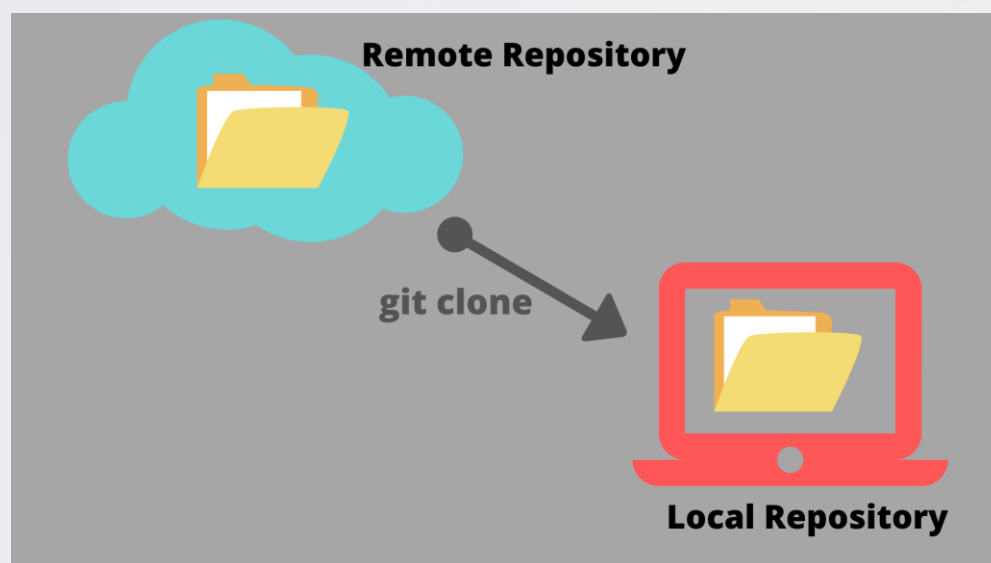
المستودع المحلي pull



# اساسيات Git



تعلية **git clone** هي أمر تُستخدم في نظام التحكم بالنسخ الموزع Git لنسخ مشروع أو مستودع Git بالكامل من مكانه البعيد (مثل مستضيف على خادم عن بُعد) إلى جهاز الكمبيوتر المحلي. هذا يسمح لك بالعمل على نسخة محلية من المشروع بحيث يمكنك إجراء التعديلات والتفاعل مع المشروع دون الحاجة إلى الاتصال المستمر بالخادم البعيد .



git أداة عمل موزع Distributed

تسجيل حساب جديد في GitHub

إنشاء مستودع في GitHub

git clone

تعريف المستودع بعيد محليا

ارسال التعديلات المحلية الى

المستودع البعيد push

تعديل الملفات في GitHub مباشرة

جلب التعديلات المستودع البعيد الى

المستودع المحلي pull



# اساسيات Git



```
git clone https://github.com/ahmadSalameh1995/Ahmad_Work.git
```

```
A. Salama@DESKTOP-409U06R MINGW64 ~/Desktop/asd
$ git clone https://github.com/ahmadSalameh1995/Ahmad_Work.git
Cloning into 'Ahmad_Work'...
remote: Enumerating objects: 6, done.
remote: Counting objects: 100% (6/6), done.
remote: Compressing objects: 100% (3/3), done.
remote: Total 6 (delta 0), reused 3 (delta 0), pack-reused 0
Receiving objects: 100% (6/6), done.
```

git أداة عمل موزع Distributed 

تسجيل حساب جديد في GitHub 

إنشاء مستودع في GitHub 


git clone 

تعريف المستودع بعيد محليا 

ارسال التعديلات المحلية الى 

المستودع البعيد push

تعديل الملفات في GitHub مباشرة 

جلب التعديلات المستودع البعيد الى 

المستودع المحلي pull





# اساسيات Git



لمزامنة المستودع المحلي الذي أعددناه في فقرة سابقة على حاسوبنا الشخصي، مع المستودع البعيد الذي أنشأناه للتو على GitHub نحتاج إلى تعريفه ضمن المستودعات البعيدة محلياً، ويتم ذلك من خلال الأمر

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

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

```
A.Salama@DESKTOP-4O9U06R MINGW64 ~/Desktop/Ahmad_Work.git (master)
$ git remote add origin https://github.com/ahmadSalameh1995/Ahmad_Work.git
```

git أداة عمل موزع Distributed 

تسجيل حساب جديد في GitHub 

إنشاء مستودع في GitHub 

git clone 

تعريف المستودع بعيد محلياً 

ارسال التعديلات المحلية الى 

المستودع البعيد push

تعديل الملفات في GitHub مباشرة 

جلب التعديلات المستودع البعيد الى 

المستودع المحلي pull



# اساسيات Git



و الآن يمكننا استعراض المستودعات البعيدة من خلال الأمر

`git remote -v`

للتأكد من أن المستودع أضيف بشكل صحيح كما يظهر في الصورة أدناه.

```
A.Salama@DESKTOP-409U06R MINGW64 ~/Desktop/Ahmad_Work.git (master)
$ git remote -v
origin https://github.com/ahmadSalameh1995/Ahmad_Work.git (fetch)
origin https://github.com/ahmadSalameh1995/Ahmad_Work.git (push)
```

git أداة عمل موزع Distributed 

تسجيل حساب جديد في GitHub 

إنشاء مستودع في GitHub 

git clone 

تعريف المستودع بعيد محليا 

ارسال التعديلات المحلية الى 

المستودع البعيد push

تعديل الملفات في GitHub مباشرة 

جلب التعديلات المستودع البعيد الى 

المستودع المحلي pull



# اساسيات Git



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

git push

بمعنى أننا سندفع التعديلات المحلية إلى المستودع البعيد ذي الاسم (origin) كما في الصورة أدناه.

```
A.Salama@DESKTOP-409U06R MINGW64 ~/Desktop/Ahmad_Work.git (master)
$ git push
Enumerating objects: 8, done.
Counting objects: 100% (8/8), done.
Delta compression using up to 4 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (6/6), 516 bytes | 258.00 KiB/s, done.
Total 6 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/ahmadSalameh1995/Ahmad_Work.git
1761c66..f1bab9e master -> master
```

**ملاحظة:** اذا كنا بغير فرع ال master نستخدم الامر

git push origin اسم الفرع

git أداة عمل موزع Distributed

تسجيل حساب جديد في GitHub

إنشاء مستودع في GitHub

git clone

تعريف المستودع بعيد محلياً

ارسال التعديلات المحلية الى

المستودع البعيد push

تعديل الملفات في GitHub مباشرة

جلب التعديلات المستودع البعيد الى

المستودع المحلي pull



# اساسيات Git

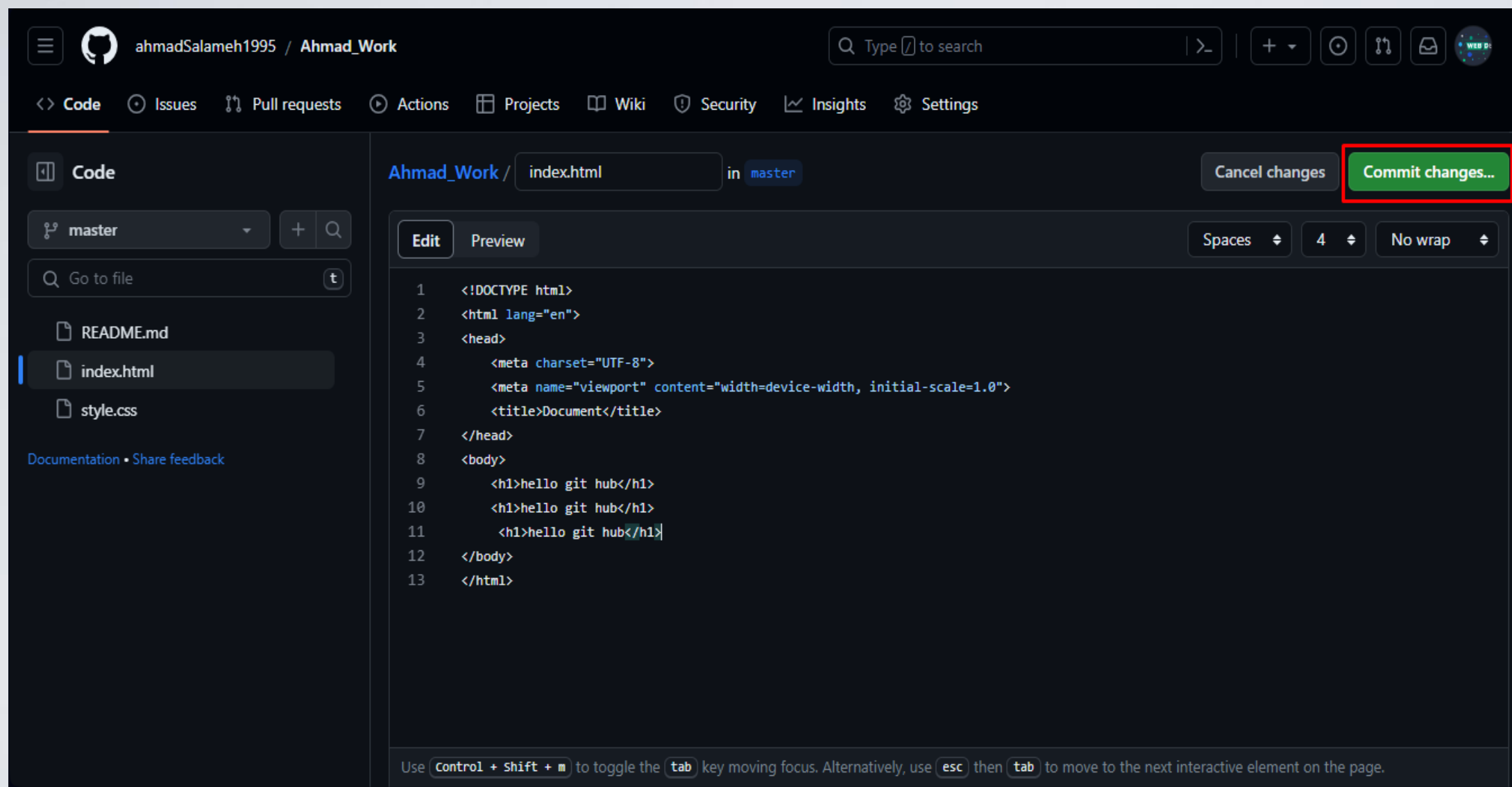


```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <title>Document</title>
7 </head>
8 <body>
9   <h1>hello git hub</h1>
10  <h1>hello git hub</h1>
11 </body>
12 </html>
```

- git أداة عمل موزع Distributed
- تسجيل حساب جديد في GitHub
- إنشاء مستودع في GitHub
- git clone
- تعريف المستودع بعيد محليا
- ارسال التعديلات المحلية الى المستودع البعيد push
- تعديل الملفات في GitHub مباشرة
- جلب التعديلات المستودع البعيد الى المستودع المحلي pull



# اساسيات Git



git أداة عمل موزع Distributed

تسجيل حساب جديد في GitHub

إنشاء مستودع في GitHub

git clone

تعريف المستودع بعيد محليا

ارسال التعديلات المحلية الى

المستودع البعيد push

تعديل الملفات في GitHub مباشرة

جلب التعديلات المستودع البعيد الى

المستودع المحلي pull

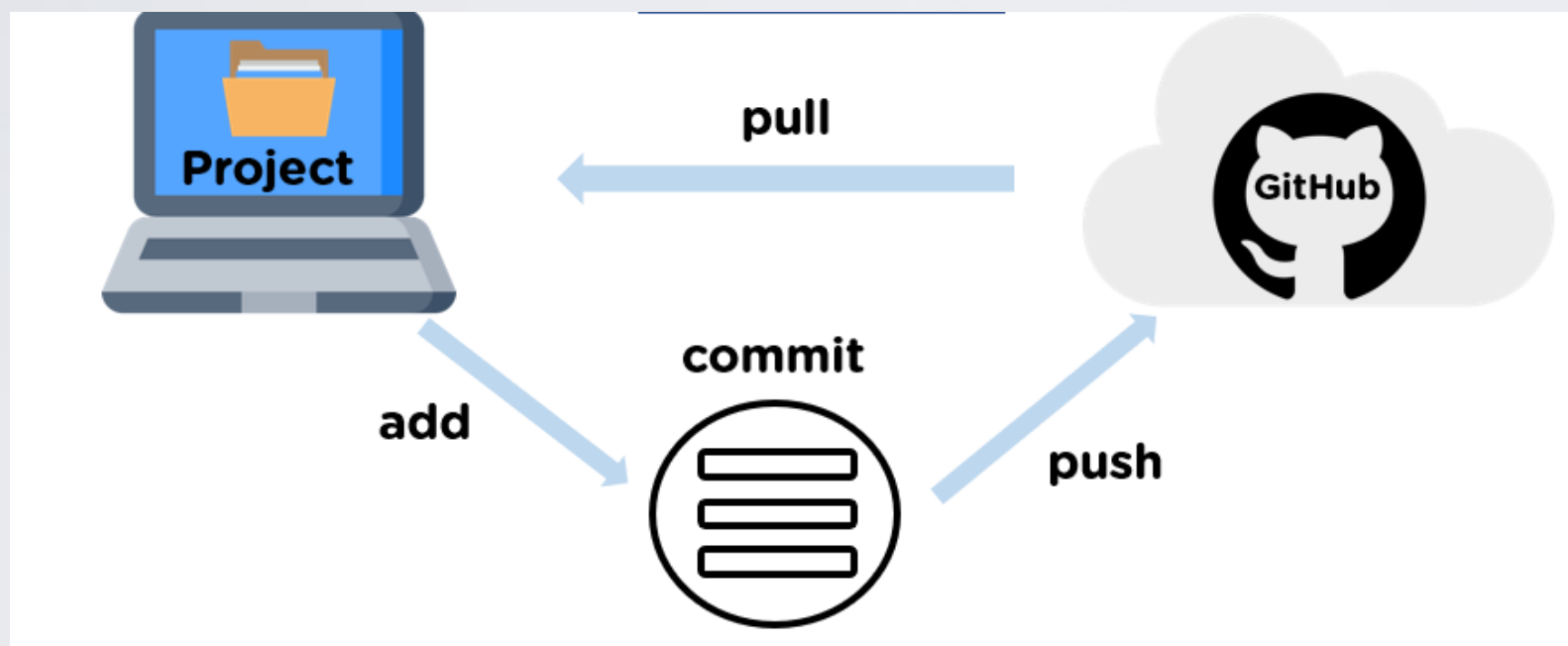




# اساسيات Git



يستخدم git في عمل موزع Distributed  
يستخدم 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



- git أداة عمل موزع Distributed
- تسجيل حساب جديد في GitHub
- إنشاء مستودع في GitHub
- git clone
- تعريف المستودع بعيد محلياً
- ارسال التعديلات المحلية الى المستودع البعيد push
- تعديل الملفات في GitHub مباشرة
- جلب التعديلات المستودع البعيد الى المستودع المحلي pull



# اساسيات Git



لتجربة ذلك فلنقم ب جلب 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.

```
A. Salama@DESKTOP-409U06R MINGW64 ~/Desktop/Ahmad_Work.git (master)
$ git pull
remote: Enumerating objects: 10, done.
remote: Counting objects: 100% (10/10), done.
remote: Compressing objects: 100% (6/6), done.
remote: Total 8 (delta 1), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (8/8), 1.95 KiB | 49.00 KiB/s, done.
From https://github.com/ahmadSalameh1995/Ahmad_Work
   f1bab9e..95764fa  master      -> origin/master
   * [new branch]    main        -> origin/main
Updating f1bab9e..95764fa
Fast-forward
 README.md | 2 ++
 index.html | 3 ++-
 2 files changed, 4 insertions(+), 1 deletion(-)
 create mode 100644 README.md
```

ملحوظة : أحياناً ، و إذا كان ثمة تعديلات على المستودع البعيد لم يتم سحبها بعد، فلن يسمح لك git بالقيام بعمل دفع push و سيطلب منك سحبها أولاً، و قد تحدث أثناء ذلك تعارضات تتطلب الحل بنفس الأسلوب الذي استعرضناه في الفصل الماضي.

git أداة عمل موزع Distributed

تسجيل حساب جديد في GitHub

إنشاء مستودع في GitHub

git clone

تعريف المستودع بعيد محلياً

ارسال التعديلات المحلية الى

المستودع البعيد push

تعديل الملفات في GitHub مباشرة

جلب التعديلات المستودع البعيد الى

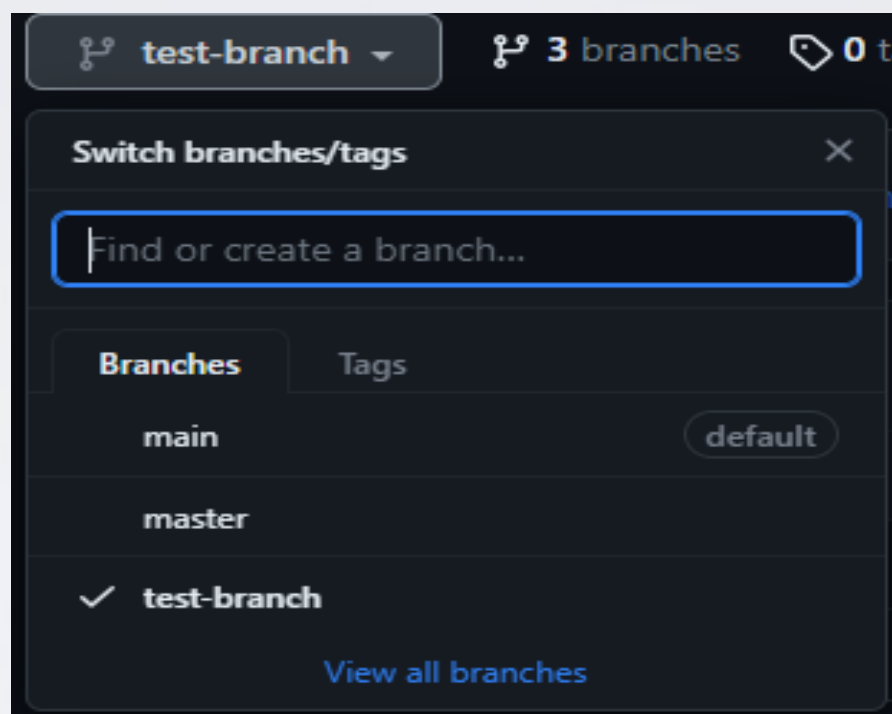
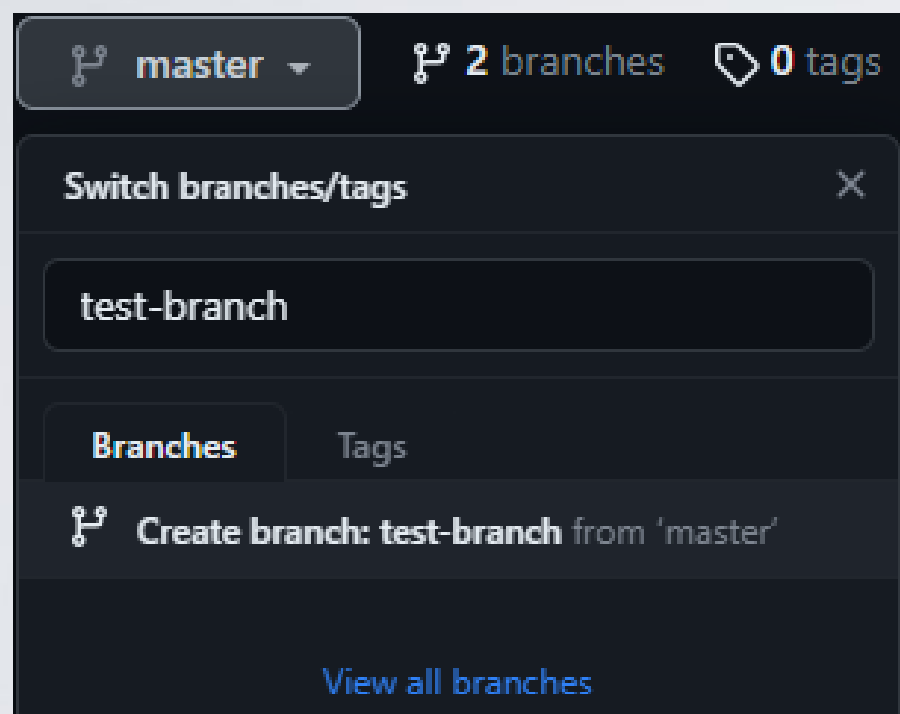
المستودع المحلي pull



# اساسيات Git



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



إنشاء الفروع في GitHub

مخطط عمل GitHub

خدمة GitHub Pages

الاشتقاق في GitHub

استنساخ المستودعات في GitHub

GitHub



# اساسيات Git



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 و الذي يركز على توظيف إمكانيات GitHub git 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 للفروع بحيث يمكن الفرق من اختبار التعديلات المزايا الجديدة بجرية و نشر البرمجيات deployment بانتظام و فكرة عمل هذا المخطط كما يلي :

- إنشاء الفروع في GitHub
- مخطط عمل GitHub
- خدمة GitHub Pages
- الاشتقاق في GitHub
- استنساخ المستودعات في GitHub



# اساسيات Git



- 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

- إنشاء الفروع في GitHub
- مخطط عمل GitHub
- خدمة GitHub Pages
- الاشتقاق في GitHub
- استنساخ المستودعات في GitHub





# اساسيات Git



- الموافقة عليها قبل أن يتم دمجها ، و في بعض البيئات تسمى طلب دمج Merge-Request و المقصود بها هنا أن عضو الفريق انتهى من المساهمات في فرع 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 و النشر : يتم دمج الفرع المطلوب و نشره Deploy بشكل آلي أو يدوي.

GitHub إنشاء الفروع في

GitHub مخطط عمل

GitHub Pages خدمة

GitHub الاشتقاق في

استنساخ المستودعات في  
GitHub



# اساسيات Git



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




github:pages

انشاء الفروع في GitHub 

مخطط عمل GitHub 

خدمة GitHub Pages 

الاشتقاق في GitHub 

استنساخ المستودعات في 

GitHub



# اساسيات Git

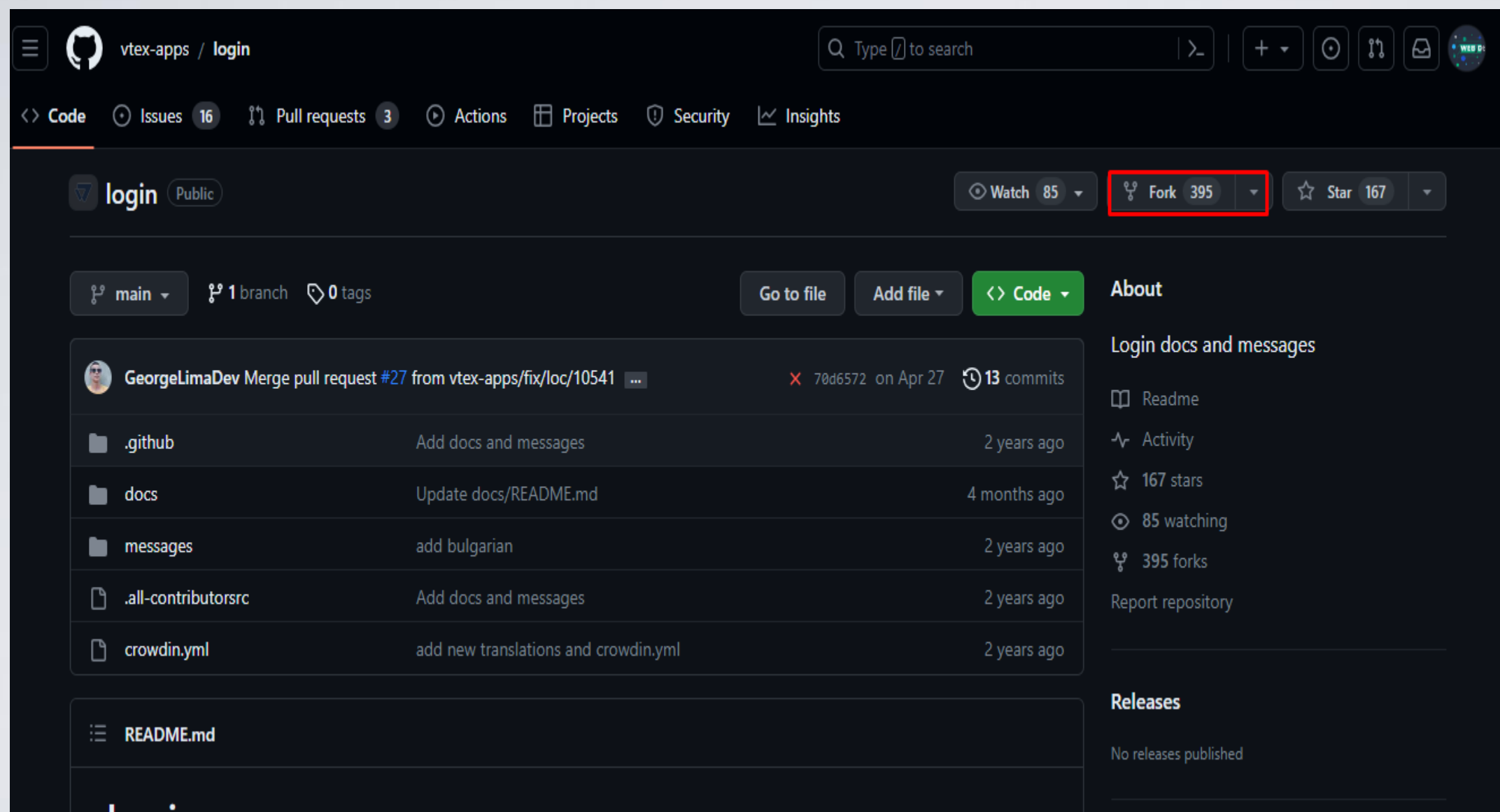


تتيح منصة GitHub لمستخدميها اشتقاق مستودعات جديدة خاصة أو عامة من مستودعات المستخدمين الآخرين العامة، وذلك بواسطة الأمر fork الذي يسمح لك بالبدء بمستودع جديد بناءً على ما تم بالفعل في مستودع مشروع آخر سواء كان المشروع الآخر مملوكاً لك أو عاماً. على أن fork نفسها ليست أحد الأوامر في نظام git، بل هي خاصية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

- انشاء الفروع في GitHub 
- مخطط عمل GitHub 
- خدمة GitHub Pages 
- الاشتقاق في GitHub 
- استنساخ المستودعات في GitHub 



# اساسيات Git



انشاء الفروع في GitHub

مخطط عمل GitHub

خدمة GitHub Pages

الاشتقاق في GitHub

استنساخ المستودعات في GitHub

GitHub