

Learn **Git** and **GitHub**

[Learn Git and GitHub](#) ملخص كورس



A lot of thanks to [Osama Elzero](#)

Created By: [Fady Alamir](#)



#01 – Whats Git & GitHub & Why U Should Learn Part 1

1. What Is Git and GitHub?

1- GIT Is Distributed Version Control System

حيث أن الـ Projects تتطور ويوجد منها versions لذا الـ GIT يستخدم لكي يـ Handle هذه الإصدارات أي يتحكم بها

2- Git Is Free and Open Source

الـ git أداة Free و Open Source Project حيث يمكن مشاركته مع أدوات أخرى دون مشاكل

3- GitHub Is Source for Project and Sources [GitLab, BitBucket]

الـ GitHub يوجد به Source جميع الـ Projects ومن الممكن المشاركة به ووضع تعديلات، والـ GitHub يقدموا نفس الخدمة التي يقدمها الـ BitBucket والـ GitLab

4- GitHub Simplify Using Git

الـ GitHub يقدم تجربة استخدام بسيطة وسهلة حيث يوفر بعد الإمكانيات الجيدة مثل الـ Issue Tracker والـ Bug Tracker وهذا المميزات لا يمكن استخدامها عند استخدام الـ git بمفرده فقط

5- You Can Use Git Without GitHub

يمكن استخدام الـ git بدون استخدام الـ GitHub حيث يمكن رفع الشغل على سيرفر آخر خاص بنا يسمى Centralite Server ويعمل عليه التيم بالكامل عن طريق الـ git

6- Git Has GUI

الـ git له واجهة يمكن استخدامها بدلاً من استخدام الـ Terminal والـ Command Line



ملحوظة: احتراف الـ Command Line مهم حيث يساعدك على دخول عالم الـ Unit System مثل الـ Mac والـ Linux حيث الشغل الـ Advanced يتم عمله باستخدام الـ Command Line

2. Why You Must Learn Git?

في الماضي كان لا يمكن أكثر من شخص أن يعدلوا على نفس الملف في نفس الوقت وعند حدوث error لا نستطيع أن نحدد من الذي قام بتعديل الكود ووقت التعديل وكيفية الرجوع لما كان قبل التعديل

وعند التعديل على الـ Header والـ Footer للصفحة الرئيسية لموقع ويوجد صفحات فرعية تستخدم نفس الـ Header والـ Footer كان يجب تعديل كل منهم على حدي ومن الممكن ان يكون تم التعديل على الـ Header والـ Footer في الصفحات الفرعية لذا كانت التعديلات تتداعل مع بعضها البعض ولكن لن يحدث هذا عند استخدام الـ git حيث:

1- Devs Contribute to The Same Project

جميع المطورين يشاركون في مشروع واحد فقط دون حدوث أي أخطاء

2- You Can Revert Changes

يمكن التراجع عن أي تعديلات ومعرفة من الذي قام بالتعديل حيث يوفر الـ git **Versions، Backup**

3- You Can Collaborate to Fix Issues

يمكن التعاون في تصليح أي Bug والتعديل على الملف في نفس الوقت

4- You Can Collaborate to Create New Features

يمكن التعاون في إنشاء ميزة جديدة بدون الحاجة لأن يكون لكل واحد من الـ Developers ملف للمشروع مختلف عن الآخر



5- You Can Solve Conflicts

يمكن باستخدام الـ git حل مشكلة الـ Conflict حيث يمكن مقارنة كود أحد المطورين بکود مطور آخر في نفس المشروع ومن ثم إنتاج النسخة السليمة من الكود

6- You Can Organize Features

من الممكن كتابة تعليق على كود لـ Feature جديدة أو تعليق على حل لـ Bug موجود في الـ System مما يسهل عملية التنظيم بين التيم القائم على المشروع ومعرفة كل ما هو جديد عن طريق التعليقات

#02 – What's Git & GitHub & Why U Should Learn Part 2

1. Words You Will Hear

1- Repository

المستودع الذي يقوم بوضع أكواد الـ Project كاملة بداخله وتخصر إلى كلمة Repo والتيم القائم على المشروع يرسل تعديلاته للـ Repo

2- Branch

القسم أو الفرع من الـ Project الذي يقوم بإضافة تعديل معين على الـ Project في هذا الفرع وعند انتهاء التعديل يقوم بإرساله

ملحوظة: تختلف آلية الـ Branch من شركة لشركة أخرى، ولكن هذا التعريف الأساسي للـ Branch

3- Local Repo

أي مستودع أكواد الـ Project الـ Local الخاص بالمبرمج على الـ PC الخاص به

4- Remote Repo

أي الشيء الـ Remotely الذي يقوم بالذهاب إليه المبرمج للتعديل على الـ Project مثل موقع GitHub أو Bitbucket أو GitLab أو سيرفر الشركة القائمة على الـ Project



5- Commit (Snapshot or Checkpoint In Your Local Repo)

ال Commit هي عندما تقوم بتعديل معين أو عند إضافة Feature أو حل Bug أو أي تعديل حيث يقوم ال Commit بعمل Checkpoint في ال Local Repo الخاص بك حيث يوجد مراحل أخرى حتى يصل لل Repository الأصلي

6- Clone [From Local Or Remote]

أي استنساخ ال Repository من ال Local Repo أو من ال Remote Repo

- حيث يمكن الاستنساخ من Repo آخر موجود في Location على ال PC وتجربة عليه تعديلات أخرى غير الموجودة على ال Repo الأول الموجود في PC آخر على ال Location أي يمكن عمل أكثر من Repo على ال Local
- وبعد الانتهاء من النسخة ال Local نقوم بأخذ النسخة النهائية ووضعها على ال Remote

7- Push [Upload Local Changes to Remote]

من خلاله نقوم برفع ما تم تعديله من ال Local Repo لل Remote Repo

8- Pull [You Pull Changes From Remote Repo To Your Local]

أي نقوم بسحب التعديلات من ال Remote Repo لل Local Repo أي تحميل المشروع بالتعديلات الجديدة الذي قام تيم المشروع بإضافتها على ال Remote Repository بالإضافة التعديلات التي نقوم بها على ال Repository النهائي

9- Pull Request [Tell Other About Your Changes To Pull It From Local To Remote]

حيث يوجد شخص Admin على ال Main Repo يقوم بمراجعة جميع تعديلات باقي مطوري ال Project، لذا عند انتهاء أحد المطوريين من تعديل معين يقوم بعمل Pull Request ثم يقوم ال Admin بمراجعةه والتأكد من التعديلات إذا كانت سليمة يقوم بقبول ال Pull Request ومن ثم يتم وضع التعديلات في ال Repo الأساسي وعند وجود خطأ في ال Code يقوم ال Admin برفض ال Pull Request ومن ثم يعود الرفض للمطور ليقوم بالتعديل عليه ويرسل Pull Request مرة أخرى



2. Important Notes

1- Create Repository For Every Project

عندما يكون لديك أكثر من Project يجب أن يكون لكل Project Repository الخاص به، لأنه من الناحية التنظيمية خطأ أن تقوم بوضع أكثر من Project في واحد فقط Repository

2- Create A New Branch For Every Feature or Enhancement

كل تعديل أو ميزة أو تحسين يجب أن نقوم بعمل Branch خاص بها وعند الانتهاء من التعديل نقوم بإرسال ال Branch ومن ثم نستطيع حذفه أو تركه

3- No Need To Connect To Remote Repo When Working

لا يوجد حاجة للاتصال بال Remote Repo عند العمل على تعديل أو إضافة جديد Project حيث تستطيع العمل على ال Local Repo وعند الانتهاء تقوم بعمل Pull لله Remote Repo بالتعديلات الجديدة

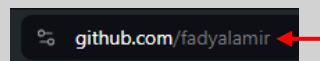
4- Anyone Can Push and Pull Depend On Permissions

كل مطور يستطيع أن يقوم بعمل Push أو Pull على حسب Remote Repository الصالحيات Permissions المقطعة له

#03 – Create Github Repository And Clone It

1- Make GitHub Account

ملاحظة: ال username يكون في رابط ال Profile هكذا



ملاحظة: عند فهم ال Concept والأوامر لن تحتاج للعمل على موقع معين سواء كان أو أي موقع حيث يمكن تطبيق نفس الشيء على GitHub أو BitBucket أو على Project الموجود في الشركة القائمة على ال Centralized Server



2- Make Repository

Screenshot 1: GitHub profile page showing the 'New' button highlighted.

Screenshot 2: 'Create a new repository' form. Labels include: 'Repo الاسم' (Repository name) pointing to 'Our_Project', 'الوصف Description' pointing to 'For Elzero Web School Git Course', 'Public' and 'Private' options with a note about Public being visible to anyone, 'Initialize this repository with:' section, and a 'Create repository' button. A note on the right says 'أي شخص يستطيع رؤيتها' (Any person can see it) for Public and 'مطوري المشروع فقط' (Only project developers) for Private.

Screenshot 3: GitHub repository page for 'fadyalamir/Our_Project'. Labels include: 'Username/Repo Name' pointing to the URL, 'Branch main هو الرئيسي' (main is the primary branch), 'ملف README.md صيغته Markdown' (README file is in Markdown format), 'تم كتابته بلغة تسمى Markdown' (Written in Markdown language), 'للتتعديل على ملف README' (To edit the README file), 'عند إنشاء Repo جديد 1 Commits سوف يوجد' (When creating a new repository, 1 commit will be available).

ملحوظة ٣: لغة الـ **Markdown** هي **Syntax** يتم كتابة الـ **Text** بها، ولكن بشكل منسق حيث يتم عمل الـ **Heading** أو الـ **List** أو خط أكبر من خط ويتم عرضها بشكل منسق على الـ **GitHub** والـ **Basic writing and formatting syntax** على هذا اللينك => [Link](#)

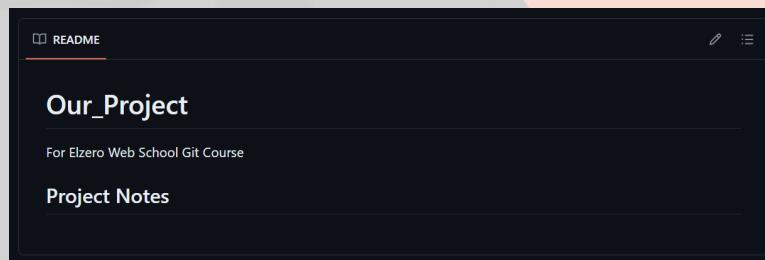


هكذا يكون تنسيق الـ Markdown

The screenshot illustrates the GitHub Markdown editor. On the left, the code editor shows a file named README.md with various Markdown elements like headings, lists, and images. On the right, the preview pane shows the formatted output. A large watermark for 'Elzero' is visible across the bottom of the image.

- عند التعديل على ملف الـ README

This screenshot shows the GitHub commit dialog after editing the README. The 'Commit changes...' button is highlighted with a red arrow, indicating the next step in the process.



3- Clone Repo From Remote Repo To Local Repo

This screenshot shows the GitHub repository page for 'Our_Project'. The 'Clone' button in the dropdown menu is highlighted with a red arrow, showing the URL for cloning the repository.



نقوم بعمل clone بواسطة الأمر `git clone` ثم رابط الـ **Repository** الذي قمت بنسخه
Cmder بواسطة GitHub

`git clone "Your Repository URL"`

```

C:\Program Files\cmder
λ cd /
C:\
λ cd Users\fadya\Downloads\
C:Users\fadya\Downloads
λ mkdir My-GitHub
C:Users\fadya\Downloads
λ cd My-GitHub\
C:Users\fadya\Downloads\My-GitHub
λ git clone https://github.com/fadylalamir/Our_Project.git
Cloning into 'Our_Project'...
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 0 (delta 0), pack-reused 0
Receiving objects: 100% (6/6), done.

C:Users\fadya\Downloads\My-GitHub
λ dir
Volume in drive C has no label.
Volume Serial Number is 5C5C-D0C8

Directory of C:\Users\fadya\Downloads\My-GitHub

03/31/2024 03:56 PM <DIR> .
03/31/2024 03:56 PM <DIR> ..
03/31/2024 03:56 PM <DIR> Our_Project
    0 File(s)      0 bytes
    3 Dir(s)  93,236,854,784 bytes free
C:Users\fadya\Downloads\My-GitHub
λ |
```



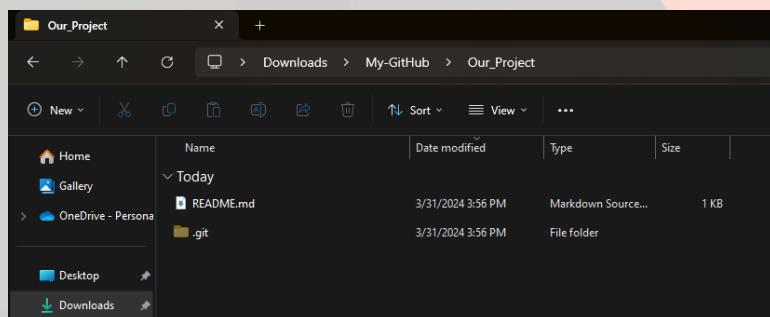
```

C:\Users\fadya\Downloads\My-GitHub
λ cd Our_Project\
C:\Users\fadya\Downloads\My-GitHub\Our_Project (main -> origin)
λ dir
Volume in drive C has no label.
Volume Serial Number is 5C5C-D0C8

Directory of C:\Users\fadya\Downloads\My-GitHub\Our_Project

03/31/2024 03:56 PM <DIR> .
03/31/2024 03:56 PM <DIR> ..
03/31/2024 03:56 PM           69 README.md
    1 File(s)      69 bytes
    2 Dir(s)  93,239,300,096 bytes free
C:\Users\fadya\Downloads\My-GitHub\Our_Project (main -> origin)
λ |
```

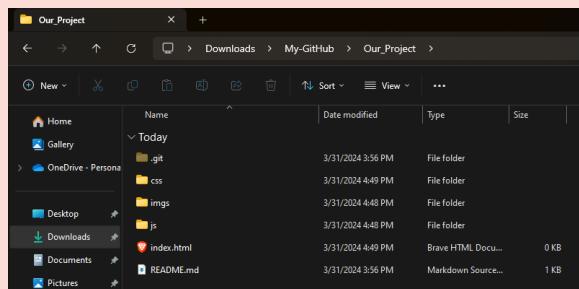
ومن ثم نستطيع تصفحه من المسار الذي قمنا بعمل clone إليه





#04 – Add And Reset And Commit & Explain Progress

بعد عمل Repo clone على جهاز الـ PC وعمل المشروع ووضعه عليه هكذا



وعند الانتهاء من المشروع ونريد ارساله للـ Remote Repo نقوم بالتالي

- Important Commands

Add Files

```
git add <File> <File>
git add *.extension
git add *
```

Show Status

```
git status
```

Commit Changes

```
git commit -m "Commit Message Here"
```

١ - **git status** : أمر يعرض حالة الـ **working directory** أي ما الموجود او ناقص او ماذا يحدث في مسار العمل الـ **working directory**

```
C:\Users\fadya\Downloads\My-GitHub\Our_Project (main -> origin)
└─ git status
On branch main
Your branch is up to date with 'origin/main'.

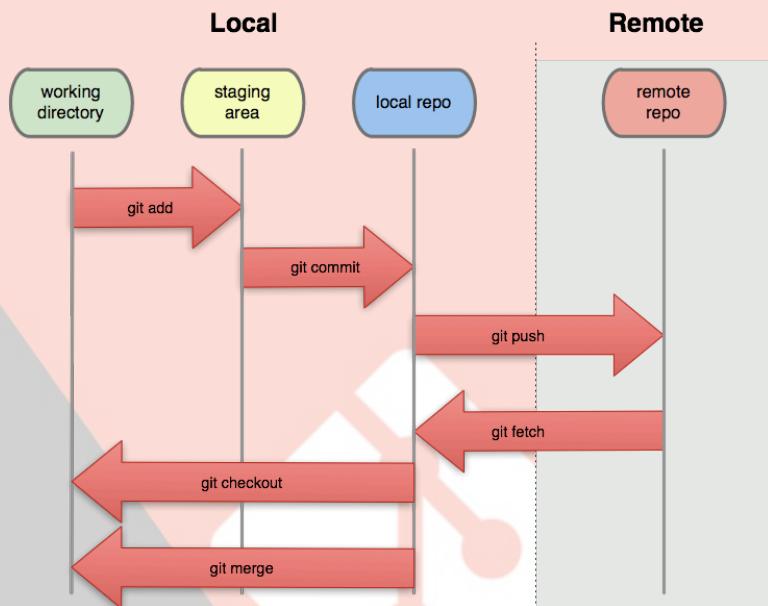
Untracked files:
  (use "git add <file>..." to include in what will be committed)
    css/
      index.html

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

C:\Users\fadya\Downloads\My-GitHub\Our_Project (main -> origin)
└─ |
```



في الصورة السابقة يوجد ملفين أي غير معرفين لذا يجب اضافتهم Remote Repo حتى نستطيع عمل commit لهم ورفعهم على الـ staging area للـ Remote Repo



حيث الـ **working directory** هو المسار الذي نقوم بالعمل فيه على الـ Task المطلوب وبناء المشروع

ثم إضافة المشروع بأمر الـ "git add" للـ **staging area**
والـ **staging area** هي منطقة التجهيز وبعد الـ **commit** تقوم بعمل
بواسطة أمر "git commit" لإضافة المشروع في الـ **Local Repo** الخاص بك
وبعد الانتهاء من الخطوات السابقة نقوم بعمل **push** للمشروع على الـ **remote repo**

ملحوظة: الـ **local repo** والـ **working directory** ليس نفس الشيء حيث أن
الـ **working directory** هو الفolder الذي تقوم بالعمل بداخله على المشروع ثم نقوم
 بإضافة المشروع للـ **staging area** ومنها للـ **local repo**

ملحوظة ٢: الـ **local repo** والـ **staging area** والـ **working directory** الثلاث
خطوات معناهم انك مازلت على الجهاز الخاص بك أي الـ **Local Repo**



staging area لـ untracked الملف git add - ٢

من الممكن كتابة اسم الملفات الذي نريد اضافتها هكذا

```
C:\Users\fadya\Downloads\My-GitHub\Our_Project (main -> origin)
λ git status
On branch main
Your branch is up to date with 'origin/main'.

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

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

C:\Users\fadya\Downloads\My-GitHub\Our_Project (main -> origin)
λ git add css\main.css index.html ←

C:\Users\fadya\Downloads\My-GitHub\Our_Project (main -> origin)
λ |
```

أو باستخدام الـ wild card هكذا

```
C:\Users\fadya\Downloads\My-GitHub\Our_Project (main -> origin)
λ git add * ←
```

وعند كتابة أمر git status مرة أخرى سوف يظهر الملفات الجديدة التي تم اضافتها

```
C:\Users\fadya\Downloads\My-GitHub\Our_Project (main -> origin)
λ git status
On branch main
Your branch is up to date with 'origin/main'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    new file:   css/main.css
    new file:   index.html

C:\Users\fadya\Downloads\My-GitHub\Our_Project (main -> origin)
λ |
```

ملاحظة ٣: من الممكن عمل لملف عن طريق الامر unstaged

```
C:\Users\fadya\Downloads\My-GitHub\Our_Project (main -> origin)
λ git status ←
On branch main
Your branch is up to date with 'origin/main'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    new file:   css/main.css
    new file:   index.html

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

C:\Users\fadya\Downloads\My-GitHub\Our_Project (main -> origin)
λ git add Fady.docx ←

C:\Users\fadya\Downloads\My-GitHub\Our_Project (main -> origin)
λ git status ←
On branch main
Your branch is up to date with 'origin/main'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    new file:   Fady.docx
    new file:   css/main.css
    new file:   index.html
```

```
C:\Users\fadya\Downloads\My-GitHub\Our_Project (main -> origin)
λ git reset head Fady.docx ←
C:\Users\fadya\Downloads\My-GitHub\Our_Project (main -> origin)
λ git status ←
On branch main
Your branch is up to date with 'origin/main'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    new file:   css/main.css
    new file:   index.html

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

C:\Users\fadya\Downloads\My-GitHub\Our_Project (main -> origin)
λ |
```



ومن ثم الملفين التي تم اضافتهم يجب عمل commit لهم لإرسالهم للـ local repo

```
C:\Users\fadya\Downloads\My-GitHub\Our_Project (main -> origin)
λ git commit -m "Created The main Project Structure" ←
[main 9aed97b] Created The main Project Structure
 3 files changed, 0 insertions(+), 0 deletions(-)
  create mode 100644 Fady.docx
  create mode 100644 css/main.css
  create mode 100644 index.html
```

local repo للـ staging area من الـ git commit -m

Syntax: git commit -m "Commit Message Here"

بعد الـ m - نترك رسالة داخل الـ double quotes نصف بها ما الذي تم اضافته للمشروع

```
C:\Users\fadya\Downloads\My-GitHub\Our_Project (main -> origin)
λ git commit -m "Created The main Project Structure"
```

بعدما قمنا بعمل commit إذن كل ملفات المشروع في local repo

```
C:\Users\fadya\Downloads\My-GitHub\Our_Project (main -> origin)
λ git status
On branch main
Your branch is ahead of 'origin/main' by 1 commit.
  (use "git push" to publish your local commits)

nothing to commit, working tree clean

C:\Users\fadya\Downloads\My-GitHub\Our_Project (main -> origin)
λ |
```

عندما نقوم بكتابة أمر git status كما في الصورة السابقة يخبرنا بوجود 1 commit ويجب استخدام الـ git push لإرسال المشروع للـ remote repo

ملاحظة: عندما ننسى أحدي التعديلات في المشروع من الممكن ان نقوم بعمل أكثر من commit هكذا

```
C:\Users\fadya\Downloads\My-GitHub\Our_Project (main -> origin)
λ git status ←
On branch main
Your branch is ahead of 'origin/main' by 1 commit.
  (use "git push" to publish your local commits)

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

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

C:\Users\fadya\Downloads\My-GitHub\Our_Project (main -> origin)
λ git add Elzero.xlsx ←

C:\Users\fadya\Downloads\My-GitHub\Our_Project (main -> origin)
λ git status ←
On branch main
Your branch is ahead of 'origin/main' by 1 commit.
  (use "git push" to publish your local commits)

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    new file:   Elzero.xlsx
```

```
C:\Users\fadya\Downloads\My-GitHub\Our_Project (main -> origin)
λ git commit -m "Added The Excel Sheet" ←
[main 3e3878d] Added The Excel Sheet
  1 file changed, 0 insertions(+), 0 deletions(-)
  create mode 100644 Elzero.xlsx

C:\Users\fadya\Downloads\My-GitHub\Our_Project (main -> origin)
λ git status ←
On branch main
Your branch is ahead of 'origin/main' by 2 commits.
  (use "git push" to publish your local commits)

nothing to commit, working tree clean

C:\Users\fadya\Downloads\My-GitHub\Our_Project (main -> origin)
λ |
```



إذن الان لدينا 2 commits ومتبقى ان نقوم بعمل git push لإرسالهم من remote repo لـ local repo

#05 – Push Local Changes To Remote Repository

- Important Commands

#Show Branches

`git branch`

`git remote -v`

#Push Changes

`git push origin main`



git

أمر نستخدمه لمعرفة جميع الـ branches الموجودة على local repo - ١

```
C:\Users\fadya\Downloads\My-GitHub\Our_Project (main -> origin)
λ git status
On branch main
Your branch is ahead of 'origin/main' by 2 commits.
  (use "git push" to publish your local commits)

nothing to commit, working tree clean

C:\Users\fadya\Downloads\My-GitHub\Our_Project (main -> origin)
λ git branch
* main ←

C:\Users\fadya\Downloads\My-GitHub\Our_Project (main -> origin)
λ |
```

وأولاً main branch يرسل لـ GitHub أي أن origin هو الـ remote repo

ـ ٢ أمر يستخدم لمعرفة اسم الـ origin الخاص بالـ remote repo

```
C:\Users\fadya\Downloads\My-GitHub\Our_Project (main -> origin)
λ git remote -v
origin https://github.com/fadyalamir/Our_Project.git (fetch)
origin https://github.com/fadyalamir/Our_Project.git (push)

C:\Users\fadya\Downloads\My-GitHub\Our_Project (main -> origin)
λ |
```



remote repo لـ local repo للمشروع من الـ Push أمر يستخدم لعمل git push -٣

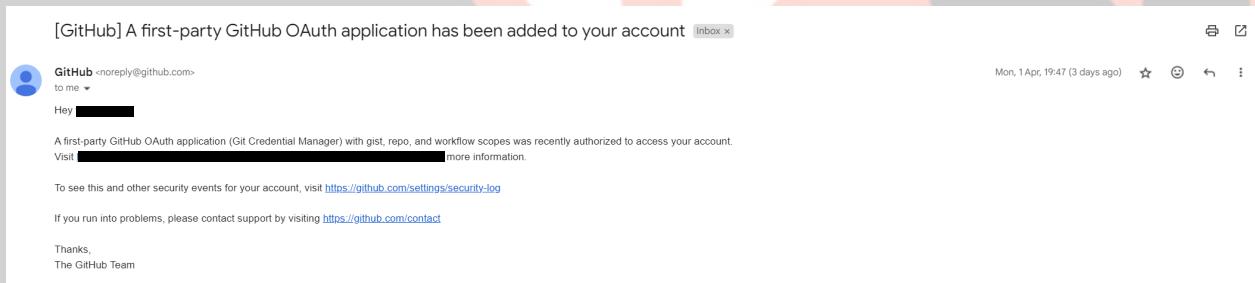
Syntax

```
get push RemoteName BranchName
```

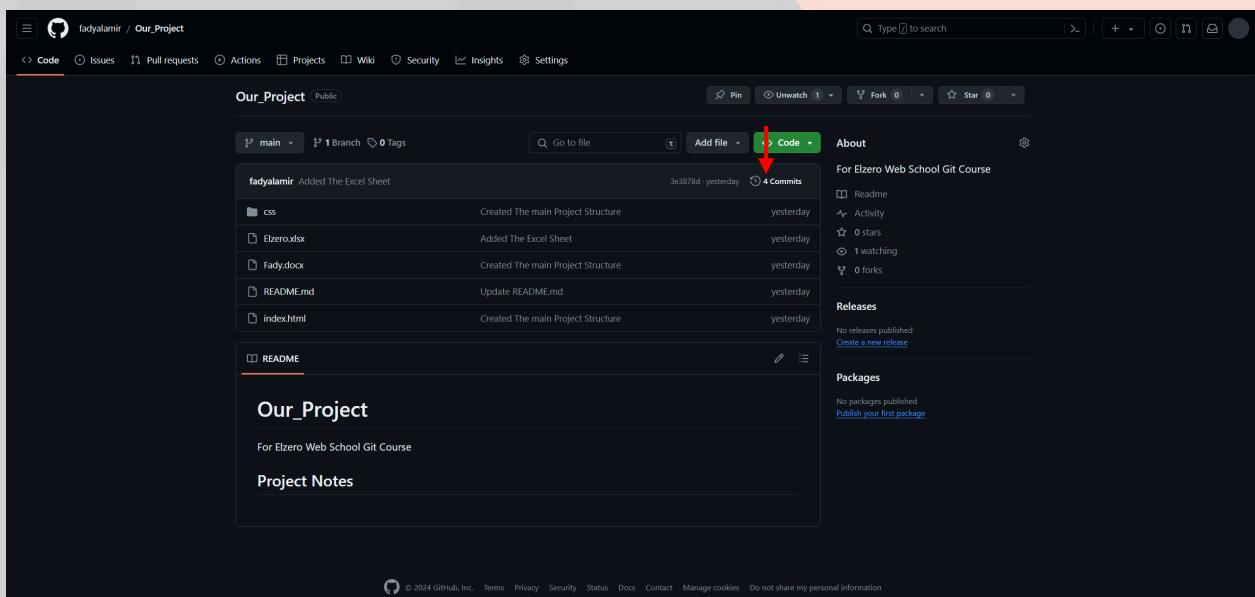
```
C:\Users\fadya\Downloads\My-GitHub\Our_Project (main -> origin)
\ git push origin main ←
Enumerating objects: 8, done.
Counting objects: 100% (8/8), done.
Delta compression using up to 12 threads
Compressing objects: 100% (5/5), done.
Writing objects: 100% (7/7), 6.15 KiB | 6.15 MiB/s, done.
Total 7 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), done.
To https://github.com/fadyalamir/Our_Project.git
 d01f960..3e3878d main -> main

C:\Users\fadya\Downloads\My-GitHub\Our_Project (main -> origin)
\ |
```

ملاحظة: عند العمل على سيرفر الشركة يكون لديه التراخيص ولا يطلب منك شيء ولكن بعد كتابة هذا الأمر يقوم الـ GitHub بطلب منك تسجيل الدخول للـ GitHub، عند كتابة بياناتك بشكل صحيح أثناء عمل Push للبيانات سوف يتم عمل Token لك على الموقع وسوف يتم إرسال رسالة إلى بريدك المسجل كما في الصورة



ومن ثم نذهب للـ GitHub أصبحت 4 بدلاً من 2 Commits





وعند الضغط على الـ Commits تظهر جميع التعديلات

Commits

main

Commits on Mar 31, 2024

Added The Excel Sheet
fadyalamir committed yesterday

Created The main Project Structure ← **commits**
fadyalamir committed yesterday

Update README.md
fadyalamir committed yesterday

Commits on Mar 30, 2024

Initial commit
fadyalamir committed 2 days ago

3e3878d · 9aed97b · d01f960 · 8a577fb

الرسالة التي قمنا بكتابتها عند عمل الـ commit
-m "Commit Message Here"

وعند الضغط على أحدي الـ commits تظهر جميع التعديلات والبيانات

Commit

Added The Excel Sheet

main

fadyalamir committed yesterday

Show 1 changed file with 0 additions and 0 deletions.

BIN +6.47 KB Elzero.xlsx

Binary file not shown.

0 comments on commit 3e3878d

Write Preview

Leave a comment

Markdown is supported Paste, drop, or click to add files

Comment on this commit

Unsubscribe You're receiving notifications because you're watching this repository.

ملاحظة ٢: عندما نقوم بالتعديل على أحد الملفات ونقوم بكتابة الامر git status سوف يخبرك بوجود ملف تم تعديله هكذا

```
C:\Users\fadya\Downloads\My-GitHub\Our_Project (main -> origin)
└─ git status
On branch main
Your branch is up to date with 'origin/main'.

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
    (use "git restore <file>..." to discard changes in working directory)
      → modified:   css/main.css

no changes added to commit (use "git add" and/or "git commit -a")

C:\Users\fadya\Downloads\My-GitHub\Our_Project (main -> origin)
└─ |
```



وبعدها لل staging area ونقوم بإضافة الملف داخل ال local repo

```
C:\Users\fadya\Downloads\My-GitHub\Our_Project (main -> origin)
\ git add css\main.css ←

C:\Users\fadya\Downloads\My-GitHub\Our_Project (main -> origin)
\ git status
On branch main
Your branch is up to date with 'origin/main'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    modified:   css/main.css

C:\Users\fadya\Downloads\My-GitHub\Our_Project (main -> origin)
\ git commit -m "Modified The CSS File" ←
[main e6505a5] Modified The CSS File
  1 file changed, 3 insertions(+)

C:\Users\fadya\Downloads\My-GitHub\Our_Project (main -> origin)
\ |
```

ومن ثم نقوم بعمل push له لل remote repo

```
C:\Users\fadya\Downloads\My-GitHub\Our_Project (main -> origin)
\ git push origin main ←
Enumerating objects: 7, done.
Counting objects: 100% (7/7), done.
Delta compression using up to 12 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (4/4), 346 bytes | 346.00 KiB/s, done.
Total 4 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/fadyalamir/Our_Project.git
  3e3878d..e6505a5  main -> main

C:\Users\fadya\Downloads\My-GitHub\Our_Project (main -> origin)
\ |
```

وعند الذهاب لل commit نجد التعديلات التي قمنا بتعديلها كاملة

The screenshot shows a GitHub commit page for a repository named 'Our_Project'. The commit message is 'Modified The CSS File' and it was made by 'fadyalamir' 7 minutes ago. The commit has 1 parent, commit '3e3878d'. The commit details show a single change in the 'css/main.css' file:

```
diff --git a/css/main.css b/css/main.css
--- a/css/main.css
+++ b/css/main.css
@@ -0,0 +1,3 @@
1 + body {
2 +   font-size: 20px;
3 + }
```

A red arrow points to the third line of the diff, which is the addition of the 'font-size' rule. Below the diff, there is a comment input field with the placeholder 'Leave a comment'.



#06 – Pull Changes From Remote Repository

ملحوظة: عند إنشاء Repo جديد سوف يوجد commit 1 وهو الـ initial commit الابتدائي هكذا

A screenshot of the GitHub 'Commits' page for a repository named 'fadyalamir / Our-Project'. The main branch is 'main'. A single commit is listed under 'Initial commit' by 'fadyalamir' committed 3 minutes ago. The commit message is 'Initial commit'. The commit hash is '07ac737'. There are filters at the top for 'All users' and 'All time'.

١- سوف نقوم بعمل الخطوات التي قمنا بعملها من الـ command line ولكن من الـ GitHub من حساب آخر كفرد من أفراد المشروع

١. سوف نقوم بعمل file جديد هكذا

A screenshot of the GitHub repository 'Our-Project' page. The 'Code' tab is selected. In the top right, there is a dropdown menu with 'Add file' highlighted with a red arrow. Below it are options '+ Create new file' and 'Upload files'. The repository has 1 branch and 0 tags. The README file shows 'Initial commit' and was created 5 hours ago. The repository summary indicates 0 stars, 0 forks, and no releases published.

٢. إضافة اسم الملف ومحتهواه

A screenshot of the GitHub code editor for the 'Fady.txt' file in the 'Our-Project' repository. A red arrow points to a warning message: 'You're making changes in a project you don't have write access to. Submitting a change will write it to a new branch in your fork markalamir/Our-Project, so you can send a pull request.' Below the editor, a note in Arabic states: 'معندها أنك تقوم بإنشاء ملف في Project ليس لديك أي صلاحيّة وصول إليه لذا قام بأخذ فرع منه في حسابك تستطيع التعديل عليه كما تشاء' (It means that you are creating a file in the Project which you do not have write access to. Submitting a change will write it to a new branch in your fork markalamir/Our-Project, so you can send a pull request.).



٣. عند الذهاب لـ repositories الخاصة بالحساب الآخر سوف نجده

The screenshot shows a GitHub user profile for 'markalamir'. In the 'Repositories' tab, there is a listing for 'Our-Project' which is described as 'Forked from fadyalamir/Our-Project'. The repository has a green and white logo. Below the repository listing, there is a section for the user 'markalamir' with an 'Edit profile' button.

٤. عند الذهاب للحساب الأساسي الذي قام بإنشاء Repo سوف نجد أن شخص قام بعمل Fork لـ Project على الحساب الخاص به هكذا

The screenshot shows a GitHub repository page for 'fadyalamir/Our-Project'. The repository has 1 branch and 1 commit. The 'Fork' button in the top right corner is highlighted with a red arrow. The repository description is 'This is Our Project'.

٥. نقوم بحذف الـ Fork الـ Repo ومن ثم نكون من افراد المشروع الـ Repo الأساسي

The screenshot shows a browser window with a dark theme. The address bar shows 'github.com/markalamir/Our-Project/settings'. A large red arrow points to the URL bar. Another red arrow points to a prominent red button labeled 'Delete this repository' at the bottom of the dialog.

نذهب للحساب الآخر وندخل على الـ Repo الـ Fork
ومن ثم لا نذهب في الأعلى ونذهب للأفل لأسفل لحذف
Delete this repository من زر Repo الـ



٦. نجعل الشخص الآخر واحد من فريق المشروع وذلك عن طريق الشخص الذي قام بعمل ال Repo الخاص بالمشروع هكذا

The screenshot shows the GitHub repository settings page for 'fadyalamir / Our-Project'. The 'Access' section is open, specifically the 'Collaborators' tab. A red arrow points to the 'Add people' button in the 'Manage access' section, which contains the message 'You haven't invited any collaborators yet'.

ومن ثم تذهب دعوة للشخص الآخر على ايميل ال Gmail هكذا

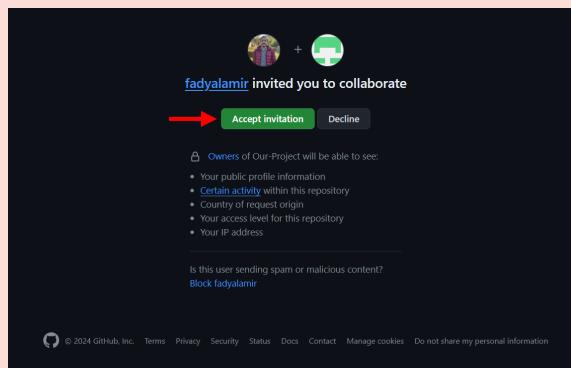
The screenshot shows an incoming Gmail email from 'fadyalamir <no-reply@github.com>' with the subject 'fadyalamir invited you to fadyalamir/Our-Project'. The email body contains the invitation message: '@fadyalamir has invited you to collaborate on the fadyalamir/Our-Project repository'. It includes a 'View invitation' button and a note about accepting or declining the invitation.

وعلي ال GitHub هكذا

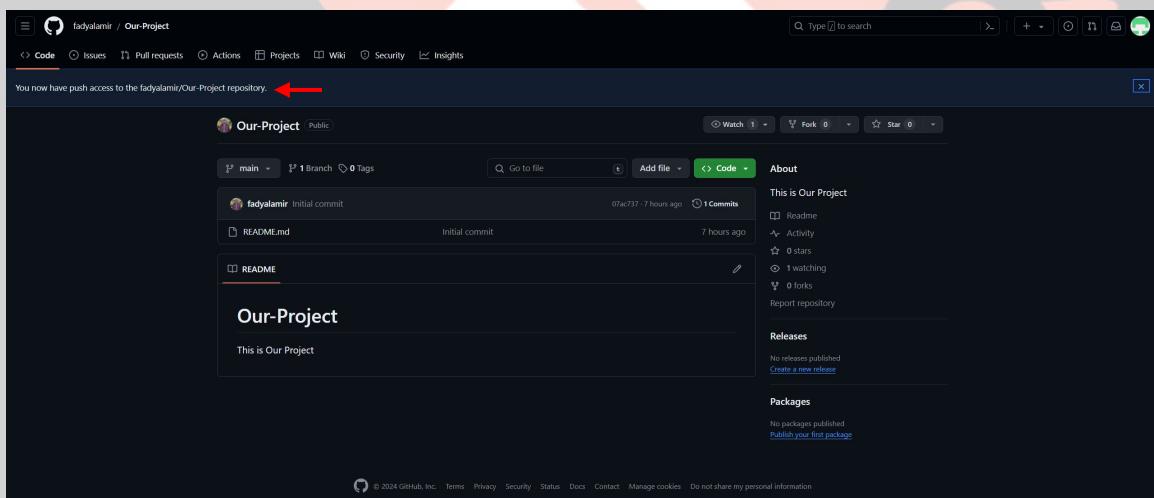
The screenshot shows the GitHub Notifications inbox. It displays an invitation email from 'fadyalamir' with the subject 'Invitation to join fadyalamir/Our-Project from fadyalamir'. The inbox also shows other notifications like 'Saved' and 'Done'.



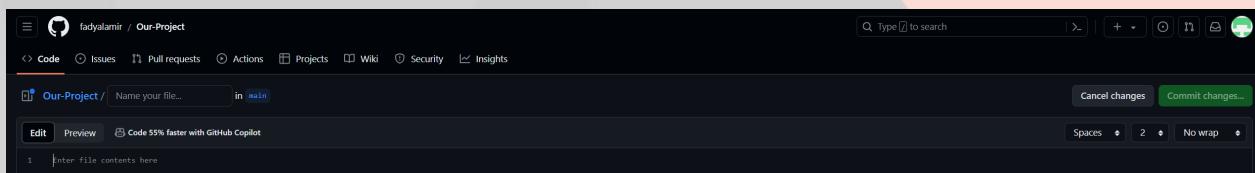
ثم نقبل الدعوة بعد الضغط على View invitation او الاشعار في GitHub هكذا



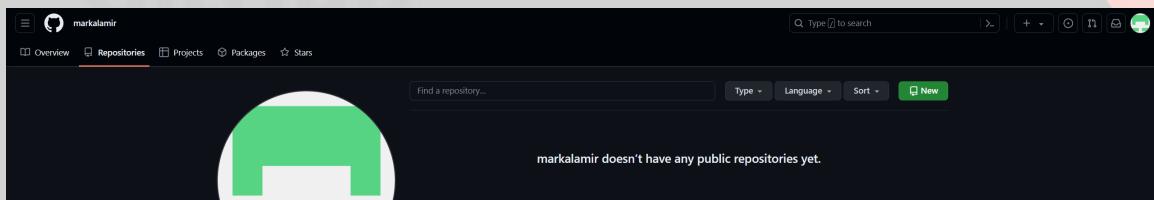
ومن ثم الحساب الآخر يكون له الصلاحيه لكي ي Push في Repo المشروع الأساسي هكذا



ملحوظة ٢: في الحساب الآخر عندما يذهب لل Repo الأن ويقوم بإضافة ملف لن تظهر الرسالة بأنه Fork لأنه أصبح من فريق المشروع قادر على التعديل في ال Repo

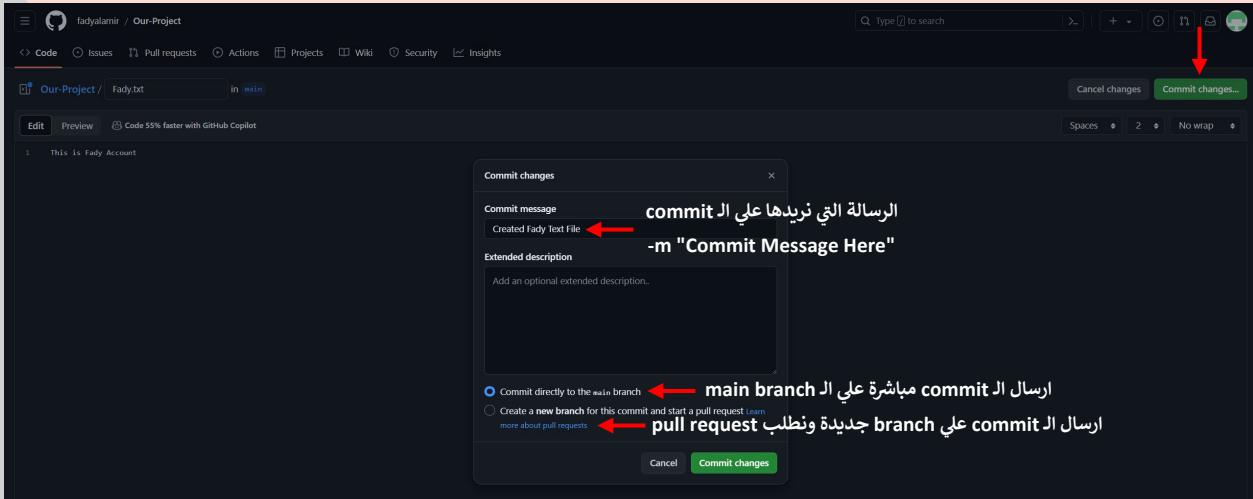


وعند الذهاب لل Repositories لن يكون من ضمنهم لأنه أصبح عضو في ال Repo الأساسي

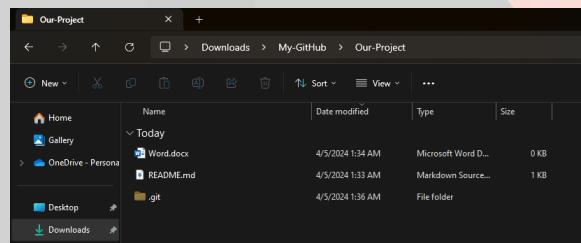
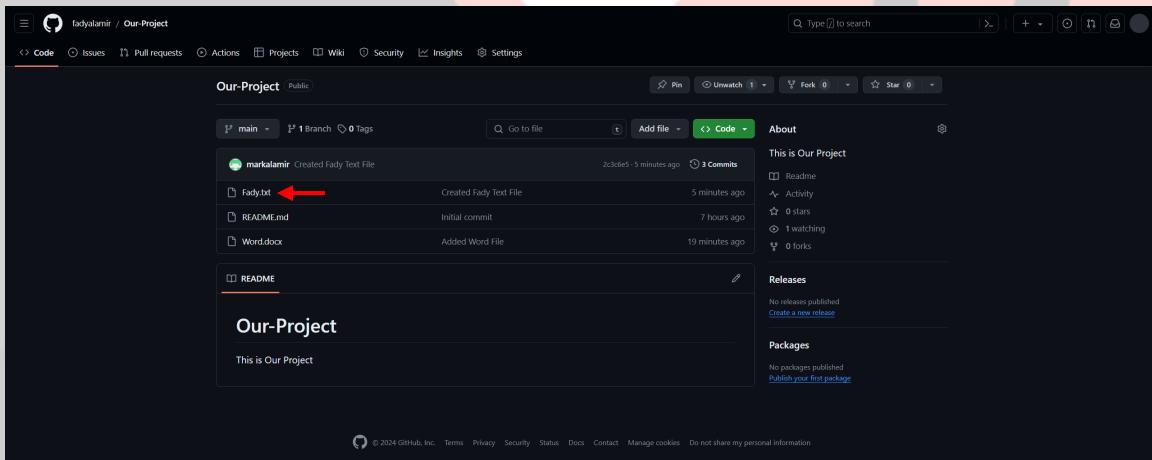




٧. عمل commit من الـ GitHub من الحساب الآخر بعدما أصبح من فريق المشروع في الـ Repo الأساسي هكذا



ملاحظة ٣: عندما يقوم الشخص الآخر بعمل commit من موقع الـ GitHub سوف تظهر التعديلات على الـ remote repo ولكن لن تظهر على الـ local repo هكذا





٨. عندما يقوم شخص من فريق المشروع بتحميل التعديلات الجديدة من ال local repo لل remote repo أمر git pull origin أي سحب التعديلات من ال origin branch وهي ال master هكذا

```
C:\Users\fadya\Downloads\My-GitHub\Our-Project (main -> origin)
\ git status ← remote repo
On branch main
Your branch is up to date with 'origin/main'.

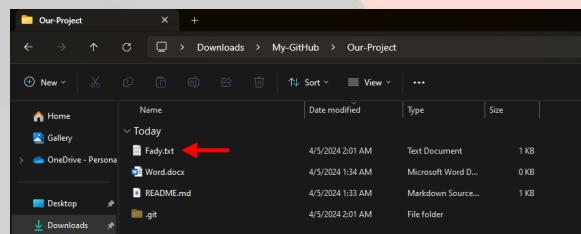
nothing to commit, working tree clean

C:\Users\fadya\Downloads\My-GitHub\Our-Project (main -> origin)
\ git pull origin ←
remote: Enumerating objects: 4, done.
remote: Counting objects: 100% (4/4), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), 984 bytes | 123.00 KiB/s, done.
From https://github.com/fadyalamir/Our-Project
  5e04333..2c3c6e5  main      -> origin/main
Updating 5e04333..2c3c6e5
Fast-forward
 Fady.txt | 1 +
 1 file changed, 1 insertion(+)
create mode 100644 Fady.txt

C:\Users\fadya\Downloads\My-GitHub\Our-Project (main -> origin)
\ |
```

حيث ال git pull يقوم بأمرين وهم git fetch يُحضر التعديلات الجديدة الموجودة في ال remote repo وأمر git merge يدمج ال remote repo مع ال local repo الخاص بك

ومن ثم نجد التعديلات الجديدة في ال local repo هكذا



#07 – Everything About Git Configurations

١ - Git Configurations: كل برنامج يكون له Configuration واعدادات معينة وبيانات خاصة بالشخص تضاف في البرنامج لكي تستعملها مستقبلاً وهكذا في ال Git نضيف الاسم والايدي الذي نحتاجهم عندما نريد عمل commit حيث يتطلب هذه البيانات لهذا سوف نثبتها في ال Configuration

ومن الممكن تغيير هذه البيانات والاعدادات على مستوى ال repo الذي نعمل عليه أو في أي repo سوف نعمل عليه Globally



- Important Commands

```
# Show All Configurations
```

```
git help config
```

```
# Show User Email
```

```
git config --global user.email
```

```
# Change User Email
```

```
git config --global user.email "Your Github Email Here"
```

```
# Edit Configuration
```

```
git config --global --edit
```

```
# Create New File
```

```
touch "File Name Here"
```

٢ - ١: **git config**: أمر نستخدمه لرؤية جميع الConfigs الموجودة في الـ Git المهمة الذي يريد منا تعديلها

```
C:\Users\fadya\Downloads\My-GitHub\Our-Project (main -> origin)
  \ git config -l
diff.astextplain.textconv=astextplain
filter.lfs.clean=git-lfs clean -- %f
filter.lfs.smudge=git-lfs smudge -- %F
filter.lfs.process=git-lfs filter-process
filter.lfs.required=true
http.sslbackend=openssl
http.sslcainfo=C:/Program Files/Git/mingw64/etc/ssl/certs/ca-bundle.crt
core.autocrlf=true
core.fscache=true
core.symlinks=false
pull.rebase=false
credential.helper=manager
credential=https://dev.azure.com/usehttppath=true
init.defaultbranch=master
core.repositoryformatversion=0
core.filemode=false
core.bare=false
core.logallrefupdates=true
core.symlinks=false
core.ignorecase=true
remote.origin.url=https://github.com/fadyalamir/Our-Project.git
remote.origin.fetch=+refs/heads/*:refs/remotes/origin/*
branch.main.remote=origin
branch.main.merge=refs/heads/main

C:\Users\fadya\Downloads\My-GitHub\Our-Project (main -> origin)
  \ |
```



٣- أمر نستخدمه لرؤية الـ **Manual** الخاص بالـ Git الموجود بداخله configurations وكل ما يخص الـ **Configs**

```
C:\Users\fadya\Downloads\My-GitHub\Our-Project (main -> origin)
λ git help config ←
```

C:\Users\fadya\Downloads\My-GitHub\Our-Project (main -> origin)
λ |

git-config(1) Manual Page

NAME
git config - set and get repository or global options

SYNOPSIS

```
git config [the option] [type-type] [fixed-value] [show-option] [c] [-m] [value] [value]
git config [the option] [type-type] [fixed-value] [add-option]
git config [the option] [type-type] [fixed-value] [replace-all-section] [value-pattern]
git config [the option] [type-type] [fixed-value] [replace-section] [value-pattern]
git config [the option] [type-type] [fixed-value] [check-value] [get-option]
git config [the option] [type-type] [fixed-value] [show-option] [c] [-m] [fixed-value] [get-option]
git config [the option] [type-type] [fixed-value] [show-option] [c] [-m] [fixed-value] [name-option]
git config [the option] [type-type] [fixed-value] [get-option] [name-option]
git config [the option] [fixed-value] [cancel-option] [value-pattern]
git config [the option] [fixed-value] [cancel-section] [value-pattern]
git config [the option] [fixed-value] [remove-section] [name-section]
git config [the option] [fixed-value] [remove-section] [name-section] [-r]
git config [the option] [get-option] [value-pattern]
git config [the option] [get-option] [value-pattern] [-r]
```

DESCRIPTION

This command sets and gets options with this command. the name is actually the section and the key separated by a dot, and the value will be accepted.

Multiple lines can be added to an option by using the `--add-option`. If you want to replace or cancel an option which can have multiple lines, a value pattern is used instead of regular expression, unless the `--fixed-value` option is given. If the option has a value pattern, it will be replaced by the new value. If the option has a fixed value, and the value pattern does not match the pattern, just ignore a single substitution made in that case due to `--EXAMPLE`, but note that this behavior is not guaranteed.

The `--type-type` option accepts `config` to ensure that incoming and outgoing values are correctly interpretable under the given type. If no `--type-type` is given, no cancellation will be performed. Values may unset an existing `--name-option`.

٤- طريقة التعديل على الـ configuration باستخدام الـ command line . ١. أمر نستخدمه لعرض الـ user email configuration الموجود في الـ configuration

```
C:\Users\fadya\Downloads\My-GitHub\Our-Project (main -> origin)
λ git config --global user.email ←
```

C:\Users\fadya\Downloads\My-GitHub\Our-Project (main -> origin)
λ |

لن يتم عرضه لأنه فارغ

لنقوم بوضع قيمة في الـ config تقوم بوضع القيمة بين double quotes

```
C:\Users\fadya\Downloads\My-GitHub\Our-Project (main -> origin)
λ git config --global user.email "my_email@gmail.com" ←
```

C:\Users\fadya\Downloads\My-GitHub\Our-Project (main -> origin)
λ git config --global user.email ←

C:\Users\fadya\Downloads\My-GitHub\Our-Project (main -> origin)
λ |



۲. الموجود في الـ configuration وتعديلاته : git config --global user.name

```
C:\Users\fadya\Downloads\My-GitHub\Our-Project (main -> origin)
\ git config --global user.name ←
C:\Users\fadya\Downloads\My-GitHub\Our-Project (main -> origin)
\ git config --global user.name "FadyAlamir" ←
C:\Users\fadya\Downloads\My-GitHub\Our-Project (main -> origin)
\ git config --global user.name ←
FadyAlamir
C:\Users\fadya\Downloads\My-GitHub\Our-Project (main -> origin)
\ |
```

وعند رؤية الـ config list نجد هم بعد التعديل

```
C:\Users\fadya\Downloads\My-GitHub\Our-Project (main -> origin)
\ git config -l ←
diff.astextplain.textconv=astextplain
filter.lfs.clean=git-lfs clean -- %f
filter.lfs.smudge=git-lfs smudge -- %f
filter.lfs.process=git-lfs filter-process
filter.lfs.required=true
http.sslbackend=openssl
http.ssicainfo=C:/Program Files/Git/mingw64/etc/ssl/certs/ca-bundle.crt
core.autocrlf=true
core.fscache=true
core.symlinks=false
pull.rebase=false
credential.helper=manager
credential.https://dev.azure.com.usehttppath=true
init.defaultbranch=master
user.email=          @gmail.com
user.name=FadyAlamir
core.repositoryformatversion=0
core.filemode=false
core.bare=false
core.logallrefupdates=true
core.symlinks=false
core.ignorecase=true
remote.origin.url=https://github.com/fadyalamir/Our-Project.git
remote.origin.fetch=+refs/heads/*:refs/remotes/origin/*
branch.main.remote=origin
branch.main.merge=refs/heads/main
C:\Users\fadya\Downloads\My-GitHub\Our-Project (main -> origin)
\ |
```

۳. اماكن وجود امر نستخدمه لعرض اماكن وجود configurations الـ : git config -l --show-origin

```
C:\Users\fadya\Downloads\My-GitHub\Our-Project (main -> origin)
\ git config -l --show-origin ←
file:C:/Program Files/Git/etc/gitconfig diff.astextplain.textconv=astextplain
file:C:/Program Files/Git/etc/gitconfig filter.lfs.clean=git-lfs clean -- %f
file:C:/Program Files/Git/etc/gitconfig filter.lfs.smudge=git-lfs smudge -- %f
file:C:/Program Files/Git/etc/gitconfig filter.lfs.process=git-lfs filter-process
file:C:/Program Files/Git/etc/gitconfig filter.lfs.required=true
file:C:/Program Files/Git/etc/openssl
file:C:/Program Files/Git/etc/ssicainfo=C:/Program Files/Git/mingw64/etc/ssl/certs/ca-bundle.crt
file:C:/Program Files/Git/etc/gitconfig core.autocrlf=true
file:C:/Program Files/Git/etc/gitconfig core.fscache=true
file:C:/Program Files/Git/etc/gitconfig core.symlinks=false
file:C:/Program Files/Git/etc/gitconfig pull.rebase=false
file:C:/Program Files/Git/etc/gitconfig credential.helper=manager
file:C:/Program Files/Git/etc/gitconfig credential.https://dev.azure.com.usehttppath=true
file:C:/Program Files/Git/etc/gitconfig init.defaultbranch=master
file:C:/Users/fadya/.gitconfig user.email=
file:C:/Users/fadya/.gitconfig user.name=FadyAlamir
file:.git/config core.repositoryformatversion=0
file:.git/config core.filemode=false
file:.git/config core.bare=false
file:.git/config core.logallrefupdates=true
file:.git/config core.symlinks=false
file:.git/config core.ignorecase=true
file:.git/config remote.origin.url=https://github.com/fadyalamir/Our-Project.git
file:.git/config remote.origin.fetch=+refs/heads/*:refs/remotes/origin/*
file:.git/config branch.main.remote=origin
file:.git/config branch.main.merge=refs/heads/main
C:\Users\fadya\Downloads\My-GitHub\Our-Project (main -> origin)
\ |
```



ملحوظة: من الممكن التعديل على config وإزالة قيمته بطريقتين

١. نقوم بترك القيمة بين الـ double quotes فارغة هكذا

```
C:\Users\fadya\Downloads\My-GitHub\Our-Project (main -> origin)
\ git config --global user.name "" ←

C:\Users\fadya\Downloads\My-GitHub\Our-Project (main -> origin)
\ git config --global user.name

C:\Users\fadya\Downloads\My-GitHub\Our-Project (main -> origin)
\
```

٢. باستخدام أمر git config --global --unset configName

```
C:\Users\fadya\Downloads\My-GitHub\Our-Project (main -> origin)
\ git config --global user.name "FadyAlamir"

C:\Users\fadya\Downloads\My-GitHub\Our-Project (main -> origin)
\ git config --global user.name
FadyAlamir

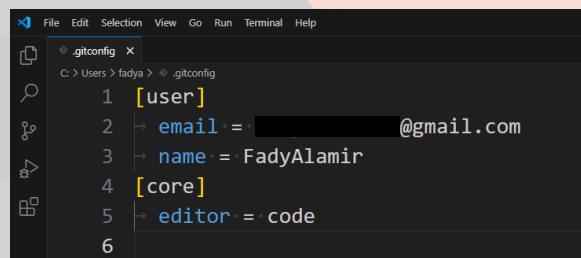
C:\Users\fadya\Downloads\My-GitHub\Our-Project (main -> origin)
\ git config --global --unset user.name ←

C:\Users\fadya\Downloads\My-GitHub\Our-Project (main -> origin)
\ git config --global user.name

C:\Users\fadya\Downloads\My-GitHub\Our-Project (main -> origin)
\
```

٥- التعديل على الـ configuration باستخدام الأمر editor باستخدام الأمر

git config --global --edit

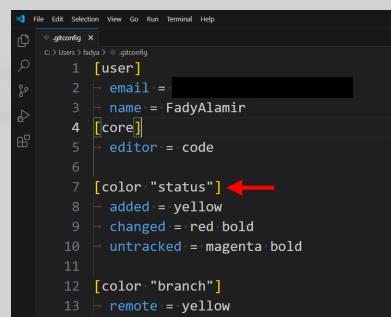


```
File Edit Selection View Go Run Terminal Help
.gitconfig
C:\Users\fadya> .gitconfig
1 [user]
2 → email = [REDACTED] @gmail.com
3 → name = FadyAlamir
4 [core]
5 → editor = code
6
```

ملحوظة٢: لتعيين الـ editor هو الـ vscode هو الـ editor الاساسي للتعديل نستخدم الامر

git config --global core.editor code

٦- التعديل على الوان ناتج الامر git status في الـ command line في الـ editor هكذا



```
File Edit Selection View Go Run Terminal Help
.gitconfig
C:\Users\fadya> .gitconfig
1 [user]
2 → email = [REDACTED]
3 → name = FadyAlamir
4 [core]
5 → editor = code
6
7 [color "status"] ←
8 → added = yellow
9 → changed = red bold
10 → untracked = magenta bold
11
12 [color "branch"]
13 → remote = yellow
```



```

λ git status
On branch master
Your branch is up to date with 'origin/master'.

Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

    new file:   cool.txt
    new file:   dsadasdasd

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

    new_file.txt

```

وعند حذف التعديلات التي قمنا بإضافتها في الـ editor تعود كما كانت هكذا

```

C:\Users\fadya\Downloads\My-GitHub\Our-Project (main -> origin)
λ git status
On branch main
Your branch is up to date with 'origin/main'.

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

nothing added to commit but untracked files present (use "git add" to track)
C:\Users\fadya\Downloads\My-GitHub\Our-Project (main -> origin)
λ |

```

: هو أمر يستخدم لعمل ملف جديد هكذا touch -v

```

C:\Users\fadya\Downloads\My-GitHub\Our-Project (main -> origin)
λ touch new_file.txt ←

C:\Users\fadya\Downloads\My-GitHub\Our-Project (main -> origin)
λ git status
On branch main
Your branch is up to date with 'origin/main'.

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

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

```

#08 – Generate And Test Github Public Key

1- هو ال Key أو Public Key أو SSH Secure Shell Key الذي ندخل بواسطته على GitHub ومن الممكن أن يكون لديك أكثر من واحد تدخل بواسطتهم من أماكن مختلفة ويسهل عليك عند عمل كل action لا تحتاج لكتابة ال username وال password

- الأمر الذي يقوم بعمل ال Shell Key هو

`ssh-keygen -t rsa -b 4096 -t "email"`

```

C:\Users\fadya\Downloads\My-GitHub\Our-Project (main -> origin)
λ ssh-keygen -t rsa -b 4096 -C "fadyalamir77@gmail.com" ←
Generating public/private rsa key pair.
Enter file in which to save the key (C:\Users\fadya\.ssh/id_rsa): ←
Created directory 'C:\Users\fadya\.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in C:\Users\fadya\.ssh/id_rsa
Your public key has been saved in C:\Users\fadya\.ssh/id_rsa.pub
The key fingerprint is:
SHA256:wcBAuQqjZXSS56i0ESBHTS+umfpv
The key's randomart image is:
+---[RSA 4096]----+
|+++.+=,           |
|.+++.o.          |
|o=o o o          |
|= . o .          |
|ooo S            |
| +.              |
|o+++.            |
|ob=B+=           |
|=E0=             |
+---[SHA256]-----+

```

مسار ال Public Key

Key لحماية ال Password

من الممكن كتابته او لا
حيث لا يظهر اثناء كتابته
ولانجوم حتى



١ . اختصار لـ ssh secure shell

- ٢ . Key Generator أي يقوم بعمل Key Generator لـ Key
- ٣ . -t اختصار لـ Algorithm نوع الـ Algorithm سوف يقوم بإنشاء الـ Key
- ٤ . الـ rsa هي الـ Algorithm حيث يوجد أكثر من نوع لـ Algorithm
- ٥ . b 4096 - تحدد عدد الـ Bits
- ٦ . وبعد إدخال الإيميل الخاص بك بين double quotes

ملاحظة: ليس اي شخص يستطيع الدخول على الـ Repo الخاص بك حيث يوجد مرحلة تسمى **Security Layer** حيث يوجد **Permission** معين من الـ Team يجب أن يتخطاه الشخص الذي يريد الدخول للـ Repo حيث يجب أن يكون واحد من الـ Team ولديه **access** للدخول على الـ Repo

٢ - أمر cat: هو أمر يُظهر محتوى أي File نريده

`cat ~/.ssh/id_rsa.pub`

```
C:\Users\fadya\Downloads\My-GitHub\Our-Project (main -> origin)
└ cat ~/.ssh/id_rsa.pub
ssh-rsa AAAAB3NzaC1yc2EAAAQABAAQDD27wVXNEkJG6G2QSKDt17WczkEUJt1DzBkHD5IyysW3qBqqJ1G1EDFzygF3q47UOfkWq1VLvGn+ExcVZXK1XLF/YoWNg
vv6BhmfrW3o7qiAdMV20YD/91NEXFwYYAZ6ePIGiJi6bcv0/ZYF8g0cuH8bN8LtoVknjuSYxIZTXDckfS7E+a200Cq7d2s9RPiY7hiqj16GpmhVYPIEzVrcsROM/OgyZu
Hd+Y94JNGazS5AGEy55eRuDPXbRHfp9BSs1ZCbrj3QURJcv2sJ+EcaP9sT9PSTCN1x7msu7ZqjRcbUP00hYGoXlnn9ckMLpkewJYMeH5ILCRrRDTKq0cyn6wo2EFHd9SVL
+SVR/3xpA7VJ8gU6SMEKQ7NzrkkmFgRNA+ITr1kduFxRpM/cz cqJYp8QGNWeMhXFZdyKfbXQAwvrQhf aq2+YyyfVY8WM8KKizqQqr7mrFkbGsqOAh6jSVscGjYWLddR8hudh
zqHj1WDiUof6dgoN9zbkcPNhizogv23NGBiRR1bxXmUICRcv3PuI9deTHE6nVEE3hJxuhQPE/K7B/Zm97d58FFoYAI1IouCOA72o8n8SBSSrG/1Y41/0XGomRBosLuIktcu
```

ثم نسخ الـ Key بالكامل والذهاب لـ GitHub ثم الـ Settings

The screenshot shows two parts of the GitHub interface. On the left, a dark sidebar menu is open with options like 'Your profile', 'Your repositories', and 'Settings' (which is highlighted with a red arrow). On the right, the 'Settings' page for the user 'Fady Alimir' is displayed. Under the 'Public profile' section, there is a list of settings including 'Account', 'Appearance', 'Accessibility', 'Notifications', 'Billing and plans', 'Emails', 'Password and authentication', 'Sessions', and 'SSH and GPG keys'. A red arrow points to the 'SSH and GPG keys' link, which is underlined.



ثم نقوم بإضافة الـ Key هكذا

SSH keys

New SSH key

This is a list of SSH keys associated with your account. Remove any keys that you do not recognize.

Add new SSH Key

Title: My Home PC

Key type: Authentication Key

Key:

```
ssh-rsa AAAAB3NzaC1yc2EAAAQAAQD2/wVXNEkIG6G2OKD1/WcrxEUH1Dz8KHDSlyyWiqBqgjTGIEdFzygf3qf/UDurkWaVtVnGr+ExXcVZKXLkYoWNgv8bmrfmWzQAdMVZOY/91NEXfwYA7sePIgj68v0/ZY8gPQCuHB8NBltvKnuJusYzXTDx5TE+eZoCqdz2sRPy7higjGpmhVPEwRz:SCRM:OgyZuhd+94NCgSz5ACeY+5eRupPxR8H4p:8585c1Zbr:3QUR,ICv31+IcaP919PSrCNw7msu7ZqgcRcup0h0YGoXin9sLMpkzWjM/HSLCR:RDtk0ynewo2EFHJ95VL+SVR/3xqA7V/bqU6SMKEQ7NjrkkmfRnRA+It1kDurRnRM/czqjY8QCNWeMhxZdrktBXQdWnrQhfqz2+YyfTYBWM8kKizqOgnr/mrFBG5qoAh6jS5scGjYWLddR8mhdhzqjzWDlUofsdgoN92hbcbPNhizOgvZ3N8BIRtBxmUICR
```

Add SSH key

٣- أمر يستخدم لعمل Test للـ Key واتصاله GitHub بالـ ssh -T git@github.com

```
C:\Users\fadya\Downloads\My-GitHub\Our-Project (main -> origin)
λ ssh -T git@github.com
The authenticity of host 'ED25519 key fingerprint' [REDACTED]
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'github.com' (ED25519) to the list of known hosts.
Enter passphrase for key 'C:/Users/fadya/.ssh/id_rsa':
Hi fadyalamir! You've successfully authenticated, but GitHub does not provide shell access.
```

أي انه متصل successfully authenticated

#09 – Create Repository From Existing Project

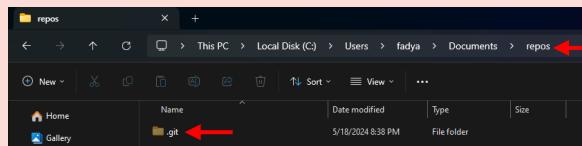
٤- هو أمر يستخدم لعمل تهيئه Repo فارغ في المكان الذي قمت بتحديده git init

```
C:\cmdr
λ cd ..
C:\ 
λ cd Users\fadya\Documents\
C:\Users\fadya\Documents
λ mkdir repos
C:\Users\fadya\Documents
λ cd repos\
C:\Users\fadya\Documents\repos
λ git init
Initialized empty Git repository in C:/Users/fadya/Documents/repos/.git/
C:\Users\fadya\Documents\repos (master) λ |
```

سوف تظهر الـ master للدلالة على وجود repository



وعند النظر على ال Folder الذي قمنا بإنشاء ال repo بداخله سوف نجد ملف ال git هكذا



ثم نقوم بإنشاء ملف داخل ال repo بإستخدام أمر touch ثم نقوم بإضافته في local repo ثم إضافته في ال commit ثم staging area

```
C:\Users\fadya\Documents\repos (master)
^ touch index.html ←

C:\Users\fadya\Documents\repos (master)
^ git status
On branch master

No commits yet

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

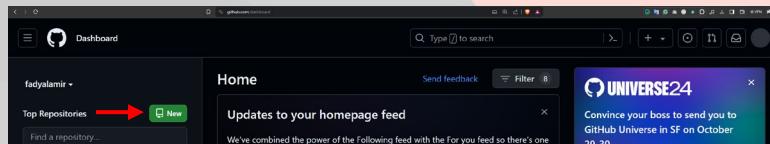
nothing added to commit but untracked files present (use "git add" to track)

C:\Users\fadya\Documents\repos (master)
^ git add index.html ←

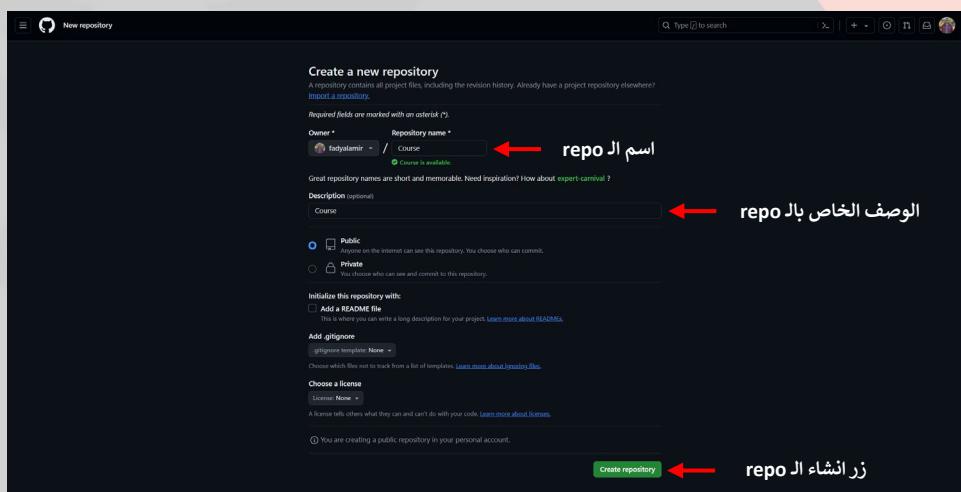
C:\Users\fadya\Documents\repos (master)
^ git commit -m "Created The Main File" ←
[master (root-commit) ce18244] Created The Main File
  1 file changed, 0 insertions(+), 0 deletions(-)
   create mode 100644 index.html

C:\Users\fadya\Documents\repos (master)
^ |
```

ثم نرفع ال repo على GitHub لإننا بدئنا ال repo على جهاز ال PC وليس على ال GitHub ومن ثم clone لجهاز ال GitHub



ومن ثم نحدد أسم ال repo وال description الخاص بال Create repository ونضغط على زر





ثم الاوامر الخاصة بعمل push من الـ local repo للـ remote repo

Quick setup — if you've done this kind of thing before

Set up GitHub Copilot
Use GitHub's AI pair programmer to autocomplete suggestions as you code.
Get started with GitHub Copilot

Add collaborators to this repository
Search for people using their GitHub username or email address.
Invite collaborators

SSH <git@github.com:fadyalamir/Course.git>

...or create a new repository on the command line

```
echo "# Course" >> README.md
git init
git add README.md
git commit -m "first commit"
git branch -M main
git remote add origin git@github.com:fadyalamir/Course.git
git push -u origin main
```

...or push an existing repository from the command line

```
git remote add origin git@github.com:fadyalamir/Course.git
git branch -M main
git push -u origin main
```

ProTip! Use the URL for this page when adding GitHub as a remote.

```
C:\Users\fadya\Documents\repos (master)
\ git remote add origin git@github.com:fadyalamir/Course.git

C:\Users\fadya\Documents\repos (master)
\ |
```

ثم أمر الـ push للـ local commits إلى الـ remote repo

```
C:\Users\fadya\Documents\repos (master)
\ git push -u origin master ←
Enter passphrase for key '/c/Users/fadya/.ssh/id_rsa':
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Writing objects: 100% (3/3), 225 bytes | 225.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To github.com:fadyalamir/Course.git
 * [new branch] master -> master
branch 'master' set up to track 'origin/master'.

C:\Users\fadya\Documents\repos (master -> origin)
\ |
```

ملحوظة: إذا كانت HTTPS نقم بتحويلها للـ SSH لأنه هو الذي يـ access public key للـ

ملحوظة ٢: الـ u- في أمر الـ git push تقوم بعمل pull ثم push لأن من الممكن عند العمل مع Team يجب سحب التعديلات التي قام بها باقي فريق التيم على تعديلاتك بواسطة الـ pull ثم وضع التعديلات جميعها معاً بالـ push



ثم تظهر التعديلات على الـ remote repo وهو الـ

الـ Email الذي قمنا بكتابته في الـ configuration

#10 – Learn Pull Request With Real Examples

Pull Request - 1: هو طلب بإضافة تعديلات جديدة يُرسل للـ Centralized Server ومن ثم الـ Admin المسؤول يرى إذا كانت هذه التعديلات مناسبة لكي يقوم بعمل Merge لها أم لا

حيث في الحياة العملية مثال الـ repo الخاص بالـ Angular Framework تقوم بعمل contribute وتشارك فيه حيث ليس من الطبيعي أن تكون من الـ Admins أي الـ Core Team كي تستطيع أن تقوم بعمل Push مباشرة ولكن بواسطة الـ Pull Request

لأن الـ Pull Request توفر بعض المراحل تساعد في تطوير الـ repo مثل أن يقوم أحد بمراجعة الكود الخاص بك وأن يقوم بعمل maintain للكود ويقوموا بالتعديل عليه قبل أن يقوموا بعمل Merge للكود الخاص بك

وعندما لن يكون لن access على الـ repo نقوم بأخذ fork منه



وبالتالي أصبح الـ repo في قائمة الـ repository في حسابك

The screenshot shows a GitHub repository page for 'Team_Admin'. At the top, there's a navigation bar with links for 'Code', 'Pull requests', 'Actions', 'Projects', 'Wiki', 'Security', 'Insights', and 'Settings'. Below the navigation bar, a red arrow points to the 'forked from fadyalamir/Team_Admin' link. The main content area displays the repository's README file, which contains the text 'Team_Admin' and 'Team_Admin'. On the right side, there are sections for 'About', 'Releases', and 'Packages'.

ثم نقوم بالتعديل على ملف الـ README لعمل Pull Request

The screenshot shows a GitHub commit dialog for the 'README.md' file in the 'Team_Admin' repository. The commit message is 'Testing Task'. In the bottom right corner of the dialog, there is a red arrow pointing to the 'Commit changes' button. The dialog also includes options for 'Commit message' and 'Extended description'.

سوف يكون التعديل في الـ repo الـ forked الخاص بي فقط ولن يكون في الـ repo الأساسي

The screenshot shows the GitHub repository page for 'Team_Admin' again. The README file now contains the text 'Team_Admin' and 'This Is Note'. A red arrow points to the word 'Note' in the README file. The rest of the repository interface is visible, including the sidebar and other repository details.



The screenshot shows a GitHub repository page for 'Team_Admin'. The top navigation bar includes links for Code, Issues, Pull requests, Actions, Projects, Wiki, Security, Insights, and Settings. The repository name 'Team_Admin' is displayed prominently, along with a public status badge. The main content area shows a file tree with 'main' as the active branch, containing 'README.md'. The README file content is displayed as 'Team_Admin'. A red arrow points to the word 'Team_Admin' in the README text. On the right side, there are sections for About, Team_Admin (with a Readme link), Activity (0 stars, 1 watching, 1 fork), Releases (no releases published, Create a new release), and Packages (no packages published, Publish your first package). The bottom of the page features a footer with copyright information for 2024 GitHub, Inc. and links for Terms, Privacy, Security, Status, Docs, Contact, Manage cookies, and a 'Do not share my personal information' checkbox.

لکی نقوم بعمل Pull Request ندخل علی قائمة الـ Pull Requests من الـ nav bar

The screenshot shows the GitHub pull request interface. At the top, there's a navigation bar with links for Code, Pull requests (which is the active tab), Actions, Projects, Wiki, Security, Insights, and Settings. Below the navigation is a search bar with placeholder text "Type / to search". To the right of the search bar are several icons: a magnifying glass, a plus sign, a circular arrow, a refresh symbol, and a gear icon. The main content area has a header "Welcome to pull requests!" with a sub-instruction: "Pull requests help you collaborate on code with other people. As pull requests are created, they'll appear here in a searchable and filterable list. To get started, you should [create a pull request](#)". At the bottom left, there's a "ProTip!" note: "Adding `notlabel` will show everything without a label." On the far right of the interface, there's a red arrow pointing to a green button labeled "New pull request".

عند الضغط على زر الـ **New pull request** يظهر هكذا

The screenshot shows a GitHub pull request page for a repository named 'fadyalamir / Team_Admin'. The pull request is titled 'Comparing changes' and is described as 'Able to merge'. It compares the 'base repository: fadyalamir/Team_Admin' with the 'head repository: markalami/Team_Admin' against the 'main' branch. The status bar at the top right indicates 'Able to merge'.

On the right side of the page, there is a green button labeled 'Create pull request'.

In the bottom right corner of the main content area, there is a green box containing the Arabic text 'التعديلات التي قمنا بها' (The changes we made).

Annotations with red arrows point to several key elements:

- An arrow points to the 'Able to merge' status bar at the top right.
- An arrow points to the 'Create pull request' button on the right.
- An arrow points to the green box containing the Arabic text 'التعديلات التي قمنا بها'.

لا توجد أي مشاكل حيث قمنا بعمل push اولاً على ال repo او fork الخاص بنا ثم سوف نقوم بعمل pull request على ال repo الاساسي



ثم عند الضغط على زر يظهر هكذا Create pull request

Open a pull request
Create a new pull request by comparing changes across two branches. If you need to, you can also compare across forks. Learn more about diff comparisons here.

Add a title
Local **Pull Request عنوان الـ Pull Request**

Add a description
Write Preview

Added Ar Lang

Markdown is supported Paste, drop, or click to add files

Allow edits by maintainers **Create pull request**

Remember, contributions to this repository should follow our [GitHub Community Guidelines](#).

Commits on May 22, 2024
Testing Task
markalamir committed 16 minutes ago
Showing 1 changed file with 2 additions and 0 deletions.

Local #1

markalamir wants to merge 1 commit into [fadyalamir/main](#) from [markalamir/main](#)

Conversation 0 → Commits 1 Checks 0 Files changed 1

markalamir commented now
Added Ar Lang

Testing Task Verified 96/99

This branch has no conflicts with the base branch Only those with write access to this repository can merge pull requests.

Add a comment Write Preview

Add your comment here... Markdown is supported Paste, drop, or click to add files

Close pull request Comment

Remember, contributions to this repository should follow our [GitHub Community Guidelines](#).

ProTip Add [patch](#) or [diff](#) to the end of URLs for Git's plaintext views.

ثم يقوم بعمل check للتأكد من عدم وجود أي اختلافات ومن ثم ارسال الـ Pull Request

الخاص بالـ repo الاساسي

Local #1

pull request from [fadyalamir/main](#)

Open

حالة الـ Pull Request هي **Open**

من الممكن عمل أكثر من commit قبل ارسال الـ Pull Request

Pull Request

conflicts التأكد من عدم وجود أي

commits

Only those with write access to this repository can merge pull requests.

Close pull request Comment

Remember, contributions to this repository should follow our [GitHub Community Guidelines](#).

ProTip Add [patch](#) or [diff](#) to the end of URLs for Git's plaintext views.



وعند الذهاب للحساب والrepo الاساسي نجد 1 Pull Request هكذا

The screenshot shows the GitHub interface for the repository 'fadyalamir / Team_Admin'. The 'Pull requests' tab is selected, showing one open pull request. A red arrow points to the 'Pull requests' tab.

The screenshot shows the details of the first pull request in the 'Team_Admin' repository. The title of the pull request is 'نقوم بالضغط عليها للدخول إلى الـ Local'. A red arrow points to the title. The pull request was opened by 'markalamir' 11 minutes ago.

The screenshot shows the merge process for the pull request. It highlights the 'Merge pull request' button, which is green and says 'Merge pull request'. A red arrow points to this button. The text 'لا يوجد أي conflict مع base branch' is displayed above the button.

عند التأكيد من الكود conflicts وعدم وجود أي conflict وأن الكود مفيد تقوم بعمل merge

The screenshot shows the confirmation dialog for merging the pull request. It asks 'Merge pull request #1 from markalamir/main'. The text 'من الممكن التعديل على الـ Merge بالاسم الذي نريده' is displayed above the 'Confirm merge' button. A red arrow points to the 'Confirm merge' button.



وعند الذهاب لل repo الـ forked سوف نجد ان الحالة تغيرت إلى Merged

A screenshot of a GitHub repository page. At the top, there's a navigation bar with links for Code, Issues, Pull requests, Actions, Projects, Security, and Insights. Below the navigation bar, a pull request card is displayed with the title "Local #1". A red arrow points to the status indicator "1 Merged" which is highlighted with a purple box. Below the card, a message says "fadyalamir merged 1 commit into fadyalamir:main from markalamir:main 1 minute ago".

وأن الـ Pull Request تم غلقه

A screenshot of a GitHub repository page. A modal window titled "Pull request closed" is open. It contains the message "If you wish, you can delete this fork of fadyalamir/Team_Admin in the [settings](#)".

للتعديل عليه فقط forked الأن لأننا نقوم بأخذ الـ forked حيث يمكن حذف الـ forked وارساله لذا من الممكن حذفه وعند الحاجة للتعديل مرة أخرى نقوم بعمل جديدة ونقوم بالتعديل عليها

وعند الذهاب لل repo الأساسي سوف نجد الـ Pull Request في الـ Closed

A screenshot of a GitHub repository page. The navigation bar shows "fadyalamir / Team_Admin". The "Pull requests" tab is selected. A search bar at the top has the query "is:pr is:closed". Below the search bar, a filter section has "1 Closed" checked, indicated by a red arrow. A message below the filters says "Now, GitHub will help potential first-time contributors [discover issues](#) labeled with [good first issue](#)". The main list shows one closed pull request: "f1 by markalamir was merged 6 minutes ago". A red arrow points to the "Closed" status in the list. A note at the bottom left says "Pull Request بارسال الـ repo الأساسي من قام".

ومن ثم في النهاية نجد التعديل الذي قمنا به عن طريق الـ Pull Request في الـ repo الأساسي هكذا

A screenshot of a GitHub repository page for "Team_Admin". The repository has 1 branch and 0 tags. A pull request from "fadyalamir" has been merged, indicated by the status "5d7e885 - 9 minutes ago". The commit count is shown as "3 Commits". A red arrow points to the merge commit. In the repository details, there is a note "This Is Note" with a red arrow pointing to it. On the right side, there's a sidebar with "Team Admin" information, "Releases" (none), "Contributors" (fadyalamir and markalamir), and a note "المساهمين في الـ repo".



وعند الذهاب لـ commits سوف نجد الـ Merge هكذا

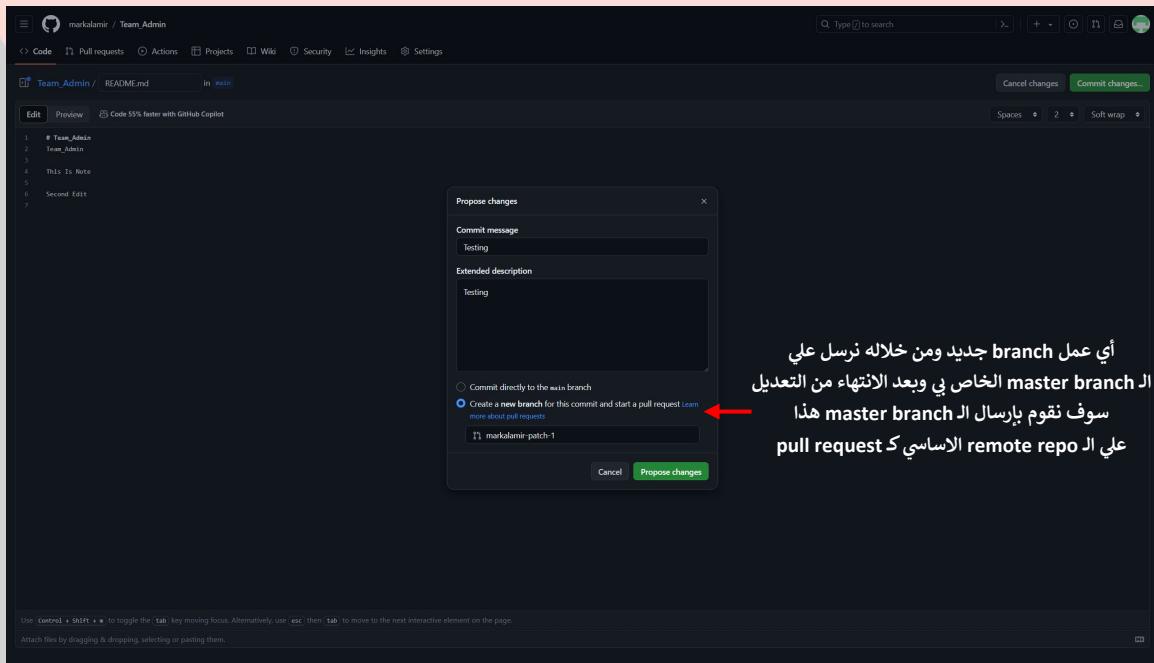
A screenshot of the GitHub 'Commits' page for the repository 'fadyalamir / Team_Admin'. The main branch is 'main'. A red arrow points to a merge commit titled 'Merge pull request #1 from markalamir/main'. The commit was made by 'fadyalamir' 11 minutes ago. Below it are two other commits: 'Testing Task' by 'markalamir' 49 minutes ago and 'Initial commit' by 'fadyalamir' 1 hour ago.

في Tab Insights يظهر من قام بعمل Pull Request أو أي حد قمت بإضافته في repo هكذا

A screenshot of the GitHub 'Insights' page for the repository 'fadyalamir / Team_Admin'. The date range is May 14, 2024 – May 21, 2024. An orange arrow points to the 'Overview' section, which shows '1 Active pull request'. Below it, a chart indicates 'Excluding merges, 2 authors have pushed 2 commits to main and 2 commits to all branches. On main, 0 files have changed and there have been 0 additions and 0 deletions.'



٢- عمل Fork آخر بعد حذف الأول ومن ثم اضافة تعديل آخر ولكن سوف نقوم بإختيار direct branch على الـ commit



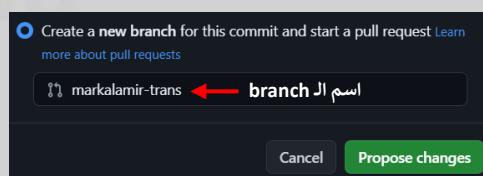
أي الفرق هنا أننا لن نقوم بالتعديل على الـ master مباشرةً وارسال الـ Pull Request على الـ master فرعياً ولكن سوف نقوم بعمل branch فرعياً

ملحوظة: نقوم بعمل الـ branch الفرعى هذا لأن عند العمل في شركة تستعمل الـ Git يقوموا بعمل branch لأى line of development أي تقوم بعمل feature معينة في وعند الانتهاء منها تقوم بإرسالها كـ Pull Request ومن ثم تقوم بحذف هذا branch الـ branch

ومثلاً قمنا بحذف الـ fork بالكامل

نقوم بحذف الـ branch بعد الانتهاء من التعديل وارسال الـ Pull Request وعمل له Merge

ملحوظة ٢: اسماء الـ branches تختلف من شركة لأخرى حيث كل شركة لها طريقة خاصة بها في تسمية الـ branches وبناءً أيضاً على الـ feature الذي سوف تقوم بإضافتها





Comparing changes

Choose two branches to see what's changed or to start a new pull request. If you need to, you can also compare across forks or learn more about diff comparisons.

branch من الـ main repo للـ fork الـ الخاص بك

base: main compare: markalamir-trans Able to merge. These branches can be automatically merged.

Create pull request

Pull Request

Discuss and review the changes in this comparison with others. Learn about pull requests.

1 commit 1 file changed 1 contributor

Commits on May 22, 2024

Testing

markalamir committed now

Showing 1 changed file with 2 additions and 0 deletions.

Split Unified

© 2024 GitHub, Inc. Terms Privacy Security Status Docs Contact Manage cookies Do not share my personal information

Testing #1

markalamir wants to merge 1 commit into main from markalamir-trans

لأحظ انك حقاً الأن داخل الـ branch والـ fork الـ main repo

ومن ثم تقوم بعمل conflicts ومن ثم يقوم بالتأكد بعدم وجود أي Pull Request

Merged

markalamir merged 1 commit into main from markalamir-trans now

Conversation 0 Commits 1 Checks 0 Files changed 1

markalamir commented 3 minutes ago Testing

markalamir merged commit cb474e8 into main now

Pull request successfully merged and closed تم الـ Merge بنجاح

Delete branch حذف الـ branch

Add a comment

Write Preview

Add your comment here...

Markdown is supported Paste, drop, or click to add files

Comment

Remember, contributions to this repository should follow our GitHub Community Guidelines.

ProTip! Add comments to specific lines under [Files changed](#).

© 2024 GitHub, Inc. Terms Privacy Security Status Docs Contact Manage cookies Do not share my personal information

وبعدها نقوم بحذف الـ branch والذهاب لـ New pull requests وعمل هكذا

Filters ispr isopen Labels 9 Milestones 0 New pull request

0 Open 1 Closed

Author Label Projects Milestones Reviews Assignees Sort

There aren't any open pull requests.



حيث أن الـ Pull Request هذا الذي سوف نقوم بإرساله للـ repo الأساسي

The screenshot shows the GitHub pull request creation interface. At the top, it says "Open a pull request" and "Create a new pull request by comparing changes across two branches. If you need to, you can also compare across forks. Learn more about diff comparisons here." Below this, there are dropdown menus for "base repository: fadyalamir/Team_Admin", "base: main", "head repository: markalamir/Team_Admin", and "compare: main". A green checkmark indicates "Able to merge. These branches can be automatically merged." There are fields for "Add a title" (with "Local" selected) and "Add a description". A note at the bottom says "أي السماح لـ Admins الـ repo الأساسي بالتعديل على الكود قبل الـ Merge member contributions to this repository should follow our GitHub Community Guidelines." A red arrow points to the "Allow edits by maintainers" checkbox. At the bottom right, there is a "Create pull request" button.

أي السماح لـ Admins الـ repoأساسي بالتعديل على الكود قبل الـ Merge

ثم إرسال الـ Pull Request وكما فعلنا في المرة الأولى نذهب للـ repo الأساسي نقوم بعمل الـ Confirm merge هكذا

The screenshot shows the GitHub pull request status page. It displays a message: "Local #2" followed by "I-Merged fadyalamir merged 2 commits into fadyalamir:main from markalamir:main now". A red arrow points to the "I-Merged" button.

وسوف نجد التعديلات في الـ repo الأساسي

The screenshot shows the GitHub repository page for "Team_Admin". It lists a commit by "fadyalamir" titled "Merge pull request #2 from markalamir/main" (d7bf948 - 1 minute ago). The commit message is "Testing". The repository has 1 branch and 0 tags. On the right, there is an "About" section for "Team_Admin" which includes a "README" file, activity, stargazing, and forking information. The "README" file content shows "Team_Admin", "This Is Note", and "Second Edit". A red arrow points to the "Second Edit" line in the README file.



#11 – Learn Everything About Aliases

1 - **Aliases**: هي الاسماء المستعارة حيث توجد في جميع الانظمة تستخدم لاختصار الـ commands

بإستخدام الأمر

`git config --global alias.st status`

```
C:\Users\fadya\Documents\repos (master -> origin)
λ git status ←
On branch master
Your branch is up to date with 'origin/master'.

nothing to commit, working tree clean

C:\Users\fadya\Documents\repos (master -> origin)
λ git config --global alias.st status ←

C:\Users\fadya\Documents\repos (master -> origin)
λ |
```

١. **--global** : حتى لا يكون علي مستوى الـ repo الخاص بك فقط

٢. **st** : هو الاختصار للأمر الذي يليه وهو الـ `status`

ويُفضل أن يكون الاختصار حرفين وليس حرف واحد حتى لا يُسبب conflicts مع اشياء أخرى

٢- بإستخدام الأمر `git config --global alias.st` يمكن عرض الاختصار الخاص بالـ `st` خاص بـ `alias`

```
C:\Users\fadya\Documents\repos (master -> origin)
λ git config --global alias.st
status
```

ويمكن استخدام الـ `command` المُختصر هكذا

```
C:\Users\fadya\Documents\repos (master -> origin)
λ git st
On branch master
Your branch is up to date with 'origin/master'.

nothing to commit, working tree clean

C:\Users\fadya\Documents\repos (master -> origin)
λ |
```



ملحوظة: عندما نريد اختصار command يتكون من كلمتين يجب وضعهم في double quotes

٣- اختصار لأمر الـ git branch

```
C:\Users\fadya\Documents\repos (master -> origin)
λ git branch ←
  main
* master

C:\Users\fadya\Documents\repos (master -> origin)
λ git config --global alias.br branch ←

C:\Users\fadya\Documents\repos (master -> origin)
λ git br ←
  main
* master

C:\Users\fadya\Documents\repos (master -> origin)
λ |
```

ملحوظة: الـ alias لا يقوم بوقف الـ command الاساسي ولكنه مجرد اسم مستعار مُختصر

```
C:\Users\fadya\Documents\repos (master -> origin)
λ git config --global alias.br branch ←

C:\Users\fadya\Documents\repos (master -> origin)
λ git br ←
  main
* master

C:\Users\fadya\Documents\repos (master -> origin)
λ git branch ←
  main
* master

C:\Users\fadya\Documents\repos (master -> origin)
λ |
```

٤- اختصار لأمر الـ commit

```
C:\Users\fadya\Documents\repos (master -> origin)
λ git config --global alias.cm "commit -m" ←

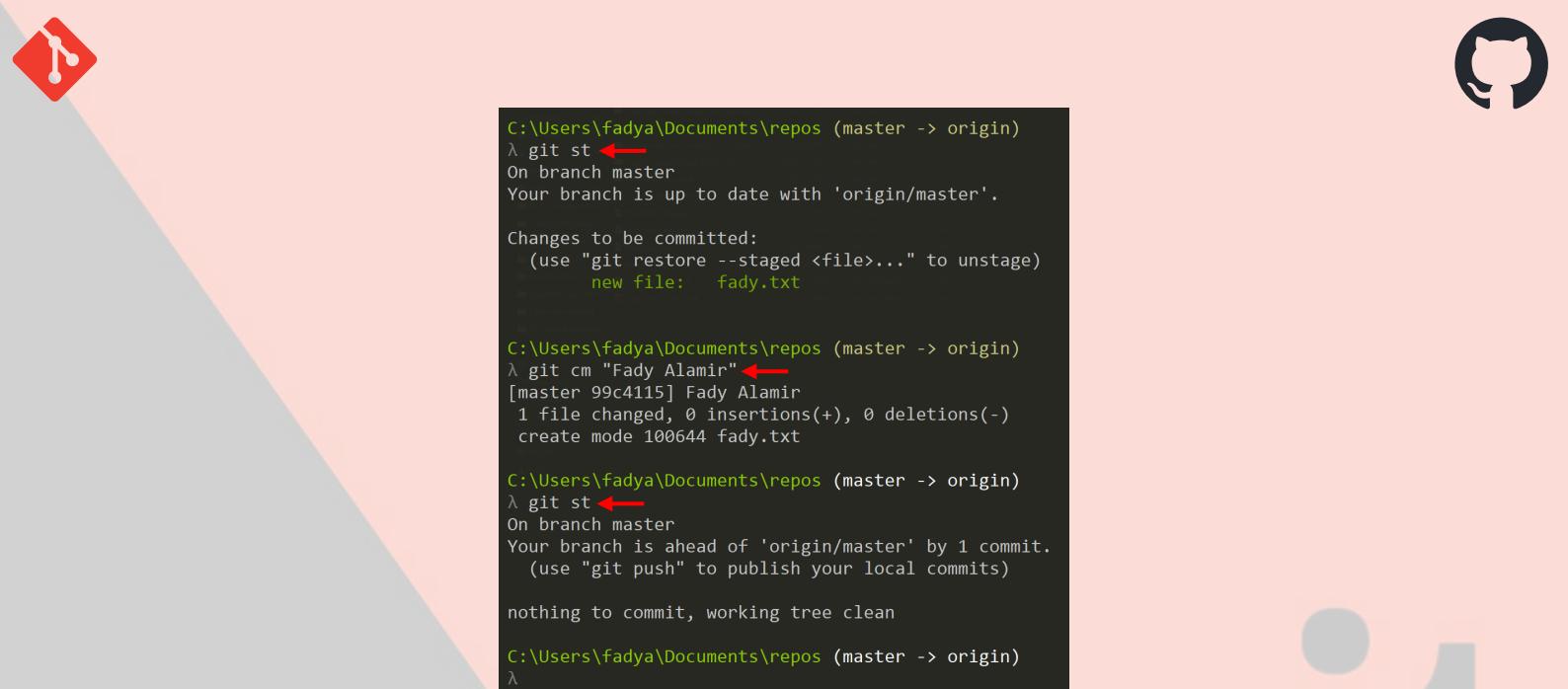
C:\Users\fadya\Documents\repos (master -> origin)
λ touch fady.txt ←

C:\Users\fadya\Documents\repos (master -> origin)
λ git st ←
On branch master
Your branch is up to date with 'origin/master'.

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

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

C:\Users\fadya\Documents\repos (master -> origin)
λ git add fady.txt ←
```



```

C:\Users\fadya\Documents\repos (master -> origin)
λ git st ←
On branch master
Your branch is up to date with 'origin/master'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    new file:  fady.txt

C:\Users\fadya\Documents\repos (master -> origin)
λ git cm "Fady Alamir" ←
[master 99c4115] Fady Alamir
  1 file changed, 0 insertions(+), 0 deletions(-)
  create mode 100644 fady.txt

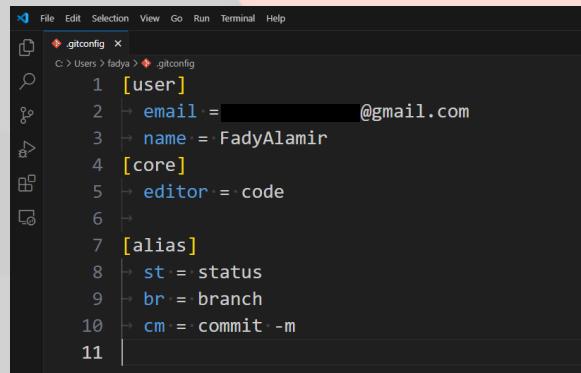
C:\Users\fadya\Documents\repos (master -> origin)
λ git st ←
On branch master
Your branch is ahead of 'origin/master' by 1 commit.
  (use "git push" to publish your local commits)

nothing to commit, working tree clean

C:\Users\fadya\Documents\repos (master -> origin)
λ

```

٥- الأمر `git config --global --edit` يستخدم لعرض الـ `alias` في ملف `text` لرؤية الأسماء المستعارة التي قمنا بإنشائها ويمكن إضافة اسم مستعار لأي `command` آخر من خلاله



#12 – Learn About Branching And Merging

١- `Branch`: عبارة عن فرع من فروع الـ `develop` تعمل عليه بمفردك لكي تقوم بشيء معين مطلوب منك أو شيء معين تريد تطويره في المشروع الخاص بك

مثال: معين `feature` اضافته او `code` في الـ `enhance` او تقوم باصلاح `bug` معين حيث تقوم بعمل `branch` وتعطيه الاسم المناسب على حسب سياسة الشركة



وعند الانتهاء من العمل على الـ **Branch** تقوم بدمج الـ **Branch** مع الـ **Master Branch** قبل أن يرسل للسيرفر الخاص بالشركة أو الـ **GitHub**

أو تقوم بإرسال الـ **pull request** للـ **origin Branch** (السيرفر أو الـ **Branch**) حتى يري المسؤولين عن المشروع الكود الخاص بك أاما ان يكون الكود الخاص بك سليم ويقوموا بعمل **Merge** له أو يكون به مشكلة في يقوموا بعمل **reject**

٢- الأوامر Commands

١ . **git branch** : أمر يستخدم لمعرفة الـ **Branches** الموجودة في الـ **repo**

الخاص بك

```
C:\Users\fadya\Documents\Elzero\Branching (master -> origin)
└ git branch ←
* master

C:\Users\fadya\Documents\Elzero\Branching (master -> origin)
└ |
```

٢ . **git branch [branchName]** : أمر يستخدم لعمل **Branch** جديد في الـ **repo** الخاص بك

```
C:\Users\fadya\Documents\Elzero\Branching (master -> origin)
└ git branch ←
* master

C:\Users\fadya\Documents\Elzero\Branching (master -> origin)
└ git branch Scroll ←

C:\Users\fadya\Documents\Elzero\Branching (master -> origin)
└ git branch
  Scroll ←
* master

C:\Users\fadya\Documents\Elzero\Branching (master -> origin)
└ |
```

٣ . **git checkout [branchName]** : أمر يستخدم للتنقل بين الـ **Branches**

```
C:\Users\fadya\Documents\Elzero\Branching (master -> origin)
└ git checkout Scroll ←
Switched to branch 'Scroll'

C:\Users\fadya\Documents\Elzero\Branching (Scroll)
└ git branch
* Scroll ←
  master

C:\Users\fadya\Documents\Elzero\Branching (Scroll)
└ |
```



٤. أمر يستخدم لحذف branch من الـ repo: `git branch -d [branchName]`.
الخاص بك

```
C:\Users\fadya\Documents\Elzero\Branching (master -> origin)
λ git branch -d Scroll ←
Deleted branch Scroll (was 166d817).

C:\Users\fadya\Documents\Elzero\Branching (master -> origin)
λ git branch
* master ←

C:\Users\fadya\Documents\Elzero\Branching (master -> origin)
λ |
```

ملحوظة: لا يمكن حذف الـ branch إذا كنت تستخدمه يجب عمل `switch` له
آخر ثم تقوم بحذفه

```
C:\Users\fadya\Documents\Elzero\Branching (Scroll)
λ git branch
* Scroll ←
  master

C:\Users\fadya\Documents\Elzero\Branching (Scroll)
λ git branch -d Scroll
error: cannot delete branch 'Scroll' used by worktree at 'C:/Users/fadya/Documents/Elzero/Branching'

C:\Users\fadya\Documents\Elzero\Branching (Scroll)
λ |
```

ملحوظة ٢: الأمر `git branch -d [branchName]` يختلف عن
الأمر `git branch -D [branchName]`

الـ d- عبارة عن `save delete` حيث يقوم بعمل branch هل الـ check هل لها تعديلات
معينة لم يتم عمل `Merge` لها فلن يقوم بحذفها

الـ D- يقوم بعمل `force delete` ويقوم بحذف الـ branch حتى لو كان به تعديلات لم
يتم عمل `Merge` لها

٥. أمر يستخدم لعمل Branch جديد: `git checkout -b [branchName]`.
ويجعل الـ branch ينتقل للـ الجديد

```
C:\Users\fadya\Documents\Elzero\Branching (master -> origin)
λ git branch ←
* master

C:\Users\fadya\Documents\Elzero\Branching (master -> origin)
λ git checkout -b Dev-Feature ←
Switched to a new branch 'Dev-Feature'

C:\Users\fadya\Documents\Elzero\Branching (Dev-Feature)
λ |
```



٦. أمر يستخدم لإعادة تسمية الـ **Branch**: `git branch -m [branchName]`.
الخاص بك ويجب أن تكون داخل الـ **branch** بواسطة الـ `git` الذي تريد ان تقوم بإعادة تسميته

```
C:\Users\fadya\Documents\Elzero\Branching (Dev-Feature)
λ git branch -m Development →

C:\Users\fadya\Documents\Elzero\Branching (Development)
λ |
```

ثم نقوم بعمل `Merge` للـ **master branch** مع الـ **Development branch**

```
C:\Users\fadya\Documents\Elzero\Branching (Development)
λ touch index.html ←

C:\Users\fadya\Documents\Elzero\Branching (Development)
λ git st ←
On branch Development
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\fadya\Documents\Elzero\Branching (Development)
λ git add index.html ←

C:\Users\fadya\Documents\Elzero\Branching (Development)
λ git commit -m "Testing Branch" ←
[Development 91ff955] Testing Branch
  1 file changed, 0 insertions(+), 0 deletions(-)
  create mode 100644 index.html

C:\Users\fadya\Documents\Elzero\Branching (Development)
λ |
```

ثم نقوم بعمل `switch` لكي نقوم بعمل `merge` للـ **master branch**

```
C:\Users\fadya\Documents\Elzero\Branching (master -> origin)
λ git merge Development →
Updating 166d817..d5798e5
Fast-forward
  index.html | 0
  1 file changed, 0 insertions(+), 0 deletions(-)
  delete mode 100644 index.html

C:\Users\fadya\Documents\Elzero\Branching (master -> origin)
λ |
```

ونقوم بحذف الـ `Development branch`

```
C:\Users\fadya\Documents\Elzero\Branching (master -> origin)
λ git branch -d Development
error: the branch 'Development' is not fully merged ←
hint: If you are sure you want to delete it, run 'git branch -D Development'
hint: Disable this message with "git config advice.forceDeleteBranch false"

C:\Users\fadya\Documents\Elzero\Branching (master -> origin)
λ git branch -D Development ←
Deleted branch Development (was 91ff955).

C:\Users\fadya\Documents\Elzero\Branching (master -> origin)
λ |
```



وبعدما قمنا بعمل الـ **Merge** نقوم برفع الـ **master branch** للـ **centralized server**

```
C:\Users\fadya\Documents\Elzero\Branching (master -> origin)
λ git push origin master ←
Enter passphrase for key '/c/Users/fadya/.ssh/id_rsa':
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Writing objects: 100% (2/2), 200 bytes | 200.00 KiB/s, done.
Total 2 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To github.com:fadyalamir/Elzero.git
  166d817..d5798e5  master -> master

C:\Users\fadya\Documents\Elzero\Branching (master -> origin)
λ |
```

٣- عمل **branch** جديد ولكن سوف نقوم بإرسال التعديلات للـ **origin** مباشرةً باستخدام الـ **merge** مع الـ **pull request** دون استخدام الـ **pull request**

```
C:\Users\fadya\Documents\Elzero\Branching (master -> origin)
λ git checkout -b Scroll ←
Switched to a new branch 'Scroll'

C:\Users\fadya\Documents\Elzero\Branching (Scroll)
λ touch main.js ←

C:\Users\fadya\Documents\Elzero\Branching (Scroll)
λ git st ←
On branch Scroll
Untracked files:
  (use "git add <file>..." to include in what will be committed)
    main.js

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

C:\Users\fadya\Documents\Elzero\Branching (Scroll)
λ git add main.js ←

C:\Users\fadya\Documents\Elzero\Branching (Scroll)
λ git commit -m "New Exper" ←
[Scroll e0b589e] New Exper
  1 file changed, 0 insertions(+), 0 deletions(-)
  create mode 100644 main.js

C:\Users\fadya\Documents\Elzero\Branching (Scroll)
λ |
```

ومن ثم نقوم بإرسال التعديلات من الـ **scroll branch** للـ **origin** مباشرةً هكذا

```
C:\Users\fadya\Documents\Elzero\Branching (Scroll)
λ git push origin Scroll ←
Enter passphrase for key '/c/Users/fadya/.ssh/id_rsa':
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Writing objects: 100% (3/3), 243 bytes | 243.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
remote:
remote: Create a pull request for 'Scroll' on GitHub by visiting:
remote:     https://github.com/fadyalamir/Elzero/pull/new/Scroll
remote:
To github.com:fadyalamir/Elzero.git
 * [new branch]      Scroll -> Scroll

C:\Users\fadya\Documents\Elzero\Branching (Scroll)
λ |
```



ثم عند الذهاب للremote repo نجد ال pull request

A screenshot of a GitHub repository page for 'Elzero'. At the top right, there is a green button labeled 'Compare & pull request'. A red arrow points to this button. The repository has 2 branches and 0 tags. It shows a commit from 'fadyalamir' titled 'New Exper' made 14 minutes ago. There is a section for adding a README, with a 'Add a README' button. On the right side, there is an 'About' section with basic repository statistics like activity, stars, forks, releases, and packages.

ومن ثم نري التعديلات عند الضغط على زر compare & pull request

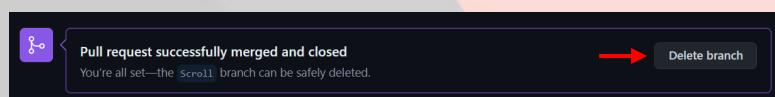
A screenshot of the 'Open a pull request' dialog box. It shows two dropdown menus: 'scroll branch' (highlighted with a red arrow) and 'master branch' (highlighted with a red arrow). Below these, there is a text input field for 'Add a title' and a rich text editor for 'Add a description'. On the right side, there are sections for 'Reviewers', 'Assignees', 'Labels', 'Projects', 'Milestone', and 'Development'. At the bottom, there is a 'Pull Request' button with a red arrow pointing to it, and a note indicating the branches are 'Able to merge'. The dialog also includes a note about GitHub Community Guidelines and a preview of the commit history.



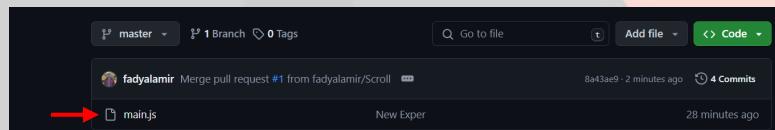
ولأنك admin لا repo سوف تقوم بعمل merge أم لا
بعض فحص وجود أي conflicts في التعديلات أم لا

The screenshot shows a GitHub repository interface. At the top, there's a navigation bar with links like 'Code', 'Issues', 'Pull requests', 'Actions', 'Projects', 'Wiki', 'Security', 'Insights', and 'Settings'. Below the navigation, a pull request is shown with the title 'New Exper #1'. A red arrow points to the text 'master branch - scroll branch' which appears above the pull request details. Another red arrow points to the green 'Merge pull request' button at the bottom of the pull request card.

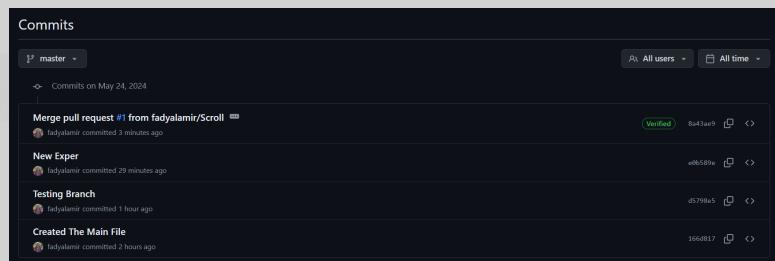
وبعد ال merged تستطيع حذف ال branch من زرار



وبعد ال merged نجد التعديل وهو الملف الذي قمنا بإنشائه في ال master branch



وجميع التعديلات سوف تكون في ال commits الخاصة بال repo





#13 – Mastering Stash Part 1

١ - **git stash**: أمر يستخدم لتخزين بعض الملفات حتى لا يتم ارسالهم مع التعديلات الجديدة أثناء العمل على الـ **branch**

حيث يعتبر كمخباً أو صندوق لتخزين الملفات

```
C:\Users\fadya\Documents\Elzero\Branching (master -> origin)
λ git status ←
On branch master
Your branch is up to date with 'origin/master'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    new file: README.md
    new file: about.html
    new file: about_readme.txt
    new file: index.html

C:\Users\fadya\Documents\Elzero\Branching (master -> origin)
λ git stash ←
Saved working directory and index state WIP on master: d5798e5 Testing Branch

C:\Users\fadya\Documents\Elzero\Branching (master -> origin)
λ git st ←
On branch master
Your branch is up to date with 'origin/master'.

nothing to commit, working tree clean

C:\Users\fadya\Documents\Elzero\Branching (master -> origin)
λ |
```

ولن تظهر الملفات او يكون لها وجود وتكون محفوظة في الـ **stash** الخاص بالـ **git**

```
C:\Users\fadya\Documents\Elzero\Branching (master -> origin)
λ ls

C:\Users\fadya\Documents\Elzero\Branching (master -> origin)
λ |
```

Text ٢ - **echo "Hello World" > about_readme.txt**: أمر يستخدم لأنشاء ملف **Hello World** بداخله جملة

```
C:\Users\fadya\Documents\Elzero\Branching (master -> origin)
λ echo "Hello World" > about_readme.txt|
```

Text ٣ - **git stash list**: أمر يستخدم لرؤية الـ **stashes** الموجودة في الـ **branch** الخاص بك والـ **stash** الخاص بكل **id**

```
C:\Users\fadya\Documents\Elzero\Branching (master -> origin)
λ git stash list ←
stash@{0}: WIP on master: d5798e5 Testing Branch

C:\Users\fadya\Documents\Elzero\Branching (master -> origin)
λ |
```



٤- أمر يستخدم لإخراج الملفات من الـ `stash` التي قمت بإنشائه لكي تكون في الـ `remote repo` ويمكن عمل `push` لها للـ `branch`

```
C:\Users\fadya\Documents\Elzero\Branching (master -> origin)
λ git stash list ←
stash@{0}: WIP on master: d5798e5 Testing Branch

C:\Users\fadya\Documents\Elzero\Branching (master -> origin)
λ git stash pop ←
On branch master
Your branch is up to date with 'origin/master'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    new file: README.md
    new file: about.html
    new file: about_readme.txt
    new file: index.html

Dropped refs/stash@{0} (56250ea3390e62862a6f047e8a3d1ea7db65ad49)

C:\Users\fadya\Documents\Elzero\Branching (master -> origin)
λ |
```

مرة اخري لن يكون موجودا stashes الخاص بالـ `list` عند محاولة رؤية الـ

```
C:\Users\fadya\Documents\Elzero\Branching (master -> origin)
λ git stash list

C:\Users\fadya\Documents\Elzero\Branching (master -> origin)
λ |
```

ومن ثم رفع الملفات التي كانت في الـ `stash` بعدما قمنا بإخراجها من الـ `stash` للـ `remote repo`

```
C:\Users\fadya\Documents\Elzero\Branching (master -> origin)
λ git commit -m "Added Files From Stash" ←
[master d17c8e2] Added Files From Stash
 4 files changed, 1 insertion(+)
  create mode 100644 README.md
  create mode 100644 about.html
  create mode 100644 about_readme.txt
  create mode 100644 index.html

C:\Users\fadya\Documents\Elzero\Branching (master -> origin)
λ git st ←
On branch master
Your branch is ahead of 'origin/master' by 1 commit.
  (use "git push" to publish your local commits)

nothing to commit, working tree clean

C:\Users\fadya\Documents\Elzero\Branching (master -> origin)
λ git push origin master ←
Enter passphrase for key '/c/Users/fadya/.ssh/id_rsa':
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 12 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 339 bytes | 339.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To github.com:fadyalamir/Elzero.git
  ec29371..d17c8e2  master -> master

C:\Users\fadya\Documents\Elzero\Branching (master -> origin)
λ |
```



#14 – Mastering Stash Part 2

أولاًً سوف نقوم بإنشاء ملف ووضعه في ال staging area ومن ثم وضعه داخل ال stash

```
C:\Users\fadya\Documents\Elzero\Branching (master -> origin)
λ touch m.txt ←

C:\Users\fadya\Documents\Elzero\Branching (master -> origin)
λ git add m.txt ←

C:\Users\fadya\Documents\Elzero\Branching (master -> origin)
λ git st ←
On branch master
Your branch is up to date with 'origin/master'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    new file:   m.txt

C:\Users\fadya\Documents\Elzero\Branching (master -> origin)
λ git stash ←
Saved working directory and index state WIP on master: d17c8e2 Added Files From Stash
```

ومن ثم نقوم بإنشاء ملف جديد ونقوم بمحاولة وضعه في ال stash ولكن لن يوضع في نفس ال stash التي وضع بها الملف السابق وسوف يوضع في stash جديدة بـ id جديد

```
C:\Users\fadya\Documents\Elzero\Branching (master -> origin)
λ touch x.txt ←

C:\Users\fadya\Documents\Elzero\Branching (master -> origin)
λ git add x.txt ←

C:\Users\fadya\Documents\Elzero\Branching (master -> origin)
λ git stash list ←
stash@{0}: WIP on master: d17c8e2 Added Files From Stash

C:\Users\fadya\Documents\Elzero\Branching (master -> origin)
λ | ←

C:\Users\fadya\Documents\Elzero\Branching (master -> origin)
λ git stash ←
Saved working directory and index state WIP on master: d17c8e2 Added Files From Stash

C:\Users\fadya\Documents\Elzero\Branching (master -> origin)
λ git stash list ←
stash@{0}: WIP on master: d17c8e2 Added Files From Stash
stash@{1}: WIP on master: d17c8e2 Added Files From Stash ←

C:\Users\fadya\Documents\Elzero\Branching (master -> origin)
λ | ←
```

1 - **stash**: أمر يستخدم لوضع الملفات في ال stash وتسمية ال stash للتفرقة بين ال stashes الموجودة في ال branch الخاص بك

```
C:\Users\fadya\Documents\Elzero\Branching (master -> origin)
λ touch osx.txt ←

C:\Users\fadya\Documents\Elzero\Branching (master -> origin)
λ git add osx.txt ←

C:\Users\fadya\Documents\Elzero\Branching (master -> origin)
λ git stash save "OSX Text" ←
Saved working directory and index state On master: OSX Text

C:\Users\fadya\Documents\Elzero\Branching (master -> origin)
λ git stash list ←
stash@{0}: On master: OSX Text ←
stash@{1}: WIP on master: d17c8e2 Added Files From Stash
stash@{2}: WIP on master: d17c8e2 Added Files From Stash

C:\Users\fadya\Documents\Elzero\Branching (master -> origin)
λ | ←
```



وعند البحث عن الملفات التي قمنا بإنشائهما لن نجدها لأنها داخل الـ **stashes**

```
C:\Users\fadya\Documents\Elzero\Branching (master -> origin)
λ ls
about.html  about_readme.txt  index.html  main.js  README.md
```

Home	Name	Date modified	Type	Size
Gallery	.git	5/27/2024 12:09 AM	File folder	
OneDrive - Personal	about.html	5/25/2024 7:10 PM	Brave HTML Doc...	0 KB
Desktop	about_readme.txt	5/25/2024 7:10 PM	Text Document	1 KB
Downloads	index.html	5/25/2024 7:10 PM	Brave HTML Doc...	0 KB
Documents	main.js	5/25/2024 6:59 PM	JSFile	0 KB
	README.md	5/25/2024 7:10 PM	Markdown Source...	0 KB

ملحوظة: أمر الـ **git stash pop** عند استخدامه يقوم بإخراج الملفات من آخر stash قمنا بإنشائه هكذا

```
C:\Users\fadya\Documents\Elzero\Branching (master -> origin)
λ git stash pop ←
On branch master
Your branch is up to date with 'origin/master'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    new file:   osx.txt ←

Dropped refs/stash@{0} (b950b2da7aaae6d639b87d7abaf8e346961fc9)

C:\Users\fadya\Documents\Elzero\Branching (master -> origin)
λ |
```

وعند رؤية الـ **stash list** لن يكون من بينهم

```
C:\Users\fadya\Documents\Elzero\Branching (master -> origin)
λ git stash list
stash@{0}: WIP on master: d17c8e2 Added Files From Stash
stash@{1}: WIP on master: d17c8e2 Added Files From Stash

C:\Users\fadya\Documents\Elzero\Branching (master -> origin)
λ |
```

٢ - **أمر يستخدم في سحب الملفات من آخر stash أنشأناه ولكن يترك الـ stash كما هو في الـ repo الخاص بك هكذا**

```
C:\Users\fadya\Documents\Elzero\Branching (master -> origin)
λ git stash list ←
stash@{0}: On master: OSX Text
stash@{1}: WIP on master: d17c8e2 Added Files From Stash
stash@{2}: WIP on master: d17c8e2 Added Files From Stash

C:\Users\fadya\Documents\Elzero\Branching (master -> origin)
λ git stash apply ←
On branch master
Your branch is up to date with 'origin/master'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    new file:   osx.txt

C:\Users\fadya\Documents\Elzero\Branching (master -> origin)
λ git stash list ←
stash@{0}: On master: OSX Text ←
stash@{1}: WIP on master: d17c8e2 Added Files From Stash
stash@{2}: WIP on master: d17c8e2 Added Files From Stash

C:\Users\fadya\Documents\Elzero\Branching (master -> origin)
λ |
```



ومن ثم يمكن رفع الملف للـ **remote repo** بدون اي مشاكل هكذا

```
C:\Users\fadya\Documents\Elzero\Branching (master -> origin)
λ git commit -m "OSX Test"
[master 39ce0ac] OSX Test
 1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 osx.txt
```

```
C:\Users\fadya\Documents\Elzero\Branching (master -> origin)
λ git push origin master
Enter passphrase for key '/c/Users/fadya/.ssh/id_rsa':
Enumerating objects: 6, done.
Counting objects: 100% (6/6), done.
Delta compression using up to 12 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (4/4), 468 bytes | 468.00 KiB/s, done.
Total 4 (delta 2), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (2/2), completed with 1 local object.
To github.com:fadyalamir/Elzero.git
  eedcf1c..09cc6d9  master -> master

C:\Users\fadya\Documents\Elzero\Branching (master -> origin)
λ |
```

```
C:\Users\fadya\Documents\Elzero\Branching (master -> origin)
λ git status
On branch master
Your branch is up to date with 'origin/master'.

nothing to commit, working tree clean
```

stash: أمر يستخدم لإخراج الملفات الموجودة في معين ومحدد عن طريق الـ **id** الخاص به ومن ثم حذف هذا الـ **repo** من الـ **stash**

```
C:\Users\fadya\Documents\Elzero\Branching (master -> origin)
λ git stash list ←
stash@{0}: On master: OSX Text
stash@{1}: WIP on master: d17c8e2 Added Files From Stash
stash@{2}: WIP on master: d17c8e2 Added Files From Stash

C:\Users\fadya\Documents\Elzero\Branching (master -> origin)
λ git stash pop stash@{2} ←
On branch master
Your branch is up to date with 'origin/master'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    new file:   m.txt

Dropped stash@{2} (50e73a8ddecd2c23975b11c84e5e0b597a25eac1)

C:\Users\fadya\Documents\Elzero\Branching (master -> origin)
λ git status ←
On branch master
Your branch is up to date with 'origin/master'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    new file:   m.txt

C:\Users\fadya\Documents\Elzero\Branching (master -> origin)
λ |
```

و عند رؤية الـ **stash list** لن يكون من بينهم

```
C:\Users\fadya\Documents\Elzero\Branching (master -> origin)
λ git stash list
stash@{0}: On master: OSX Text
stash@{1}: WIP on master: d17c8e2 Added Files From Stash
```



٤- أمر `git stash drop`: أمر يستخدم لحذف آخر stash قمنا بإنشائه من الـ repo الموجودة في الـ repo الخاص بك

```
C:\Users\fadya\Documents\Elzero\Branching (master -> origin)
λ git stash list ←
stash@{0}: On master: OSX Text ←
stash@{1}: WIP on master: d17c8e2 Added Files From Stash

C:\Users\fadya\Documents\Elzero\Branching (master -> origin)
λ git stash drop ←
Dropped refs/stash@{0} (8e4d76400c7457a77ad2974e35c6dfb8e434b2b4)

C:\Users\fadya\Documents\Elzero\Branching (master -> origin)
λ git stash list ←
stash@{0}: WIP on master: d17c8e2 Added Files From Stash
```

ملاحظة ٢: أمر `git stash drop` يقوم بحذف الـ `stash` بالملفات الموجودة بداخله بدون أي إنذار لوجود ملفات داخل الـ `stash`

-٥- أمر `git stash drop stash@{id}`: أمر يستخدم لحذف stash من الـ repo من الـ `stash` الخاص به ومن ثم حذف هذا الـ `id`

```
C:\Users\fadya\Documents\Elzero\Branching (master -> origin)
λ git stash save "OSX Text" ←
Saved working directory and index state On master: OSX Text

C:\Users\fadya\Documents\Elzero\Branching (master -> origin)
λ git stash list
stash@{0}: On master: OSX Text ←
stash@{1}: WIP on master: d17c8e2 Added Files From Stash

C:\Users\fadya\Documents\Elzero\Branching (master -> origin)
λ git stash drop stash@{1} ←
Dropped stash@{1} (e8d68aa65e9b4f596edd0e756ccbb8603ff9c080)

C:\Users\fadya\Documents\Elzero\Branching (master -> origin)
λ git stash list ←
stash@{0}: On master: OSX Text

C:\Users\fadya\Documents\Elzero\Branching (master -> origin)
λ |
```

٦- أمر `git stash show`: أمر يستخدم لاظهار آخر stash قمنا بإنشائه وما بداخل هذا stash الموجود في الـ repo الخاص بك

```
C:\Users\fadya\Documents\Elzero\Branching (master -> origin)
λ git stash list
stash@{0}: WIP on master: ccf8dc4 remove osx.txt ←
stash@{1}: On master: OSX Text

C:\Users\fadya\Documents\Elzero\Branching (master -> origin)
λ git stash show ←
  .txt | 0
  1 file changed, 0 insertions(+), 0 deletions(-)
```



أمر يستخدم لإظهار ما بداخل stash معين ومحدد عن طريق id الخاص به: `git stash show stash@{id} -v`

```
C:\Users\fadya\Documents\Elzero\Branching (master -> origin)
λ git stash list ←
stash@{0}: WIP on master: ccf8dc4 remove osx.txt
stash@{1}: On master: OSX Text

C:\Users\fadya\Documents\Elzero\Branching (master -> origin)
λ git stash show stash@{1} ←
osx.txt | 0
1 file changed, 0 insertions(+), 0 deletions(-)
```

أمر يقوم بحذف جميع stash الموجودة في repo الخاص بك بالملفات الموجودة بداخلها هكذا: `git stash clear -v`

```
C:\Users\fadya\Documents\Elzero\Branching (master -> origin)
λ git stash list ←
stash@{0}: WIP on master: ccf8dc4 remove osx.txt
stash@{1}: On master: OSX Text

C:\Users\fadya\Documents\Elzero\Branching (master -> origin)
λ git stash clear ←

C:\Users\fadya\Documents\Elzero\Branching (master -> origin)
λ git stash list ←

C:\Users\fadya\Documents\Elzero\Branching (master -> origin)
λ |
```

#15 – Restore And Clean

أمر يستخدم لعمل `unstage` لملف قمنا بإضافته لل staging area مسبقاً: `git restore --staged [filename]`

```
C:\Users\fadya\Downloads\My-GitHub\Git_Course_Continue (master -> origin)
λ touch new.txt ←

C:\Users\fadya\Downloads\My-GitHub\Git_Course_Continue (master -> origin)
λ git status ←
On branch master
Your branch is up to date with 'origin/master'.

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

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

C:\Users\fadya\Downloads\My-GitHub\Git_Course_Continue (master -> origin)
λ git add new.txt ←

C:\Users\fadya\Downloads\My-GitHub\Git_Course_Continue (master -> origin)
λ git status ←
On branch master
Your branch is up to date with 'origin/master'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    new file:   new.txt

C:\Users\fadya\Downloads\My-GitHub\Git_Course_Continue (master -> origin)
λ |
```



```
C:\Users\fadya\Downloads\My-GitHub\Git_Course_Continue (master -> origin)
λ git restore --staged new.txt ←

C:\Users\fadya\Downloads\My-GitHub\Git_Course_Continue (master -> origin)
λ git status ←
On branch master
Your branch is up to date with 'origin/master'.

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

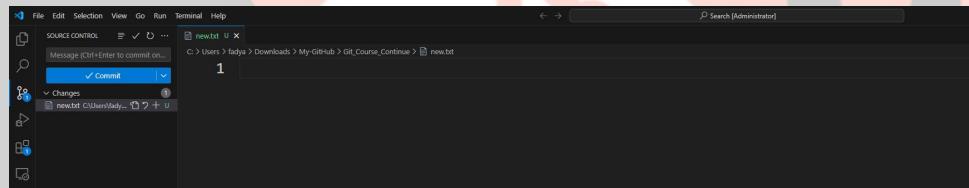
nothing added to commit but untracked files present (use "git add" to track)

C:\Users\fadya\Downloads\My-GitHub\Git_Course_Continue (master -> origin)
λ |
```

**: أمر يستخدم لفتح ملف معين في الـ repo الخاص بك باستخدام الـ code [filename] -٢
هكذا vs code**

```
C:\Users\fadya\Downloads\My-GitHub\Git_Course_Continue (master -> origin)
λ code new.txt

C:\Users\fadya\Downloads\My-GitHub\Git_Course_Continue (master -> origin)
λ |
```



ملحوظة: من الممكن عمل ما نقوم به باستخدام الأوامر عن طريق الـ git نقوم بعمله في
الـ vs code عن طريق الـ graphical user فقط دون استخدام
الأوامر

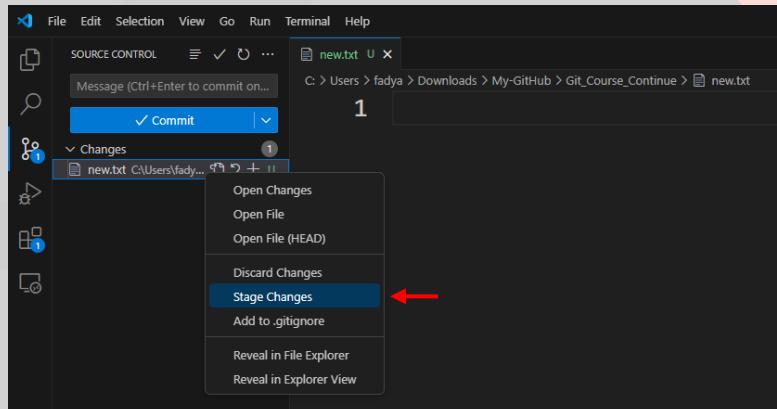
حيث يمكن وضع الملف في الـ staging area عن طريق الـ graphical user هكذا

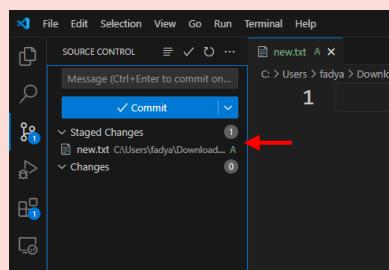
```
C:\Users\fadya\Downloads\My-GitHub\Git_Course_Continue (master -> origin)
λ git status
On branch master
Your branch is up to date with 'origin/master'.

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

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

C:\Users\fadya\Downloads\My-GitHub\Git_Course_Continue (master -> origin)
λ |
```



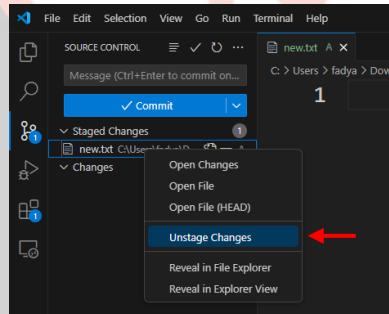


```
C:\Users\fadya\Downloads\My-GitHub\Git_Course_Continue (master -> origin)
λ git status
On branch master
Your branch is up to date with 'origin/master'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    new file:   new.txt

C:\Users\fadya\Downloads\My-GitHub\Git_Course_Continue (master -> origin)
λ |
```

أيضاً يمكن عمل **unstage** للملف من الـ graphical user



```
C:\Users\fadya\Downloads\My-GitHub\Git_Course_Continue (master -> origin)
λ git status
On branch master
Your branch is up to date with 'origin/master'.

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

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

C:\Users\fadya\Downloads\My-GitHub\Git_Course_Continue (master -> origin)
λ |
```

ملاحظة ٢: من الممكن عمل **unstage** لأكثر من ملف باضافته لـ **"git restore --staged *"** هكذا

```
C:\Users\fadya\Downloads\My-GitHub\Git_Course_Continue (master -> origin)
λ git status
On branch master
Your branch is up to date with 'origin/master'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    new file:   new.txt
    new file:   new2.txt

C:\Users\fadya\Downloads\My-GitHub\Git_Course_Continue (master -> origin)
λ git restore --staged *

C:\Users\fadya\Downloads\My-GitHub\Git_Course_Continue (master -> origin)
λ git status
On branch master
Your branch is up to date with 'origin/master'.

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    new.txt
    new2.txt

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

C:\Users\fadya\Downloads\My-GitHub\Git_Course_Continue (master -> origin)
λ |
```



أمر يقوم بحذف الملفات الـ **untracked** التي لم يتم اضافتها في
الـ **staging area** هكذا

```
C:\Users\fadya\Downloads\My-GitHub\Git_Course_Continue (master -> origin)
λ touch new3.txt ←

C:\Users\fadya\Downloads\My-GitHub\Git_Course_Continue (master -> origin)
λ git status ←
On branch master
Your branch is up to date with 'origin/master'.

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    new.txt
    new2.txt
    new3.txt

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

C:\Users\fadya\Downloads\My-GitHub\Git_Course_Continue (master -> origin)
λ git add new2.txt ←

C:\Users\fadya\Downloads\My-GitHub\Git_Course_Continue (master -> origin)
λ git clean -n ←
Would remove new.txt
Would remove new3.txt

C:\Users\fadya\Downloads\My-GitHub\Git_Course_Continue (master -> origin)
λ git clean -f ←
Removing new.txt
Removing new3.txt

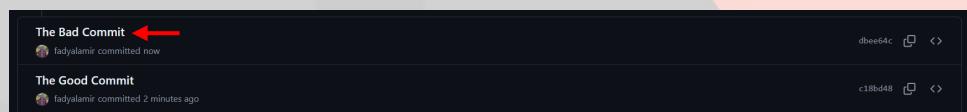
C:\Users\fadya\Downloads\My-GitHub\Git_Course_Continue (master -> origin)
λ git status ←
On branch master
Your branch is up to date with 'origin/master'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    new file:   new2.txt

C:\Users\fadya\Downloads\My-GitHub\Git_Course_Continue (master -> origin)
λ |
```

#16 – Resetting The Head

1 - **head**: هو مصطلح في الـ **git** يُعبر عن الـ **pointer** الذي يُشير للأخر **commit** قمنا
بإنشاءه



وهو الـ **Bad Commit**



٢- git log : أمر يستخدم لرؤية ال commits التي قمنا بإنشائها في ال branch

```
C:\Users\fadya\Downloads\My-GitHub\Git_Course_Continue (master -> origin)
λ git log
commit dbree64c755a6e60b784b0e4da4a5cc5088303f4f (HEAD -> master, origin/master)
Author: FadyAlamir <[REDACTED].com>
Date:   Wed May 29 21:11:07 2024 +0300

    The Bad Commit

commit c18bd489118ad3b5fc43f4f23f9ad1942a356746
Author: FadyAlamir <[REDACTED].com>
Date:   Wed May 29 21:08:48 2024 +0300

    The Good Commit

commit edf72a1a3c58143a65aa65de1372e2a5c3e87a59
Author: FadyAlamir <[REDACTED].com>
Date:   Wed May 29 00:48:25 2024 +0300

    Initial commit

C:\Users\fadya\Downloads\My-GitHub\Git_Course_Continue (master -> origin)
λ |
```

ال HEAD هو ال commit الاخير معناها أن هذا ال commit المكتوبة في ال HEAD هو ال الاخير في ال master branch

ملاحظة: كل commit له hash code هكذا

The screenshot shows a GitHub commit page for the 'Code' tab of the 'fadyalamir / Git_Course_Continue' repository. The commit is titled 'The Good Commit' and was made by 'fadyalamir' 23 minutes ago. The commit message includes the hash code 'c18bd489118ad3b5fc43f4f23f9ad1942a356746'. A red arrow points to this hash code.

```
commit c18bd489118ad3b5fc43f4f23f9ad1942a356746 ←
Author: FadyAlamir <[REDACTED]@gmail.com>
Date:   Wed May 29 21:08:48 2024 +0300

    The Good Commit
```

٣- git reset --hard [commitHashCode] : هو أمر يقوم بحذف ال head commit أخر هو ال commit الذي تم تعديله ويجعل ال head بجميع تعديلاته

```
C:\Users\fadya\Downloads\My-GitHub\Git_Course_Continue (master -> origin)
λ git reset --hard c18bd489118ad3b5fc43f4f23f9ad1942a356746
HEAD is now at c18bd48 The Good Commit ←

C:\Users\fadya\Downloads\My-GitHub\Git_Course_Continue (master -> origin)
λ |
```

ومن ثم نقوم بعمل update لل remote repository

```
C:\Users\fadya\Downloads\My-GitHub\Git_Course_Continue (master -> origin)
λ git push origin master --force ←
Enter passphrase for key '/c/Users/fadya/.ssh/id_rsa':
Total 0 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To github.com:fadyalamir/Git_Course_Continue.git
 + dbree64c...c18bd48 master -> master (forced update)

C:\Users\fadya\Downloads\My-GitHub\Git_Course_Continue (master -> origin)
λ |
```



ملحوظة ٢: الـ `--force` لعمل `forced update`

وعند الذهاب ورؤيه الـ `commits` سنجد انه قام بحذف الـ `bad commit` هكذا

```

Commits
master
Commits on May 29, 2024
The Good Commit
fadyalamir committed 33 minutes ago
c18bd48 [ ] <...> All users All time

C:\Users\fadya\Downloads\My-GitHub\Git_Course_Continue (master -> origin)
└ git log
commit c18bd489118ad3b5fc43f4f23f9ad1942a356746 (HEAD -> master, origin/master)
Author: FadyAlamir <...@gmail.com>
Date:   Wed May 29 21:08:48 2024 +0300

    The Good Commit

commit edf72a1a3c58143a65aa65de1372e2a5c3e87a59
Author: FadyAlamir <...@gmail.com>
Date:   Wed May 29 00:48:25 2024 +0300

    Initial commit

C:\Users\fadya\Downloads\My-GitHub\Git_Course_Continue (master -> origin)
└

```

ملحوظة ٣: أمر الـ `git reset --hard [commitHashCode]` من الممكن وضع الـ `soft` فقط ولا تمحى الملفات التي تم تعديلها بدلاً من الـ `hard` لكي تقوم بحذف الـ `commit` فقط ولا تمحى الملفات التي تم تعديلها

ملحوظة ٤: لحذف أكثر من `commits` متتاليين نقوم بجعل الـ `commit` الذي قبلهم هو الـ `head` الـ `commit`

وبالتالي سوف يتم حذف هذه الـ `commits` بملفاتهم والتعديلات الموجودة بداخلهم

#17 – Ignoring Files And Directories

Nodejs: حيث عند العمل بمكتبة معينة مثل الـ `Ignore Files and Directories` - ١ سوف تعمل بملفات `System Backages` سوف تعمل بواسطتها على المشروع ثم ترفع الـ `Code` الخاص بالمشروع ولكن من غير المنطقي رفع الـ `Backages` الخاص بالـ `remote repo` مع كود المشروع للـ `Nodejs`

والـ `IDE` او الـ `Text Editor` يكون له اعدادات معينة او الـ `Config` الخاص بها تخزن في ملف الـ `.xml`. حيث انه ليس من الملفات التي تكون ضمن المشروع لرفعها على الـ `remote repo` وملفات الـ `Hidden System File` مثل الملف الذي يكون في الـ `MAC` وملف `DS_Store` وهو في الـ `Windows` `thums.db`.



٢- ignore files: هو ملف لإنشاء إهمال الخاص بعمل بعض الـ remote repo لعدم إرسالها مع ملفات المشروع

```
C:\Users\fadya\Downloads\My-GitHub\Git_Course_Continue (master -> origin)
λ touch .gitignore ←

C:\Users\fadya\Downloads\My-GitHub\Git_Course_Continue (master -> origin)
λ ls ←
good.txt index.html main.css main.js new2.txt osama.txt README.md

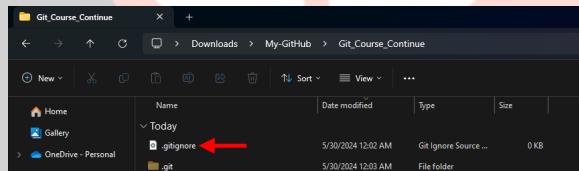
C:\Users\fadya\Downloads\My-GitHub\Git_Course_Continue (master -> origin)
λ git status ←
On branch master
Your branch is up to date with 'origin/master'.

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

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

C:\Users\fadya\Downloads\My-GitHub\Git_Course_Continue (master -> origin)
λ |
```

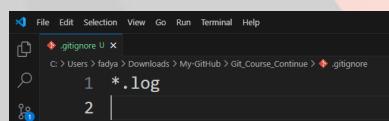
وعند رؤية الملفات باستخدام أمر ls لن يظهر ملف الـ .gitignore. ولكن يكون موجود وسط الملفات على الـ File Browser الخاص بالـ Windows



وبعدها نقوم بالتعديل على الملف باستخدام الـ vs code

```
C:\Users\fadya\Downloads\My-GitHub\Git_Course_Continue (master -> origin)
λ code .gitignore

C:\Users\fadya\Downloads\My-GitHub\Git_Course_Continue (master -> origin)
λ |
```



=> لجعل الـ git يتتجاهل اي ملف امتداده .log

وبالتالي عندما ننشئ ملف امتداده log. سوف يتتجاهله الـ git ولن يظهر في الـ git status

```
C:\Users\fadya\Downloads\My-GitHub\Git_Course_Continue (master -> origin)
λ touch fady.log ←

C:\Users\fadya\Downloads\My-GitHub\Git_Course_Continue (master -> origin)
λ git status ←
On branch master
Your branch is up to date with 'origin/master'.

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

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

C:\Users\fadya\Downloads\My-GitHub\Git_Course_Continue (master -> origin)
λ |
```



ملحوظة: يمكن جعل الـ git يتتجاهل جميع الملفات التي امتدادها log. بإستثناء ملف معين امتداده log.

```
C:\Users\fadya\Downloads\My-GitHub\Git_Course_Continue (master -> origin)
λ touch vip.log ←

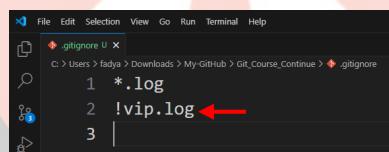
C:\Users\fadya\Downloads\My-GitHub\Git_Course_Continue (master -> origin)
λ git status ←
On branch master
Your branch is up to date with 'origin/master'.

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

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

C:\Users\fadya\Downloads\My-GitHub\Git_Course_Continue (master -> origin)
λ |
```

باستخدام الـ not هكذا



وسوف يظهر باستخدام الامر git status ثم لن يتتجاهله الـ

```
C:\Users\fadya\Downloads\My-GitHub\Git_Course_Continue (master -> origin)
λ git status ←
On branch master
Your branch is up to date with 'origin/master'.

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    .gitignore
    text.txt
    vip.log ←

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

C:\Users\fadya\Downloads\My-GitHub\Git_Course_Continue (master -> origin)
λ |
```

٣- لجعل الـ git يتتجاهل folder معين داخل الـ branch هكذا

```
C:\Users\fadya\Downloads\My-GitHub\Git_Course_Continue (master -> origin)
λ mkdir node_packs
```

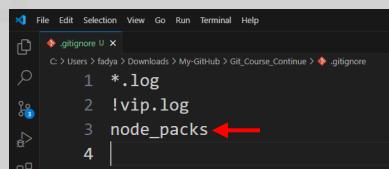


```
C:\Users\fadya\Downloads\My-GitHub\Git_Course_Continue (master -> origin)
λ git status
On branch master
Your branch is up to date with 'origin/master'.

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    .gitignore
    node_packs/ ←
    text.txt
    vip.log

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

C:\Users\fadya\Downloads\My-GitHub\Git_Course_Continue (master -> origin)
λ |
```





بعدما نقوم بوضع اسم الملف في ملف الـ `.gitignore`. سوف يتتجهله الـ `git`

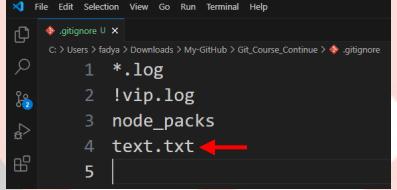
```
C:\Users\fadya\Downloads\My-GitHub\Git_Course_Continue (master -> origin)
λ git status
On branch master
Your branch is up to date with 'origin/master'.

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    .gitignore ←
    text.txt
    vip.log

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

C:\Users\fadya\Downloads\My-GitHub\Git_Course_Continue (master -> origin)
λ |
```

٤- يمكن تحديد ملف معين بامتداده داخل ملف الـ `.gitignore`. ليتجاهله الـ `git` هكذا



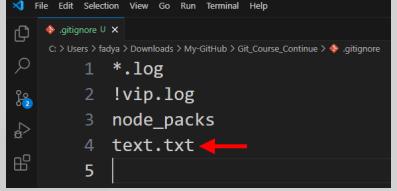
```
C:\Users\fadya\Downloads\My-GitHub\Git_Course_Continue (master -> origin)
λ git status
On branch master
Your branch is up to date with 'origin/master'.

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

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

C:\Users\fadya\Downloads\My-GitHub\Git_Course_Continue (master -> origin)
λ |
```

٥- يمكن عمل `add` أو `push` لملف محدد داخل الـ `.gitignore`. غصبن عن الـ `gitignore` باستخدام الـ `force` هكذا



```
C:\Users\fadya\Downloads\My-GitHub\Git_Course_Continue (master -> origin)
λ git add text.txt
The following paths are ignored by one of your .gitignore files:
text.txt
hint: Use -f if you really want to add them.
hint: Turn this message off by running
hint: "git config advice.addIgnoredFile false"

C:\Users\fadya\Downloads\My-GitHub\Git_Course_Continue (master -> origin)
λ |
```

حيث عندما تريid عمل `add` لملف للـ `git` سوف يقوم الـ `git` بإخبارك بأن `staging area` `ignored` بواسطة الـ `.gitignore`. وإذا كنت تريid اضافته للـ `staging area` استخدم الـ `-f` وهي الـ `flag` الخاصة بالـ `git`



```
C:\Users\fadya\Downloads\My-GitHub\Git_Course_Continue (master -> origin)
λ git add -f text.txt ←

C:\Users\fadya\Downloads\My-GitHub\Git_Course_Continue (master -> origin)
λ git status ←
On branch master
Your branch is up to date with 'origin/master'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    new file:   text.txt

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

C:\Users\fadya\Downloads\My-GitHub\Git_Course_Continue (master -> origin)
λ |
```

في الأمر `git add --force` او `-f`- حيث يمكن وضع

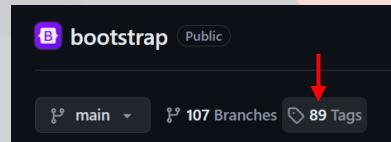
#18 – Tagging And Releasing Part 1

1 - Tag: هي شيء يساعدك في تميز جزء معين من المشروع الخاص بك خلال مدة معينة تستخدم في المشاريع للـ `versions` الخاصة بالمشروع

مثل الـ **Bootstrap Framework** له عدة اصدارات `1 < 2 < 3 < 4` وهكذا

حيث الاصدار الاول مثلاً نقوم بعمل Tag له لتمييزه علي أنه الاصدار الاول من الـ **Bootstrap** وكل اصدار تقوم بعمل Tag مُخصص له

ومن مميزات الـ Tag تستطيع تحميل كل اصدار منفصل بمفرده للمشروع



يوجد للـ **Bootstrap Framework** 89 Tags حتى الأن

Tag Version	Published Date	Downloads	Verified
v5.3.3	Jun 20, 2023	1.4M	Verified
v5.3.2	Sep 14, 2023	1.4M	Verified
v5.3.1	Jul 28, 2023	2.3M	Verified
v5.3.0	May 30, 2023	4.6M	Verified
v5.3.0-alpha3	Apr 2, 2023	1.4M	Verified
v5.3.0-alpha2	Mar 24, 2023	1.4M	Verified
v5.3.0-alpha1	Dec 24, 2022	1.4M	Verified
v5.2.3	Nov 21, 2022	1.4M	Verified
v5.2.2	Oct 1, 2022	9.6M	Verified
v5.2.1	Sep 1, 2022	2.4M	Verified



وال Releases الاصدارات التي اطلقها هي التي نستطيع تحميلها

٢- ينقسم ال Tag إلى نوعين هما ال lightweight tag & annotated tag

١. هي ال Tag المرتبطة بال commit وتأخذ نفس last commit بال message الخاصة بال

```
C:\Users\fadya\Downloads\My-GitHub\Git_Course_Continue (master -> origin)
λ touch first.txt ←

C:\Users\fadya\Downloads\My-GitHub\Git_Course_Continue (master -> origin)
λ git add first.txt ←

C:\Users\fadya\Downloads\My-GitHub\Git_Course_Continue (master -> origin)
λ git commit -m "Message From Commit" ←
[master 1fbabc8] Message From Commit
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 first.txt

C:\Users\fadya\Downloads\My-GitHub\Git_Course_Continue (master -> origin)
λ git push origin master ←
Enter passphrase for key '/c/Users/fadya/.ssh/id_rsa':
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Delta compression using up to 12 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (2/2), 243 bytes | 243.00 KiB/s, done.
Total 2 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To github.com:fadyalamir/Git_Course_Continue.git
  f55234b..1fbabc8  master -> master

C:\Users\fadya\Downloads\My-GitHub\Git_Course_Continue (master -> origin)
λ |
```

٣- git tag: أمر يستخدم لرؤية ال الموجودة في repo الخاص بك tags

```
C:\Users\fadya\Downloads\My-GitHub\Git_Course_Continue (master -> origin)
λ git tag ←

C:\Users\fadya\Downloads\My-GitHub\Git_Course_Continue (master -> origin)
λ git tag v1.0 ←

C:\Users\fadya\Downloads\My-GitHub\Git_Course_Continue (master -> origin)
λ git tag ←
v1.0

C:\Users\fadya\Downloads\My-GitHub\Git_Course_Continue (master -> origin)
λ |
```

٤- git tag [tagName]: أمر يستخدم لعمل lightweight tag لإصدار من المشروع

ملاحظة: يوصي بتسمية ال versions بهذا الشكل (v1.0 – v2.0 – v2.3)

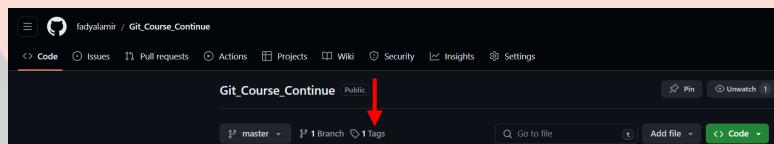


أمر يستخدم لعمل push لل tag هكذا : git push origin [tagName] -٥

```
C:\Users\fadya\Downloads\My-GitHub\Git_Course_Continue (master -> origin)
λ git push origin v1.0
Enter passphrase for key '/c/Users/fadya/.ssh/id_rsa':
Total 0 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To github.com:fadyalamir/Git_Course_Continue.git
 * [new tag]           v1.0 -> v1.0

C:\Users\fadya\Downloads\My-GitHub\Git_Course_Continue (master -> origin)
λ |
```

وعند الذهاب لل GitHub سوف نجد ال tag في ال repo



وعند الضغط على ال Tags



٢. هي Tag غير مرتبطة بال last commit و تقوم بكتابة المESSAGE الخاص بها كما تريد

```
C:\Users\fadya\Downloads\My-GitHub\Git_Course_Continue (master -> origin)
λ touch new_feature.txt ←

C:\Users\fadya\Downloads\My-GitHub\Git_Course_Continue (master -> origin)
λ git add new_feature.txt ←

C:\Users\fadya\Downloads\My-GitHub\Git_Course_Continue (master -> origin)
λ git commit -m "New Version" ←
[master 4e5d551] New Version
 1 file changed, 0 insertions(+), 0 deletions(-)
  create mode 100644 new_feature.txt

C:\Users\fadya\Downloads\My-GitHub\Git_Course_Continue (master -> origin)
λ git push origin master ←
Enter passphrase for key '/c/Users/fadya/.ssh/id_rsa':
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Delta compression using up to 12 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (2/2), 243 bytes | 243.00 KiB/s, done.
Total 2 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To github.com:fadyalamir/Git_Course_Continue.git
 1fbabc8..4e5d551  master -> master

C:\Users\fadya\Downloads\My-GitHub\Git_Course_Continue (master -> origin)
λ |
```



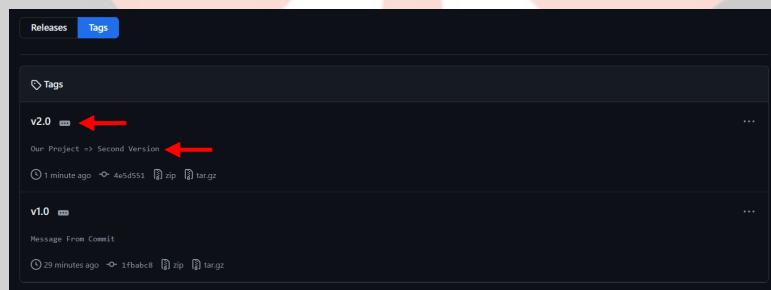
أمر يستخدم لعمل tag في repo لإصدار من المشروع annotated tag

```
C:\Users\fadya\Downloads\My-GitHub\Git_Course_Continue (master -> origin)
λ git tag -a v2.0 -m "Our Project => Second Version" ↗

C:\Users\fadya\Downloads\My-GitHub\Git_Course_Continue (master -> origin)
λ git push origin v2.0 ↗
Enter passphrase for key '/c/Users/fadya/.ssh/id_rsa':
Enumerating objects: 1, done.
Counting objects: 100% (1/1), done.
Writing objects: 100% (1/1), 180 bytes | 180.00 KiB/s, done.
Total 1 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To github.com:fadyalamir/Git_Course_Continue.git
 * [new tag]           v2.0 -> v2.0

C:\Users\fadya\Downloads\My-GitHub\Git_Course_Continue (master -> origin)
λ |
```

وعند الذهاب لل GitHub repo سوف نجد ال tag في repo



وعند الضغط على الإصدار سوف نجد ملفات إصدار المشروع بالكامل وجميع التعديلات التي تمت على الإصدار حديثاً هكذا

```
C:\Users\fadya\Downloads\My-GitHub\Git_Course_Continue (master -> origin)
λ touch new.txt ↗

C:\Users\fadya\Downloads\My-GitHub\Git_Course_Continue (master -> origin)
λ git add new.txt ↗

C:\Users\fadya\Downloads\My-GitHub\Git_Course_Continue (master -> origin)
λ git commit -m "Added New Feature" ↗
[master fdcd214] Added New Feature
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 new.txt

C:\Users\fadya\Downloads\My-GitHub\Git_Course_Continue (master -> origin)
λ git push origin master ↗
Enter passphrase for key '/c/Users/fadya/.ssh/id_rsa':
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Delta compression using up to 12 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (2/2), 236 bytes | 236.00 KiB/s, done.
Total 2 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To github.com:fadyalamir/Git_Course_Continue.git
 4e5d51..fdcd214 master -> master

C:\Users\fadya\Downloads\My-GitHub\Git_Course_Continue (master -> origin)
λ |
```



ملاحظة ٢: عند إنشاء commit جديد سوف يتم إرساله لجميع tags التي قمنا بإنشاءها لننا نعمل على master branch واحد وهو الـ

لذا يجب أن نعمل على المشروع في كل إصدار على branch مختلف

#19 – Tagging And Releasing Part 2

أمر يستخدم لإظهار الإصدارات الأولى من المشروع

```
C:\Users\fadya\Downloads\My-GitHub\Git_Course_Continue (master -> origin)
λ git tag -l "v1.*"
v1.0

C:\Users\fadya\Downloads\My-GitHub\Git_Course_Continue (master -> origin)
λ |
```

مثل v1.1، v1.2، v1.5، ...

من موقع repo من داخل GitHub نقوم بالضغط على Create release - ٢
الـ ٣ نقط بجانب Tag للإصدار الذي نريد عمل release له ومن ثم نضغط على زر هذا Create release

The screenshot shows the GitHub repository interface. The 'Tags' tab is selected. Two tags are listed: 'v2.0' and 'v1.0'. To the right of the 'v1.0' tag, there is a context menu with options 'Create release' and 'Delete tag'. A red arrow points to the 'Create release' option.

The screenshot shows the 'Create release' dialog box. It includes fields for 'Release name' (set to 'Our Awesome Release'), 'Description' (set to 'This is Notes For Our Awesome Release'), and a 'Tag' dropdown set to 'v2.0'. There is a checkbox labeled 'Set as a pre-release' with the note 'This release will be labeled as non-production ready'. At the bottom, there are buttons for 'Publish release' and 'Save draft'. Red arrows highlight the 'Release name' field, the 'Tag' dropdown, the 'Set as a pre-release' checkbox, and the 'Publish release' button.



وبعد الضغط على زر Publish release سوف يظهر الـ release للإصدار الثاني هكذا

ملحوظة: بداخل الـ Releases داخل كل اصدارات يتم وضع كل ما حدث في هذا الاصدار من تعديلات وجميع الـ commits والـ features التي تم اضافتها والمشكلات التي تم حلها والـ Documentation وبعض الملفات للتوضيح عن المشروع

ومن الممكن المقارنة بين الاصدارات المختلفة للمشروع من زر الـ Compare

وبعدما قمنا بعمل Release للإصدار سوف نجد الـ Notes اسفل الاصدار الثاني في تبويب الـ Tags



وعند انشاء الـ Releases يكونوا هكذا في تبويب الـ Releases بتوقيت انشاءهم والملفات واسم الـ Release ووصفه هكذا

The screenshot shows the GitHub repository interface for 'fadyalamir/Git_Course_Continue'. The 'Releases' tab is selected. Two releases are listed:

- Our Awesome Release** (Latest): Published 28 minutes ago by fadyalamir. It contains two assets: 'Source code (zip)' and 'Source code (tar.gz)', both uploaded 2 days ago.
- First Try**: Published 12 minutes ago by fadyalamir. It contains two assets: 'Source code (zip)' and 'Source code (tar.gz)', both uploaded 2 days ago.

ملاحظة ٢: عند عمل Release لاصدار من المشروع يوجد اختيار لكي تجعله الـ latest release اي الاصدار الاحدث هكذا



ملاحظة ٣: عندما نقوم بعمل اي تعديل علي المشروع سوف يتم تعديلها في جميع الـ Releases التي تم انشائها في الـ repo لإننا نعمل علي branch واحد

Draft a new release -٣: تشبه المسودة أي عمل نسخة من المشروع لن يتم اطلاقها في الوقت الحالي ولكن نعمل عليها حتى ننتهي من التعديل ثم نقوم بعمل release لها

The screenshot shows the GitHub repository interface for 'fadyalamir/Git_Course_Continue'. The 'Releases' tab is selected. A red arrow points to the 'Draft a new release' button located at the top right of the page.



ويمكن حذف الـ Release من زر الـ Delete هكذا

The first screenshot shows the GitHub interface for a release named 'Our Awesome Release'. A red arrow points to the delete icon (a trash can) in the top right corner of the release card. The second screenshot shows a modal dialog box titled 'Delete this release?' with a red arrow pointing to the 'Delete this release' button.

ويمكن حذف الـ Tags هكذا

The screenshot shows the GitHub interface for tags. A red arrow points to the 'Delete tag' button in the dropdown menu next to a tag entry labeled 'v1.0'.

ملحوظة: بعد حذف الـ Tags من الـ GitHub سوف تكون موجودة ايضاً **locally** لذا يجب حذفها من الـ command line هكذا

أمر يستخدم لحذف الـ Tag من الـ local repo: `git tag -d [tagName]`

```
C:\Users\fadya\Downloads\My-GitHub\Git_Course_Continue (master -> origin)
λ git tag -d v1.0 ←
Deleted tag 'v1.0' (was 1fbabc8)

C:\Users\fadya\Downloads\My-GitHub\Git_Course_Continue (master -> origin)
λ git tag -d v2.0 ←
Deleted tag 'v2.0' (was 83ab0c5)

C:\Users\fadya\Downloads\My-GitHub\Git_Course_Continue (master -> origin)
λ git tag ←

C:\Users\fadya\Downloads\My-GitHub\Git_Course_Continue (master -> origin)
λ |
```

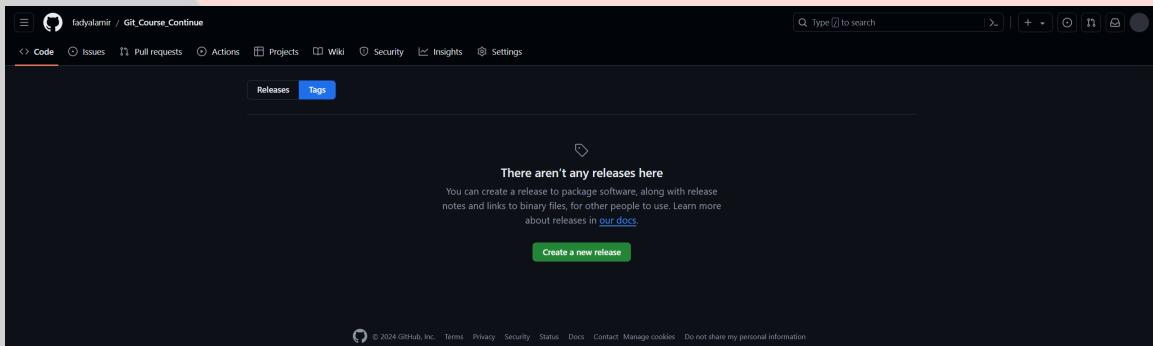


أمر يستخدم لحذف الـ Tag من remote repo -
git push origin --delete [tagName]

```
C:\Users\fadya\Downloads\My-GitHub\Git_Course_Continue (master -> origin)
λ git push origin --delete v2.0
Enter passphrase for key '/c/Users/fadya/.ssh/id_rsa':
To github.com:fadyalamir/Git_Course_Continue.git
  - [deleted]           v2.0

C:\Users\fadya\Downloads\My-GitHub\Git_Course_Continue (master -> origin)
λ |
```

وعند الذهاب للـ GitHub سوف لن نجد الـ Tag التي قمنا بحذفه في تبويب الـ Tags



#20 – The End And Advices

- يجب أن تقوم بالإستعانة بالـ Git & GitHub في جميع مشاريعك حتى إن كنت مبتدء أو كان مشروع صغير لأن مهارة الـ Git & GitHub مهمة في سوق العمل
- يجب أن تتعود على استخدام الـ Git & GitHub لأنها مهارة مهمة ولا ترك استخدامهم لفترات طويلة للحفاظ على شغلك ومشاريعك