



Cloud Computing Git

المحاضرة
الثانية-الثالثة

أ/ نوال الرجوي

- يُعرّف Git بأنه نظام لإدارة النسخ، يسمح بحفظ الملفات وتتبع نسخها المختلفة سواء كانت هذه الملفات تخص شخصًا واحدًا أو عدة أشخاص مشاركين في الملفات.

- فعلى سبيل المثال: لنفترض أن لديك ملفًا يحتوي على سطر نصي، وقمت بحفظ هذا الملف في Git ثم أضفت إلى الملف سطرين آخرين وحفظته مرة أخرى، فإن Git سيسمح لك بالرجوع إلى حالة الملف الأولى عندما كان يحتوي على سطر واحد فقط وسيسمح لك أيضًا بالذهاب مرة أخرى إلى آخر تعديل، وطالما تقوم بحفظ جميع التغييرات في Git فإن جميع النسخ التي عملت عليها ستكون متاحة لك متى شئت.

- التراجع إلى أي نسخ سابقة بما يتيح تلافي أي خطأ يحدث أثناء العمل.
- تتبع جميع النسخ التي تم العمل عليها بحرية.
- يتيح Git لأفراد الفريق البرمجي العمل على وقت واحد على نفس الملفات بحيث يكون لكل فرد في الفريق نسخته.
- إمكانية خلق فروع (نسخ) من ملف العمل الأصلي والتطوير عليها باستقلالية واختبارها دون التأثير على الملف الأصلي ومن ثم دمجها معه.
- تتيح كثير من خدمات الاستضافة الشهيرة كـ [Heroku](#) و [Netlify](#) ربط مواقع الويب فيها بالـ Git وهو ما يعني سهولة تحديث الموقع دون الحاجة لإجراء عمليات الرفع التقليدية بواسطة الـ FTP.



5

مميزات

نظام لا مركزي



يسمح بإنشاء
تفرعات متعددة



سريع جدا في تتبع
التعديلات



يحتوي على مجتمع كبير
من المستخدمين



عيوب

يحتاج الى وقت لفهم
الأوامر الأساسية



إدارة الدمج صعب
ومعرض للتعقيدات



الحاجة للتنظيم
الجيد



لايعتمد بكفاءة مع الملفات
الثنائية مثل الصور
والفيديوهات



GIT



استخدام GIT

6

إدارة المشاريع الشخصية

يساعد الافراد على تنظيم مشاريعهم
وحفظ نسخ من كل مرحلة

تطوير المهارات التقنية

يساهم في تطوير الأدوات من خلال التعاون المستمر



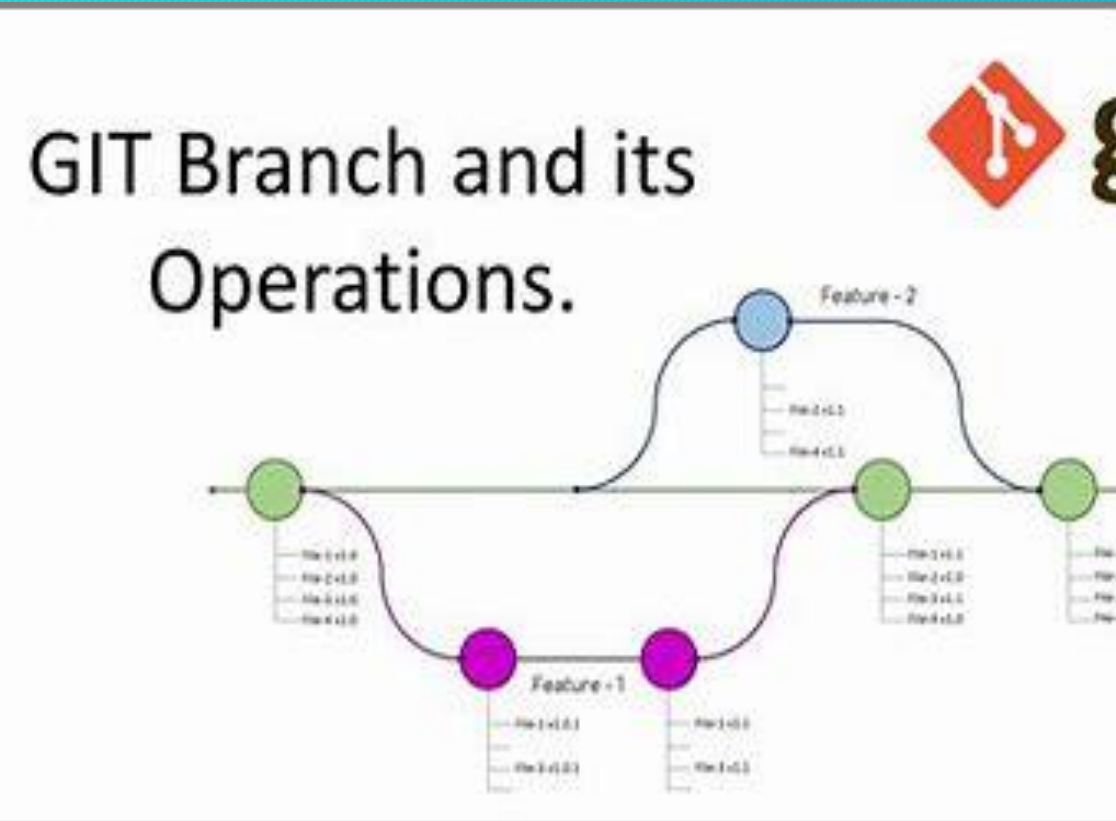
تتبع العمل الشخصي

يستخدم في تتبع العمل ومراجعة التقدم الشخصي
في المشاريع

العمل الحر

يستخدمه الافراد لتقديم مشاريعهم بمرونة
ومراجعة التعديلات بسهولة

١. **Repository** أو المستودع، ويمثل المشروع الذي يتم العمل عليه بكافة ملفاته ونسخه وفروعه المختلفة، ويُطلق عليه اختصارًا **Repo**.
٢. **Stage** الطور أو المرحلة، يتعامل **Git** مع حفظ النسخ على مرحلتين وتمثل الـ **Stage** المرحلة الأولى حيث يتم إعلام **Git** بالتغييرات التي يُراد حفظها.
٣. **Commit** أو الحفظ، وتشبه نقطة الحفظ في الألعاب **Save Point** حيث يمكن من خلالها العودة إلى نسخة معينة تم الحفظ عندها.



- 4 - **Branch** أو الفرع، وهو نسخة من محتوى المستودع الرئيسي، يتم العمل عليها بشكل منفصل تمامًا، ويمكن اعتبارها عملية نسخ للمشروع بكافة محتوياته.
- سنتطرق بالتفصيل إلى فائدة كل عنصر من العناصر السابقة وكيفية التعامل معه.

◀ Git Installation

- Go to this link to install git for windows
<https://git-scm.com/download/win>
- Follow the installation in instructions to start using Git
- Make sure to download the latest version of Git .
- Choose what is suitable for your device : 64 or 32 bit .



◀ Steps to Start using Git



1

Install
Git

2
Choose
any
simple
project



3

Upload it

4

Use Git
commands
on it



التعامل مع أوامر Git

11

- Git - اظهر كل أوامر ال git
- Clear مسح الشاشة
- Git init انشاء مستودع
- Git touch txt1.txt انشاء ملف
- Git add txt1.txt إضافة ملف الى المستودع
- Git status حالة الملف - هل في تغييرات او لا
- Git commit --m "name change" حفظ التغييرات
- push رفع التغييرات الى مصدر خارجي (server)
- Git pull سحب التغيير من مصدر خارجي (server) الى git

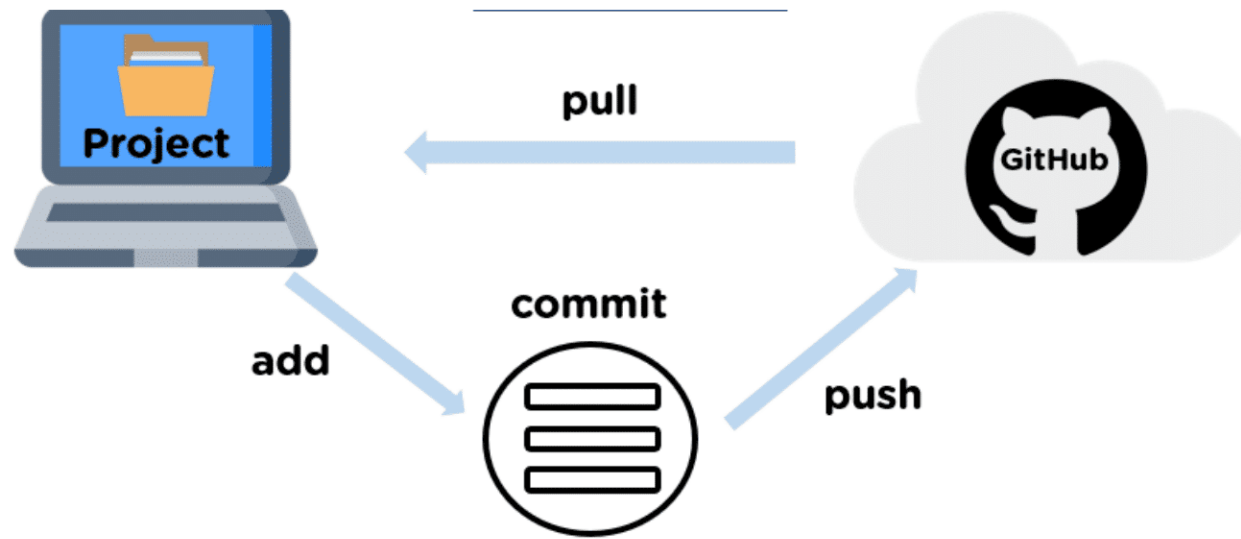
Git push

يُستخدم لإرسال (رفع) التغييرات التي قمت بها محليًا (على جهازك) إلى المستودع البعيد (Remote Repository) مثل GitHub أو GitLab أو Bitbucket.

• بمعنى آخر:

بعد أن تنشئ commits في المستودع المحلي، فأنت بحاجة إلى رفعها إلى المستودع البعيد حتى يتمكن الآخرون من رؤيتها أو حتى تحفظ نسخة احتياطية منها، وهذا يتم باستخدام `git push`

"هو أمر هام في Git يُستخدم لجلب (استرداد) التحديثات من الفرع البعيد remote branch ودمجها تلقائيًا مع الفرع الحالي. عند العمل مع فرق teams- أو تعاون الفرق في مشروع واحد على المستودع، فإن git pull يسمح للمطورين بالحصول على التحديثات الأخيرة من المستودع البعيد وجلبها إلى الفرع الحالي.



git pull

- يستخدم لجلب التحديثات جديدة من مستودع على GitHub ودمجها مع الفرع المحلي الحالي.

الفرق بين git pull و git push

أمر git push

- الوظيفة: رفع (إرسال) التغييرات من المستودع المحلي (على جهازك) إلى المستودع البعيد (GitHub)
- كأنك تقول "يا GitHub، هذا آخر كود عندي، احفظه عندك".
- يُستخدم بعد:

١. git add إضافة الملفات

٢. git commit حفظ التغييرات محليًا

٣. مثال:

• git push origin main

👉 يرفع التغييرات إلى فرع main على GitHub

الفرق بين git pull و git push

16

أمر git pull

. الوظيفة: جلب (استقبال) التغييرات من المستودع البعيد (GitHub) إلى المستودع المحلي على جهازك كأنك تقول "يا GitHub، أعطني آخر نسخة من الكود اللي عندك".

. يقوم بعمليتين:

◦ git fetch جلب التغييرات

دمجها مع كودك المحلي git merge .

مثال: .

. git pull origin main

👉 يسحب آخر التغييرات من فرع main على GitHub ودمجها مع الكود عندك.

التطبيق العملي

□ تطبيق الاوامر الاساسيه في ال git

- تسجيل حساب
- انشاء مستودع
- اضافه الملف للمستودع
- حفظ التغيرات للملف

□ تطبيق مفهوم ال Branch

□ تطبيق عملية Pull-push

بتسجيل حساب في ال git

18

Specify username and email and main branch

```
MINGW64:/c/Users/xenon

xenon@A1-Anowd MINGW64 ~
$ git config --global user.name "alanowdfawaz"

xenon@A1-Anowd MINGW64 ~
$ git config --global user.email "alanowdfawas123@gmail.com"

xenon@A1-Anowd MINGW64 ~
$ git config --global init.default branch main

xenon@A1-Anowd MINGW64 ~
$ |
```

بتسجيل حساب في ال git

19

- git config --global user.name "NawalAlragwi"
- git config --global user.email "nawaldreem2020@gmail.com"
- git config --global init.default branch main


- للتأكد مما إذا كان اسم المستخدم البريد الإلكتروني قد تمت إضافتهما أم لا، استخدم هذا الأمر

git config -list

```
lenovo@DESKTOP-J07HM2E MINGW64 ~/desktop/exp (master)
$ git config --list
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
user.name=riadbokirh
user.email=riadbokirh2002@gmail.com
init.default=branch
init.default=branch
core.repositoryformatversion=0
core.filemode=false
core.bare=false
core.logallrefupdates=true
core.symlinks=false
core.ignorecase=true
init.default=branch
```

تحديد مسار الملف

21

 MINGW64:/c/Users/xenon/desktop/e

```
xenon@A1-Anowd MINGW64 ~  
$ cd desktop
```

```
xenon@A1-Anowd MINGW64 ~/desktop  
$ cd e
```

```
xenon@A1-Anowd MINGW64 ~/desktop/e  
$
```

Go to your project file location

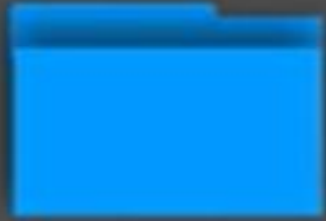
Create a git repository

git init

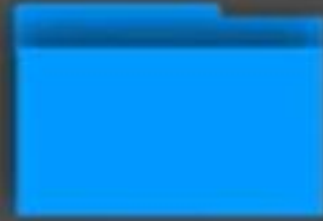
• التهيئة

- تعني عملية التهيئة عملية إنشاء المستودع أو الـ Repository وتتم العملية بكتابة الأمر **git init** في نفس مجلد المشروع الذي يتم العمل عليه، وهنا تم إنشاء مجلد باسم **e** لتنفيذ عمليات Git عليه..

project-x



file-x



file-y

`git init`

create a repository

24

```
xenon@Al-Anowd MINGW64 ~/desktop/e
$ git init
Initialized empty Git repository in C:/Users/xenon/Desktop/e/.git/

xenon@Al-Anowd MINGW64 ~/desktop/e (master)
$
```

الاسم	تاريخ التعديل	النوع	الحجم
.git	12/22/2023 03:11 م	مجلد ملفات	
contact.html	11/5/2023 01:03 م	Chrome HTML Do...	١ كيلوبايت
hi.txt	12/21/2023 11:09 ...	مستند نص	١ كيلوبايت
image.html	12/12/2023 10:53 م	Chrome HTML Do...	٢٩ كيلوبايت
index.html	10/31/2023 11:13 م	Chrome HTML Do...	١ كيلوبايت
MoveDiv.html	11/1/2023 12:19 ص	Chrome HTML Do...	٢ كيلوبايت

Git status

```
xenon@A1-Anowd MINGW64 ~/desktop/e (master)
$ git status
On branch master

No commits yet

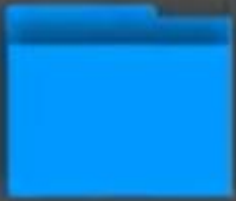
Untracked files:
  (use "git add <file>..." to include in what will be committed)
    MoveDiv.html
    contact.html
    hi.txt
    image.html
    index.html

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

xenon@A1-Anowd MINGW64 ~/desktop/e (master)
$
```

Go back git bash , write
this command to see your
branch status

Untracked



file-y

`git add file-x`



Tracked

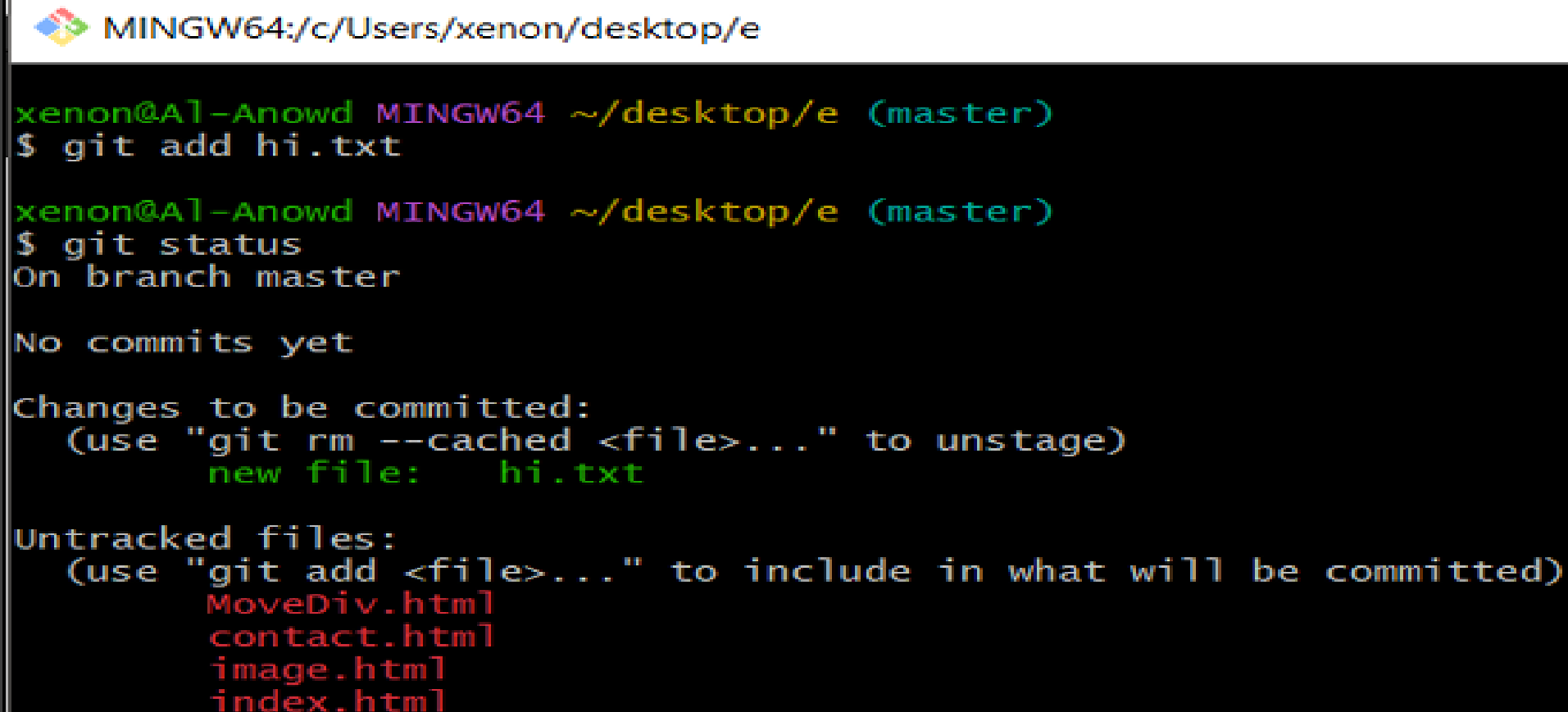


file-x

Git add

27

Write this command to track any file



```
MINGW64:/c/Users/xenon/desktop/e

xenon@Al-Anowd MINGW64 ~/desktop/e (master)
$ git add hi.txt

xenon@Al-Anowd MINGW64 ~/desktop/e (master)
$ git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
    new file:   hi.txt

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

```
Tenovo@DESKTOP-J07HM2E MINGW64 ~/desktop/proo (master)
$ git add --all
```

```
Tenovo@DESKTOP-J07HM2E MINGW64 ~/desktop/proo (master)
$ git status
On branch master
```

```
No commits yet
```

```
Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
```

```
new file:   a.html
new file:   b.css
new file:   contact.html
new file:   gitignore
new file:   image.html
new file:   index.html
```

• تابع

مجلد ملفات	٢٠٢٤/١٢/٠٩ م ١٠:١٩	.git
Microsoft Edge HT...	٢٠٢٤/١٢/٠٩ م ١٠:٠٤	a.html
... مستند ورقة أنماط	٢٠٢٤/١٢/٠٩ م ١٠:٠٤	b.css
Microsoft Edge HT...	٢٠٢٤/١٢/٠٩ م ١٠:٠٥	contact.html
ملف	٢٠٢٤/١٢/٠٩ م ١٠:١٢	gitignore
Microsoft Edge HT...	٢٠٢٤/١٢/٠٩ م ١٠:٠٦	image.html
Microsoft Edge HT...	٢٠٢٤/١٢/٠٩ م ١٠:٠٦	index.html

• نلاحظ اعلاه انه تم تتبع جميع ملفات المشروع

حفظ التغييرات للملف

```
xenon@Al-Anowd MINGW64 ~/desktop/e (master)
$ git commit -m "first commit"
[master (root-commit) fc42a12] first commit
5 files changed, 1246 insertions(+)
create mode 100644 MoveDiv.html
create mode 100644 contact.html
create mode 100644 hi.txt
create mode 100644 image.html
create mode 100644 index.html

xenon@Al-Anowd MINGW64 ~/desktop/e (master)
$
```

Unmodified

Tracked

Modified

Staged



file-x

`git commit -m "some message"`

some changes on file-x
----->

----->
`git add file-x`

◀ Commit Log

You can see all commits and all details about them using this command:

```
xenon@Al-Anowd MINGW64 ~/desktop/e (master)
$ git log
commit 78dab9e0f301f330fe4d3fcb2eaabe5c27628e98 (HEAD -> master)
Author: alanowdfawaz <alanowdfawas123@gmail.com>
Date:   Fri Dec 22 15:50:34 2023 +0300
```

updating home page

```
commit fc42a12ca8f283822c16bb1a9857c4a7854325ab
Author: alanowdfawaz <alanowdfawas123@gmail.com>
Date:   Fri Dec 22 15:36:00 2023 +0300
```

first commit

```
xenon@Al-Anowd MINGW64 ~/desktop/e (master)
$ git log --oneline
78dab9e (HEAD -> master) updating home page
fc42a12 first commit
```

```
xenon@Al-Anowd MINGW64 ~/desktop/e (master)
$ |
```


تابع العملي

```
lenovo@DESKTOP-JO7HM2E MINGW64 ~/desktop/proo (master)
$ git status
On branch master
Changes not staged for commit:
  (use "git add/rm <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        deleted:    index.html

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

lenovo@DESKTOP-JO7HM2E MINGW64 ~/desktop/proo (master)
$ git restore index.html

lenovo@DESKTOP-JO7HM2E MINGW64 ~/desktop/proo (master)
$ git status
On branch master
nothing to commit, working tree clean
```

- وإذا تم حذف ملف من المشروع نقوم باستعادته باستخدام الأمر :
git restore
-

تابع العملي

```
lenovo@DESKTOP-JO7HM2E MINGW64 ~/desktop/proo (master)
$ git diff
diff --git a/a.html b/a.html
index e69de29..aedef6cd 100644
--- a/a.html
+++ b/a.html
@@ -0,0 +1,3 @@
+<html>
+  <h1>"fgfdsffgxfxdf"</h1>
+</html>
\ No newline at end of file

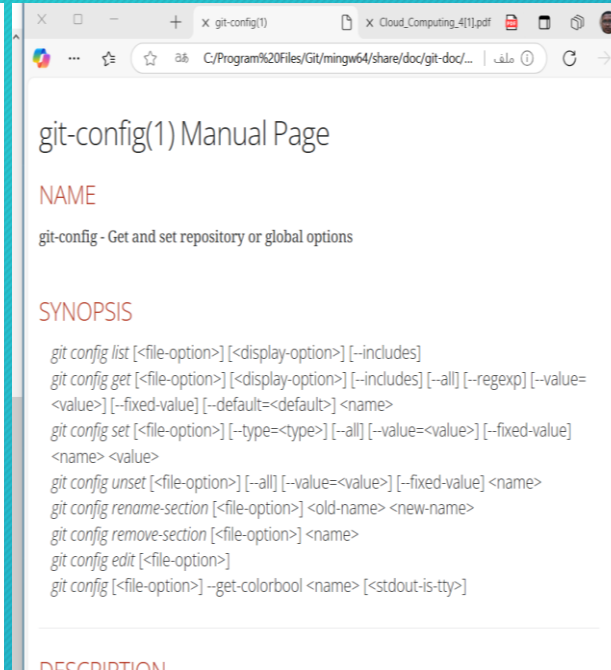
lenovo@DESKTOP-JO7HM2E MINGW64 ~/desktop/proo (master)
$
```

اظهار التغيرات :

تريد معرفة ما تم تغييره او اضافته الى مشروعك (من خلال الامر (git diff):

- للحصول على مساعدة من git
- يظهر رابط للانتقال الى المتصفح
- سيقوم هذا الأمر بمسح screen :

```
xenon@A1-Anowd MINGW64 ~  
$ git config -h
```



git-config(1) Manual Page

NAME

git-config - Get and set repository or global options

SYNOPSIS

git config list [*<file-option>*] [*<display-option>*] [*--includes*]
git config get [*<file-option>*] [*<display-option>*] [*--includes*] [*--all*] [*--regex*] [*--value=<value>*] [*--fixed-value*] [*--default=<default>*] *<name>*
git config set [*<file-option>*] [*--type=<type>*] [*--all*] [*--value=<value>*] [*--fixed-value*] *<name>* *<value>*
git config unset [*<file-option>*] [*--all*] [*--value=<value>*] [*--fixed-value*] *<name>*
git config rename-section [*<file-option>*] *<old-name>* *<new-name>*
git config remove-section [*<file-option>*] *<name>*
git config edit [*<file-option>*]
git config [*<file-option>*] *--get-colorbool* *<name>* [*<stdout-is-ty>*]

DESCRIPTION

```
xenon@A1-Anowd MINGW64 ~  
$ git help config
```

```
xenon@A1-Anowd MINGW64 ~  
$ clear
```


إنشاء فرع آخر:

```
Tenovo@DESKTOP-J07HM2E MINGW64 ~/desktop/proo (master)
$ git branch tests
```

```
Tenovo@DESKTOP-J07HM2E MINGW64 ~/desktop/proo (master)
$ git branch
* master
tests
```

```
Tenovo@DESKTOP-J07HM2E MINGW64 ~/desktop/proo (master)
$ git switch tests
D      index.html
A      riad.html
Switched to branch 'tests'
```

```
Tenovo@DESKTOP-J07HM2E MINGW64 ~/desktop/proo (tests)
$ git branch
master
* tests
```

١. يمكنك إنشاء فرع آخر لإجراء بعض التغييرات عليه
٢. ثم يمكنك دمج جميع التغييرات على الفرع الرئيسي
٣. استخدم هذه الأوامر.
٤. إنشاء فرع جديد : `git branch test`
٥. تحقق من عدد الفروع لديك : `git branch`
٦. التبديل إلى الفرع الجديد : `git checkout <branch-name>`
٧. قم بإجراء أي تغييرات في مشروعك
٨. الالتزام : `git commit -m "رسالة الالتزام"`

تابع العملي -الفروع

- حذف و دمج الفروع :

١. الدمج في الفرع الرئيسي

٢. يمكنك حذف الفرع الجديد بعد الانتهاء من استخدامه

```
lenovo@DESKTOP-JO7HM2E MINGW64 ~/desktop/proo (tests)
$ git merge -m "merge back too master" tests
Already up to date.
```

```
lenovo@DESKTOP-JO7HM2E MINGW64 ~/Desktop/proo (master)
$ git branch -d tests
Deleted branch tests (was 0c1326d).
```

رفع الملف على مستودع في git hub

5. الآن ، ستقوم بإضافة الملفات التي نريد رفعها في المستودع الذي قمنا بإنشائه. الرجاء استبدال عنوان url في الأمر التالي بمستودع Github الذي قمت بإنشائه على Github.

```
git remote add origin https://github.com/username/new-repo.git
```

```
$ git remote add origin https://github.com/imintweb/new-repo.git  
outs7@DESKTOP-NESGORI MINGW64 ~/github/Test folder (master)
```


سنقوم الآن برفع المجلد بداخل المستودع الذي أنشأناه.

```
git commit -m "أسم الملف"
```

```
outs7@DESKTOP-NESG0RI MINGW64 ~/github/Test folder (master)
$ git commit -m "Test folder"
[master (root-commit) acce9da] Test folder
1 file changed, 14 insertions(+)
create mode 100644 index.html.html
```

بعد ذلك ، نقوم بإضافة أمر رفع الملفات ، عن طريق الكتابة.

```
git push -u origin master
```

```
outs7@DESKTOP-NESGORI MINGW64 ~/github/Test folder (master)
$ git push -u origin master
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Delta compression using up to 2 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 392 bytes | 196.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0)
To https://github.com/imintweb/new-repo.git
 * [new branch]      master -> master
Branch 'master' set up to track remote branch 'master' from 'origin'.
```

خطوات رفع ملف موجود على الجهاز مرتبط بال git الى git hub

- `git remote add origin https://github.com/Nawa1Alragwi/w2.git`
- `$ git branch -M main`
- `git push -u origin main`
- -----
- `PS C:\Users\USERW\OneDrive\Desktop\pro2> git add html1.html`
- `PS C:\Users\USERW\OneDrive\Desktop\pro2> git commit -m "fff"`
- `[main 7a926f2] fff`
- 1 file changed, 4 insertions(+), 1 deletion(-)
- `PS C:\Users\USERW\OneDrive\Desktop\pro2> git push origin main --force`

جميع اوامر عملية الرفع

```
git init
```

```
. git add
```

```
git remote add origin https://github.com/username/nama_repo.git
```

```
git commit -m "أسم الملف"
```

```
git push -u origin master --force
```

تطبيق عملي

١. قم بتسجيل حساب في ال git
٢. باستخدام ال git قم انشاء مستودع
٣. انشاء ملف من نوع TEXT يحتوى على اسمك و اضف الملف الى المستودع
٤. قم بعمل إضافة التخصص الى الملف واحفظ التغيير باسم CHANG 1
٥. قم بعمل إضافة المؤهل الى الملف واحفظ التغيير تغيير باسم CHANG 2
٦. اظهر جميع التغييرات

تطبيق عملي

- قم بتسجيل حساب في ال git باسمك cloud+
- باستخدام ال git قم انشاء مستودع
- وانشاء ملف من نوع TEXT يحتوى على اهم أوامر git (٣) واضف الملف الى المستودع
- إنشاء فرع جديد باسم cloud
- تحقق من عدد الفروع لديك
- التبديل إلى الفرع الجديد
- قم باضافة امر جديد من اوامر git الى text واحفظ التعديلات
- دمج الفرع الجديد مع الفرع الرئيسي
- اعمل لقطة شاشه كامل الخطوات المنفذه -واحفظها صورته وأرسلها للايميل
- nawaldreem2020@gmail.com