

**Learn Git and GitHub**

# WHAT IS GIT AND GITHUB?

GIT IS DISTRIBUTED  
VERSION CONTROL SYSTEM

حيث أن الـ **Projects** تتطور  
ويوجد منها **versions** لذا الـ  
**GIT** يستخدم لكي يـ **Handle**  
هذه الإصدارات يعني يتحكم  
فيها

YOU CAN USE GIT  
WITHOUT GITHUB

يمكن استخدام الـ **GIT** بدون استخدام الـ  
**GITHUB** حيث يمكن رفع الشغل علي  
سيرفر آخر ويعمل عليه التيم بالكامل عن  
طريق الـ **GIT**

GIT IS FREE AND OPEN  
SOURCE

الـ **GIT** أداة **FREE** و **PROJECT**  
**SOURCE OPEN** حيث يمكن  
مشاركته مع أدوات أخرى  
دون مشاكل

GITHUB IS SOURCE FOR  
PROJECT AND SOURCES  
[GITLAB, BITBUCKET]

الـ **GITHUB** يوجد به **SOURCE**  
جميع الـ **PROJECTS** ومن  
الممكن المشاركة به ووضع  
تعديلات والمواقع الثانية  
بتقدم نفس الخدمة

GIT HAS GUI

الـ **GIT** له واجهة يمكن استخدامها بدلا من  
**TERMINAL**

GITHUB SIMPLIFY  
USING GIT

الـ **GITHUB** يقدم تجربة استخدام  
بسيطة وسهلة حيث يوفر بعد  
الإمكانيات الجيدة مثل ن الـ  
**TRACKER BUG** والـ **TRACKER ISSUE**



ايام زمان مكنش ينفع اكثر من شخص  
يعدلوا علي نفس الملف ف نفس  
الوقت ولما يحص **error** مش بنقدر  
نحدد مين اللي عدل علي الملف او  
امتي عمل ده ومينفعش نرجع للكود  
القديم اللي قبل التعديل غير كدا مثلا  
لما نعدل **header or footer on**  
**home page** لازم اعدل علي كل  
الصفحات الفرعية هنشوف مميزات  
ال **GIT**

# WHY YOU MUST LEARN GIT?

YOU CAN COLLABORATE  
TO FIX ISSUES

YOU CAN REVERT  
CHANGES

DEVS CONTRIBUTE TO  
THE SAME PROJECT

YOU CAN COLLABORATE  
TO CREATE NEW  
FEATURES

يمكن نتعاون ف تصليح **bugs**  
واعدل فنفس الوقت

يمكن التراجع عن اي تعديلات  
ومعرفة من اللي عدل علشان ال **git**  
بيوفر ملفات **Backup** و **Versions**

كل **developer** يشاركوا ف  
مشروع واحد من غير **errors**

YOU CAN ORGANIZE  
FEATURES

ده علشان تكتب كومنت لو  
هتضيف ميزه جديده او حل  
لمشكلة

YOU CAN SOLVE  
CONFLICTS

يمكن باستخدام ال **git** حل مشكلة ال  
**Conflict** حيث يمكن مقارنة كود أحد  
المطورين بكود مطور اخر في نفس المشروع  
ومن ثم انتاج النسخة السليمة من الكود

يمكن التعاون ف اشاء مميزات  
جديده من غير محتاج ان نعمل  
ملف جديد

REPOSITORY

ده المستودع اللي بنحط فيه

ال **Project** وبنختصرها **Repo**

والتيم بيبيع التعديلات لل **Repo**

## WORDS YOU WILL HEAR

PULL [YOU PULL CHANGES FROM REMOTE REPO TO YOUR LOCAL]

نقوم بسحب التعديلات من ال

**Remote Repo** لل **Local Repo**

REMOTE REPO

ده اللي بيروحله المبرمج

للتعديل علي ال **Project** زي

**GitHub** او سيرفر الشركة

CLONE [FROM LOCAL OR REMOTE]

أي استنساخ ال **Repository** من

ال **Local Repo** أو من ال **Remote**

**Remote**

BRANCH

القسم او الفرع من ال **Project**

PULL REQUEST [TELL OTHER ABOUT YOUR CHANGES TO PULL IT FROM LOCAL TO REMOTE]

حيث يوجد شخص **Admin**

علي ال **main Repo** لل

**Project** ويراجع التعديلات

بتاع باقي التيم

LOCAL REPO

ده مستودع أكواد **Project**

الخاص بالمبرمج علي ال **PC**

COMMIT (SNAPSHOT OR CHECKPOINT IN YOUR LOCAL REPO)

ده لما بنعمل تعديل او بنضيف

ميزه او حل مشكلة بيعمل

**checkpoint** في ال **Local**

**Repo**

PUSH [UPLOAD LOCAL CHANGES TO REMOTE]

من خلاله نقوم برفع ما تم تعديله

من ال **Local Repo** لل **Remote**

**Remote**

Windows PowerShell

Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! <https://aka.ms/PSWindows>

PS C:\Users\Amar> **cd** /

PS C:\> **mkdir** firstgithub

Directory: C:\

Mode	LastWriteTime	Length	Name
----	-----	-----	----
d-----	11/23/2024 6:49 PM		firstgithub

PS C:\> **cd** firstgithub

PS C:\firstgithub> **git** clone https://github.com/Amarr15/myresume.git

Cloning into 'myresume'...

remote: Enumerating objects: 5, done.

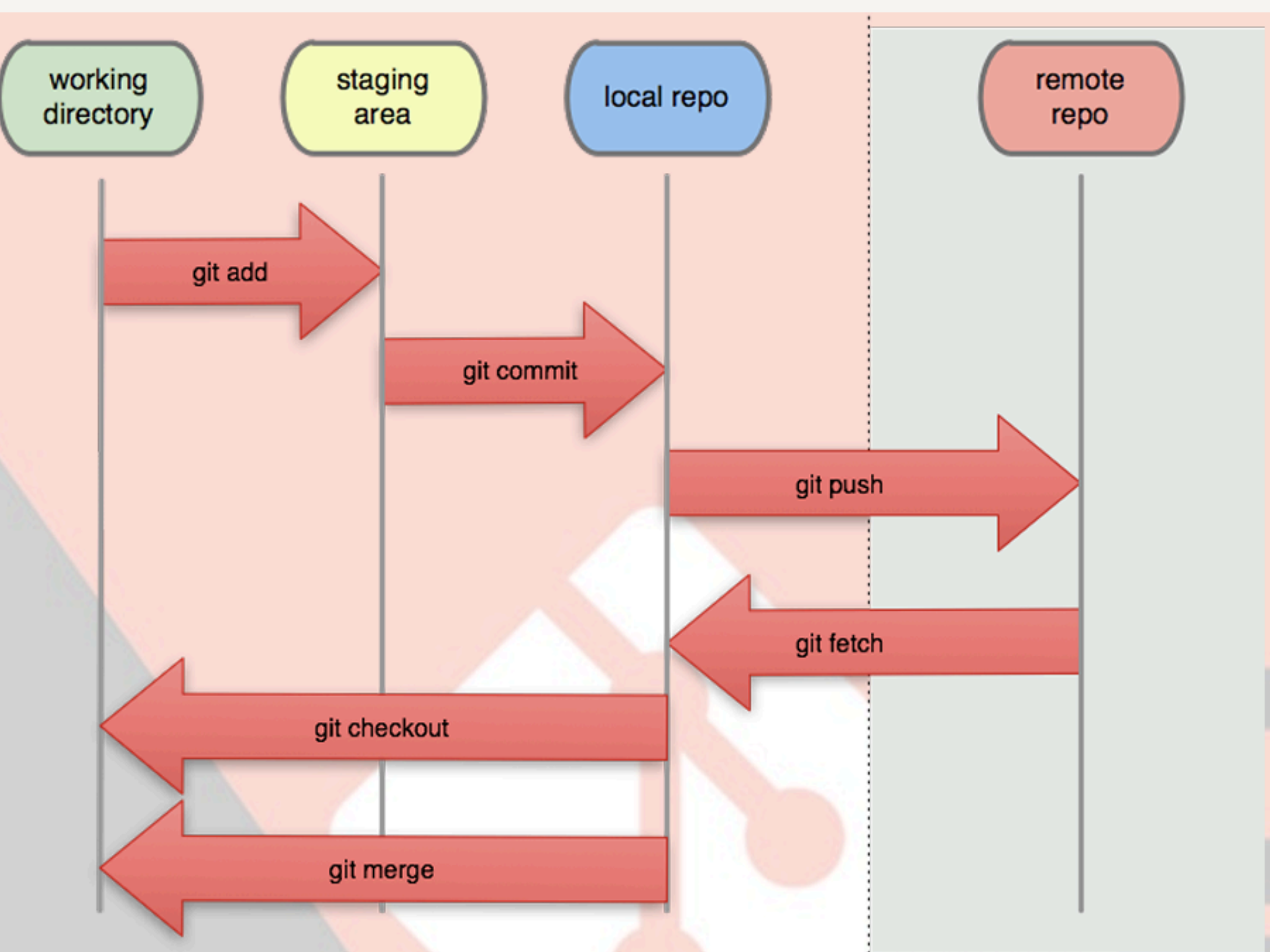
remote: Counting objects: 100% (5/5), done.

remote: Compressing objects: 100% (5/5), done.

remote: Total 5 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)

Receiving objects: 100% (5/5), 23.12 MiB | 2.96 MiB/s, done.

PS C:\firstgithub> |



*- IMPORTANT COMMANDS*

*#SHOW BRANCHES*

*GIT BRANCH*

*GIT REMOTE -V*

*#PUSH CHANGES*

*GIT PUSH ORIGIN MAIN*

*PULL CHANGES FROM A  
REMOTE REPOSITORY*



*MERGE*

*FETCH*



دمج التحديثات من  
المستودع البعيد مع فرعك  
المحلي

جلب التحديثات من المستودع  
البعيد دون دمجها مع فرعك  
الحالي



# *GIT CONFIGURATIONS*

*EX*

*DEFINITION*

**git config --global user.name "Your Name"**  
**git config --global user.email**  
**"your.email@example.com"**

تُستخدم إعدادات **Git** لتحديد كيفية عمل **Git** وكيفية التفاعل مع المستودعات. يمكن تطبيق هذه الإعدادات على مستوى النظام، أو المستخدم، أو المستودع  
**git config --global**

*SECURE SHELL (SSH)  
PUBLIC KEY*

```
graph TD; A([SECURE SHELL (SSH) PUBLIC KEY]) --> B([EX]); A --> C([DEFINITION]); B --> D[ssh-keygen -t rsa -b 4096 -C "your_email@example.com"]; C --> E[A Secure Shell (SSH) key is a cryptographic key that allows you to securely access and interact with remote systems like GitHub without needing to enter your username and password each time. Using SSH keys is a recommended way to authenticate Git operations in GitHub.]
```

*EX*

**ssh-keygen -t rsa -b 4096 -C  
"your\_email@example.com"**

*DEFINITION*

**A Secure Shell (SSH) key is a cryptographic key that allows you to securely access and interact with remote systems like GitHub without needing to enter your username and password each time. Using SSH keys is a recommended way to authenticate Git operations in GitHub.**

*ALIASES*



*EX*

**git config --global  
alias.st status**

*DEFINITION*

**The meaning of ALIAS is an assumed or additional name that a person sometimes uses; sometimes, specifically : a pseudonym (such as a pen name or stage name) that is associated with a person's professional identity**