# Introduction to GitHub

## What is GitHub?

GitHub is a cloud-based platform where developers can host, manage, and collaborate on software projects.   
It is built on top of Git, which is a powerful distributed version control system. Git tracks every change you make to your files,   
enabling you to revert, review, and coordinate work with others without losing track of history.  
  
In simple terms:  
- Git manages versions of your code on your computer.  
- GitHub stores those versions online and provides collaboration tools.

## Why should you use GitHub?

- Safeguard your work: Your code is backed up on GitHub’s servers, protecting against data loss.  
- Track changes over time: View a complete history of edits, see who changed what, and when.  
- Collaborate with others: Multiple developers can work on the same project without overwriting each other’s work.  
- Get feedback: Use issues and pull requests to discuss changes.  
- Showcase your portfolio: Recruiters and peers can see your public repositories, making it easier to demonstrate your skills.

## Core concepts explained

Repository (repo): A project folder on GitHub containing your code, files, and history. Like a directory snapshot tracked over time.  
Commit: A record of changes, with a message describing what was modified. Like a saved checkpoint.  
Branch: A parallel version of your code. Used to develop features or fix bugs without affecting the main code until you're ready.  
Merge / Pull Request (PR): A process to propose integrating changes from one branch into another, typically after review.  
Clone: Download a copy of a GitHub repository to your local machine.  
Push / Pull: Push sends your commits to GitHub. Pull retrieves changes from GitHub to your local machine.

## Getting started step-by-step

### 1. Sign up for a GitHub account

Go to https://github.com and sign up with your email. Choose a username and set up your profile.

### 2. Install Git on your computer

Download it from https://git-scm.com.  
After installation, open your terminal (or Command Prompt on Windows) and type:  
git --version  
to verify it’s installed.

### 3. Configure Git

Set your name and email (they appear in your commits):  
git config --global user.name "Your Name"  
git config --global user.email "you@example.com"

### 4. Create a repository on GitHub

Click the + in the top right → New repository.  
Name it, add a description, and decide if it’s public or private. Optionally, add a README file.

### 5. Clone the repository

Copy the HTTPS URL from your new GitHub repo page.  
In your terminal, navigate to where you want the project folder and run:  
git clone https://github.com/your-username/your-repo.git  
This creates a local copy on your computer.

### 6. Make changes locally

Edit or add files in this directory using your code editor.

### 7. Stage, commit, and push changes

Stage your changes:  
git add .  
  
Commit them with a descriptive message:  
git commit -m "Add feature X or fix bug Y"  
  
Push them to GitHub:  
git push  
  
Now your GitHub repository is updated with your new changes.

## Helpful practices

- Use .gitignore: To exclude files like logs or local config.  
- Write meaningful commit messages: So collaborators understand what was changed.  
- Open issues or pull requests: To discuss features or track bugs.  
- Regularly pull: Keep your local repo up to date with:  
git pull

## Summary

GitHub, together with Git, helps you safely store your code, manage its history, and work seamlessly with others.   
By learning these basics, you can confidently start contributing to projects or even launch your own.