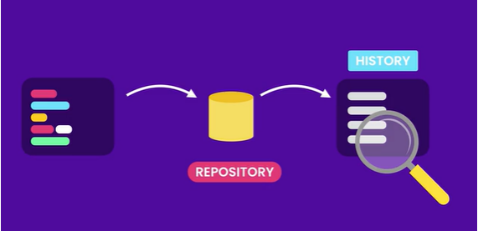
1. **Introduction:**

* Fundamental concepts
* Creating snapshots
* Browsing project history
* Branching and merging
* Collaborating using GitHub
* Rewriting history

1. **What is Git**

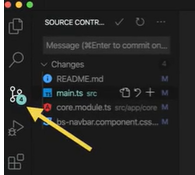
Version control system



1. Centralized : Microsoft
2. Distributed system: Git and Mercurial

Know how to track history and work together

1. Using git
2. Command line
3. Code editor and IDEs



GitLens: also extension of it, https://marketplace.visualstudio.com/items?itemName=eamodio.gitlens

1. GUI: GUI Clients

GitKraken Git GUI: <https://www.gitkraken.com/>

Sourcetree: completely free

1. Installing Git

CMD terminal

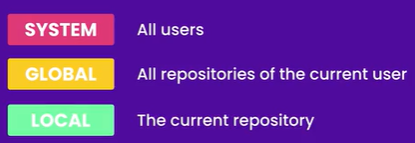
>git –version

To download git: <https://git-scm.com/downloads>

https://gitforwindows.org/

1. Configuring Git

Levels:



>git config --global user.name “Ghh84”

>git config --global user.email [ghiwethh@gmail.com](mailto:ghiwethh@gmail.com)

# To download visual studio

<https://code.visualstudio.com/>

>code // to open vs

>git config --global core.editor “code --wait” // to set default editor

>git config --global -e // to edit the config file

// configure end of line

In windos : \r (carriage return) and \n (line feed)

>git config --global core.autocrlf true //mac/linix use input

1. Getting help

Google: git configuration,

[git-config Documentation;](git-config Documentation; https://git-scm.com/docs/git-config) **[https://git-scm.com/docs/git-config](git-config Documentation; https://git-scm.com/docs/git-config)**

>git config –help // q to exit

>git config -h

// cheat sheet

> git commit -m “Refactor code.” //

> git add file1.txt

1. Seting of GitHub

**…or create a new repository on the command line**

echo "# my-project" >> README.md

git init

git add README.md

git commit -m "first commit"

git branch -M main

git remote add origin https://github.com/Ghh84/my-project.git

git push -u origin main

git remote add origin-push

### Front-end Development

All the courses you need to build beautiful websites. HTML, CSS, JavaScript, React, and more!

https://codewithmosh.com/p/front-end

### Back-end Development

All the courses you need to build powerful APIs for web and mobile apps. Node, Django, ASP.NET MVC, MySQL, and more!

### <https://codewithmosh.com/p/back-end>

### Mobile Development

All the courses you need to build professional, cross-platform mobile apps using React Native.

<https://codewithmosh.com/p/mobile-development>

1. Introduction

Creating snapshots:

1. Workflow:
2. Storing area

>> echo ocean >> app1.js

1. Initializing a Repository

We can create any where in the directory

>> mkdir Moon

>> cd Moon

>> git init

>> ls -a // list directory

// first download posh-git and install

// it is a repository power shell

>> open .git

>> rm -rf .git // to remove the repository

1. Git Work flow

|  |  |
| --- | --- |
|  |  |

1. Staging Files

>> echo add > hello file1.txt

>> git status

// to show all needs to add the fils

>> git add . // to add all files

>> git add file1.txt

1. Committing Changes

>> git commit

>> git commit -m “Initial comment”

1. Committing Best Practices
2. Skipping the Staging Area

>> git commit -am “Fix the bug that prevented the user from sign up.”

1. Removing files

Remove files from working area and staging

>> rm file2.txt

>>git status

>>git ls-files

>>git add file2.txt // check file ls and status

>> git commit -m “remove unused code.” // remove from both staging and work area

1. Renaming or moving files

>>mv file1.txt main.js

>> git add file1.txt //delete

>> git add mai.js // creating

//single command

>> git mv file1.txt main.js

//commit

>> git commit -m “Refactor code.”

1. Ignoring files

>> echo logs/ > .gitignore

>> echo logs/ > .gitignore

>> git add .gitignore // add to staging are

>>git commit -m “ ignoring files”