

Introduction To Git

Objectives

- Getting to know Git
- How to use Git
- Getting familiar with Git commands
- Interacting with remote and local repositories

What is Git?

- Git is an open source distributed version control system, which means that git is a content tracker so it can be used to store content like code for example.
- As code changes or added, even in parallel, Git tracks it.
- Git tracks the changes between two repositories, a remote one (on Github for example) and another local repository, the local repository is a copy of the remote one and when change are made to it Git can be used to reflect these changes on the remote repository and vice versa

Advantages Of Git

- Open Source
- Distributed architecture, repo has the full history of the changes
- Flexibility, works in multiple workflows
- Popular, Everyone knows it

Using Git

- Git can be used by inputting commands to the terminal.
- Terminal refers to Git bash on Windows and to terminal on Mac and Linux
- To make sure you got Git installed on the terminal and check the version use the following command `git --version`

Git Commands (Local Repository)

Git Init

- To initialise a git repository in the root folder run `git init` in the terminal
- Most of Git commands can only be used in a git repository so it would make sense to run `git init` first.

Git Status

- To check the state of the working directory run `git status`
- Git status shows the changes that has been staged or not
- It also shows the files that are not being tracked by git

Git Add

- To add the changes from the working directory to the staging area run `git add <file name>`
- An example would be `git add index.html`
- To add all the changes from the working directory run `git add .`

Git Commit

- To create a snapshot of the of the staged anges run `git commit -m "commit message"`
- An example would be `git commit -m "created index.htm"`
- After committing git state goes back to unmodified

Git Log

- To list the history of the git commits run `git log`

Git Commands (Remote Repository)

Git Config

- To set the configuration for git to work with remote repositories use the following command `git config`
- To set the username run `git config --global user.name "replace with your username"`
- To set the email run `git config --global user.email "replace with your email"`

Access Token (GitHub)

- To generate an access token from github follow this [guide](#)
- Access token is used to authenticate the user
- Make sure to save the access token since it will only be shown once

Git Clone

- To copy a remote repository locally run `git clone <repo url>`
- To copy the git repository url from Github do the following
 - Open the repository
 - Click on the drop down menu “code”
 - Pick the HTTPS tab
 - Copy the link

Git Push

- To upload the committed changes from the local repository to the remote repository run `git push <remote> <branch>`
- When working with your own repository the remote would be called `origin`
- The branch name by default will either main or master, open the github repository to check it out
- A complete example would be `git push origin master`
- When asked for the password enter the access token

Summary

- What is git
- Git advantages
- Git Commands:
 - git config
 - git clone
 - git status
 - git add
 - git commit
 - git push
- Github access token
- Local and remote repositories

Resources

- [Git Documentation](#)
- [Git cheat sheet](#)

Questions?