**Lecture:** Introduction to Git and GitHub (Part 1)

* **Teacher:** Sir Arsalan Shah
* **Typist:** Yousuf Naveed

# Git and GitHub Basics

## Git: **Version Control System**

Definition: Git is a distributed version control system used to track changes in source code and collaborate on software development projects.

## GitHub: **Online Platform for Git Repositories**

Definition: GitHub is a web-based platform that provides hosting for Git repositories, collaborative tools, and more.

## Configuration

## **Git Configuration:** Customize Git settings globally.

* `git config --global user.name "YourUserName"`
* `git config --global user.email [YourEmail@example.com](mailto:YourEmail@example.com)`
* `git config --list: List Git configuration settings.`

## Understanding Git Status

* Untracked: New files.
* Modified: Previously tracked files with changes.
* Stage (Staging Area): Files ready for commit.
* Unmodified: Files after a commit.

## Checking Status

* Short status: `git status -s`
* Full status: `git status`

## File Management and Committing

* To upload all files: `git add .` or `git add -A`
* To upload a single file: `git add yourfilename`
* Commit: `git commit`
* Type "i," write your commit message, press "Esc," then `:wq`, and press Enter.
* Commit with message: `git commit -m "your commit`"

## Adding & Committing

To write add and commit commands in short: `git commit -a -m"your commit`"

## Working Tree

Working tree: The area where you work on your files.

## Committing Changes

Committing is the process of saving your changes in Git.

## Remote Process

* Add a remote repository: `git remote add origin YourRepoURL`

**"origin"** is a name you assign; it can be customized.

* View remote URLs: `git remote -v`
* Change remote URL: `git remote set-url origin NewURL`

## Pushing to Remote Repository

* Push to the "master" branch: `git push origin master`
* Universal push: `git push -u origin master`

After the initial setup, you can use git push without specifying the branch name.

## Terminal Commands

Clear your terminal: clear

## Recovering Data

### Restore data from the last commit:

* For one file: `git checkout yourfilename`
* For all files: `git checkout -f`

## Viewing Commit History

* View commit history: `git log`
* Exit log: Type "Q"
* View a specific commit: `git log -1` (Can replace with the desired commit number)

## Git Extra Commands

### Basic File Operations:

* `touch filename`: Create a new file.
* `ls`: List the contents of the current directory.
* `ls -lart`: Detailed directory listing, including hidden files.
* `pwd`: Display the current directory.
* `cd abc/`: Navigate to another folder.
* `cd ..`: Navigate back to the parent folder.

## Differences

View differences between the working directory and the repository: `git diff`

View differences between staged and repository: `git diff –staged`

## Branches

* View existing branches: `git branch`
* Create a new branch: `git checkout -b YourBranchName`
* Switch to an existing branch: `git checkout YourAnyExistingBranchName`
* Delete a branch: `git branch -d YourAnyExistingBranchName`
* Rename a branch: `git branch -m NewName`

**YouTube Video:** <https://www.youtube.com/watch?v=s979g_WW-40>

**Primary Cheat Sheet:** <https://github.com/SyedMuhammadArsalanShah/Git-Course/tree/master/Git%2001%20Primary%20Cheat%20Sheet>

**Secondary Cheat Sheet:** <https://github.com/SyedMuhammadArsalanShah/Git-Course/tree/master/Git%2001%20Primary%20Cheat%20Sheet>