ChatGPT

PowerPoint Presentation: Introduction to Git, GitHub, and Common Git Commands

---

Slide 1: Title Slide

- Title: *Introduction to Git, GitHub, and Git Commands*
- Subtitle: *Understand the core concepts and commonly used commands in Git version control.*
- Presented by: [Your Name]

Slide 2: What is Git?

- Git is a **distributed version control system**.
- It helps track changes in source code during software development.
- Allows multiple developers to work together.
- Created by Linus Torvalds in 2005.

Slide 3: What is GitHub?

- GitHub is a **web-based platform** for hosting Git repositories.
- Provides collaboration features like issues, pull requests, wikis.
- Popular for open-source and team collaboration.
- Owned by Microsoft.

Slide 4: Git Workflow Basics

- **Working Directory**: Local files you are currently working on.
- **Staging Area (Index)**: Where you place files you want to commit.
- **Repository (Commited Area)**: Where committed changes are stored permanently.

Slide 5: Visual of Git Workflow [Include a diagram showing: Working Directory → Staging Area → Repository]

Slide 6: Git Config Setup

- **Command**:

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

- **Purpose**: Sets your Git identity globally on your system.
- Required for commit author info.

Slide 7: Git Command - git init

- **Command**: `git init`
- **Purpose**: Initializes a new Git repository.
- **Usage**:

```
git init
```

- Creates a hidden `.git` directory in your project folder.

Slide 8: Git Command - git add

- **Command**: `git add <filename>`
- **Purpose**: Adds files to the staging area.
- **Usage**:

```
git add index.html
```

- Use `git add .` to add all files.

Slide 9: Git Command - git commit -m

- **Command**: `git commit -m "message"`
- **Purpose**: Saves changes from staging to local repository.
- **Usage**:

```
git commit -m "Initial commit"
```

Slide 10: Git Command - git status

- **Command**: `git status`
- **Purpose**: Shows current status of the working directory and staging area.

- **Usage**:

```
git status
```

- Helps identify untracked, modified, or staged files.

Slide 11: Git Command - git branch -M main

- **Command**: `git branch -M main`
- **Purpose**: Renames the current branch to `main`.
- **Usage**:

```
git branch -M main
```

- `main` is now the default branch in many repositories.

Slide 12: Git Command - git remote add origin

- **Command**: `git remote add origin <repo-url>`
- **Purpose**: Links local repo to a remote GitHub repo.
- **Usage**:

```
git remote add origin https://github.com/user/repo.git
```

Slide 13: Git Command - git push -u origin main

- **Command**: `git push -u origin main`
- **Purpose**: Pushes committed changes to remote repository.
- **Usage**:

```
git push -u origin main
```

- `-u` sets `origin main` as default push/pull location.

Slide 14: Git Command - git clone

- **Command**: `git clone <repo-url>`
- **Purpose**: Creates a copy of an existing repository.

- **Usage**:

```
git clone https://github.com/user/repo.git
```

- Clones the full project with history.

Slide 15: Git Command - git log

- **Command**: `git log`
- **Purpose**: Shows a list of recent commits.
- **Usage**:

```
git log
```

- Can be customized with options like `--oneline`.

Slide 16: Git Command - git log --all

- **Command**: `git log --all`
- **Purpose**: Displays commits from all branches.
- **Usage**:

```
git log --all
```

- Useful to visualize complete commit history.

Slide 17: Git Command - git checkout

- **Command**: `git checkout <branch-name>`
- **Purpose**: Switches to another branch or commit.
- **Usage**:

```
git checkout main
```

- Can also be used to recover previous file versions.

Slide 18: Git Pull and Merge Conflicts

- **Command**: `git pull origin main`

- **Purpose**: Fetches and merges changes from remote main branch.
- **Usage**:

```
git pull origin main
```

- **Merge Conflicts**: Occur when local and remote changes conflict.
- Git will pause and require manual resolution.
- Use `git status` to see conflicted files.
- After fixing, use:

```
git add <resolved-file>
git commit
```

Slide 19: Summary

- Git is essential for version control.
- GitHub provides collaboration tools.
- Understand workflow: Working Directory → Staging → Commit.
- Know key commands for setup, commit, push, pull, and branching.

Slide 20: Thank You

- Questions?
- Contact: [Your Email / Contact Info]

End of Presentation