Master Git: Manage Your Versions and Collaborate Effectively

Module 1: Introduction to Git

Objective: Understand the usefulness of Git in version control and project development.

- Introduction to version control in the context of a static website project.
- Install and configure Git as well as Python, Pip, GitHub, GitLab, and MkDocs.
- Overview of Git and its ecosystem (GitHub, GitLab, Bitbucket, etc.).

Module 2: Working Alone with Git

Objective: Master the fundamentals of Git for an individual static website project using Python and Markdown.

- Basic commands: git init, git config.
- Track and record changes: git add, git status, git commit.
- Explore previous versions: git log, git diff.
- Manage versions: git tag.
- Publish online: git push.
- Discover MkDocs, some Python, and Markdown to run the site locally and host it online.
- Discover other concepts: branches (git branch, git checkout), a YAML file, a README.md, on GitHub and GitLab.
- Clone an online project: git clone.
- Work with command aliases.

Module 3: Working in a Team with Git

Objective: Learn how to use Git in a collaborative environment.

- Discover team types: owner and collaborator for a 2-person project, or equal collaborators.
- Manage teams and define permissions.
- Review commands and add updates to synchronize the local repository with the central repository: git fetch, git pull.
- Best practices for teamwork with Git.

Module 4: Pull Request (Merge Request)

Objective: Master the process of requesting code integration through a Pull Request.

- Introduction to Pull Requests, contributing to a project, and the code review / merge request process.
- Create and submit a Pull Request.
- Handle a conflict-free request: approve or reject the request.
- Update your repository: git fetch, git pull.
- Strategies for managing Pull Requests within a team.

Module 5: Managing Merge Conflicts

Objective: Learn how to handle and resolve version conflicts.

- Understand merge conflicts: causes and how to detect them.
- Handle a request with conflicts: proceed or reject the request.
- Resolve conflicts manually: git status, git mergetool.
- Use conflict resolution tools (VS Code, Meld, etc.).
- Best practices to avoid conflicts: use the stash git stash.
- Manage the stash: add, apply, drop, and more.

Module 6: Synchronization and Continuous Integration

Objective: Manage team integration workflows.

- Differentiate between origin and upstream repositories.
- Introduction to CI/CD concepts, continuous integration, and triangular workflows.
- Introduction to branches: git branch, git checkout -b.
- Introduction to HTTP and SSH remotes and managing SSH encryption keys.
- Discover backtracking, history, and other user-friendly tools.
- Explore the CI/CD pipeline, continuous integration, and continuous deployment.