Mercurial

Mercurial

- Mercurial is a free, distributed source control management tool.
- It is easy to use and hard to break, making it ideal for anyone working with versioned files.
- Efficiently handle projects of any size while using an intuitive interface.

Features of Mercurial

- **Distributed Architecture:** Mercurial is a distributed system where every user has a full copy of the repository and can work offline.
- **Simplicity:** The command set is more user-friendly, which makes it easier for beginners to pick up.
- **Fast:** Mercurial's implementation and data structures are optimized for speed, allowing you to generate diffs between revisions or revert to previous versions in just seconds.

- **Branching Model:**

  1.Mercurial has a simpler branching model compared to Git. It doesn’t require creating a separate directory for branches; branches are created within the same repository.

  2.Named branches and bookmarks are used to track different lines of development.

- **Platform independent**:Mercurial was developed with platform independence as a priority, which is why most of it is written in Python, with a portion in portable C for performance optimization.
- **Extensible:** Mercurial also supports extensions, which allow additional features to be added. Some of these are part of the core, while others can be enabled as needed.

Key Differences:

- **Branching and History**: Git allows more flexibility with branches and rewriting history, while Mercurial offers a more straightforward, linear development model with an emphasis on keeping history immutable.
- **Ease of Use**: Mercurial tends to be simpler and easier to use, especially for beginners, while Git provides more power and flexibility but with added complexity.
- **Community and Ecosystem**: Git has a larger ecosystem and more widespread adoption, while Mercurial is used in specific communities (e.g., Mozilla, large legacy projects).

1. Initialize a Repository

- Command: hg init

- Creates a new repository in the current directory.

2. Clone a Repository

- Command: hg clone <repository-url>

- Clones an existing repository from a remote location.

3. Check Repository Status

- Command: hg status

- Shows the status of files in the working directory (untracked, modified, etc.).

4. Add Files to the Repository

- Command: hg add <file>

- Adds a new file to be tracked by Mercurial.

5. Commit Changes

- Command: hg commit -m "Commit message"
- Commits changes with a descriptive message.

6. View Commit History

- Command: hg log
- Displays a history of commits made in the repository.

7. View Differences (Diffs)

- Command: hg diff

- Shows the differences between the working directory and the last commit.

8. Revert Changes

- Command: hg revert <filename>

- Reverts changes in a file to the last committed state.

9. Update to a Specific Revision

- Command: hg update -r <revision-number>

- Updates the working directory to a specific revision.

10. Branching

- Command: hg branch <branch-name>

- Creates or switches to a branch.

11. Merging Branches

- Command: hg merge <branch-name>

- Merges changes from another branch into the current branch.

12. Push Changes

- Command: hg push
- Pushes local commits to a remote repository.

13. Pull Changes

- Command: hg pull

- Pulls changes from a remote repository to your local repository.

14. Resolve Conflicts

- Command: hg resolve <file>

- Resolves merge conflicts in a specific file.

15. Create a Tag

- Command: hg tag <tag-name>
- To create a tag for a specific revision

- To view all tags:
- Command: hg tags

16. Remove a File from the Repository

- Command: hg remove <file>
- To remove a file from the repository

17. Undo a Commit (Roll Back)

- Command: hg rollback

- This can only undo the most recent commit, so use with caution.

18. Shelve Changes

- Command: hg shelve
- To temporarily set aside uncommitted changes (save them for later).

- Command: hg unshelve
- To restore shelved changes

19. Merge Tool

- Command: hg merge --tool <tool-name>

- If a merge conflict occurs, Mercurial can launch a merge tool to help resolve conflicts.

20. List All Branches

- Command: hg branches

- To list all branches in the repository.

21. Abort an Ongoing Merge

- Command: hg merge -- abort
- To abort an in-progress merge.