Mercurial Cheat Sheet

A comprehensive guide to Mercurial, covering essential commands for version control, branching, merging, and collaboration.

Basic Commands

Repository Operations

`hg init [repository]`	Create a new repository in the specified directory. If no directory is specified, it initializes one in the current directory. **Example:** `hg init myrepo`
`hg clone [source] [destination]`	Clone an existing repository from a source (local or remote) to a destination. **Example:** `hg clone https://example.com/repo mylocalrepo`

Working Directory Commands

`hg add [file(s)]`	Add the specified file(s) to the repository for tracking.
	Example:
	`hg add myfile.txt`
	`hg add *.txt`

`hg remove [file(s)]`	Remove the specified file(s) from the repository. The files are removed from the next commit, but are still present in your working directory until you delete them. **Example:** `hg remove myfile.txt`
`hg status`	Show the status of files in the working directory (modified, added, removed, unknown, ignored). **Example:** `hg status`
`hg revert [file(s)]`	Revert the specified file(s) to the last committed version. **Example:** `hg revert myfile.txt`

Commit Operations

`hg commit -m "[message]"`	Commit the changes in the working directory with a descriptive message. **Example:** `hg commit -m "Fixed a bug in the login page"`
`hg log`	Show the commit history of the repository. **Example:** `hg log`
`hg diff [file(s)]`	Show the differences between the working directory and the last committed version. **Example:** `hg diff myfile.txt`

Branching and Merging

Branch Management

`hg branch [name]`	Create a new named branch. **Example:** `hg branch feature-x`
`hg branches`	List existing branches. **Example:** `hg branches`
`hg update [branch]`	Switch to a different branch. **Example:** `hg update feature-x`
`hg merge [branch]`	Merge changes from a specified branch into the current branch. **Example:** `hg merge feature-x`
`hg resolve [file(s)]`	Mark conflicts as resolved after a merge. **Example:** `hg resolve -m myfile.txt`
`hg commit -m "Merged branch feature-x"`	Commit the merge after resolving conflicts. **Example:** `hg commit -m "Merged branch feature-x"`

Named Branches vs. Bookmarks

Mercurial offers both named branches and bookmarks for managing concurrent development. Named branches are permanent and shared with other repositories, while bookmarks are local and lightweight, ideal for tracking experimental changes.

Bookmarks

`hg bookmark [name]`	Create a new bookmark at the current revision. **Example:** `hg bookmark my-experiment`
`hg bookmarks`	List existing bookmarks. **Example:** `hg bookmarks`
`hg update [bookmark]`	Switch to a different bookmark. **Example:** `hg update my-experiment`
`hg bookmark -d [bookmark]`	Delete a bookmark. **Example:** `hg bookmark -d my-experiment`

Remote Repositories

Synchronization

`hg pull [source]`	Pull changes from a remote repository to the local repository. **Example:** `hg pull https://example.com/repo`
`hg push [destination]`	Push changes from the local repository to a remote repository. **Example:** `hg push https://example.com/repo`
`hg incoming`	Show incoming changes from the default remote repository without applying them. **Example:** `hg incoming`
`hg outgoing`	Show outgoing changes to the default remote repository. **Example:** `hg outgoing`

Configuration

Remote repository URLs can be configured in the `.hg/hgrc` file for easier access. This allows you to use aliases instead of full URLs.

```
[paths]
default = https://example.com/repo
origin = ssh://hg@bitbucket.org/user/repo
```

Example Workflow

A common workflow involves pulling changes, updating the working directory, making changes, committing them, and then pushing the changes back to the remote repository.

```
hg pull && hg update && ... (make changes) ... && hg commit -m "Your message" && hg push
```

Advanced Features

Ignoring Files

`.hgignore`	Create a `.hgignore` file in the root of your repository to specify files and patterns that should be ignored by Mercurial. **Example:** `syntax: glob *.log tmp/*`
Syntax	`.hgignore` supports glob and regexp syntax. Use `syntax: glob` or `syntax: regexp` at the top of the file to specify the syntax. Glob is the default.

Revisions

`hg update -r [revision]`	Update to a specific revision number or tag.
	Example:
	`hg update -r 123`
	`hg update -r mytag`

`hg diff -r [rev1] -r [rev2]`	Show the differences between two revisions. **Example:** `hg diff -r 10 -r 20`
`hg revert -r [revision] [file(s)]`	Revert specific files to a specific revision. **Example:** `hg revert -r 5 myfile.txt`

Shelving

`hg shelve`	Shelve the current changes in the working directory. **Example:** `hg shelve`
`hg unshelve`	Unshelve the shelved changes and apply them to the working directory. **Example:** `hg unshelve`
`hg shelve -l "[description]"`	Shelve with a description. **Example:** `hg shelve -l "My important changes"`