Mercurial Distributed version control

START HERE

`hg status`
What changed?

`hg push`
Share changes

`hg add <file>`
Track a new file

`hg log -G`
See history graph

`hg commit -m "<msg>"`
Save a changeset

`hg diff`
Show edits

`hg pull -u`
Get changes and update

`hg update <rev>`
Switch revision

Working dir your files → `hg commit` → **Local repo** changesets → `hg push` → **Remote** default path

CONCEPTS

Unlike git there is no staging area: commit takes all changed tracked files, or just the files you name.

hg pull only fetches; hg pull -u also updates the working directory (or run hg update after).

Phases (draft, public) mark which changesets are safe to rewrite: public ones are not.

SETUP

`hg init`
Create a repository here

`hg clone <url> [<dir>]`
Copy a remote repository

`hg config --edit`
Open your user hgrc

`hg config ui.username`
Show the configured author

`hg paths`
List remote aliases

`hg help <command>`
Help for one command

`hg help -e <ext>`
Help for an extension

SNAPSHOT

`hg status`
List changed files

`hg add [<file>]`
Track new files

`hg commit -m "<msg>"`
Commit all changes

`hg commit -m "<msg>" <file>`
Commit only some files

`hg commit -i`
Pick hunks interactively

`hg commit --amend`
Fold changes into the last commit
Only for draft changesets that have not been pushed.

`hg diff`
Show uncommitted changes

`hg diff -c <rev>`
Show what one changeset did

`hg diff -r <a> -r <b>`
Compare two revisions

`hg addremove`
Add new, forget missing files

`hg remove <file>`
Delete and untrack a file

`hg forget <file>`
Untrack but keep the file

SNAPSHOT (cont.)

`hg rename <old> <new>`
Move or rename a tracked file

`hg copy <src> <dst>`
Copy and keep history

`hg revert <file>`
Discard edits to a file
Leaves a .orig backup unless --no-backup is given.

`hg revert --all`
Discard all uncommitted edits

`hg purge`
Remove untracked files
Needs the purge extension enabled.

HISTORY

`hg log`
Show revision history

`hg log -G`
Show history as a graph

`hg log -l <n>`
Limit to n entries

`hg log <file>`
History of one file

`hg log -r "<revset>"`
Select with a revset

`hg log -k <text>`
Search commit messages

`hg annotate <file>`
Who changed each line

`hg cat -r <rev> <file>`
Show a file at a revision

`hg summary`
Summarize working directory

`hg identify`
Current revision and branch

`hg heads`
List branch heads

`hg grep <pattern>`
Search tracked files

`hg bisect --bad`
Mark the revision as bad

`hg bisect --good <rev>`
Mark a known good revision

REMOTES

`hg pull`
Fetch changesets only

`hg pull -u`
Fetch and update

`hg push`
Send changesets to default

`hg push -r <rev>`
Push up to one revision

`hg incoming`
Preview what a pull brings

`hg outgoing`
Preview what a push sends

REMOTES (cont.)

`hg push --new-branch`
Allow pushing a new branch

`hg serve`
Serve the repo over HTTP

UPDATE AND BRANCHES

`hg update <rev>`
Switch the working dir

`hg update -C <rev>`
Switch, discarding local edits

`hg branch`
Show the current branch name

`hg branch <name>`
Name a new branch
Named branches are permanent in history; bookmarks are lighter.

`hg branches`
List named branches

`hg bookmark <name>`
Create a bookmark here

`hg bookmarks`
List bookmarks

`hg update <bookmark>`
Switch to a bookmark

`hg bookmark -d <name>`
Delete a bookmark

`hg push -B <bookmark>`
Push a bookmark

`hg tag <name>`
Tag the current revision

`hg tags`
List tags

MERGE AND UNDO

`hg merge`
Merge the other head into working dir

`hg merge <rev>`
Merge a specific revision

`hg resolve --list`
List files and merge state

`hg resolve --mark <file>`
Mark a conflict as resolved

`hg resolve --all --tool <tool>`
Re-run merge on conflicts

`hg commit`
Finish the merge with a commit

`hg graft <rev>`
Copy a changeset here

`hg backout <rev>`
Undo a changeset with a new one

`hg shelve`
Set aside local changes

`hg unshelve`
Restore shelved changes

MERGE AND UNDO (cont.)

`hg abort`
Abort an unfinished operation

`hg continue`
Resume an interrupted operation

REWRITE HISTORY

`hg rebase -d <dest>`
Move current branch onto dest
Needs the rebase extension enabled.

`hg rebase -s <src> -d <dest>`
Rebase a subtree

`hg histedit <rev>`
Edit, fold or drop commits
Needs the histedit extension enabled.

CONFIG AND EXTENSIONS

Revset shortcuts

Revset	Meaning
`tip`	newest changeset
`.`	working dir parent
`draft()`	not yet public
`a::b`	a to b with ancestors
`branch(<name>)`	on a branch
`user(<name>)`	by an author

Common extensions

Extension	Adds
`rebase`	hg rebase
`histedit`	hg histedit
`shelve`	hg shelve
`purge`	hg purge
`evolve`	safe history rewriting

Basic ~/.hgrc

```
[ui]
username = Your Name
<you@example.com>

[extensions]
rebase =
histedit =
purge =
```

RECIPES

Daily loop

```
hg pull -u
# edit files
hg status
hg commit -m "Fix parser"
hg push
```

Feature with a bookmark

```
hg bookmark feature-x
# work and commit
hg push -B feature-x
```

Merge two heads

```
hg pull
hg heads
hg merge
hg resolve --list
hg commit -m "Merge"
```

Tidy before pushing

```
hg log -r "draft()"
hg histedit <first-draft>
hg rebase -d default
```