

Version control

or “revision control” or “source control”, part of “[software configuration management](https://en.wikipedia.org/wiki/Software_configuration_management)” (SCM)
en.wikipedia.org/wiki/Version_control

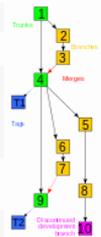
Why I use it:

- to keep track of changes as I develop a code (document, project)
- possibility to go back if things go bad
- in combination with online repository (see below), backup + availability

Much richer possibilities which I don't harvest (e.g., multi-user collaboration, branches).

Centralized (client-server) vs. ***distributed*** (peer-to-peer) source-management models

Version control software		[hide]
Years, where available, indicate the date of first stable release. Systems with names <i>in italics</i> are no longer maintained or have planned end-of-life dates.		
Local only	Free/open-source	SCCS (1972) · RCS (1982)
	Proprietary	PVCS (1985) · QVCS (1991)
Client-server	Free/open-source	CVS (1986, 1990 in C) · CVSNT (1998) · QVCS Enterprise (1998) · Subversion (2000)
	Proprietary	Software Change Manager (1970s) · Panvalet (1970s) · Endeavor (1980s) · Dimensions CM (1980s) · DSEE (1984) · Synergy (1990) · ClearCase (1992) · CMVC (1994) · Visual SourceSafe (1994) · Perforce (1995) · StarTeam (1995) · Integrity (2001) · Surround SCM (2002) · AccuRev SCM (2002) · SourceAnywhere (2003) · Vault (2003) · Team Foundation Server (2005) · Team Concert (2008)
Distributed	Free/open-source	<i>GNU arch</i> (2001) · Darcs (2002) · DCVS (2002) · ArX (2003) · Monotone (2003) · SVK (2003) · Codeville (2005) · Bazaar (2005) · Git (2005) · Mercurial (2005) · Fossil (2007) · Veracity (2010)
	Proprietary	<i>TeamWare</i> (1990s?) · Code Co-op (1997) · BitKeeper (1998) · Plastic SCM (2006)
Concepts	Branch · Fork · Changeset · Commit · Interleaved deltas · Delta compression · Data comparison · Merge · Repository · Tag · Trunk	
Category · Comparison · List		



Git (git-scm.com)

initially designed & developed by Linus Torvalds for Linux kernel development in 2005 ([wiki](https://en.wikipedia.org/wiki/Git))

Mercurial (mercurial.selenic.com)

started in 2005 ([wiki](https://en.wikipedia.org/wiki/Mercurial))

Both are command-line tools. (GUI additions on some platforms.)

Mercurial seems a bit simpler, talks less by default.

Git most used (42.9% of professional software developers as of May 2014).

[Git cheat sheet](#) (GitHub oriented)

[Mercurial cheat sheet](#), combined from [here](#) (Adrian's) and [there](#)

Markdown for READMEs etc: [Markdown cheat sheet](#), [Bitbucket Supported Markdown](#)

Software repositories

Bitbucket (bitbucket.org)

Free account allows private repositories.

Github (github.com)

Only public repos.

HTTPS or SSH access. Possible to configure RSA keys to password-less pushes/pulls.

Mercurial

```
apt-get install mercurial
```

```
vi ~/.hgrc
```

```
[ui]
```

```
username = Ondrej Sramek <ondrej.sramek@gmail.com>
```

```
hg init (in a empty directory)
```

```
creates subdirectory .hg/
```

```
create some files
```

```
hg status
```

```
hg add file1 file2
```

```
hg forget file2
```

```
hg commit -m 'some descriptive message'
```

Keep adding files, making changes to files, committing changes...

```
hg diff
```

```
hg tip
```

```
hg summary
```

```
hg log (default -r tip:0 ; last 5 commits: -l 5 ; see tree -G)
```

```
hg cp
```

```
hg mv
```

```
hg push ssh://hg@bitbucket.org/UncleDrO/jednooci
```

```
vi .hg/hgrc
```

```
[paths]
```

```
default = ssh://hg@bitbucket.org/UncleDrO/jednooci
```

```
hg incoming
```

```
hg outgoing
```

```
hg update 1
```

```
hg identify -n
```

```
hg update tip
```

Update to an older revision, make some changes, commit – which creates a new head. Now the two heads need to be merged into one while conflicts are resolved.

```
hg heads
```

```
hg resolve --list
```

```
hg resolve file1
```

```
hg merge
```

```
hg update --clean ... undo an uncommitted merge if things go bad
```

```
hg clone ssh://hg@bitbucket.org/UncleDrO/jednooci ... get another working copy, e.g., on a different machine
```

Git

```
apt-get install git
```

...

Documenting code

Sphinx (sphinx-doc.org)

Python documentation generator, also C/C++, plans to add support for other languages.
[reStructuredText](#) (*.rst) sources

Doxygen (www.doxygen.org)

C++ documentation generator, also C, Objective-C, C#, PHP, Java, Python, IDL, **Fortran**, VHDL, Tcl, and to some extent D.
Commented code + additional Markdown (*.md) sources.

MOOCs

= [Massive open online course](#)

Many providers, non-profit and commercial, usually associated with Universities, e.g.,
[Coursera](#), [edX](#), [Stanford Online](#), [Udacity](#)

You may want to enroll in a class (for free), get access to videos, materials. Even if you don't "take" the class. Quick(?) way to get exposed to a topic.