



[Hacking on Mnemosyne](#) [1]

We use [Launchpad](#) [2] to coordinate the development. It's good to first take the tour of Launchpad (on its front page) to get a feel for how this works.

When you've picked a project you want to work on, post a message on the [developer mailing list](#) [3]. Next, install the [bazaar version control tool](#) [4], get familiar with it, and create your own local copy of the Mnemosyne code by typing:

```
bzr branch lp:mnemosyne-proj
```

This will check out the main line of development. You will only need to do this once. (It's best not to do this from your home directory, but using a subdirectory, otherwise you'll mislead a feature making it easier to run Mnemosyne from an USB key.) For an overview of the different other branches, see [our Launchpad code page](#) [5].

Then, start hacking on the code to your heart's content, occasionally saving your progress by doing a bzr commit.

If you want to pull in the updates other developers make, the command to use is bzr pull.

It is important to also publish your changes to the world, so that you can get feedback on your work. To do this, sign up for an account on Launchpad, register your SSH key there, and type

```
bzr push bzr+ssh://john.doe@bazaar.launchpad.net/~john.doe/mnemosyne-proj/mybranchname
```

replacing john.doe with your own Launchpad username, and mybranchname with the name of the branch you are working on. To keep your published version up to date, you can at later times suffice with the shorter version

```
bzr push
```

It is also important to periodically pull in the changes that the other developers make on the trunk (use `bzr merge lp:mnemosyne-proj`), so that it becomes easier to merge your work later.

Once you think your work is ready for merging into the mainline, a developer will review your code, and if it's approved it will be merged.

To get an overview of how all the different bits of the library fit together, see the documentation in the code at [mnemosyne/mnemosyne/libmnemosyne/docs/build/html/index.html](#)

In order to keep the code looking uniform, please following the standard Python style guides [PEP8](#) [6] and [PEP257](#) [7].

To start working on Mnemosyne, you need at least the following software: easyinstall, virtualenv, PyQt5, matplotlib, cheroot, sphinx, webob including their



dependencies. All of these can be located by a simple Google search. Note that Mnemosyne uses Python 3.x.

You can either run a development version of Mnemosyne by using your system-wide Python installation, or by using a virtual environment with virtualenv. If your distribution provides and packages all necessary libraries in a recent enough version, using the system-wide Python install is probably easier and the recommended way.

First, install all dependencies with your distribution's package manager. Then, run `make` from the top-level mnemosyne directory. This will generate all the needed auxiliary files and start Mnemosyne with a separate datadir under `dot_mnemosyne2`. If you want to use mnemosyne interactively from within a python shell, run `python` from the top-level mnemosyne directory. You can check if the correct local version was imported by running `import mnemosyne; print(mnemosyne.__file__)`.

If your distribution does not provide all required libraries, or if the libraries are too old, create a virtual environment in the top-level directory (`virtualenv venv`), activate it (`. venv/bin/activate`) and install all the required dependencies with `pip install`. Then, follow the steps of the previous paragraph.

A separate page with more info to set up a development environment under Windows can be found [here](#) [8].

Source URL: <https://mnemosyne-proj.org/hacking-mnemosyne>

Links

- [1] <https://mnemosyne-proj.org/hacking-mnemosyne>
- [2] <https://launchpad.net>
- [3] <http://groups.google.com/group/mnemosyne-proj-devel>
- [4] <http://bazaar-vcs.org/>
- [5] <https://code.launchpad.net/mnemosyne-proj>
- [6] <http://www.python.org/dev/peps/pep-0008/>
- [7] <http://www.python.org/dev/peps/pep-0257/>
- [8] <http://mnemosyne-proj.org/mnemosyne-development-under-windows>