How to install Cadnano on macOS

Getting started

This guide describes how to set up and configure your local **macOS** desktop or laptop so you can begin designing DNA origami nanostructures with **Cadnano**. If you have a PC, please see the **Windows** version of this document. This tutorial matches how we set up our own machines in the Douglas Lab as of December 2021.

Note to users who previously installed douglaslab/cadnano2

You may have **homebrew**, **python3**, and a clone of the **cadnano2** repo installed in an existing virtual environment. If it's working fine, <u>do not modify it</u>! These instructions should allow you to install the latest PyQt6-compatible cadnano2 (v2.4.0) alongside any previous installations. (Note that we now use `venv` instead of `virtualenvwrapper`.)
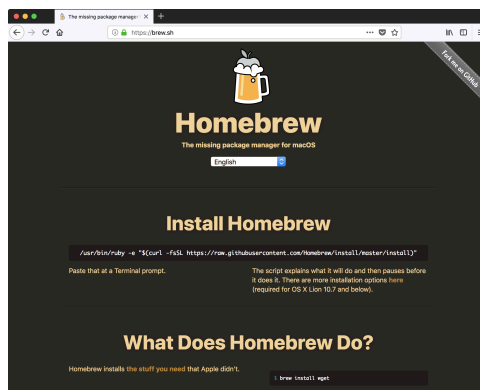

Step 1. Install Homebrew

Homebrew is a package manager for macOS that facilitates **python3** installation.

1. Open a **Terminal** (the app can be found by typing `terminal` into Spotlight search, or navigating via Finder to /Applications/Utilities/ Terminal).
2. Open a web browser and visit the URL https://brew.sh/
3. Copy & paste the "Install Homebrew" command into the **Terminal**, it will be something like:

```
/usr/bin/ruby -e "$(curl -fsSL https://
raw.githubusercontent.com/Homebrew/install/master/install)"
```

4. Press RETURN when prompted. You may need to enter your password to install **Xcode Command Line Tools** or fix file permissions.
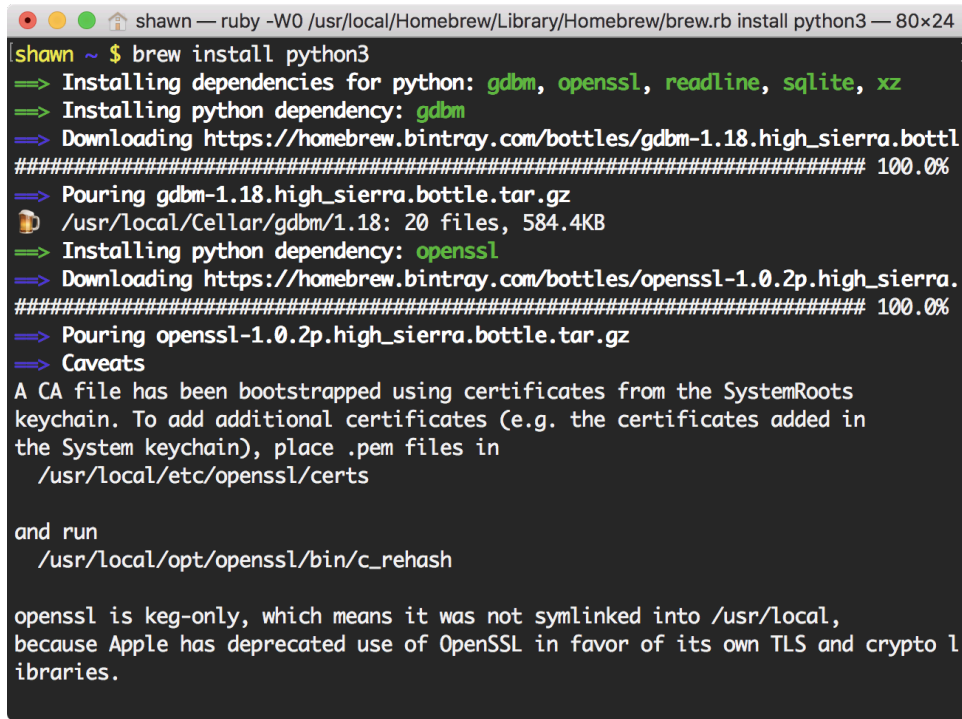
Step 3. Install Python3

1. Once **Homebrew** is successfully installed, type the following command to install python3:
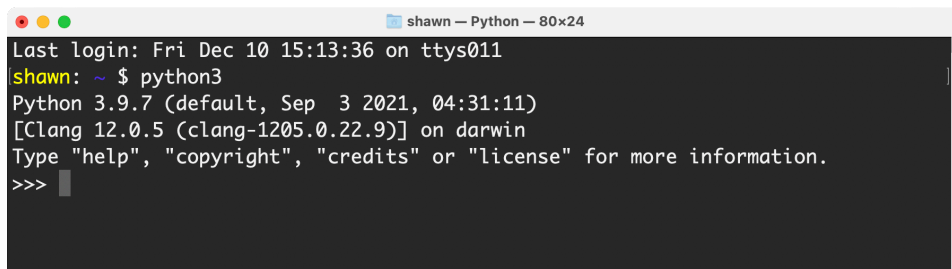
    `brew install python3`

    ```
    [shawn ~ $ brew install python3
    ==> Installing dependencies for python: gdbm, openssl, readline, sqlite, xz
    ==> Installing python dependency: gdbm
    ==> Downloading https://homebrew.bintray.com/bottles/gdbm-1.18.high_sierra.bottl
    ################################################################## 100.0%
    ==> Pouring gdbm-1.18.high_sierra.bottle.tar.gz
    🍺  /usr/local/Cellar/gdbm/1.18: 20 files, 584.4KB
    ==> Installing python dependency: openssl
    ==> Downloading https://homebrew.bintray.com/bottles/openssl-1.0.2p.high_sierra.
    ################################################################## 100.0%
    ==> Pouring openssl-1.0.2p.high_sierra.bottle.tar.gz
    ==> Caveats
    A CA file has been bootstrapped using certificates from the SystemRoots
    keychain. To add additional certificates (e.g. the certificates added in
    the System keychain), place .pem files in
      /usr/local/etc/openssl/certs

    and run
      /usr/local/opt/openssl/bin/c_rehash

    openssl is keg-only, which means it was not symlinked into /usr/local,
    because Apple has deprecated use of OpenSSL in favor of its own TLS and crypto l
    ibraries.
    ```

2. If successful, you should be able to launch **python3** from the terminal by typing:

    `python3`

    ```
    Last login: Fri Dec 10 15:13:36 on ttys011
    [shawn: ~ $ python3
    Python 3.9.7 (default, Sep  3 2021, 04:31:11)
    [Clang 12.0.5 (clang-1205.0.22.9)] on darwin
    Type "help", "copyright", "credits" or "license" for more information.
    >>>
    ```

Step 4. Set up a virtual environment

We recommend using a virtual environment to manage your Cadnano2 installation. Fortunately, python3 provides native support for virtual environments. Virtual environments simplify the process of managing multiple packages that may have incompatible dependencies. They also protect your root Python3 installation — if you make a mistake, you can simply delete the virtual environment folder and start over.
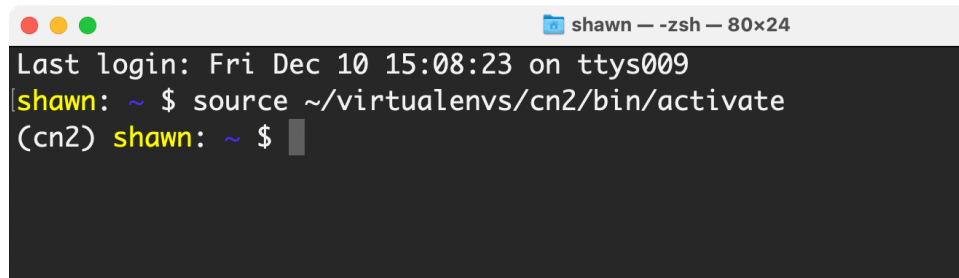
1. Create a new virtual environment called `cn240` (based on the latest cadnano2 version, v2.4.0) by running this command from the **Terminal**:

   `python3 -m venv ~/virtualenvs/cn240`

2. Run this command to confirm that you can successfully activate the new virtual environment:

   `source ~/virtualenvs/cn240/bin/activate`

   If everything worked, your terminal prompt should include a `(cn240)` prefix.

   ```
   ● ● ●                          🔲 shawn — -zsh — 80×24
   Last login: Fri Dec 10 15:08:23 on ttys009
   [shawn: ~ $ source ~/virtualenvs/cn2/bin/activate
   (cn2) shawn: ~ $ ▋
   ```

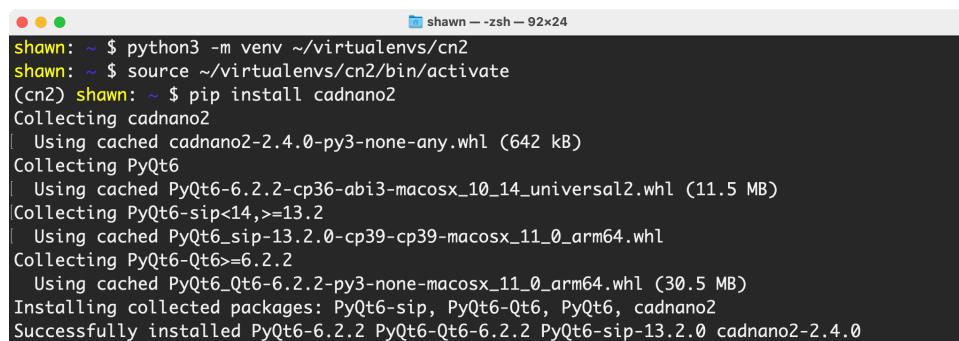   (I called my virtual environment `cn2`, so these screenshots differ from the instructions)

Step 5. Install Cadnano2 package

We will be using **pip**, a built-in command-line python package installer, to install **cadnano2**.

1. With your `cn240` virtual environment active in the **Terminal**, run the following command:

   `pip install cadnano2`

   You should see up to four packages install: `PyQt6`, `PyQt6-sip`, `PyQt6-Qt6`, and `cadnano2`

   ```
   ● ● ●                          🔲 shawn — -zsh — 92×24
   shawn: ~ $ python3 -m venv ~/virtualenvs/cn2
   shawn: ~ $ source ~/virtualenvs/cn2/bin/activate
   (cn2) shawn: ~ $ pip install cadnano2
   Collecting cadnano2
     Using cached cadnano2-2.4.0-py3-none-any.whl (642 kB)
   Collecting PyQt6
     Using cached PyQt6-6.2.2-cp36-abi3-macosx_10_14_universal2.whl (11.5 MB)
   Collecting PyQt6-sip<14,>=13.2
     Using cached PyQt6_sip-13.2.0-cp39-cp39-macosx_11_0_arm64.whl
   Collecting PyQt6-Qt6>=6.2.2
     Using cached PyQt6_Qt6-6.2.2-py3-none-macosx_11_0_arm64.whl (30.5 MB)
   Installing collected packages: PyQt6-sip, PyQt6-Qt6, PyQt6, cadnano2
   Successfully installed PyQt6-6.2.2 PyQt6-Qt6-6.2.2 PyQt6-sip-13.2.0 cadnano2-2.4.0
   ```
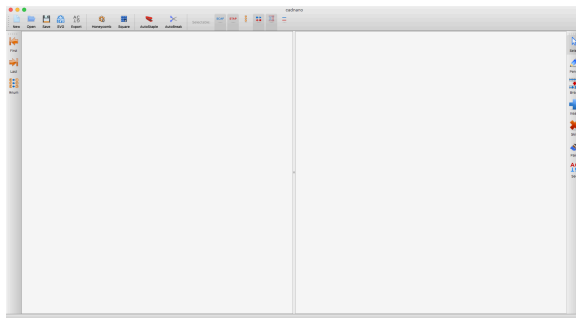
   (Note: the package versions may differ.)

Step 6: Run Cadnano2

1. The python package adds a console script to the path, so this command should launch Cadnano2 when run within your active virtual environment:

   `cadnano2`

Hopefully, the **cadnano2** window launched!

Both honeycomb and square lattice designs are handled by the same version of **cadnano2**, so we are pretty much ready to start designing! You can click on the **[Honeycomb]** or **[Square]** button in the interface to get started, or visit http://cadnano.org/gallery.html to download some published DNA origami designs to open and play with.
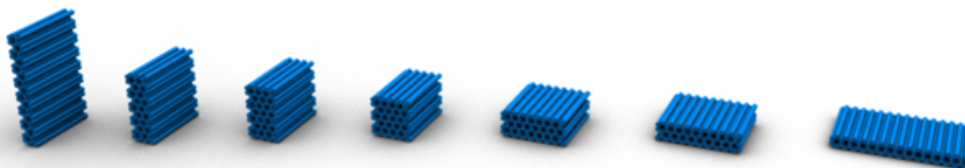
Additional Notes

What is the deal with all these versions of Cadnano?

Admittedly, versioning in the Cadnano ecosystem is a bit of a mess. Sorry about that! When we refer to **Cadnano1**, we mean two separate "legacy" versions of Cadnano. The first is **cadnano v0.2.3** which supports honeycomb-lattice designs, and the second is **cadnanoSQ v0.2.4**, which supports square-lattice designs). **Cadnano1** was written in the ActionScript language (similar to JavaScript) for the **Adobe AIR** platform.

Cadnano2 is the Python port of Cadnano1. It uses **PyQt6**, a library that provides Python bindings for **Qt** (pronounced "cute"), a cross-platform GUI framework, which replaces the function of Adobe AIR. **Cadnano2**

Cadnano2.5 is a complete rewrite of **Cadnano2**, led by Nick Conway. It introduced a very powerful and important feature: a scriptable Python API, that has enabled our lab to prototype the next generation of tools (some not yet released). It also served as a testbed for several new data model (e.g. non-lattice designs) and GUI features. Unfortunately it never received enough attention for the new GUI to stabilize. Thus, our lab uses it primarily for the Python API. Note that it also introduced a new file format (sometimes output with a **_cn25** suffix**)**. Cadnano2.5 scripts can also save a **legacy** file format that is backwards compatible with **Cadnano1** and **Cadnano2**.

How to Install honeycomb and square-lattice versions of Cadnano1

Practically speaking, we find that it is useful to have a more than one copy of **Cadnano** available on your system. For example, you might want to open two versions of a design and edit them side-by-side. Also, while an OS update may often cause one version to stop working, it is rare that both will break simultaneously.

1. Install **Adobe AIR** from: https://airsdk.harman.com/runtime
2. Download **cadnano** and **cadnanoSQ** from http://cadnano.org/legacy
3. Double-click each installers and follow the instructions. You should see an Application Install dialog window. Don't worry about the warning, it just means that we didn't pay a tax to become a known publisher. When you proceed, you can choose to add a desktop shortcut, or install in a different location.
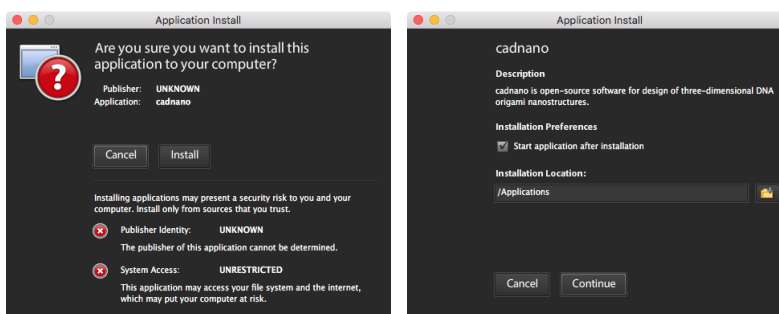
How to create an alias to launch cadnano2 from a single command

Once cadnano2 is working in your virtual environment, open a new **Terminal** and run the following:

```
cat <<END >> ~/.zprofile
alias cn240="source ~/virtualenvs/cn240/bin/activate && cadnano2"
END
```

The command above creates an alias that first activates your virtual env, and then runs the cadnano2 script. If successful, you should be able to open another new Terminal window and run one command to launch:

```
cn240
```

If you see an error, make sure that you didn't copy any special characters from the PDF.

Reporting Bugs

The **douglaslab/cadnano2** application is not bug free. You should get in the habit of saving your designs frequently, and reporting bugs when it crashes so we can make it more stable over time. Typically, you will get a Traceback in the **Terminal** window with some information about the line number and function that was executing when an exception was thrown.

If you encounter this situation, please copy the full traceback and paste it into new "Issue" along with detailed steps to reproduce at https://github.com/douglaslab/cadnano2/issues. If you have any example files that will be useful in reproducing the bug, please upload them too.

If you poke around in the source code and think that you've fixed the bug, please submit a pull request along with your Issue. If we incorporate your fix, you will be added to the list of contributors.

Updating Cadnano2

If you submit any bugs, hopefully, we will be able to patch the source code quickly. Once that is done, you'll need to synchronize your local copy of **cadnano2** with the latest release.

The safest approach is to create a new virtualenv, just in case you still want to use the previous release. Suppose we patch Cadnano2 and release v2.4.1, here's how we would install the new release alongside our existing one:

```
python3 -m venv ~/virtualenvs/cn241
source ~/virtualenvs/cn241/bin/activate
pip install cadnano2
```

If you want to upgrade your current virtualenv (e.g. cn240) in place, use the pip upgrade:

```
source ~/virtualenvs/cn240/bin/activate
pip install --upgrade cadnano2
```

Installing a specific version of Cadnano2

If you want to install a specific release, use the following commands to create a new venv and

```
python3 -m venv ~/virtualenvs/cn240
source ~/virtualenvs/cn240/bin/activate
pip install cadnano2==2.4.0
```