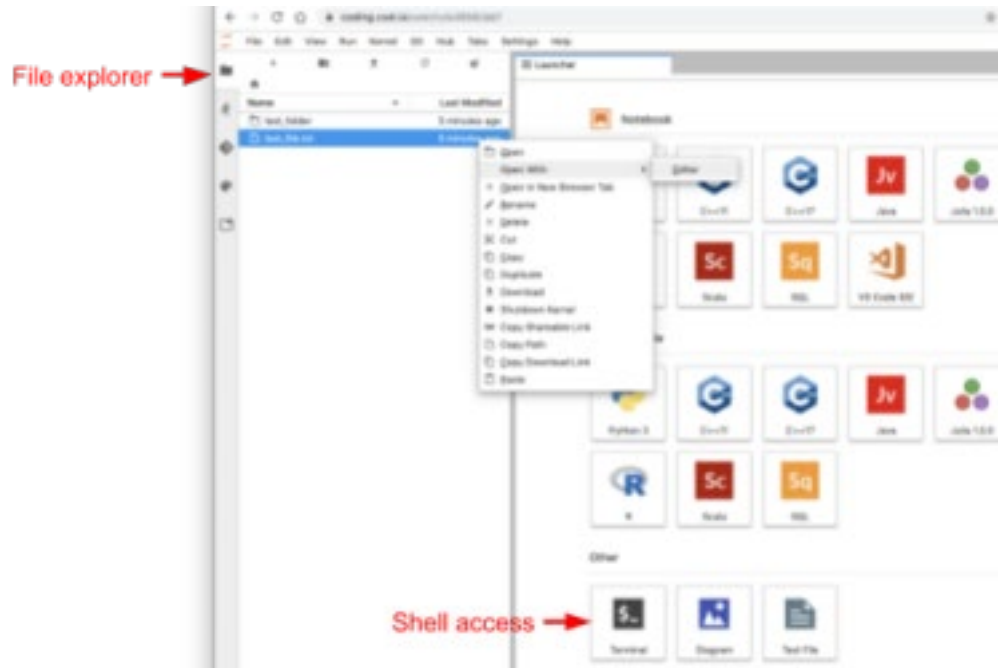


# Development Environments

## Software Engineering for Scientists

The reference environment is the CU CS JupyterHub at <https://coding.csel.io/>

Choose “Default Coding Environment”. This interface includes a file explorer for creating and editing files and a command-line interface.



### Initialize the CU CS JupyterHub setup (this you will only have to do once)

Essentially all software has dependencies (i.e. external, independent code libraries or configurations on which the software relies). Ensuring that all software dependencies are met is a major obstacle to reproducible scientific software. There are many ways to help communicate these dependencies and simplify the installation process. In this class, we will use conda (<https://conda.io/>), which is an environment manager and a package manager. Conda is already installed in this environment, but we need to do a few setup steps. NOTE: These steps are specific to this environment. Keep reading for instructions for getting conda installed in other environments (i.e., your laptop).

To set up conda we must first run the initialization:

```
jovyan@jupyter-jast1849:~$ conda init
no change      /opt/conda/condabin/conda
no change      /opt/conda/bin/conda
no change      /opt/conda/bin/conda-env
```

```
no change      /opt/conda/bin/activate
no change      /opt/conda/bin/deactivate
no change      /opt/conda/etc/profile.d/conda.sh
no change      /opt/conda/etc/fish/conf.d/conda.fish
no change      /opt/conda/shell/condabin/Conda.psm1
no change      /opt/conda/shell/condabin/conda-hook.ps1
no change      /opt/conda/lib/python3.8/site-packages/xontrib/conda.xsh
no change      /opt/conda/etc/profile.d/conda.csh
modified       /home/jovyan/.bashrc

==> For changes to take effect, close and re-open your current shell. <==

jovyan@jupyter-jast1849:~$
```

We've created a file called `.bashrc`. The new contents of this file is code that initializes our conda environments. As you can see below:

```
jovyan@jupyter-jast1849:~$ cat .bashrc

# >>> conda initialize >>>
# !! Contents within this block are managed by 'conda init' !!
__conda_setup="$('/opt/conda/bin/conda' 'shell.bash' 'hook' 2> /dev/null)"
if [ $? -eq 0 ]; then
    eval "$__conda_setup"
else
    if [ -f "/opt/conda/etc/profile.d/conda.sh" ]; then
        . "/opt/conda/etc/profile.d/conda.sh"
    else
        export PATH="/opt/conda/bin:$PATH"
    fi
fi
unset __conda_setup
# <<< conda initialize <<<

jovyan@jupyter-jast1849:~$
```

This file **\*SHOULD\*** run everytime you open a new shell window (it is the configuration file for the shell environment --- “rc” stands for “run commands”), **however** it doesn't in the case of the CS JupyterHub (idiosyncratic compared to other Linux machines). So we need to configure the shell environment so that it does.

There is another file `.bash_profile` in the home directory. This file does run every time you open a shell terminal. Therefore we just need to put a command to run the `.bashrc` file in there:

```
jovyan@jupyter-jast1849:~$ echo ". $HOME/.bashrc" >> $HOME/.bash_profile
```

```
jovyan@jupyter-jast1849:~$
```

Now, if you exit your terminal and reopen it, you should see the command prompt has changed. There is now “(base)” at the start. This means conda has been properly configured and you are in the base conda environment:

```
(base) jovyan@jupyter-jast1849:~$
```

## Installing and updating conda on your machine

**NOTE:** You will not have to do this for CSEL

If you don't have conda installed or you want to install it on another computer, you will need to install what's called **Miniconda**. This is one portion of a much larger library of software called **Anaconda**, which is a python distribution that is meant to make data science easier. Anaconda provides the command line environment management tool conda. Conda has become popular independent of anaconda, which lead to Miniconda, which cuts out all of the extra stuff in Anaconda. Miniconda can be installed from the following page: <https://docs.conda.io/en/latest/miniconda.html>

Once conda is installed (either by you or by default on the CS JupyterHub), be sure to update it. This may take a few minutes.

```
conda update -n base conda
```

## CSEL specific: Change you default environment directory

Before creating a new environment, we will first need to change the default directory where conda stores its environments. By default, conda stores environment information in `/opt/conda/envs/`. However, the contents of this directory are cleared every time you end your CSEL session. It is the contents of the home directory `/home/jovyan/` that persists between sessions. Therefore we need to create a file `.condarc` in the home directory and include in it a new default environment location.

```
(base) jovyan@jupyter-jast1849:~$ echo -e "envs_dirs:\n- /home/jovyan/.conda/envs/" > ~/.condarc
(base) jovyan@jupyter-jast1849:~$ cat ~/.condarc
envs_dirs:
- /home/jovyan/.conda/envs/
(base) jovyan@jupyter-jast1849:~$
```

Now, when the CSEL session loads, it will run the `.condarc` file and set the environment directory to `/home/jovyan/.conda/envs/` and any newly created environment will be stored in there.

## Creating a new environment

There exists a directory `/opt/conda/envs/` where the info for any new directory will be stored. When you create a new environment a new directory corresponding to the environment name will be created here, thus, when you go to install python libraries you want to use in your own code (dependencies) they will automatically be installed in this directory.

```
(base) jovyan@jupyter-jast1849:~$ conda create -n sweefs
Collecting package metadata (current_repodata.json): done
Solving environment: done

## Package Plan ##

environment location: /opt/conda/envs/sweefs
```

You will then be prompted to confirm the creation of the environment.

```
Proceed ([y]/n)? y

Preparing transaction: done
Verifying transaction: done
Executing transaction: done
#
# To activate this environment, use
#
#     $ conda activate sweefs
#
# To deactivate an active environment, use
#
#     $ conda deactivate

(base) jovyan@jupyter-jast1849:~$
```

Now, activate your environment and we can begin to install the packages we will use. You will see that you're in the new environment because the name in the parentheses at the start of the command prompt will change to that of the environment.

```
(base) jovyan@jupyter-jast1849:~$ conda activate sweefs
(sweefs) jovyan@jupyter-jast1849:~$
```

## Installing packages/libraries with conda

From within your environment you can install a python library as follows:

```
(sweefs) jovyan@jupyter-jast1849:~$ conda install pycodestyle
Collecting package metadata (current_repodata.json): done
Solving environment: done

...
```

Here we are installing a package called `pycodestyle` which is a common one for checking our code to make sure it conforms to good formatting. Whenever you install a package as above, conda will attempt to install the most up-to-date version of that package that is consistent with all the other libraries already in the environment. It will also update/downgrade or newly install any dependencies needed for this particular package. This may take a few minutes and it will prompt you if you want to proceed after it lists the changes it has determined it will make.

At any point you can look at the list of packages you have installed in an environment. You will see the package name, version, build ID, and the source from which it was downloaded:

```
(sweefs) jovyan@jupyter-jast1849:~$ conda list
# packages in environment at /opt/conda/envs/sweefs:
#
# Name                                Version                                Build                                Channel
_libgcc_mutex                         0.1                                    conda_forge                         conda-forge
_openmp_mutex                         4.5                                    1_gnu                              conda-forge
ca-certificates                       2021.5.30                             ha878542_0                          conda-forge
certifi                               2021.5.30                             py39hf3d152e_0                      conda-forge
ld_impl_linux-64                     2.36.1                                hea4e1c9_2                          conda-forge
libffi                                3.3                                    h58526e2_2                          conda-forge
libgcc-ng                             11.1.0                                hc902ee8_8                          conda-forge
libgomp                               11.1.0                                hc902ee8_8                          conda-forge
libstdcxx-ng                          11.1.0                                h56837e0_8                          conda-forge
ncurses                               6.2                                    h58526e2_4                          conda-forge
openssl                               1.1.1k                                 h7f98852_0                          conda-forge
pip                                   21.2.4                                pyhd8ed1ab_0                        conda-forge
pycodestyle                           2.7.0                                pyhd8ed1ab_0                        conda-forge
python                                3.9.6                                 h49503c6_1_cpython                  conda-forge
python_abi                            3.9                                    2_cp39                              conda-forge
readline                              8.1                                    h46c0cb4_0                          conda-forge
setuptools                            49.6.0                                py39hf3d152e_3                      conda-forge
sqlite                                 3.36.0                                h9cd32fc_0                          conda-forge
tk                                    8.6.10                                h21135ba_1                          conda-forge
tzdata                                2021a                                  he74cb21_1                          conda-forge
wheel                                 0.37.0                                pyhd8ed1ab_0                        conda-forge
xz                                    5.2.5                                  h516909a_1                          conda-forge
zlib                                  1.2.11                                h516909a_1010                      conda-forge
(sweefs) jovyan@jupyter-jast1849:~$
```

If you want to install a *specific version* of a package you would run:

```
conda install <package>=<version>
```

## Documenting and reproducing your environment

One of the principal purposes of conda is to help make your environment (and thus your software) more reproducible. To this end, you can save the exact state of all the packages in your environment and use that record to install an identical environment on another computer. You can export the condensed version of the environment info as follows:

```
(swefs) jovyan@jupyter-jast1849:~$ conda env export
name: swefs
channels:
  - conda-forge
dependencies:
  - _libgcc_mutex=0.1=conda_forge
  - _openmp_mutex=4.5=1_gnu
  - ca-certificates=2021.5.30=ha878542_0
  - certifi=2021.5.30=py39hf3d152e_0
  - ld_impl_linux-64=2.36.1=hea4e1c9_2
  - libffi=3.3=h58526e2_2
  - libgcc-ng=11.1.0=hc902ee8_8
  - libgomp=11.1.0=hc902ee8_8
  - libstdcxx-ng=11.1.0=h56837e0_8
  - ncurses=6.2=h58526e2_4
  - openssl=1.1.1k=h7f98852_0
  - pip=21.2.4=pyhd8ed1ab_0
  - pycodestyle=2.7.0=pyhd8ed1ab_0
  - python=3.9.6=h49503c6_1_cpython
  - python_abi=3.9=2_cp39
  - readline=8.1=h46c0cb4_0
  - setuptools=49.6.0=py39hf3d152e_3
  - sqlite=3.36.0=h9cd32fc_0
  - tk=8.6.10=h21135ba_1
  - tzdata=2021a=he74cb21_1
  - wheel=0.37.0=pyhd8ed1ab_0
  - xz=5.2.5=h516909a_1
  - zlib=1.2.11=h516909a_1010
prefix: /opt/conda/envs/swefs
(swefs) jovyan@jupyter-jast1849:~$
```

Which can be written to a file (commonly labeled with the extension `.yaml`)

```
(swefs) jovyan@jupyter-jast1849:~$ conda env export > swefs_env.yaml
```

This file can then be provided with your software and used to create an environment identical to your development environment on a new machine. This improves the reproducibility of your software. To create the environment from the `.yaml` file is as simple as the following:

```
(base) jovyan@jupyter-jast1849:~$ conda env create -f swefs_env.yaml
```

## Conda vs Pip

**Recommendation:** Use pip only after conda. Install as many requirements as possible with conda before resorting to using pip.

[Pip](#) is the basic Python Packaging Authority's recommended tool for installing packages from the [Python Package Index](#), PyPI. You can use pip to install python packages on your computer, without being in a conda environment. Pip installs Python software packaged as wheels (pre-compiled) or source distributions (not pre-compiled). The latter may require that the system have compatible compilers, and possibly libraries, installed before invoking pip to succeed.

[Conda](#) is a cross-platform package *and* environment manager that installs and manages conda packages from the [Anaconda repository](#). You can have different Conda environments for different projects, it is a nice way to keep all the dependencies your code needs together, distinct from other projects, and to be able to replicate the environment on another machine later.

Conda packages are binaries. There is never a need to have compilers available to install them. Additionally, conda packages are not limited to Python software. They may also contain C or C++ libraries, R packages, or any other software.

If you install packages with Conda, in your Conda environment, you do not need to ALSO pip install. However, if you change environments, you *will* need to install packages in the new environment as well. They do not transfer. We will be using one conda environment for this course so you do not need to worry about this.