**When to use pip VS conda?**

Differences between pip and conda:

1. Package Availability

Pip installs packages from the Python Package Index (PyPI), which hosts a vast array of Python libraries. Almost any Python library can be installed using pip.

On the other hand, conda installs packages from the Anaconda distribution and other channels. While the number of packages available through conda is smaller than pip, conda can install packages for multiple languages and not just Python.

2. Environment Management

While pip can be used in conjunction with virtualenv to create isolated environments, conda has this feature built-in. Conda environments can have different versions of Python and other languages, making it a powerful tool for managing complex projects.

3. Binary Packages

Conda installs binary packages, which means the packages include compiled code. This can make the installation process faster and more reliable, especially for packages with complex dependencies.

Pip, by contrast, often installs packages from source, which means the code is compiled during the installation process. This can be slower and more prone to errors, especially on Windows.

When to Use Pip or Conda?

Use pip if you are working with pure Python projects and need access to the vast array of libraries available on PyPI.

Use conda if you are working with projects that use multiple languages, need different versions of Python, or require complex binary dependencies.

In many cases, you can use both tools in the same project. For example, you can use conda to manage environments and install binary packages, and pip to install Python libraries that are not available through conda.