PIP

Pip is a standard package manager for Python. It allows you to install and manage additional libraries and dependencies not included in the standard Python library. Here's a basic tutorial on how to use pip:

Check if pip is installed: Pip is installed by default with Python 2.7.9 and later, and Python 3.4 and later. To check if pip is installed, open your terminal or command prompt and run:

```
pip --version
```

If pip is installed, you'll see the version of pip you have. If it's not installed, you'll need to install it.

Install a package: To install a package, you can use the `pip install` command followed by the name of the package. For example, to install the `requests` package, you would run:

```
pip install requests
```

This command installs the latest version of the package. If you want to install a specific version, you can specify it like this:

```
pip install requests==2.18.4
```

Upgrade a package: If you want to upgrade an already installed package to the latest version, you can use the `pip install --upgrade` command. For example, to upgrade the `requests` package, you would run:

```
pip install --upgrade requests
```

Uninstall a package: To uninstall a package, you can use the `pip uninstall` command followed by the name of the package. For example, to uninstall the `requests` package, you would run:

```
pip uninstall requests
```

List installed packages: To list all the packages that are currently installed, you can use the `pip list` command. This will display a list of installed packages along with their versions.

Search for packages: To search for packages by name, you can use the `pip search` command followed by a search query. For example, to search for packages related to "http", you would run:

```
pip search http
```

Show package details: To show more information about a specific installed package, you can use the `pip show` command followed by the name of the package. For example, to show information about the `requests` package, you would run:

```
pip show requests
```

This will display information like the version of the package, the directory where it's installed, its dependencies, and more.

Note: Depending on your system setup and whether you're using a virtual environment, you might need to use `pip3` instead of `pip` or add `python -m` before `pip` like `python -m pip install requests`.

That's a basic guide on how to use pip. Pip has many more features and options that you can explore by running `pip --help` or checking the official pip documentation.