PIP COMMANDS

1: Install a package: **pip install package_name**. This command installs the specified package and all its dependencies.

2: Uninstall a package: **pip uninstall package_name**. This command removes the specified package and all its dependencies.

3: Upgrade a package: **pip install --upgrade package_name**. This command upgrades the specified package to the latest version.

4: List installed packages: **pip list**. This command displays a list of all the packages installed in your Python environment.

5: Search for a package: **pip search package_name.** This command searches the Python Package Index for packages that match the specified name or keyword.

6: Show package information: **pip show package_name**. This command displays detailed information about the specified package, including its version, location, dependencies, and installed files.

7: Install a package from a requirements file: **pip install -r requirements.txt**. This command installs all the packages listed in the requirements.txt file in the current directory.

8: Create a requirements file: **pip freeze > requirements.txt.** This command creates a requirements.txt file that lists all the packages installed in your Python environment, along with their versions.

9: Check for outdated packages: **pip list --outdated**. This command displays a list of all the packages installed in your Python environment that have newer versions available.

10: Install a package in editable mode: **pip install -e path/to/package**. This command installs the specified package in "editable" mode, which means that any changes you make to the source code of the package are immediately reflected in your Python environment.