



Environment Management

Creating Environments

`conda create -n <env_name>` - Creates a new environment named `<env_name>`.

Example:

```
conda create -n myenv
```

`conda create -n <env_name> python=<version>` - Creates an environment with a specific Python version.

Example:

```
conda create -n myenv python=3.8
```

`conda create -n <env_name> --clone <existing_env>` - Clones an existing environment.

Example:

```
conda create -n newenv --clone myenv
```

Activating/Deactivating Environments

`conda activate <env_name>` - Activates the specified environment.

Example:

```
conda activate myenv
```

`conda deactivate` - Deactivates the current environment.

Listing and Removing Environments

`conda env list` - Lists all conda environments.

Example:

```
conda env list
```

`conda remove -n <env_name> --all` - Removes the specified environment.

Example:

```
conda remove -n myenv --all
```

Package Management

Installing Packages

`conda install <package_name>` - Installs a package in the current environment.

Example:

```
conda install numpy
```

`conda install -n <env_name> <package_name>` - Installs a package in a specific environment.

Example:

```
conda install -n myenv numpy
```

`conda install <package_name>=<version>` - Installs a specific version of a package.

Example:

```
conda install numpy=1.20.0
```

`conda install --file requirements.txt` - Installs packages from a requirements file.

Example:

```
conda install --file requirements.txt
```

Listing Packages

`conda list` - Lists installed packages in the current environment.

`conda list -n <env_name>` - Lists packages in a specific environment.

Example:

```
conda list -n myenv
```

Updating and Removing Packages

`conda update <package_name>` - Updates a package in the current environment.

Example:

```
conda update numpy
```

`conda update --all` - Updates all packages in the current environment.

`conda remove <package_name>` - Removes a package from the current environment.

Example:

```
conda remove numpy
```

Channel Management

Adding Channels

`conda config --add channels <channel_name>` - Adds a channel to the channel list.

Example:

```
conda config --add channels conda-forge
```

`conda config --prepend channels <channel_name>` - Adds a channel to the top of the channel list, giving it priority.

Example:

```
conda config --prepend channels conda-forge
```

Listing Channels

`conda config --get channels` - Lists configured channels.

Removing Channels

`conda config --remove channels <channel_name>` - Removes a channel from the channel list.

Example:

```
conda config --remove channels conda-forge
```

Configuration & Troubleshooting

Configuration Management

`conda config --set <key> <value>` - Sets a configuration option.

Example:
`conda config --set auto_update_conda false`

`conda config --get` - Lists all configuration options.

Troubleshooting

Problem: Conda is not recognized as a command.
Solution: Ensure Conda's bin directory is in your system's PATH environment variable.

Problem: Package installation fails due to conflicts.
Solution: Try creating a new environment with minimal dependencies, or use `conda update --all` in the existing environment to resolve dependency issues.

Problem: Slow package downloads.
Solution: Configure Conda to use a faster mirror or add the `conda-forge` channel, which often provides optimized packages.

Tips and Tricks

Use `conda clean --all` to remove unused packages and caches to free up disk space.

Export environment to file `conda env export > environment.yml`

Create from file `conda env create -f environment.yml`