# Conda Environment Setup and Jupyter Integration Guide

## Section 1: Cleanup Phase

Objective: Safely deactivate and remove old Conda environments and Jupyter kernels before setting up a fresh one.

# 1. Deactivate any active Conda environment  
conda deactivate

# 2. List all existing environments  
conda env list

# 3. Remove unwanted environments  
conda remove --name myenv --all

# 4. List existing Jupyter kernels  
jupyter kernelspec list

# 5. Uninstall orphaned or broken kernels  
jupyter kernelspec uninstall myenv

# 6. Restart terminal session  
exec "$SHELL"

## Section 2: Environment Setup Phase

Objective: Create a clean Conda environment from a YAML file and register it as a Jupyter kernel.

# 7. Create Conda environment from YAML  
conda env create -f myenv.yaml

# 8. Activate the new environment  
conda activate myenv

# 9. Install ipykernel inside the environment  
pip install ipykernel

# 10. Register the environment with Jupyter  
python -m ipykernel install --user --name=myenv --display-name "Python (myenv)"

## Section 3: Jupyter Notebook Integration

Objective: Verify the environment is correctly used within Jupyter.

# 11. Launch Jupyter Notebook from the environment  
jupyter notebook

In the Notebook UI:  
- Go to Kernel > Change Kernel > Python (myenv)

import sys  
print(sys.executable

# OR

!which python

✅ Expected Output: Should point to `myenv` path (e.g., /Users/sri/miniconda3/envs/myenv/bin/python)

## Final Notes

- Restart both Jupyter and the terminal after cleanup.  
- Always launch Jupyter from the target Conda environment to ensure proper linkage.  
- Use `conda env export > env.yaml` to back up environments for future reuse.