



PYTHON ENVIRONMENTS

SEPARATED, CONFLICT FREE, SHARABLE, REPRODUCIBLE, PERSISTENT ...

Simeon Beinlich
beinlich@fhi.mpg.de
PP&B Computer Support Group
Fritz-Haber-Institut der Max-Planck-Gesellschaft

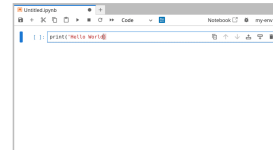
TERMINAL

```
(base) ~$: conda activate my-env  
(my-env) ~$: python3 my_script.py
```

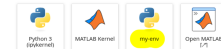
EXECUTABLE SCRIPT

```
#!/home/myusername/.conda/envs/my-env/bin/python3  
print("Hello World!")  
...
```

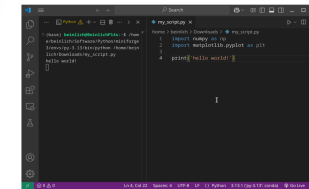
NOTEBOOKS (CHANGE)



NOTEBOOKS (CREATE)



IDE - VS CODE ETC.





WHY PYTHON ENVIRONMENTS?

WHY NOT THE SYSTEM'S PYTHON?

- **Limited user control**
- Python and package **versions fixed by the operating system**
 - **Compatibility issues**
- **Required updates** might break your packages
 - *numpy 2.x* vs *numpy 1.x*
 - *setup.py* vs. *pyproject.toml* (*python > 3.10*)
- ...



WHY PYTHON ENVIRONMENTS?

WHY NOT THE SYSTEM'S PYTHON?

- **Limited user control**
- Python and package **versions fixed by the operating system**
 - **Compatibility issues**
- **Required updates** might break your packages
 - *numpy 2.x vs numpy 1.x*
 - *setup.py vs. pyproject.toml (python > 3.10)*
- ...

BENEFITS OF ENVIRONMENTS

- Never touch a running system: **Set up once, never touch again!**
- **Full user control**
- **Exportable & Importable**
 - **Backup** or **Share** with colleagues / group
- Some modules require **specific python & package versions**
 - **Old scripts** / code (*Scientific packages aren't always well maintained!*)
 - **Compatibility issues** between conflicting packages
- **Extra features** (*conda can install entire isolated software stacks for a single user*) `conda install gcc=8.5 ...`
- **Multiple isolated setups.**
- ...



WHY PYTHON ENVIRONMENTS?

WHY NOT THE SYSTEM'S PYTHON?

- **Limited user control**
- Python and package **versions fixed by the operating system**
 - **Compatibility issues**
- **Required updates** might break your packages
 - *numpy 2.x vs numpy 1.x*
 - *setup.py vs. pyproject.toml (python > 3.10)*
- ...

**Conda is the default Python Setup
on all Compute Servers!**

*Use Python environments now instead of having
troubles later!*

BENEFITS OF ENVIRONMENTS

- Never touch a running system: **Set up once, never touch again!**
- **Full user control**
- **Exportable & Importable**
 - **Backup** or **Share** with colleagues / group
- Some modules require **specific python & package versions**
 - **Old scripts** / code (*Scientific packages aren't always well maintained!*)
 - **Compatibility issues** between conflicting packages
- **Extra features** (*conda can install entire isolated software stacks for a single user*) `conda install gcc=8.5 ...`
- **Multiple isolated setups.**
- ...



MANAGING YOUR ENVIRONMENTS

CREATING (*ONCE*)

```
(base) ~$: conda create --name=my-env python=3.12
```

ACTIVATING (*ALWAYS*)

```
(base) ~$: conda activate my-env
```

REGISTERING (*ONCE*)*

For using it in jupyterhub, vscode, pycharm, ...

```
(my-env) ~$: conda install ipykernel  
python3 -m ipykernel install --user --name=my-env
```

DEACTIVATING (*WHEN DONE*)

```
(my-env) ~$: conda deactivate
```

INSTALLING (*ON DEMAND*)*

```
(my-env) ~$: conda install ipython scipy ipykernel  
jupyterlab=3.2 matplotlib=3.5 numpy=1.21
```



MANAGING YOUR ENVIRONMENTS

CREATING (*ONCE*)

```
(base) ~$: conda create --name=my-env python=3.12
```

ACTIVATING (*ALWAYS*)

```
(base) ~$: conda activate my-env
```

REGISTERING (*ONCE*)*

For using it in jupyterhub, vscode, pycharm, ...

```
(my-env) ~$: conda install ipykernel  
python3 -m ipykernel install --user --name=my-env
```

DEACTIVATING (*WHEN DONE*)

```
(my-env) ~$: conda deactivate
```

INSTALLING (*ON DEMAND*)*

```
(my-env) ~$: conda install ipython scipy ipykernel  
jupyterlab=3.2 matplotlib=3.5 numpy=1.21
```

*You can also use `pip install something somethingelse==1.2.3` but try to not mix pip and conda within the same environment!



MANAGING YOUR ENVIRONMENTS

CREATING (*ONCE*)

```
(base) ~$: conda create --name=my-env python=3.12
```

ACTIVATING (*ALWAYS*)

```
(base) ~$: conda activate my-env
```

REGISTERING (*ONCE*)*

For using it in jupyterhub, vscode, pycharm, ...

```
(my-env) ~$: conda install ipykernel  
python3 -m ipykernel install --user --name=my-env
```

DEACTIVATING (*WHEN DONE*)

```
(my-env) ~$: conda deactivate
```

INSTALLING (*ON DEMAND*)*

```
(my-env) ~$: conda install ipython scipy ipykernel  
jupyterlab=3.2 matplotlib=3.5 numpy=1.21
```

LIST EXISTING

```
(base) ~$: conda env list
```

REMOVING

```
(base) ~$: conda remove --name=my-new-env --all
```

EXPORTING / BACKUP

```
(my-env) ~$: conda env export > environment.yml
```

IMPORTING / RESTORING

```
(base) ~$: conda env create -f environment.yml
```

UNDO CHANGES (*IF NEEDED*)

```
# List your changes  
(my-env) ~$: conda list --revisions  
  
# Restore some state  
(my-env) ~$: conda install --revision=REVNUM
```

* You can also use `pip install something somethingelse==1.2.3` but try to not mix pip and conda within the same environment!



MANAGING YOUR ENVIRONMENTS

CREATING (*ONCE*)

```
(base) ~$: conda create --name=my-env python=3.12
```

ACTIVATING (*ALWAYS*)

```
(base) ~$: conda activate my-env
```

REGISTERING (*ONCE*)*

For using it in jupyterhub, vscode, pycharm, ...

```
(my-env) ~$: conda install ipykernel  
python3 -m ipykernel install --user --name=my-env
```

... not much additional effort ...

*You can also use `pip install something somethingelse==1.2.3` but try to not mix pip and conda within the same environment!

USAGE IN THE TERMINAL, SCRIPTS, NOTEBOOKS ETC.



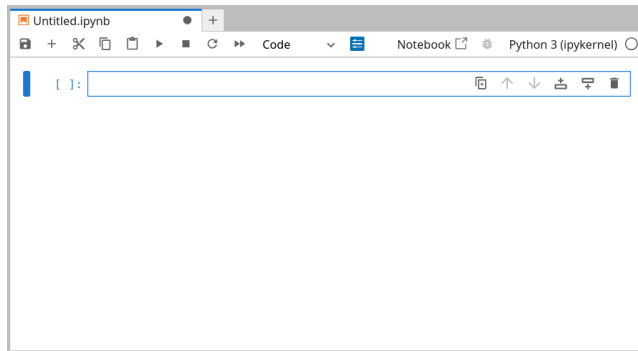
TERMINAL

```
(base) ~$: conda activate my-env  
  
(my-env) ~$: python3 my_script.py
```

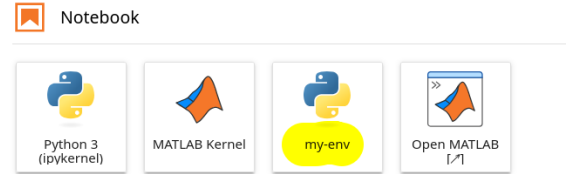
EXECUTABLE SCRIPT

```
#!/home/myusername/.conda/envs/my-env/bin/python3  
print("Hello World!")  
...
```

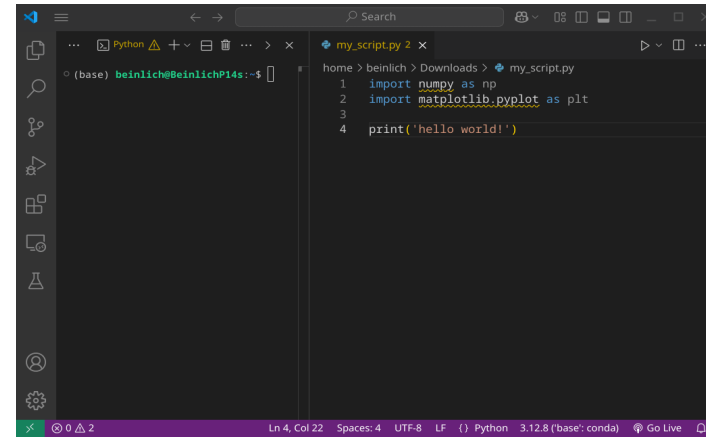
NOTEBOOKS (CHANGE)



NOTEBOOKS (CREATE)



IDE - VSCODE ETC.





FURTHER TOPICS & LINKS

MORE TRAINING MATERIAL

Visit root.compute.fhi.mpg.de/training

PPB WIKI | HPC

General Information.

FHI GITLAB | HPC DOCUMENTATION

Jupyterhub Wiki | Virtual Desktop Wiki | Storage Wiki

Questions, issues, needing help?

Please mail: beinlich@fhi.mpg.de

or call: +49 30 8431 5270