

# JASMIN Workshop: Exercise 07: Build your own python 3 environment

## Scenario

I need to use a Python 3 environment for my analysis work. I have seen that CEDA has Python 3 available in its "Jaspy" environment but I need a specific package called "fixnc" (fix netCDF files).

## Objectives

I want to build my own Python 3 environment on top of the Jaspy environment and then install "fixnc" into it. I can use the "venv" module to create my own virtual environment and install new packages into that.

## JASMIN resources

- A scientific analysis server
- Access to my `$HOME` directory to install the environment
- Access to the Python 3 ("Jaspy") environment on JASMIN.

## Local resources

- SSH client (to login to JASMIN)

## Instructions

1. Start ssh-agent session and add JASMIN private key
2. SSH to a scientific analysis server
3. Activate the Jaspy Python 3 environment with the "module" command
4. Create a Python 3 virtual environment in your `$HOME` directory
5. Activate the virtual environment
6. Pip install the "fixnc" package from the PyPI remote repository
7. Test that the package can be imported in a python session
8. Deactivate the virtual environment and test the import again
9. Write a setup script ("`~/setup-workshop-env.sh`") so that you can activate the virtual environment in a single line each time you login
10. Now whenever you login you can run "`source ~/setup-workshop-env.sh`" and your own Python 3 environment will be enabled

## Review

This exercise demonstrates how to:

- Activate the default Jaspy Python environment on JASMIN
- Create a Python 3 "virtual environment"
- Install additional packages into your virtual environment
- Create a setup script for activating your virtual environment when you login

## Alternative approaches and best practice

- Sharing your environment with others
- Set up virtual environment without "system site packages"

# Cheat sheet for Exercise 07: Build your own python 3 environment

1. Start ssh-agent session and add JASMIN private key

```
exec ssh-agent $SHELL  
ssh-add ~/.ssh/id_rsa_jasmin
```

2. SSH to a scientific analysis server

```
ssh -A <username>@jasmin-login1.ceda.ac.uk  
ssh jasmin-sci5 # Could use any of sci[123456]
```

3. Activate the Jaspy Python 3 environment with the "module" command

```
module load jasper
```

4. Create a Python 3 virtual environment in your \$HOME directory

```
python -m venv ~/my-workshop-venv --system-site-packages
```

5. Activate the virtual environment

```
source ~/my-workshop-venv/bin/activate
```

6. Pip install the "fixnc" package from the PyPI remote repository

```
pip install fixnc
```

7. Test that the package can be imported in a python session

```
python -c 'import fixnc; print(dir(fixnc))'
```

8. Deactivate the virtual environment and test the import again

```
deactivate # Deactivates the virtual environment  
python -c 'import fixnc' # Now fails to import because cannot find "fixnc"  
Traceback (most recent call last):  
  File "<string>", line 1, in <module>  
ModuleNotFoundError: No module named 'fixnc'
```

9. Write a setup script ("~/setup-workshop-env.sh") so that you can activate the virtual environment in a single line each time you login

```
echo "module load jasper" > ~/setup-workshop-env.sh  
echo "source ~/my-workshop-venv/bin/activate" >> ~/setup-workshop-env.sh
```

10. Now whenever you login you can run "source ~/setup-workshop-env.sh" and your own Python 3 environment will be enabled.

## Alternative approaches and best practice

- Sharing your environment with others:
  - If you need to create your own environment it is important to be aware of which file system you are working on:
    - SOF (e.g. `/gws/nopw/j04`): does not perform well with small files at present.
    - SSD (e.g. `$HOME` and `/gws/smf/j04`): performs much better with small files.
  - If you are building an environment for your use only then it makes sense to create it under your `$HOME` directory.
  - If you need to share an environment with other JASMIN users you can:
    - Request a "small files" Group Workspace (GWS).
    - Install the software environment within the "small files" GWS.
    - Then all users with access to that GWS will be able to access the environment.
- Request that your software dependencies are added to the common Python 3 "Jaspy" environment on JASMIN:
  - See more details at:  
<https://help.jasmin.ac.uk/article/4729-jaspy-envs-py3-rhel6-rhel7#request-updates>
- Set up virtual environment without "system site packages":
  - We called the `"venv"` module with this argument: `--system-site-packages`
  - That means that all the packages in the base Jaspy Python 3 environment are available in the virtual environment.
  - However, you might prefer to only keep the core Python 3 packages. If that is the case then simply remove the `"--system-site-packages"` flag.