# Login to Cannon

ssh <username>@login.rc.fas.harvard.edu

# Go to a compute node on the test partition:

salloc –-partition test --nodes=1 --cpus-per-task=2 --mem=12GB --time=00:30:00

# Check Python modules available on Cannon:

module spider python

# Get detailed information on specific module, e.g.:

module spider python/3.10.13-fasrc01

# Load the latest (usually also the default) Python module:

module load python

# Check current location & change if desired for this training: pwd

cd <desired-location>

# Clone FASRC User Codes repository: https://github.com/fasrc/User\_Codes/tree/master

SSH - git clone git@github.com:fasrc/User\_Codes.git

HTTPS - git clone https://github.com/fasrc/User\_Codes.git

# Create a training folder & go to that folder:

mkdir python-training

cd python-training

# Copy Python folders from the User Codes directory:

cp -r ../User\_Codes/Languages/Python .

cp -r ../User\_Codes/Parallel\_Computing/Python/Python-Multiprocessing-Tutorial .

# Check Python version

python --version

# Invoke Python interpreter & write a program

def square(x):

"""square a number"""

return x \*\* 2

for N in range(1, 4):

print(N, "squared is", square(N))

exit()

# Run a Python program

python myscript.py

# Go to Example1 folder

cd Python/Example1

# Submit job

sbatch run.sbatch

# Go to a compute node

salloc -p test --nodes=1 --cpus-per-task=2 --mem=12GB --time=01:00:00

# Create a vanilla mamba environment on a desired location

module load python

mamba create --prefix=/n/holylabs/LABS/<desired-folder>/multiproc\_env python=3.11 -y

# Create a vanilla mamba environment on default $HOME

module load python

mamba create --name multiproc\_env python=3.11 -y

# Activate the environment based on the location

mamba activate /n/holylabs/LABS/<desired-folder>/multiproc\_env

#OR

mamba activate multiproc\_env

# Install relevant packages

mamba install numpy pandas matplotlib -y

pip install jupyterlab swifter

# Deactivate environment

mamba deactivate

# Go to Example2 folder

cd ../Python/Example2

# Submit job

sbatch run.sbatch