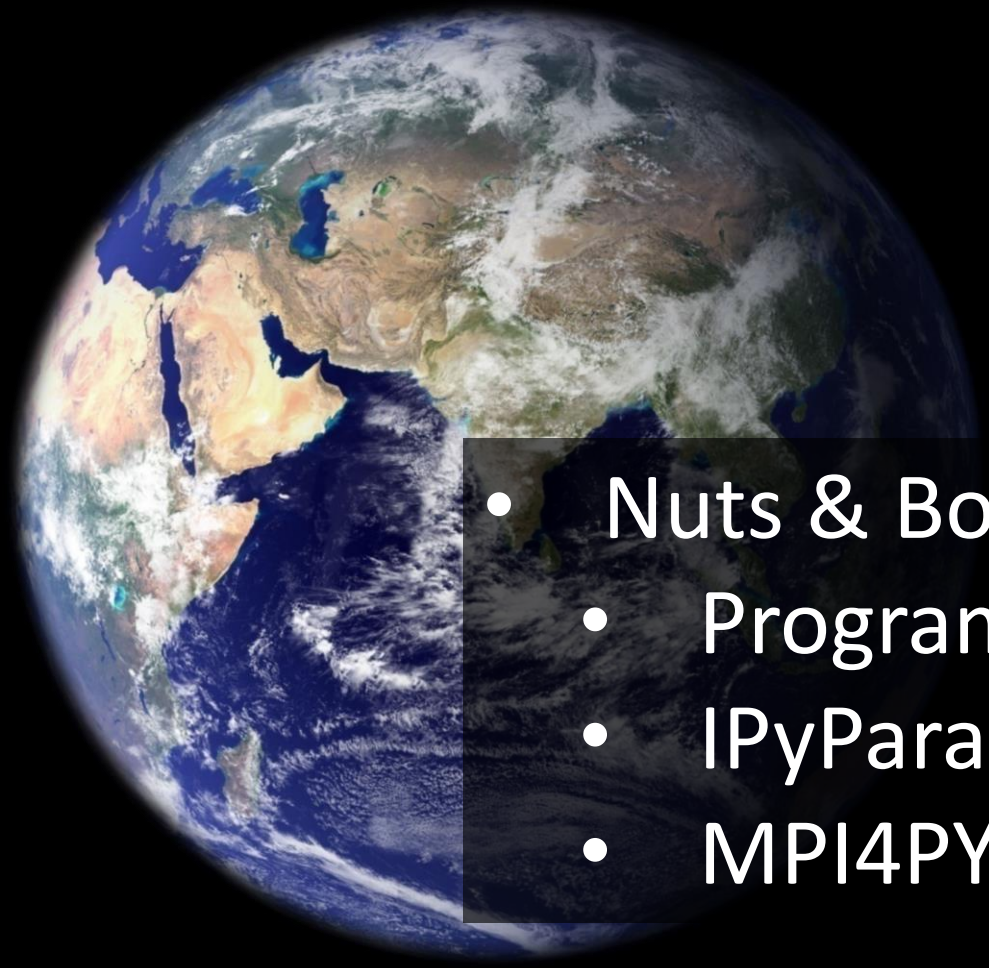
An Introduction to Parallel Programming in Python

Nick Featherstone
CU Research Computing

Web Link to These Slides



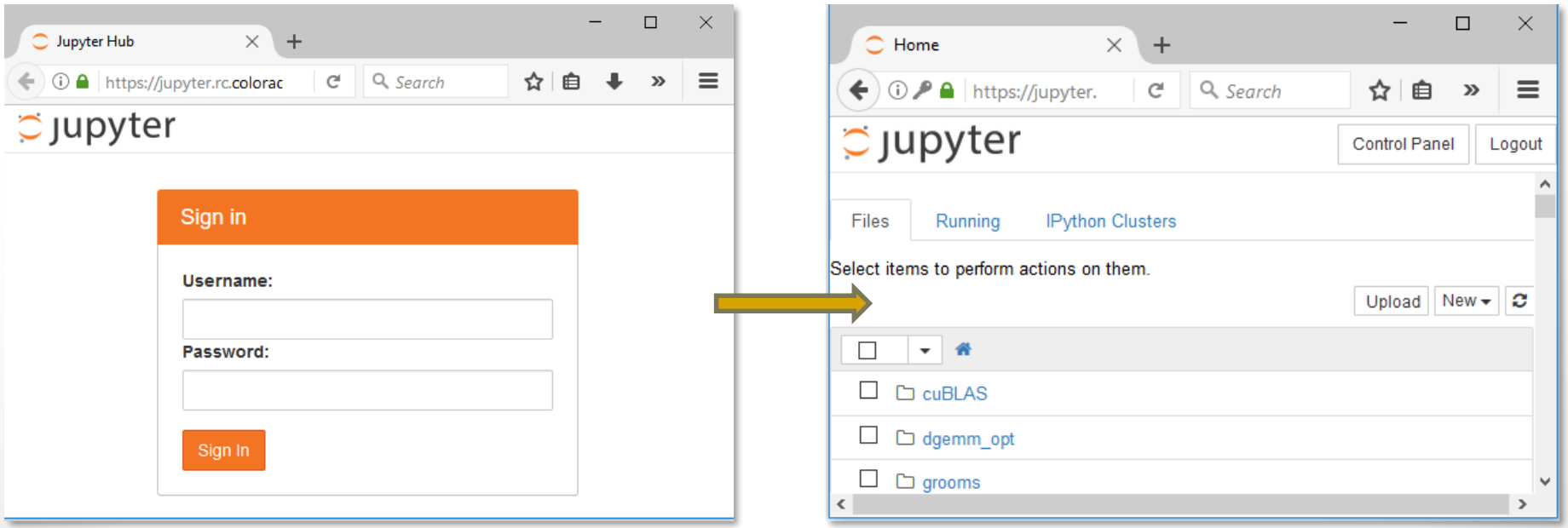
Who are we?
Why are we here?

- Nuts & Bolts of Parallel Python
  - Programming with NumPy
  - IPyParallel
  - MPI4PY

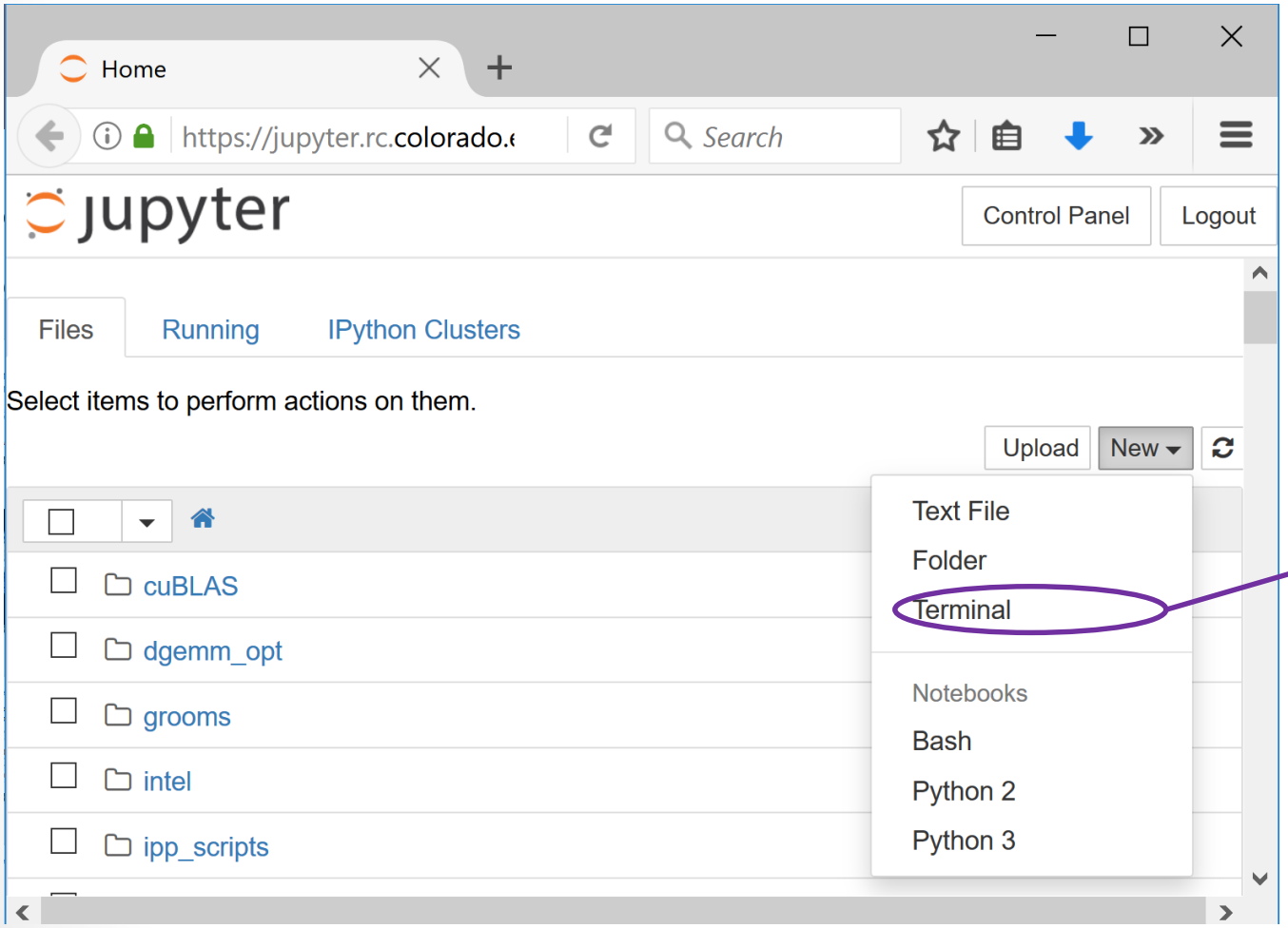
Getting Started

- Login to the RC Jupyter Hub:

https :// jupyter . rc . colorado . edu



Getting Started…



Open a terminal

Getting Started…

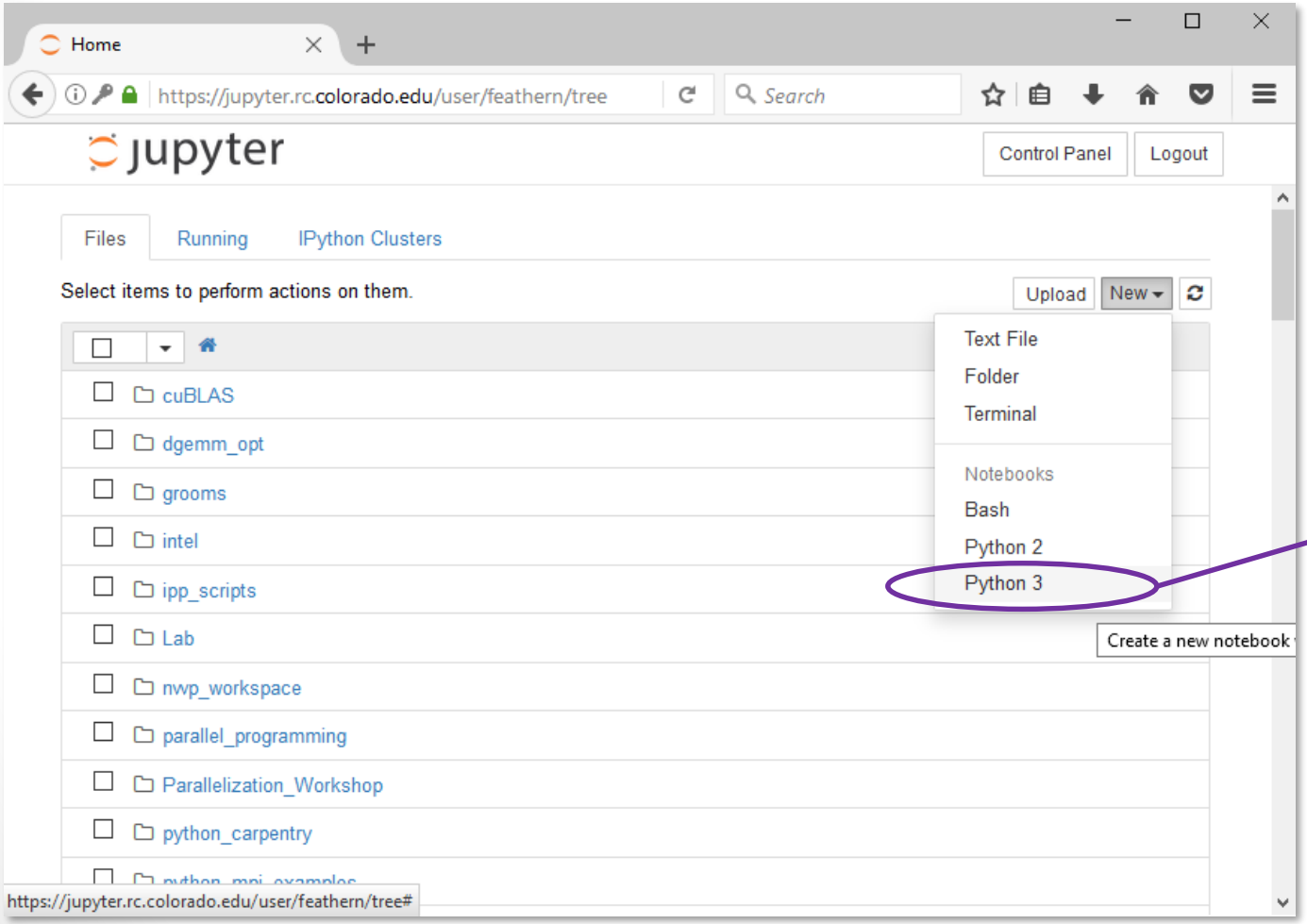
- Clone the repository (type all one line):

  git clone git@github.com:
  ResearchComputing/Parallelization_Workshop.git

- Type ‘exit’

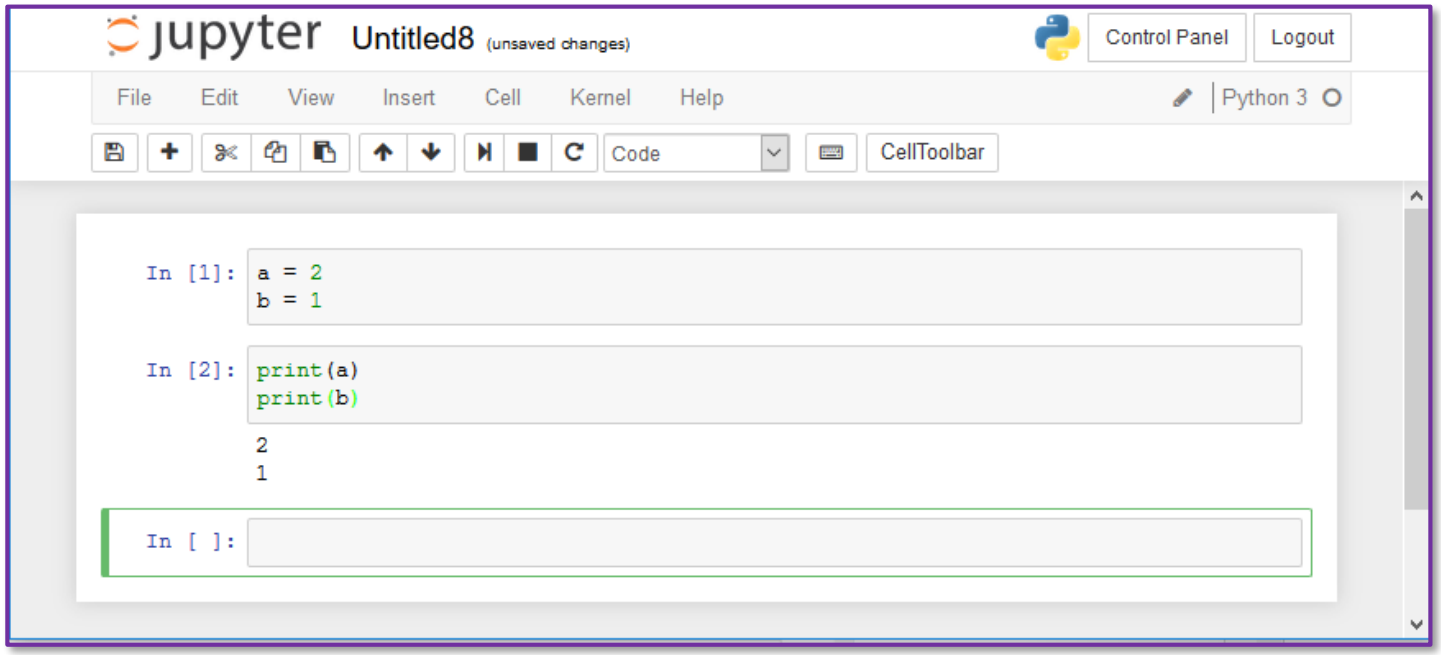
- Close your terminal tab

Getting Started…



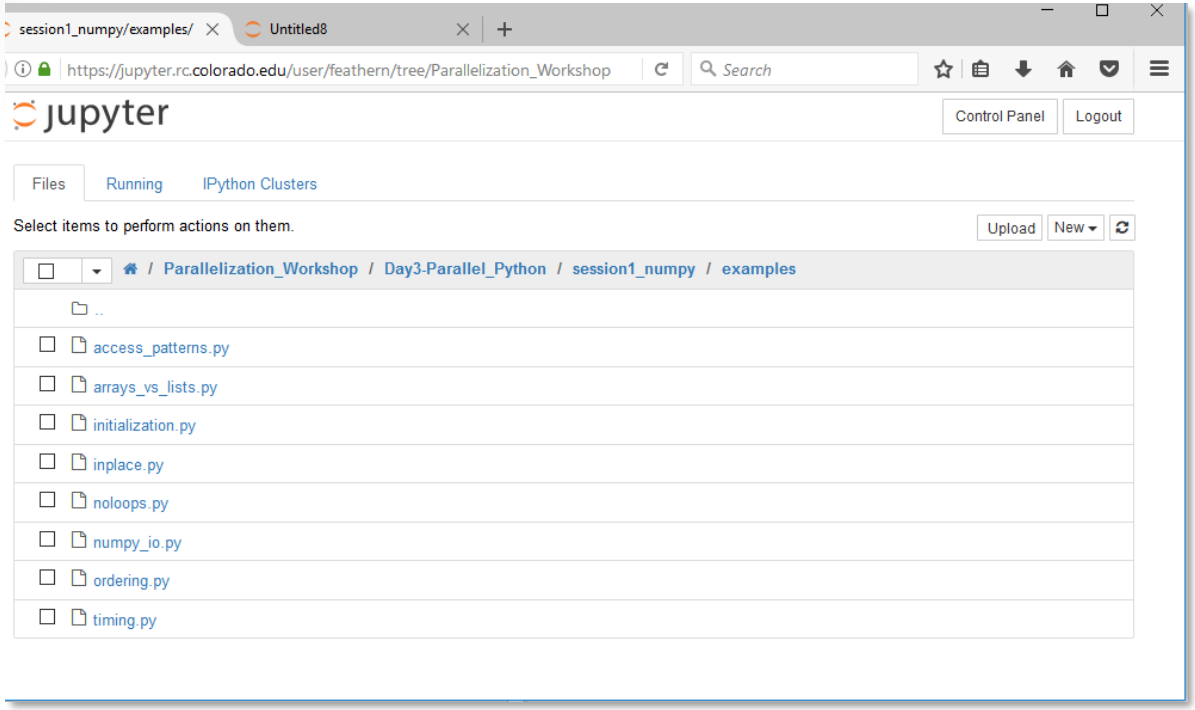
Start a Python 3 Notebook

So how does this work?



- Pressing “enter” moves to next line
- Pressing “shift” + “enter” executes code block
- Variables remain in memory between blocks…

File browser tab remains open…



Open this file:

Parallelization Workshop /

Day3-Parallel_Python /

session1_numpy /

examples /

timing.py

Workflow for today:

- Open file in file browser
- Cut + paste into notebook tab
- “shift” + “enter”

Timing in Python…

- Timing via “time” module
- Let’s look at **timing.py**
- time() returns seconds elapsed since some reference time.

usage pattern

```
import time

t0 = time.time()

… code you want to time …

t1 = time.time()
dt = t1-t0
print (‘Calculation time in seconds: ‘, dt)
```

Open this file:

Parallelization Workshop /

Day3-Parallel_Python /

session1_numpy /

examples /

timing.py