

Python crash course with numpy and scipy

NumPy, Scipy

L. Kos¹ M.Brank¹

¹Faculty of Mechanical Engineering
University of Ljubljana

HPC in Data Science: focus on Big Data and AI, June 2023

Overview

Demonstration and hands-on `login1.karolina.it4i.cz`

- Running a Jupyter notebook session
- Demonstrating numpy aspects
- Performing computations
- SciPy

Examples for this session:

`https://github.com/kosl/python-training/`

Running a Jupyter notebook session on a LOGIN node

```
# On your local Linux or MacOS machine
ssh it4i-username@login1.karolina.it4i.cz
# Clone the tutorial
git clone https://github.com/kosl/python-training.git
# Load anaconda module on a compute node
module load Anaconda3
# Run the Jupyter lab
jupyter-lab
...
Or copy and paste one of these URLs:
    http://localhost:8888/lab?token=34fa853873fd58aa1
```

The port number after the localhost: differs according to the free port availability. We need to forward this port to our localhost (laptop) through SSH connection.

Note

Running long computation on a login node is not recommended but can be used for the introductory part although running on a compute node is preferred.

Port forwarding from the login node to the localhost

To establish the port forwarding tunnel from the compute node through the login node to the local machine (laptop) press Enter while the lab process is running to get a clear new line then press ~ (tilda), then C, to get the ssh> prompt. Then enter the port forwarding command

```
ssh> -L 8888:localhost:8888
Forwarding port.
```

Note

Port number differs by user. While typing SSH escape command ~C, to get the ssh> prompt, nothing is echoed!

Alternative method to establish a SSH tunnel is to run another terminal connection from the localhost with above connection like

```
ssh -TN -L 8888:localhost:8888 \  
-f it4i-username@login1.karolina.it4i.cz
```

Running a Jupyter notebook session on a compute node

This method is preferred but can take a while for compute node to be allocated for max 1 hour.

```
# On your local Linux or MacOS machine
```

```
ssh it4i-username@login1.karolina.it4i.cz
```

```
# Clone the tutorial
```

```
git clone https://github.com/kosl/python-training.git
```

```
# Acquire your own compute node for an hour
```

```
python-training/jupyter-lab.sh
```

```
...
```

Or copy and paste one of these URLs:

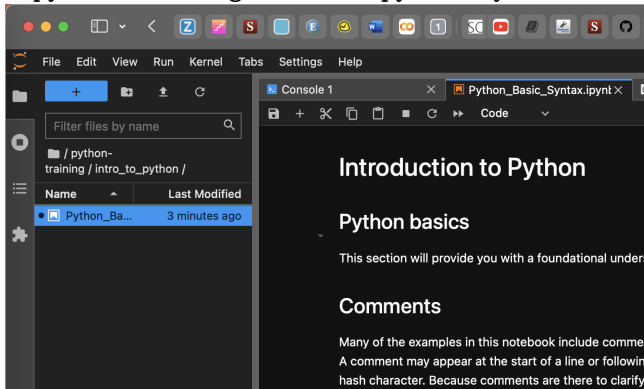
```
http://localhost:8000/lab?token=c659716f2c10612b1
```

The script establishes a new port forwarding tunnel from the compute node to the the login node. User needs only to establish tunnel to local machine (laptop) as described in the previous slide. Key sequence <Enter>~C and then `ssh> -L 8000:localhost:8000` in the above port forwarding example.

After port forwarding is established at our localhost we can open a connection printed out together with token in any browser. For example

```
http://localhost:8000/lab?token=c659716f2c10612bb6bd6502
```

and immediately the jupyter-lab appears where we select default kernel and first example in file browser on the left, where we open and run step by step `~/python-training/intro_to_python/Python_Basic_Syntax.ipynb`



The screenshot shows the JupyterLab interface. On the left is a file browser with a search bar and a table of files. The file `Python_Ba...` is selected, with a last modified time of 3 minutes ago. The main area displays a notebook titled `Python_Basic_Syntax.ipynb`. The notebook content includes the following sections:

Introduction to Python

Python basics

This section will provide you with a foundational unders

Comments

Many of the examples in this notebook include commen
A comment may appear at the start of a line or followin
hash character. Because comments are there to clarify

NumPy fast array interface

- Standard Python is not well suitable for numerical computations
 - lists are flexible but slow for using in numerical computations
- NumPy adds a new numpy **array** data type
 - Static, multidimensional
 - All elements of the array have the same type
 - Fixed number of elements in array, it's shape can be changed

Advantages of Numpy arrays over Python lists:

- Array operators: +, *, /, **
- Vectorization: Less for loops to write, better looking code
- Array operations and functions

Numpy acts as a base for writing numerical code in Python. Numpy arrays can be passed to C, C++ or Fortran, thus extending python with C/C++/Fortran.

Scipy, numerical library that provides user-friendly and efficient numerical routines for numerical integration, interpolation, optimization, linear algebra and statistics.

Written to use numpy arrays, thus functions take and operate on numpy arrays. For linear algebra it calls the BLAS/LAPACK functions, therefore it is recommended to have an optimised BLAS (ATLAS, MKL, OpenBLAS) installed.

Pandas, a high-performance, easy-to-use data structure and data analysis tools for Python.

Used for work with relational or labeled data easy and intuitive. It provide tools for inspection, statistics and visualization of data.

Hands-on session

Prepared Jupyter notebooks explain the basics needed for handling data in the following sessions found under `~/python-training/intro_to_numpy_scipy` directory:

[numpy_examples](#) Basics of NumPy: Operators, array manipulation,...

[numpy_mandelbrot](#) Calculating the Mandelbrot set using NumPy

[scipy_find_minimum](#) Search extrema with scipy

[scipy_fitting_function](#) Fitting a function through data using scipy

[tasMaxMin](#) ECAS data handling examples