Python for HPC

Dr. Simone Bacchio

Computational Scientist

CaSToRC, The Cyprus Institute

National Competence Center in HPC

THE CYPRUS INSTITUTE

RESEARCH•TECHNOLOGY•INNOVATION

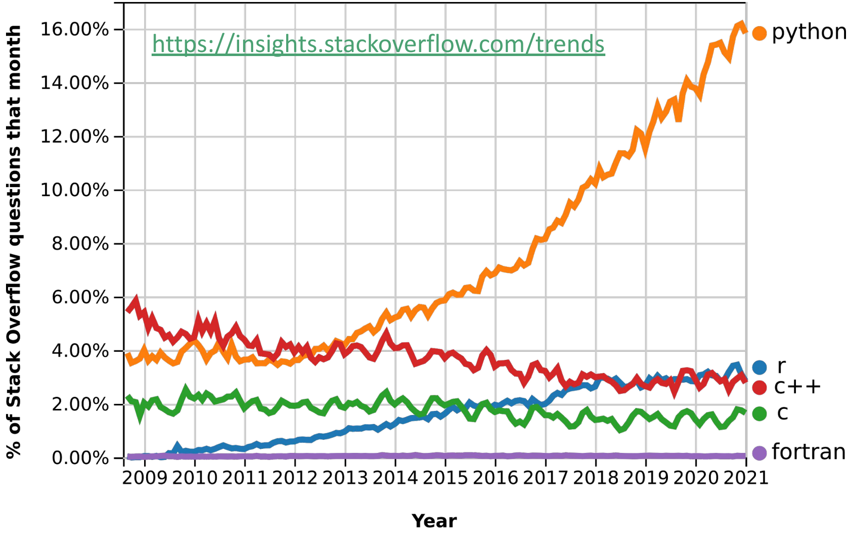
Today’s program

10:00-11:00	Performance Optimization, with numpy
11:00-11:30	Break
11:30-12:30	Performance Optimization, with numba
12:30-13:15	Lunch Break
13:15-14:15	Linking to C/C++ code, with cppyy
14:15-14:45	Break
14:45-15:45	MPI in Python, with mpi4py

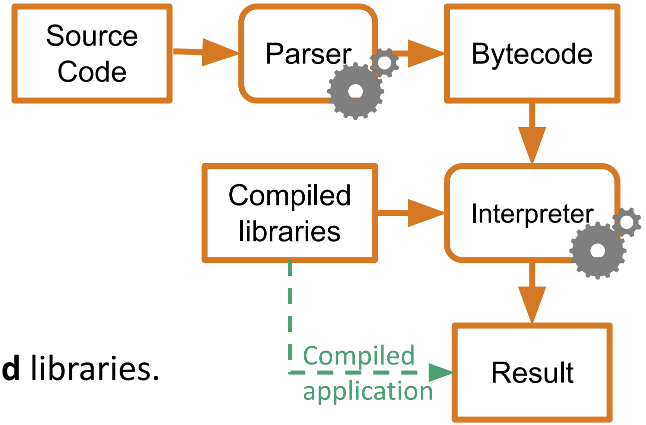
16:00 - 17:00	Seminar: “The CYGNUS Models for the Spectral Energy Distributions of Galaxies and their Supermassive Black Holes”, Prof. Andreas Efstathiou

Before starting... Why Python?

- **Interpreted** and object oriented programming language
- Science- and data-oriented
- Easy to Learn and Use
- Huge community
- Hundreds of Python Libraries and Frameworks
- First choice for Big Data and **Machine learning**
- User-friendly and great **APIs**
- Easy deployment of software (PyPI)
- Build with a scientific approach (PEPs)
- **Performance issues?** They can be overcome



Performance in Python



Python is a very powerful and flexible programming language, but…

- interpreted = bad (computational) performance
- it is important to know the strengths and the weaknesses!
- By its own it is not mean for High-Performance computing.

Built-in functions and HPC modules are based on **compiled** and **optimized** libraries.

Use as much as possible:

- built-in functions
- numerical modules (Numpy, Scipy, Pandas, …)
- compile your kernels (Cython, Pythran, Numba, …)

NEVER do for-loops on data!