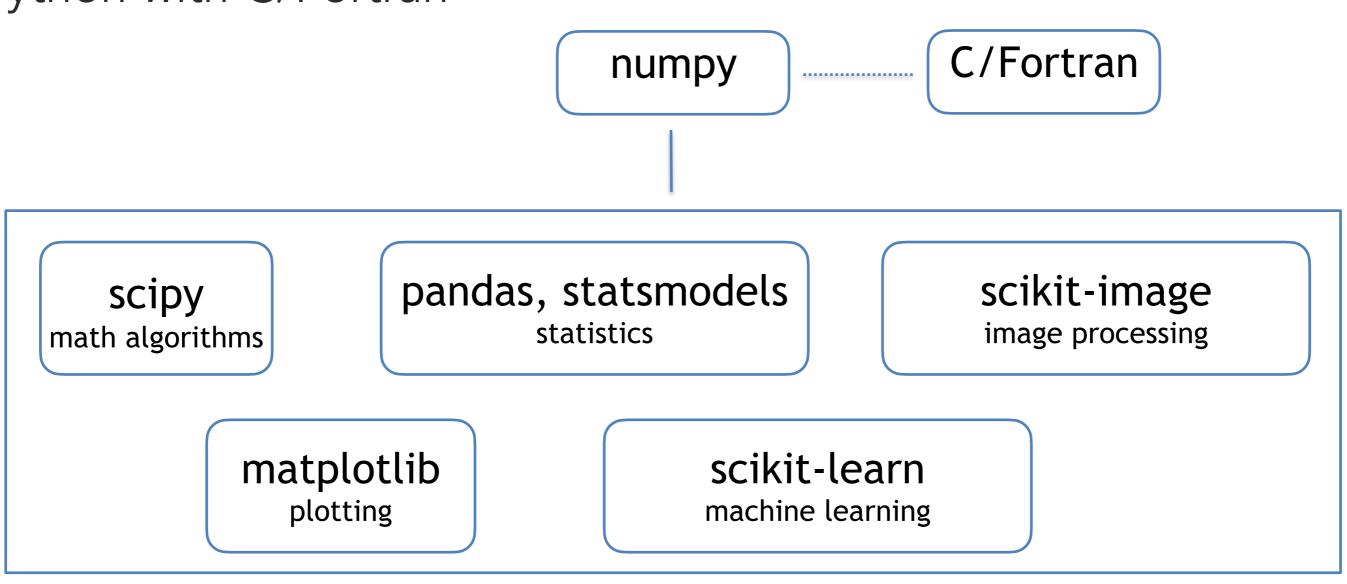
NumPy

- The fundamental package for numerical computing with Python
- Fast, memory-efficient N-dimensional arrays
- Excellent choice for large, homogeneous data sets
- A foundation for many mathematical packages, and to integrate Python with C/Fortran

NumPy

• The foundation for many numerical packages, and to integrate Python with C/Fortran



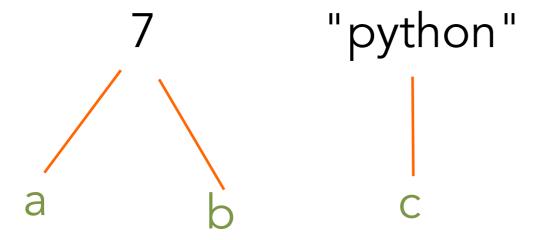
NumPy

- Deep learning frameworks reproduce the NumPy array interface
- Specialized array libraries are interoperable with NumPy

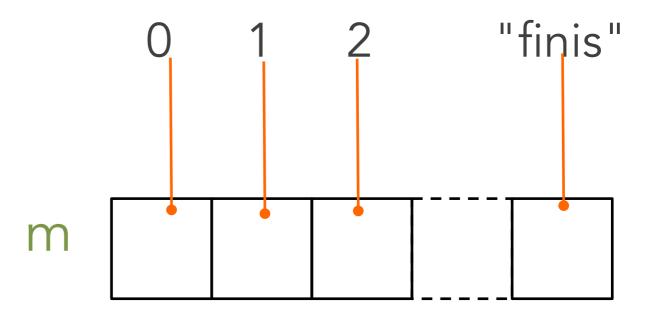
numpy deep-learning frameworks advanced array libraries Dask xarray distributed arrays labeled arrays Tensorflow **PyTorch** CuPy PyData/Sparse **MXNet** JAX arrays on GPUs sparse arrays

Python variables and lists

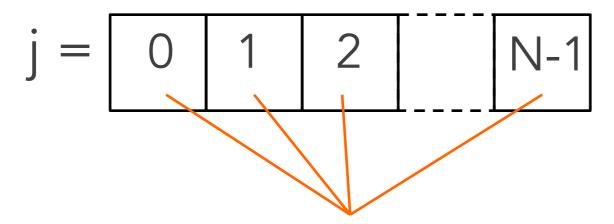
Python objects



python list



NumPy Arrays



actual data items; all have same byte size

v[j] ndim = 1 shape = (N,)

j,
$$k = \begin{bmatrix} 0.0 & 0.1 & 0.M-1 \\ 1.0 & 1.1 & 1.M-1 \\ N-1.0 & N-1.1 & N-1.M-1 \end{bmatrix}$$

$$ndim = 2$$

shape =
$$(N,M)$$

NumPy Data Types (dypes)

- Integers: numpy.int8, numpy.int16, numpy.int32, numpy.int64. Unsigned integers: numpy.uint8, etc.
- Floating point: numpy.float32, numpy.float64, numpy.float128.
 Complex floating point: numpy.complex64, etc.
- Others: numpy.bool_; numpy.string_, numpy.unicode_ (fixed length);
 numpy.void_ (composite); numpy.object_ (reference)