

*An array library for **nested, variable-sized data**, including arbitrary-length lists, records, mixed types, and missing data, using **NumPy-like idioms**.*

Awkward Array



a 3-minute update

*Arrays are **dynamically typed**, but operations on them are **compiled and fast**.*

Coincides with NumPy when arrays are regular; generalizes when they're not.

Peter Fackeldey and Ianna Osborne, Princeton University



Awkward Array Library Integrations

Awkward Array is a part of larger scientific Python ecosystem. It must work well with other libraries.



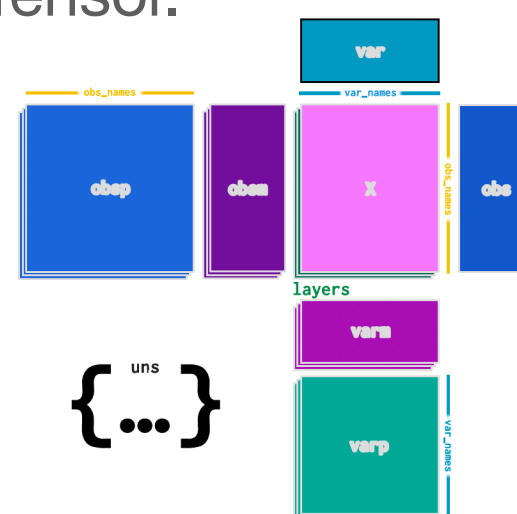
Lossless, bidirectional, zero-copy conversions between Awkward Array, Apache Arrow, Feather, and Parquet.



Bidirectional conversion between Awkward Array and TensorFlow RaggedTensor.



dask-awkward is a high-level collection type for Dask, alongside Array and DataFrame.



Integrated into AnnData for single-cell genomics.



JIT-compiled as a C++ iterator in ROOT's RDataFrame workflow.



Auto-differentiate through expressions involving Awkward Arrays, using JAX



Read any Kaitai Struct format into Awkward Arrays.



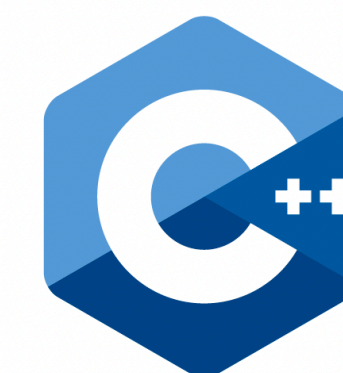
Awkward Arrays on GPUs, backed by CuPy.



Efficiently iterate over or build Awkward Arrays in code JIT-compiled with Numba, on CPUs and GPUs



Store Awkward Arrays in the Tiled database and slice them in the cloud.



JIT-compile Awkward Arrays in C++ with cppy. Header-only library to build Awkward Arrays in C++ and transfer them to Python.



RAPIDS

Akimbo: work with Awkward Arrays as DataFrame columns



Exchange data with the Julia language through a reimplement of the Awkward Array memory layouts.



Awkward Release Highlights

Wait, What? (v2.6.7 → v2.8.4)

- Named axis support for `ak.Array` - making Awkward Array's multidimensional irregular data manipulation readable and robust
- Virtual Arrays support - “read what you need”
- Enhanced error reporting and performance improvements
- Improved and more complete GPU support - CUDA backend
- Major progress on segmented reducers: `argmin`, `argmax`, `count_nonzero`
- Expanded Tensor framework support (TensorFlow, PyTorch, JAX, cuDF)
- Continuous compatibility maintenance (Python 3.13, Numpy 2.3, etc.)



Awkward Plan



Path Forward to Awkward Array Version 3

- We will streamline and unify the codebase to improve maintainability and make future development more efficient.
- Our focus will be on optimizing performance and ensuring the library scales with large and complex datasets.
- We aim to provide comprehensive GPU support, enabling acceleration for high-performance computing tasks.

We warmly invite the community to actively contribute, share feedback, and collaborate to help shape the future of Awkward Array.