# Awkward Arrays: Accelerating scientific data analysis on irregularly shaped data

Framework, OAC-2103945 PI: J. Pivarski (P. Elmer)

## What Awkward Array does

### Example of an “awkward” array

Record structures with differently typed fields

Array of variable-length lists (“ragged” or “jagged” arrays)

Nested variable-length lists

Missing data

Heterogenous data (union/variant types)

```
array = ak.Array([
    [{"x": 1.1, "y": [1]}, {"x": 2.2, "y": [1, 2]}, {"x": 3.3, "y": [1, 2, 3]}],
    [],
    [{"x": 4.4, "y": [1, 2, 3, 4]}, {"x": 5.5, "y": [1, 2, 3, 4, 5]}]
])
```

### NumPy-like expression

```
output = np.square(array["y", ..., 1:])
```

### Result

```
output.to_list()
  [
      [[], [4], [4, 9]],
      [],
      [[4, 9, 16], [4, 9, 16, 25]]
  ]
```

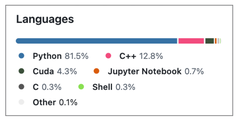
### Equivalent Python

```
output = []
for sublist in python_objects:
    tmp1 = []
    for record in sublist:
        tmp2 = []
        for number in record["y"][1:]:
            tmp2.append(np.square(number))
        tmp1.append(tmp2)
    output.append(tmp1)
```

(for a similar calculation 10 million times larger, single threaded)

**1.5 seconds**
**2.1 GB of memory**

**140 seconds**
**22 GB of memory**




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

# Awkward Array

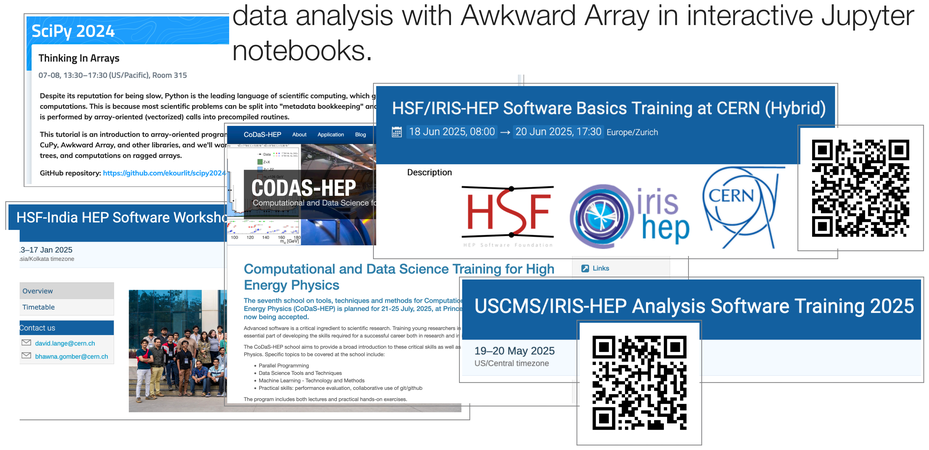
*Arrays are **dynamically typed**, but operations on them are **compiled and fast**. Coincides with NumPy when arrays are regular; generalizes when they're not.*

**Awkward Releases Highlights (Aug 2024 - now)**

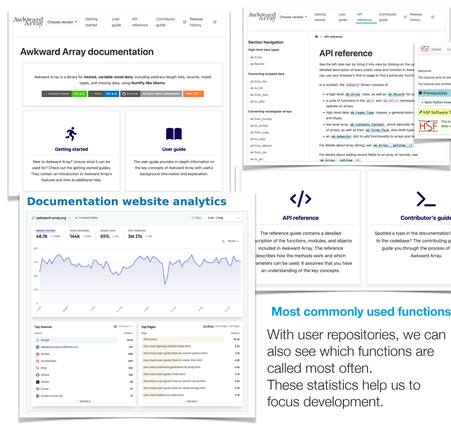
Named axis support for ak.Array - making Awkward Array's multidimensional irregular data manipulation readable and robust
Virtual Arrays support - “read what you need”
Enhanced error reporting and performance improvements
Improved and more complete GPU support - CUDA backend
Major progress on segmented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

## Training, Education and Outreach

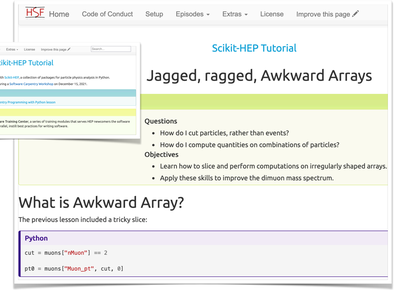
The most recent events that include teaching columnar data analysis with Awkward Array in interactive Jupyter notebooks.




## Documentation




## Self-study Tutorials




## Awkward Array Library Integrations

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

**Apache Arrow / Parquet**

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

**ROOT Data Analysis Framework**

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

**CuPy**

Awkward Arrays on GPUs, backed by CuPy.

**C++**

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

**TensorFlow**

Bidirectional conversion between Awkward Array and TensorFlow RaggedTensor.

**Numba**

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

**Auto-differentiate through expressions involving Awkward Arrays, using JAX**

**pandas / polars / RAPIDS**

Akimbo: work with Awkward Arrays as DataFrame columns

**DASK**

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

**bluesky**

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

**Integrated into AnnData for single-cell genomics.**

**Read any Kaitai Struct format into Awkward Arrays.**

**Awkward Array.jl**

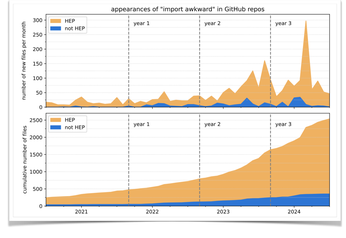
Exchange data with the Julia language through a reimplementation of the Awkward Array memory layouts.

## Adoption

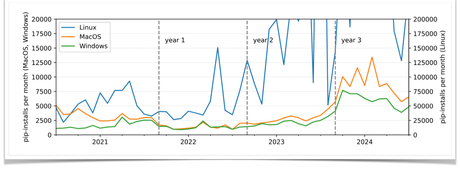
Awkward Array was developed in the context of particle physics

"Non-HEP" use-cases include
• neutrinos: P-ONE, CIEMAT-Neutrino, Harvard-Neutrino, LEGEND, DUNE, ICARUS, IceCube, nEXO, GraphNeT
• statistics: errors in annotated data (UKPLab/nessie), Baysean analysis (bat/batty), cmpatino/optimal-observables multivariate shapelets (bianchimario/MARS), text embedding (taylorai/mlx_embedding_models)
• climate science: Lagrangian ocean probes (Cloud-Drift/clouddrift), cloud microphysics (yoctoyotta1024/CLEO)
• biology: single-cell genetics (scverse), genotyping (kage-genotyper/kage), astrocytes (janreising/astroCAST)
• astronomy: active galactic nuclei (Zstone19/TempMap), dark matter direct detection (XENONnT/fuse)
• defense: identifying missile launches through ionic disturbances in GPS (tylerni7/missile-tid)
• other: simulation of fire (silvxlabs/DripTorch), food orders (EASS-HIT-PART-A-2022-CLASS-II/food-ordering), analyzing lastFM data (delannoy/lastfm-explorer), software (WayScience/software-landscape-analysis), polygon meshes (dewloosh/PolyMesh), CAD (j8sr0230/codelink, AxisVM/pyaxisvm), adarshchbs/pdf2md

Analysis of user code
We clone all our users' non-fork repositories and search for "import awkward" in Python and Jupyter files.




Download statistics
Frequency of "pip install awkward" is affected by batch jobs and automated testing that installs Awkward Array as a first step, but more so on Linux than MacOS and Windows.




**Most commonly used functions**

With user repositories, we can also see which functions are called most often. These statistics help us to focus development.

| calling pattern | count |
|---|---|
| ak.flatten(-) | 2832 |
| ak.num(-) | 2498 |
| ak.to_numpy(-) | 2193 |
| ak.sum(-, axis=1) | 874 |
| ak.flatten(-, axis=None) | 865 |
| ak.sum(-) | 564 |
| ak.ones_like(-) | 455 |
| ak.Array(-) | 406 |
| ak.concatenate(-) | 283 |

| function | #unique visitors | #views | avg. time |
|---|---|---|---|
| ak.Array | 785 | 1100 | 3m33s |
| ak.concatenate | 223 | 293 | 4m35s |
| ak.count | 210 | 265 | 4m20s |
| ak.flatten | 203 | 242 | 4m23s |
| ak.where | 202 | 262 | 3m54s |
| ak.num | 184 | 235 | 3m07s |
| ak.to_numpy | 181 | 218 | 3m25s |
| ak.mask | 178 | 231 | 3m52s |
| ak.zip | 163 | 221 | 5m02s |

## Team

Regular contributors to Awkward Array during the time of this grant, September 2021 – present.

52 contributors on GitHub