

# YOU DON'T HAVE TO BE AN EXPERT

Stories from the Open Source  
Frontlines

Alenka Frim

# OPEN SOURCE

... open source software (OSS), a global public good that plays a vital role in the economy and is foundational for most technology we use today ...

[https://www.hbs.edu/ris/Publication%20Files/24-038\\_51f8444f-502c-4139-8bf2-56eb4b65c58a.pdf](https://www.hbs.edu/ris/Publication%20Files/24-038_51f8444f-502c-4139-8bf2-56eb4b65c58a.pdf)

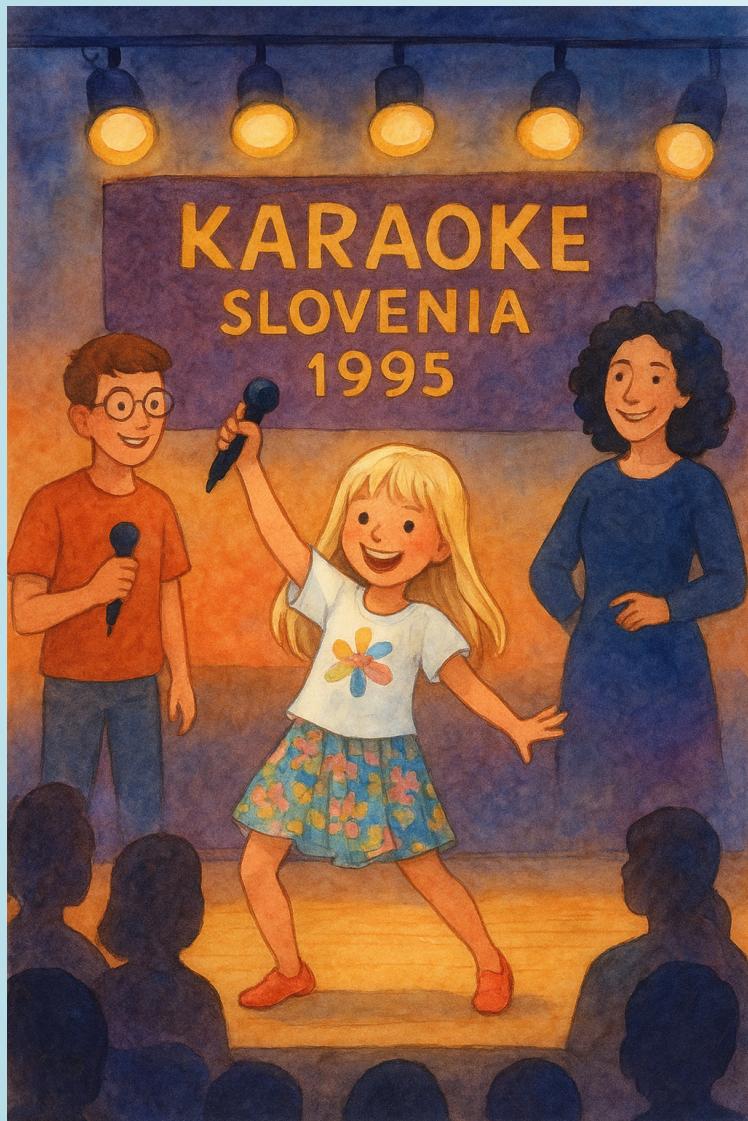
Open source fosters a collective approach to problem-solving.

<https://www.turing.ac.uk/blog/open-source-software-why-it-matters-and-how-get-involved>

Developing and deploying open source software is no longer just a novel idea. It's a strategic necessity in a fast-changing digital world.

<https://www.bcg.com/publications/2021/open-source-software-strategy-benefits>

# KARAOKE



# CAN THE DATA ECOSYSTEM BE OVERWHELMING?

Do you remember a time when you ...

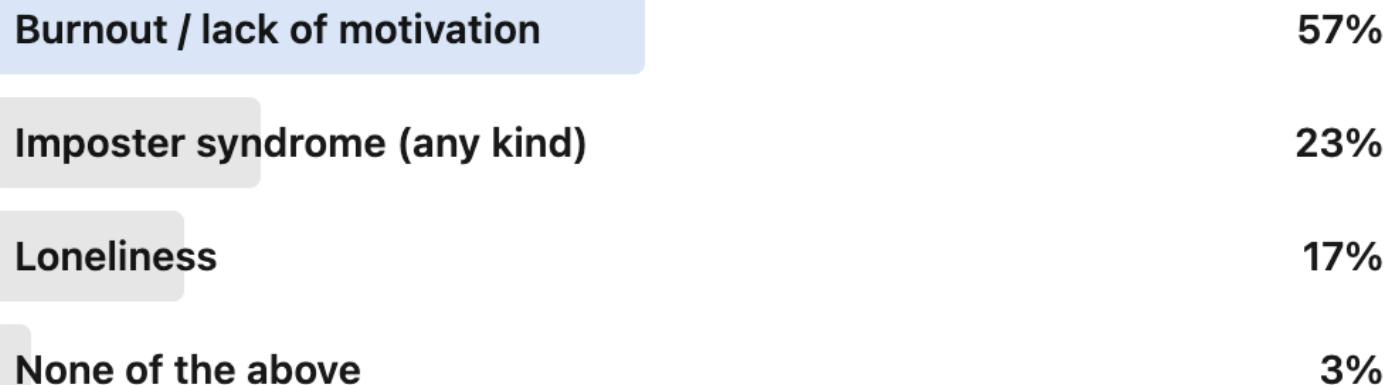
... felt you were not smart enough?

... you were afraid of being wrong?

# CAN THE DATA ECOSYSTEM BE OVERWHELMING?

Have you [a data scientist, engineer, developer, educator, open-source contributor/maintainer], in the past year, felt one of the following?

You can see how people vote. [Learn more](#)



**30 votes** • Poll closed

# DR. CAT HICKS

Psychologist for Software Teams

## Developer Thriving Learning Culture

People are always looking for cues to whether or not they are in a safe place for learning





# BUILDING PLAYFULNESS

# TYPE CHECKING SUPPORT

- **Rok Mihevc** wanted to find a solution for the Type Checking support in PyArrow
- took the opportunity to brainstorm ideas on EuroPython sprints



- Worked towards POC with **Patrick J. Roddy**

# TYPE ANNOTATIONS IN PYARROW

## [Python] Gradually add type checks to Arrow, initial step #45

 Open   rok wants to merge 10 commits into [main](#) from [gradual\\_pyarrow\\_stubs](#) 

 Conversation 11    Commits 10    Checks 24    Files changed 18

 rok commented last month • edited  ...

This proposes adding type annotation to pyarrow by adopting pyarrow-stubs into pyarrow. To do so we copy a subset of pyarrow-stubs's stubfiles into pyarrow tree. We then add annotation checks for some stubsfiles and some test files and make sure checks pass. Annotation checks should be expanded until all (or most) project files are covered in future work.

PR introduces:

1. adds [pyarrow-stubs](#) into `arrow/python/pyarrow/`
2. fixes pyarrow-stubs to pass mypy and pyright check
3. adds mypy and pyright check to CI (crudely)
4. adds a tool (`update_stub_docstrings.py`) to keep annotation docstrings in sync with source docstrings
5. adds docstring sync check to CI (crudely)

 1    1    1    1

# TYPE ANNOTATIONS IN PYARROW

```
import pyarrow.compute as pc
```

```
pc.
```

- ↑ ⚡ all(array, skip\_nulls, min\_count, options) → pyarrow.compute
- ↓ ⚡ scalar(value) → pyarrow.compute
- ↑ ⚡ array\_take(array, indices, boundscheck=True) → pyarrow.compute
- ↑ ⚡ sum(array, skip\_nulls, min\_count, options) → pyarrow.compute
- ↑ ⚡ array\_filter(array, selection\_filter, safe=True) → pyarrow.compute
- ↓ ⚡ cast(arr, target\_type, safe, options) → pyarrow.compute
- ↓ ⚡ abs(x, memory\_pool) → pyarrow.compute
- ↓ ⚡ abs\_checked → pyarrow.compute

# TYPE ANNOTATIONS IN PYARROW

```
import pyarrow.compute as pc
pc.strptime()
```

⌚ pyarrow.compute

```
def strftime(timestamps: Date32Scalar | Date64Scalar | Time32Scalar[Any] | TimestampScalar[Any] | DurationScalar[Any] | MonthDayNanoIntervalScalar | Date64Scalar | Time32Scalar[Any] | Time64Scalar[Any] | TimestampScalar[Any] | MonthDayNanoIntervalScalar) | ChunkedArray[Date32Scalar | Date64Scalar | Time32Scalar[Any] | Time64Scalar[Any] | TimestampScalar[Any] | DurationScalar[Any] | MonthDayNanoIntervalScalar] | MemoryPool) -> StringScalar | String[] | String[MemoryPool]
```

Format temporal values according to a format string.

For each input value, emit a formatted string. The time format string and locale can be set using the "%S" (seconds) format code depends on the input time precision: it is an integer for timestamps with the required number of fractional digits for higher precisions. Null values emit null. An error occurs if the specified timezone is invalid or if the specified locale does not exist.

Params:

- `timestamps` – Argument to compute function.

- `format` – Pattern for formatting input values.

# APACHE ARROW

Would I be any good at software development in Open source?

## Official Implementations

The Apache Arrow project houses a collection of libraries for different programming languages. Use the links in the table below to access the documentation and source code for these libraries.

Language	Docs	Source
C++	<a href="#">C++ Docs</a>	<a href="#">C++ Source</a>
C GLib	<a href="#">C GLib Docs</a>	<a href="#">C GLib Source</a>
C#	<a href="#">C# Docs</a> ↗	<a href="#">C# Source</a>
Go	<a href="#">Go Docs</a> ↗	<a href="#">Go Source</a>
Java	<a href="#">Java Docs</a>	<a href="#">Java Source</a>
JavaScript	<a href="#">JavaScript Docs</a> ↗	<a href="#">JavaScript Source</a>
Julia	<a href="#">Julia Docs</a> ↗	<a href="#">Julia Source</a>
MATLAB	<a href="#">MATLAB Docs</a> ↗	<a href="#">MATLAB Source</a>
Python	<a href="#">Python Docs</a>	<a href="#">Python Source</a>
R	<a href="#">R Docs</a> ↗	<a href="#">R Source</a>
Ruby	<a href="#">Ruby Docs</a> ↗	<a href="#">Ruby Source</a>
Rust	<a href="#">Rust Docs</a> ↗	<a href="#">Rust Source</a>
Swift	<a href="#">Swift Docs</a> ↗	<a href="#">Swift Source</a>



# ARROW R PACKAGE

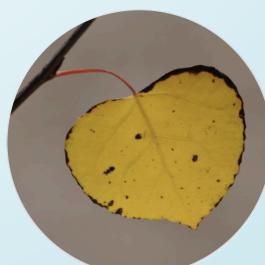
Lots of good first issues!

<input type="checkbox"/> Open	15	Closed	45	Author	Labels	Projects	Milestones	Assignees
<input type="checkbox"/>	<span>●</span>	[R] Update R package to use R 4.1+ native forward pipe syntax	<span>Component: R</span>	<span>good-first-issue</span>	<span>Type: enhancement</span>			
		#47106 · amoeba opened on Jul 14 · 22.0.0						
<input type="checkbox"/>	<span>●</span>	[R] concat_tables on a record_batch causes segfault	<span>Component: R</span>	<span>good-first-issue</span>	<span>Type: bug</span>			
		#47000 · warnes opened on Jul 5						
<input type="checkbox"/>	<span>●</span>	[R] Update the docs to show how to avoid situations like data loss with leading zero in partition column	<span>Component: R</span>	<span>good-first-issue</span>	<span>Type: enhancement</span>			
		#39660 · arthurgaines opened on Jan 17, 2024						

# Lots of support from the maintainers!

<a href="#">XX apache/arrow ARROW-13137: [C++][Documentation] Make in-table references consistent</a>	5
#10630 by AlenkaF was closed on Jul 5, 2021 • Approved	
<a href="#">XX apache/arrow ARROW-12867: [R] Bindings for abs() Component: R</a>	24
#10519 by AlenkaF was closed on Jun 21, 2021 • Changes requested	
<a href="#">XX apache/arrow ARROW-12198: [R] bindings for strftime Component: R</a>	46
#10334 by AlenkaF was closed on Jun 4, 2021 • Approved	

# HELPED ME BUILD MY COMMUNITY



# BUILDING CURIOSITY



# APACHE ARROW FORMAT

Apache Arrow defines a language-independent columnar memory format for flat and nested data, organized for efficient analytic operations on modern hardware like CPUs and GPUs.

# APACHE ARROW FORMAT

The Arrow memory format also supports zero-copy reads for lightning-fast data access without serialization overhead.

# APACHE ARROW FORMAT

Apache Arrow defines a language-independent columnar memory format for flat and nested data, organized for efficient analytic operations on modern hardware like CPUs and GPUs.

The Arrow memory format also supports zero-copy reads for lightning-fast data access without serialization overhead.

# YOGIC APHORISMS

| Yogaś citta vṛtti nirodhaḥ

Chitta is the consciousness which includes the mind, the intellect and the ego. Yoga is a method of silencing the vibrations of the chitta.

~ <https://bksiyengar.com/modules/iyoga/iyoga.htm>

# Single reading

vs

## re-reading

## reading multiple texts



# DOCUMENTATION IN OPEN SOURCE

- do the documentation!
- elevate documentation skills
- don't look down on documentation work!

# ★ DOCUMENTATION IN OPEN SOURCE

- 💡 aphorism structure

Aphorism → Translation → Commentary

## PyArrow Tables

Instances of `pyarrow.Table`, a logical table data structure in which each column consists of one or more `pyarrow.Array` objects of the same type.

<http://arrow.apache.org/docs/python/data.html#tables>

# BUILDING BRAVERY

## IMPACT OF OS

and open standards!

# ★ TABULAR DATA STRUCTURES

Python

- select a dataframe library to work with tabular data

R

- built in columnar tabular data structure called  
`data.frame`

| Fragmentation of the Python dafatframe libraries.

# DATAFRAME INTERCHANGE PROTOCOL

In 2021 **Ralf Gommers** from Quansight took on the task to:

- facilitate the discussion
- to construct a **dataframe interchange protocol**
- allow converting one type of dataframe into another

# DATAFRAME INTERCHANGE PROTOCOL



Array API

DataFrame API

Blog

Annual reports

## Consortium for Python Data API Standards

Consortium includes representatives from both open source Python libraries and the industry.

# LIBRARY INSTEAD OF A PROTOCOL

**Marco Gorelli**

Narwhals, the dataframe  
“translator”

compatibility layer  
between Python  
dataframe libraries

Narwhals



# ADOPTING A WIDER STANDARD

**Joris Van den Bossche and Will Jones**

- wanted to use the Apache Arrow columnar format
- de facto standard
- all about zero-copy

Enable data interchange without the need to depend on PyArrow.

# ARROW PYCAPSULE PROTOCOL

## C data interface

Implemented in 2020 by **Antoine Pitrou**

Inspired by the Python buffer protocol.

Later additions: C stream interface and C device interface

## Capsules

part of Python C API

Envelopes for the C Data Interface structs

# More content on the Arrow PyCapsule Protocol: The expanding Apache Arrow universe

## Joris Van den Bossche | PyData Paris 2024



# EXAMPLE BEFORE

```
1 df_polars = pl.DataFrame({"a": [1,2], "b": [0.1, 0.2]})  
2  
3 # Convert polars do datafusion  
4 ctx = datafusion.SessionContext()  
5 df_datafusion = ctx.from_arrow(df_polars.to_arrow())  
6  
7 # and convert back to polars  
8 pl.DataFrame(df_datafusion.to_arrow_table())
```

# EXAMPLE AFTER

```
1 df_polars = pl.DataFrame({"a": [1,2], "b": [0.1, 0.2]})  
2  
3 # Convert polars do datafusion  
4 ctx = datafusion.SessionContext()  
5 df_datafusion = ctx.from_arrow(df_polars)  
6  
7 # and convert back to polars  
8 pl.DataFrame(df_datafusion)
```

# COMMUNITY BENEFITS

 Open [Python] Promote usage of the Arrow PyCapsule Protocol (for the C Data Interface) #39195

 kylebarron on Jul 23, 2024 · edited by kylebarron Edits Contributor ...

I've been working a bit to promote the protocol; here's a running tally:

**implemented:** (at least partially, return objects with pycapsule protocol and/or check for existence of protocol in constructors)

- pandas: ↗ ENH: support the Arrow PyCapsule Interface on pandas.DataFrame (export) pandas-dev/pandas#56587, ↗ ENH: support the Arrow PyCapsule Interface on pandas.Series (export) pandas-dev/pandas#59518, ↗ ENH: support Arrow PyCapsule Interface on Series for export pandas-dev/pandas#59587, ↗ ENH: add basic DataFrame.from\_arrow class method for importing through Arrow PyCapsule interface pandas-dev/pandas#59696
- ibis: ↗ feat(pyarrow): support Arrow PyCapsule interface on ibis.Table objects ibis-project/ibis#9143, ↗ feat(pyarrow): support \_\_arrow\_c\_schema\_\_ on ibis.Schema objects ibis-project/ibis#9665
- pyarrow
- nanoarrow
- arro3
- ionboard
- pyogrio
- gdal ↗ Python bindings: add a ogr.Layer.WriteAll() method consuming \_\_arrow\_c\_stream\_\_ or \_\_arrow\_c\_array\_\_ interfaces OSGeo/gdal#9133, ↗ Python bindings: implement \_\_arrow\_c\_stream\_\_() interface for ogr.Layer OSGeo/gdal#9043

# THANK YOU!

GitHub profile



Slides

