My NumPy year

From no CPython C API experience
to shipping a new DType in NumPy 2.0

Nathan Goldbaum (Quansight Labs)

My NumPy year

- Why did NumPy string arrays need to be fixed?
- How did the community fund this work?
- How did I become a NumPy maintainer?
- How do I start working on a big project?
- What cool new feature did I add to NumPy?

A brief history of strings in NumPy

String arrays in Python 2

```
Python 2.7.18 (default, Jul  1 2024, 10:27:04)

>>> import numpy as np
```

String arrays in Python 2

```
Python 2.7.18 (default, Jul  1 2024, 10:27:04)

>>> import numpy as np


>>> np.array(["hello", "world"])

array(['hello', 'world'], dtype='|S5')
```

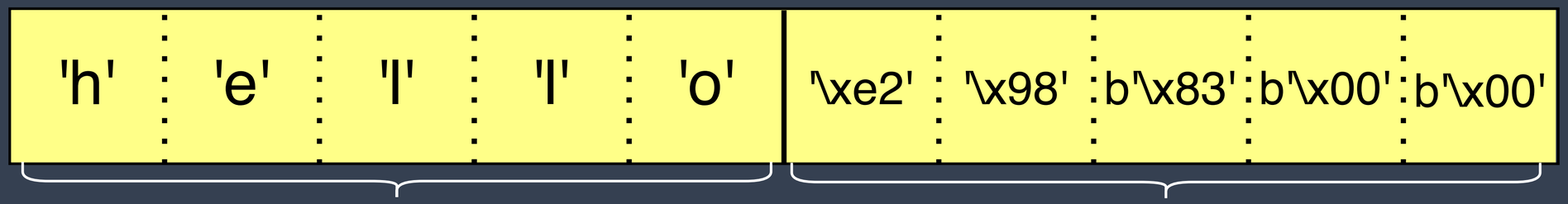
String arrays in Python 2

```
Python 2.7.18 (default, Jul  1 2024, 10:27:04)

>>> import numpy as np


>>> np.array(["hello", "world"])

array(['hello', 'world'], dtype='|S5')


>>> np.array(['hello', '☃'])

array(['hello', '\xe2\x98\x83'], dtype='|S5')
```

String arrays in Python 2

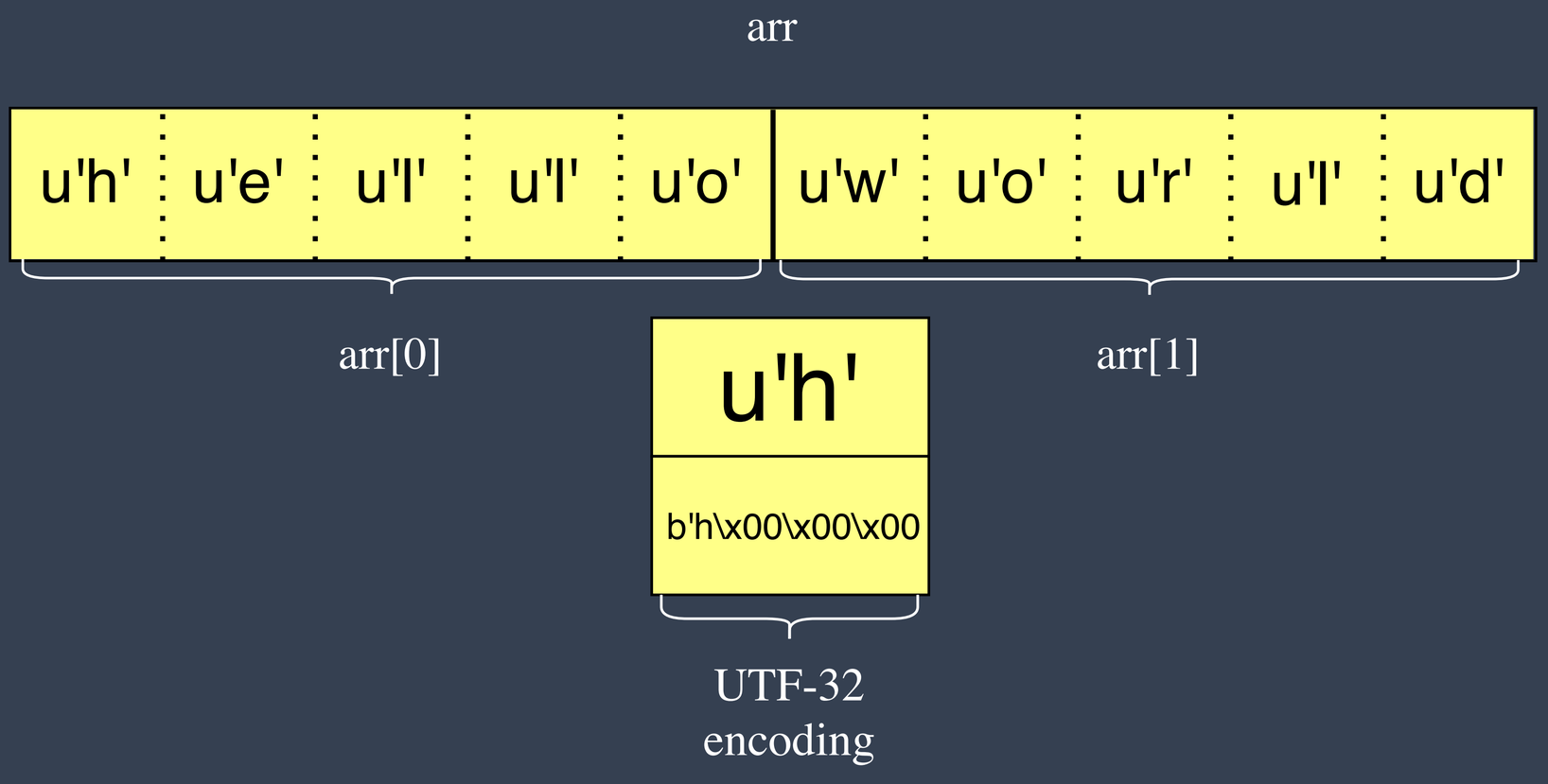
```
Python 2.7.18 (default, Jul  1 2024, 10:27:04)

>>> import numpy as np


>>> np.array(['hello', '☃'])

array(['hello', '\xe2\x98\x83'], dtype='|S5')
```

Memory layout of a NumPy string array



String Arrays in Python 2

```
>>> arr = np.array([u'hello', u'world'])
>>> arr
array([u'hello', u'world'], dtype='<U5')
```



String Arrays in Python 3

```
>>> arr = np.array(['hello', 'world'])
>>> arr
array(['hello', 'world'], dtype='<U5')
```

Pragmatism: Make existing unicode DType the default!

String Arrays in Python 3

```
>>> arr = np.array(['hello', 'world'])
>>> arr
array(['hello', 'world'], dtype='<U5')

>>> arr.tobytes()
```

String Arrays in Python 3

```
>>> arr = np.array(['hello', 'world'])
>>> arr
array(['hello', 'world'], dtype='<U5')

>>> arr.tobytes()
b'h\x00\x00\x00e\x00\x00\x00l\x00\x00\x00l\x00\x00\x00o\x00\x0
0\x00w\x00\x00\x00o\x00\x00\x00r\x00\x00\x00l\x00\x00\x00d\x00
\x00\x00'
```

40 bytes to store 10 ASCII characters!

String operations were inefficient too!

Writing fast string ufuncs for NumPy 2.0

Published May 21, 2024

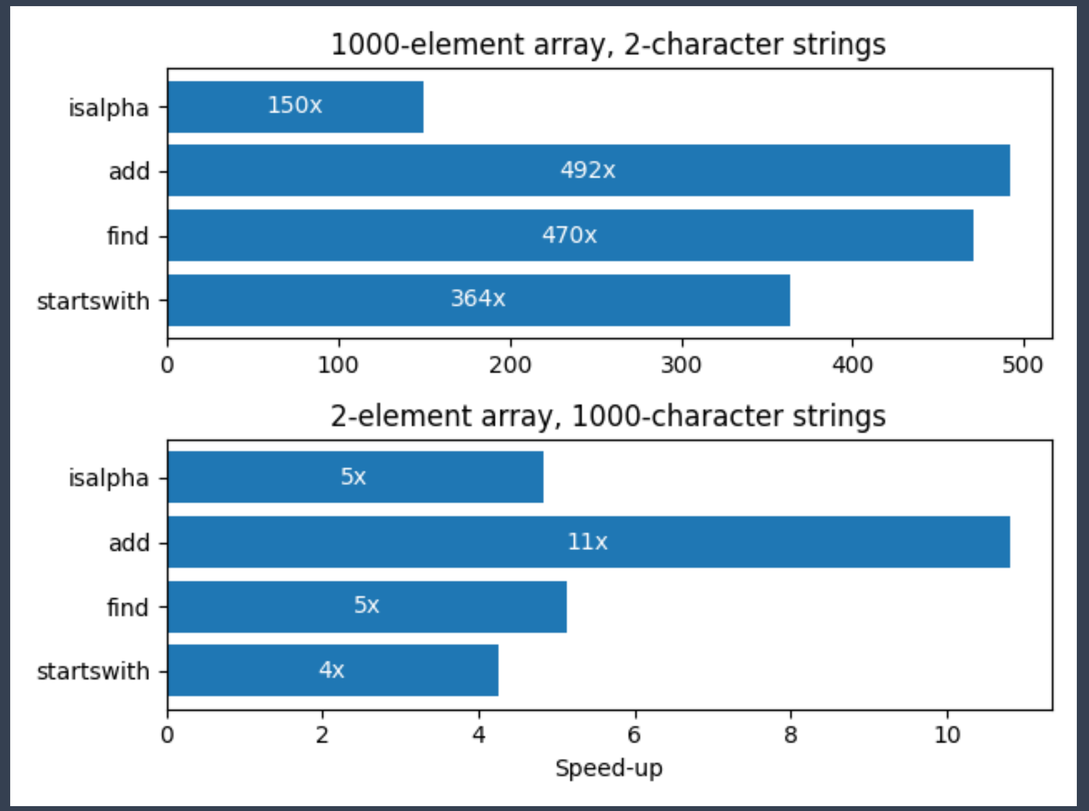
labs.quansight.org/blog/numpy-string-ufuncs

lysnikolaou

Lysandros Nikolaou

Quansight Labs

```
// Get the corresponding str method
method = PyObject_GetAttr((PyObject *)&PyUnicode_Type, method_name);

<loop over the array> {
  // Create Python object out of the array item
  PyObject* item = PyArray_ToScalar(in_iter->dataptr, in_iter->ao);

  // call corresponding str method
  PyObject *item_result = PyObject_CallFunctionObjArgs(method, item, NULL);

  // add result to output array
  PyArray_SETITEM(resultarr, PyArray_ITER_DATA(out_iter), item_result);
}
```



Object arrays

```
>>> arr = np.array(
    ['this is a very long string', np.nan, 'another string'],
    dtype=object
)
```

Object arrays

```
>>> arr = np.array(
    ['this is a very long string', np.nan, 'another string'],
    dtype=object
)

>>> arr
array(['this is a very long string', nan, 'another string'],
      dtype=object)
```

Object arrays

```
>>> arr = np.array(
    ['this is a very long string', np.nan, 'another string'],
    dtype=object
)

>>> np.isnan(arr[1])
np.True_
```

Object arrays

```
>>> arr = np.array(
    ['this is a very long string', np.nan, 'another string'],
    dtype=object
)

>>> np.isnan(arr[1])
np.True_

>>> type(arr[0])
str
```

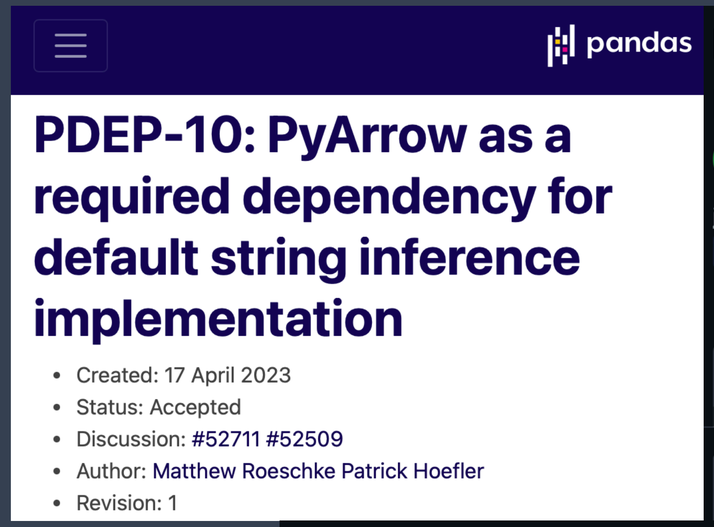
Ecosystem-wide technical debt

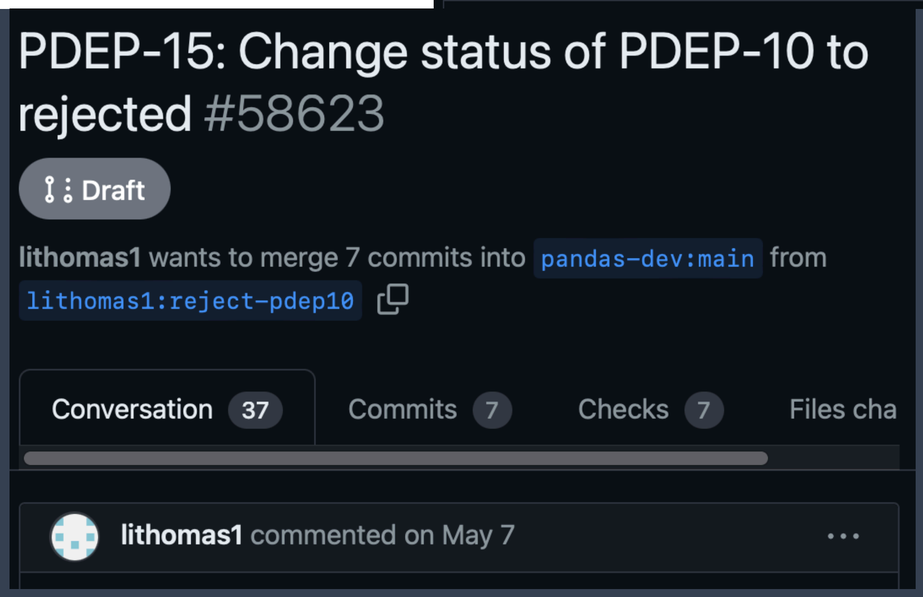
Community Impact: pandas

```
In [1]: df = pd.DataFrame(
   ...: {'names':
   ...:       [
   ...:          'Marie Curie',
   ...:          'Jane Goodall',
   ...:       ]
   ...: })

In [2]: df['names']
Out[2]:
0     Marie Curie
1    Jane Goodall
Name: names, dtype: object
```

Community Impact: pandas

```
In [1]: df = pd.DataFrame(
   ...: {'names':
   ...:       [
   ...:           'Marie Curie',
   ...:           'Jane Goodall',
   ...:       ]
   ...: })

In [2]: df['names']
Out[2]:
0      Marie Curie
1    Jane Goodall
Name: names, dtype: object
```





Community Impact: astropy

Community Impact: astropy

Table unicode sandwich - make 'S' type useful in Python 3 #5700

Merged

taldcroft merged 16 commits into `astropy:master` from `taldcroft:table-unicode-sandwich` on May 30, 2017

Conversation 66 | Commits 16 | Checks 0 | Files ch

taldcroft commented on Jan 14, 2017 ···

Failure of unicode sandwich for multi-dimensional columns #9614

Open **mhvk** opened this issue on Nov 17, 2019 · 3 comments

Labels: Bug | Effort-medium | Package-expert | table

taldcroft commented on Apr 2, 2020 ···

I suspect this is a bit of a rabbit hole... with new numpy dtypes is there hope for a single-byte character type anywhere on the horizon before I'm retired?

Collective Brainstorming

The only string type available in Python3 is np.unicode which uses 4-byte utf-32 encoding which is deemed to use too much memory to actually see much use.

What people apparently want is a string type for Python3 which uses less memory for the common science use case.

Julian Taylor jtaylor.debian at googlemail.com
Thu Apr 20 09:15:27 EDT 2017

Collective Brainstorming

The only string type available in Python3 is np.unicode which uses 4-byte utf-32 encoding which is deemed to use too much memory to actually see much use.

What people apparently want is a string type for Python3 which uses less memory for the common science use case.

Julian Taylor jtaylor.debian at googlemail.com
Thu Apr 20 09:15:27 EDT 2017

`numpy-discussion` mailing list thread

NumPy Roadmap

This is a live snapshot of tasks and features we will be investing resources in. It may be used to encourage and inspire developers and to search for funding.

- Allow adding (a) new string dtype(s). This could be encoded strings with fixed-width storage (e.g., `utf8` or `latin1`), and/or a variable length string dtype. The latter could share an implementation with `dtype=object`, but be explicitly type-checked. One of these should probably be the default for text data. The current string dtype support is neither efficient nor user friendly.

https://numpy.org/neps/roadmap.html

YEARS LATER...

NASA ROSES Grant

- Memory usage improvements
- Support new NumPy String DType
- Numba

- Support Pandas dataframes
- Array API support
- Cython modernization

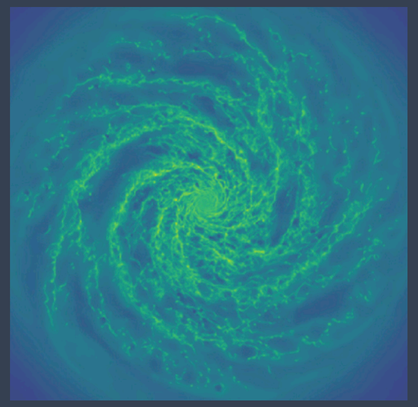
- Linear programming optimizations
- Array API support

- Variable-length String DType
- SIMD improvements

All projects

- Maintenance
- Build and packaging
- Benchmarking infrastructure improvements

My background



Simulations of galaxies on supercomputers

2009-2015

1990s C++
Python

Open Source Maintainer

2015-2019

Cython
Maintainer Skills

Recurse Center

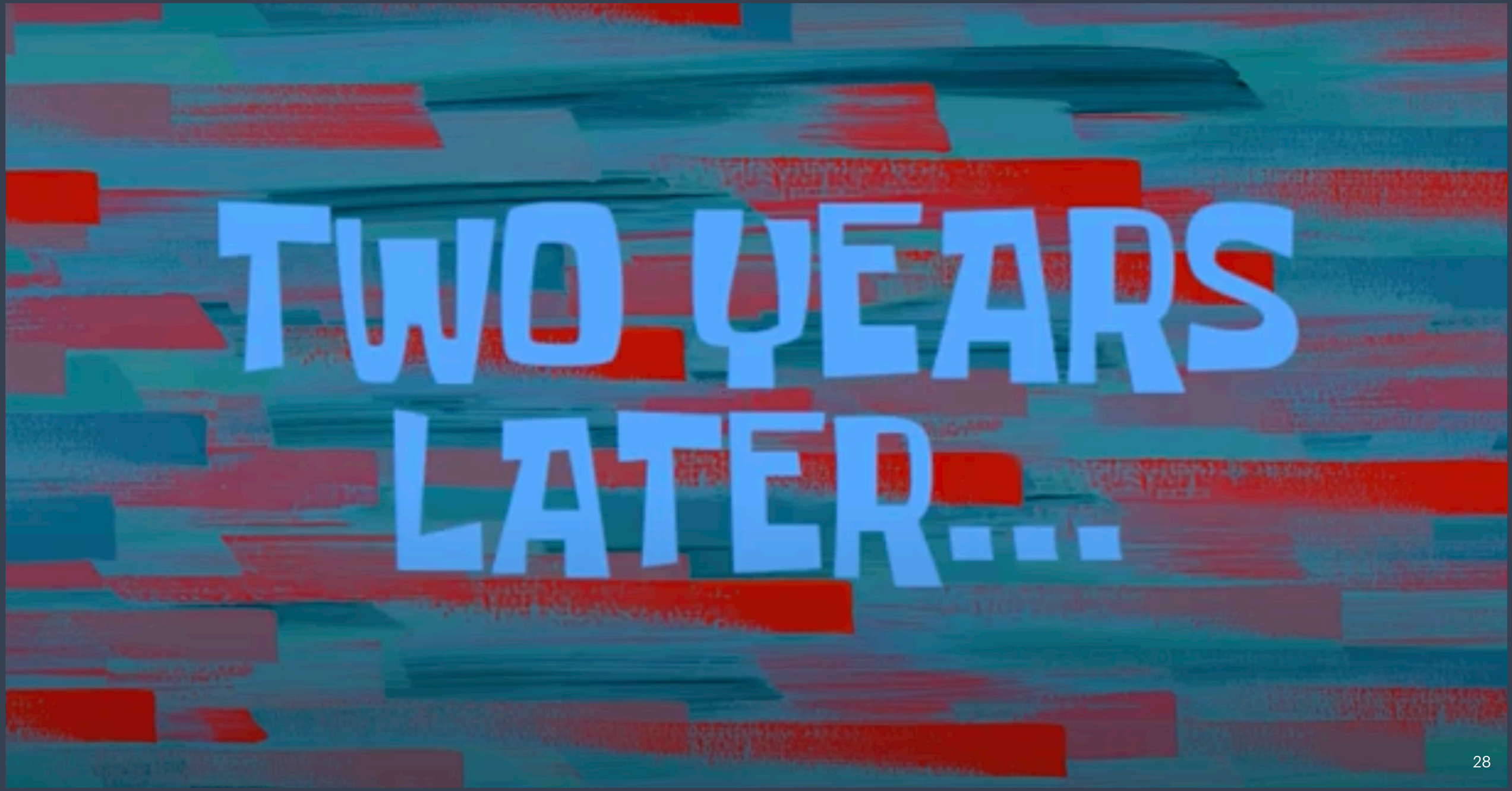
2019

Rust

Software Engineer

2019-2020

Modern C++
Pybind11


TWO YEARS
LATER...

Software roles at Quansight?

Inbox ×

Nathan <nathan.goldbaum@gmail.com> Oct 3, 2022, 12:06 PM

to ralf.gommers

Hi Ralf,

It's been a while, I hope you're doing well!

I'm checking in on the off chance you're looking for someone with my background for software roles at Quansight. Since I left back in 2020 I've been trying to focus on my mental health as well

My task

- Build a variable-width string DType
- In C, using the NumPy C API
- Ship the DType in NumPy

How to draw an owl

1.

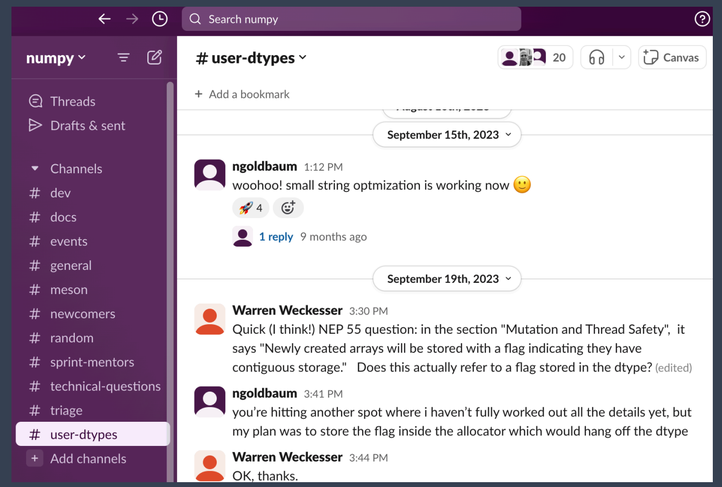
2.

1. Draw some circles

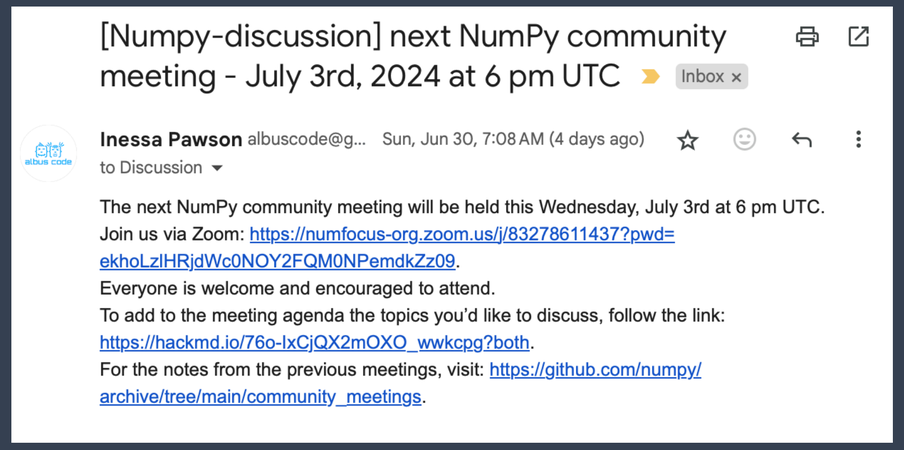
2. Draw the rest of the f !?&@ g owl

Getting started and Getting Help

- Ask questions in public channels with no expectation of an answer.
- Look for interesting-sounding or project-relevant bugs and try to fix them.
- Try to meet people face-to-face.
  - Projects often have regular meetings
  - Scheduled call with an interested maintainer.



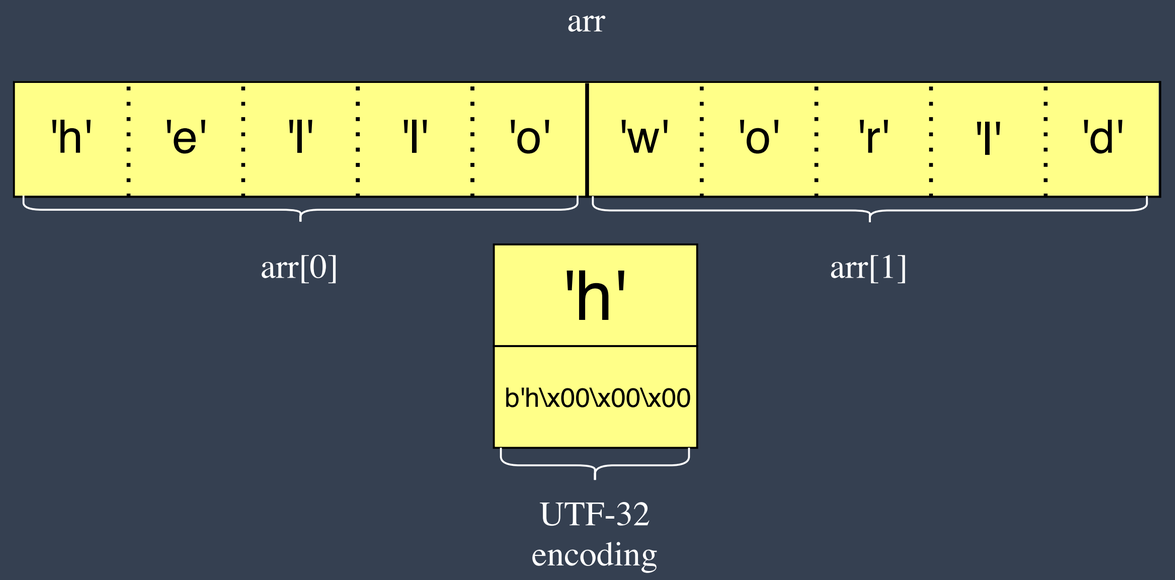
Channels for async chat



Face-to-face zoom meetings

How to store variable-width strings in an array?

- Variable-width strings break the assumption that each array element has a fixed size in memory.
  - How does indexing work?
  - For other arrays,
    a[i] = a + i*a.dtype.itemsize
- Store an array of offsets into the array in a sidecar buffer?
  - NumPy only stores data in the array buffer, no way to store data elsewhere.

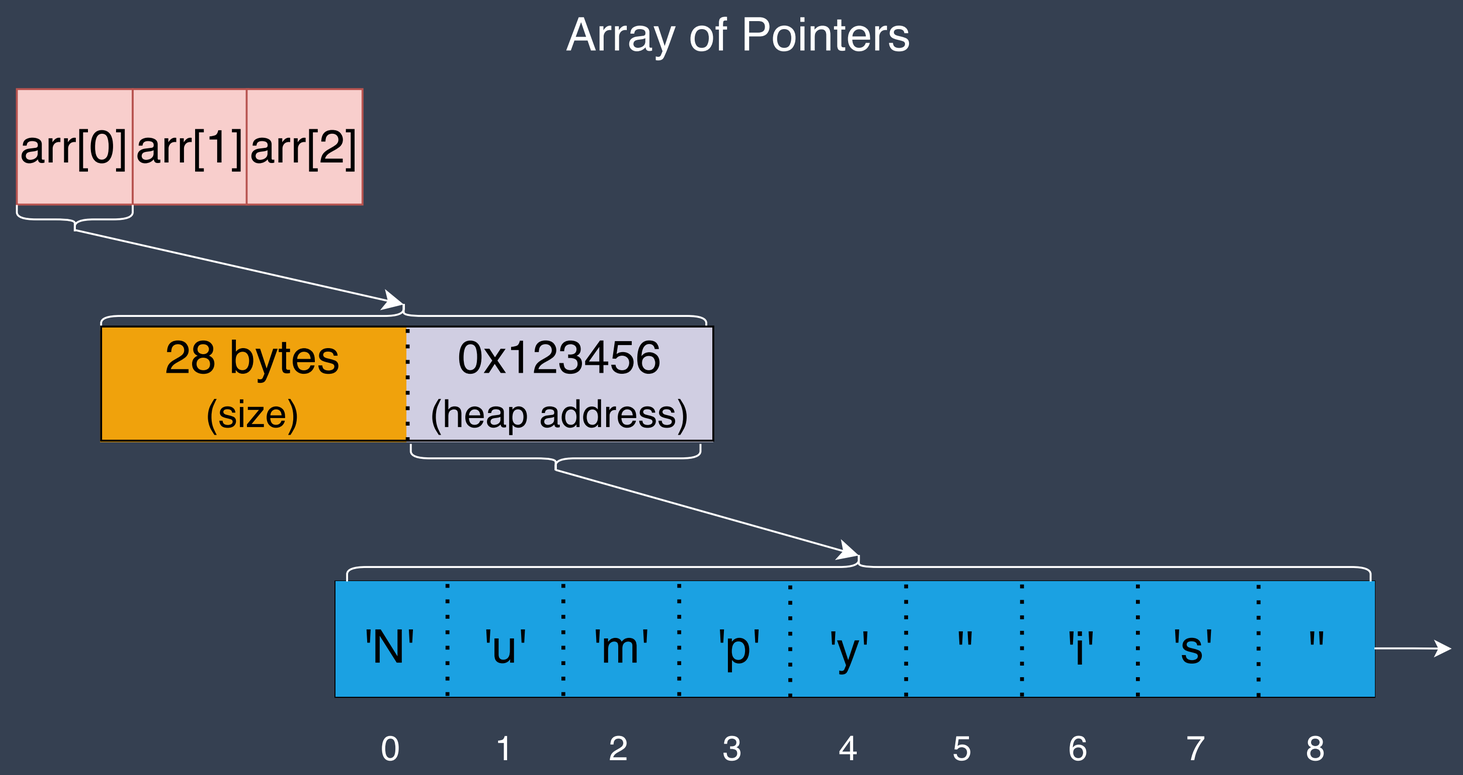


Sebastian Berg

Peyton Murray

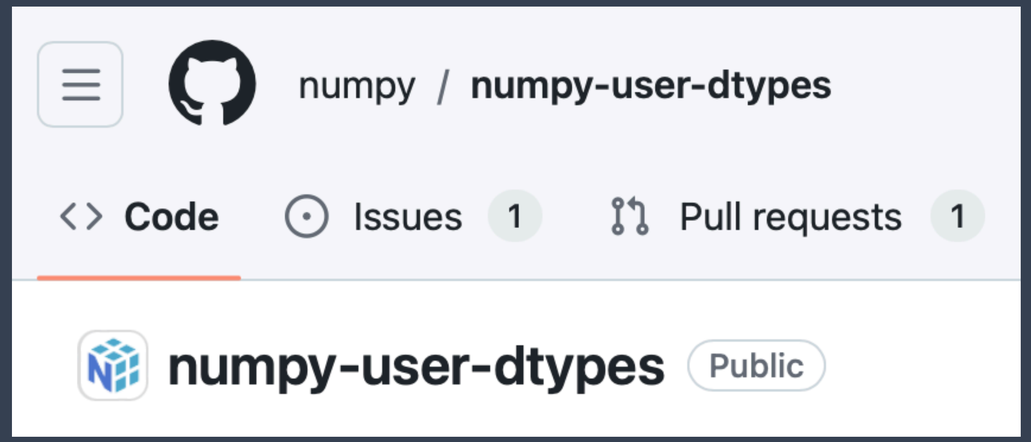
Idea: array of pointers

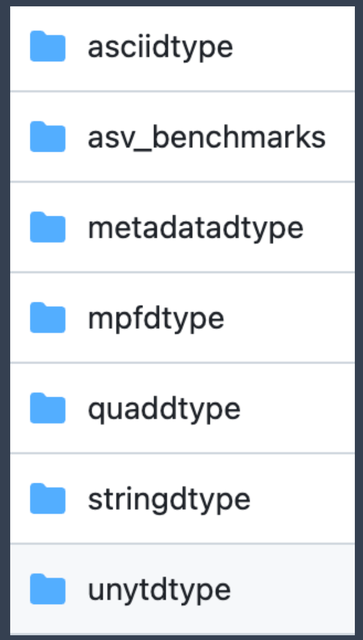
```
>>> arr = np.array(['numpy is a very cool library', 'hello', 'world'],
                   dtype=np.dtypes.StringDType())
>>> arr
array(['numpy is a very cool library', 'hello', 'world'],
      dtype=StringDType())
```



Building a DType from Scratch

- Start simple and build on that
- Prototypes in a quiet repo with code review from interested developers.
- MetadataDType
  - Attach a dict to a float64 DType
- ASCIIDType
  - Fixed-width ASCII text DType
- Finally, I built a real UTF-8 string DType





About

Repository for example user DTypes using the new API

The NumPy 2.0 DType API

- C API for registering user-defined DTypes and operation loops.
  - It exists! It's public now!
- Possibilities for DTypes
  - Physical units
  - Reduced, extended, or arbitrary-precision floats
  - Categoricals

Sound interesting?

Come to the NumPy Sprint!

Getting it done

Real thoughts I had working on this project

- This is too hard.
- I'll never be able to finish
- I don't know how to proceed on any of the unfixed issues.
- I don't want to write code today.
- I don't understand why this bug is happening.
- I asked a question to try and get some help but the answer I got doesn't make any sense.

Ways forward from mental blocks

- Go for a walk.
- Hard things take time. It's OK to struggle.
- Take notes.
- It's OK to talk to people about your problem and ask questions to help clarify things.
- Improve your programming workflow
- Talk to people about the problems you're running into. Be open to new ideas.

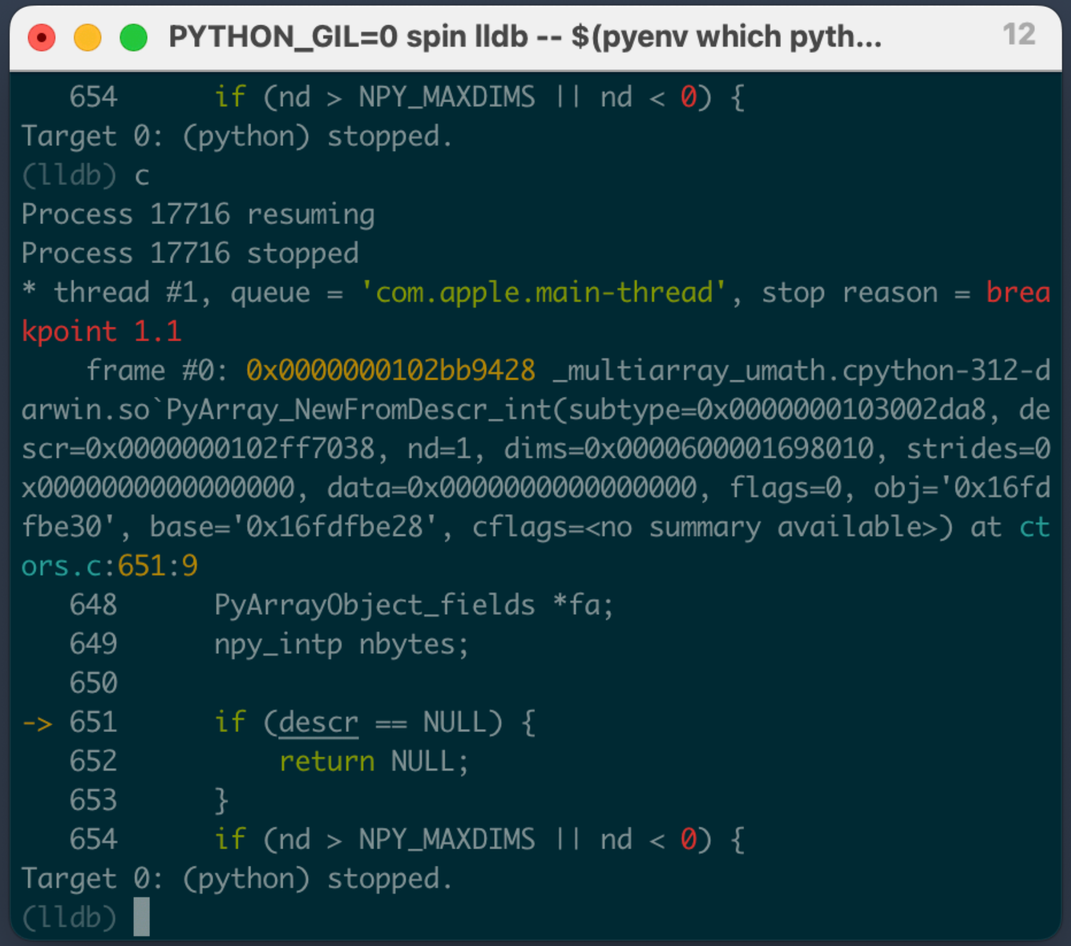
Ways forward from mental blocks

- Go for a walk.
- Hard things take time. It's OK to struggle.
- Take notes.
- It's OK to talk to people about your problem and ask questions to help clarify things.
- Improve your programming workflow
- Talk to people about the problems you're running into. Be open to new ideas.

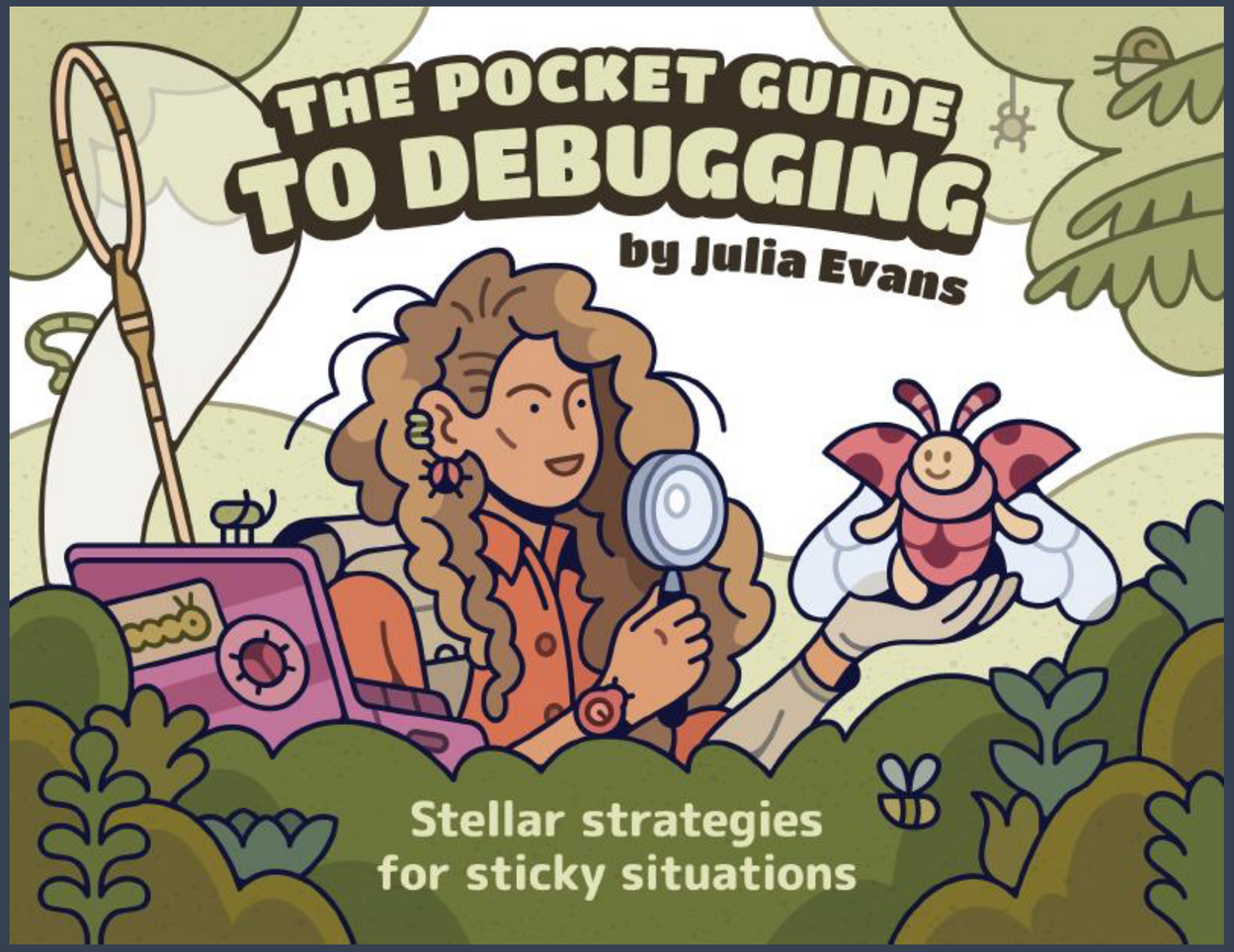
(I just wanted to have a picture of my cats in this talk)

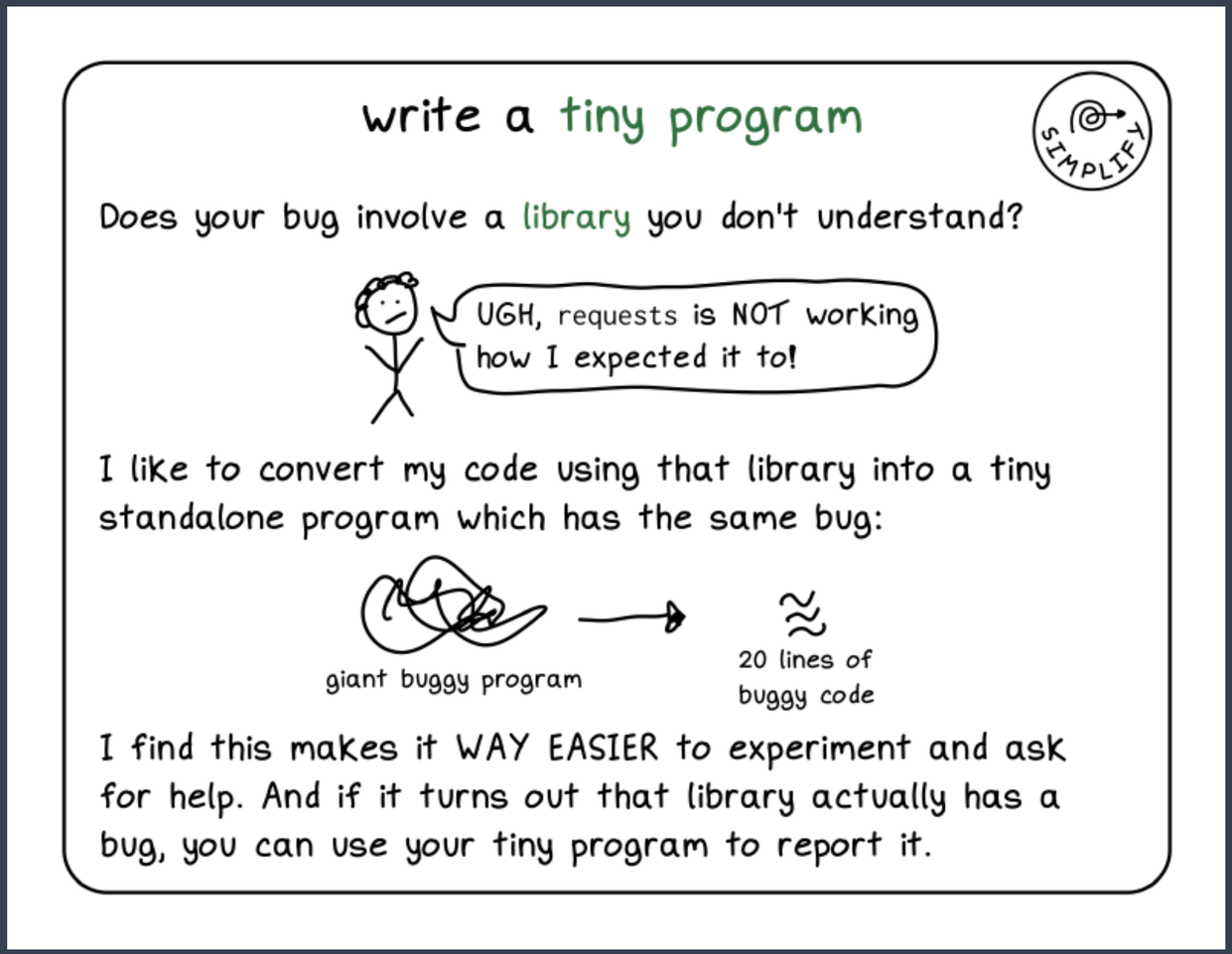
Debugging tips

- Debugging is experimentation, you are asking questions and getting answers by printing things and changing the code.
- Use a debugger!
- You are making assumptions about the world. Sometimes they're wrong! Maybe something you think couldn't possibly break is actually broken.



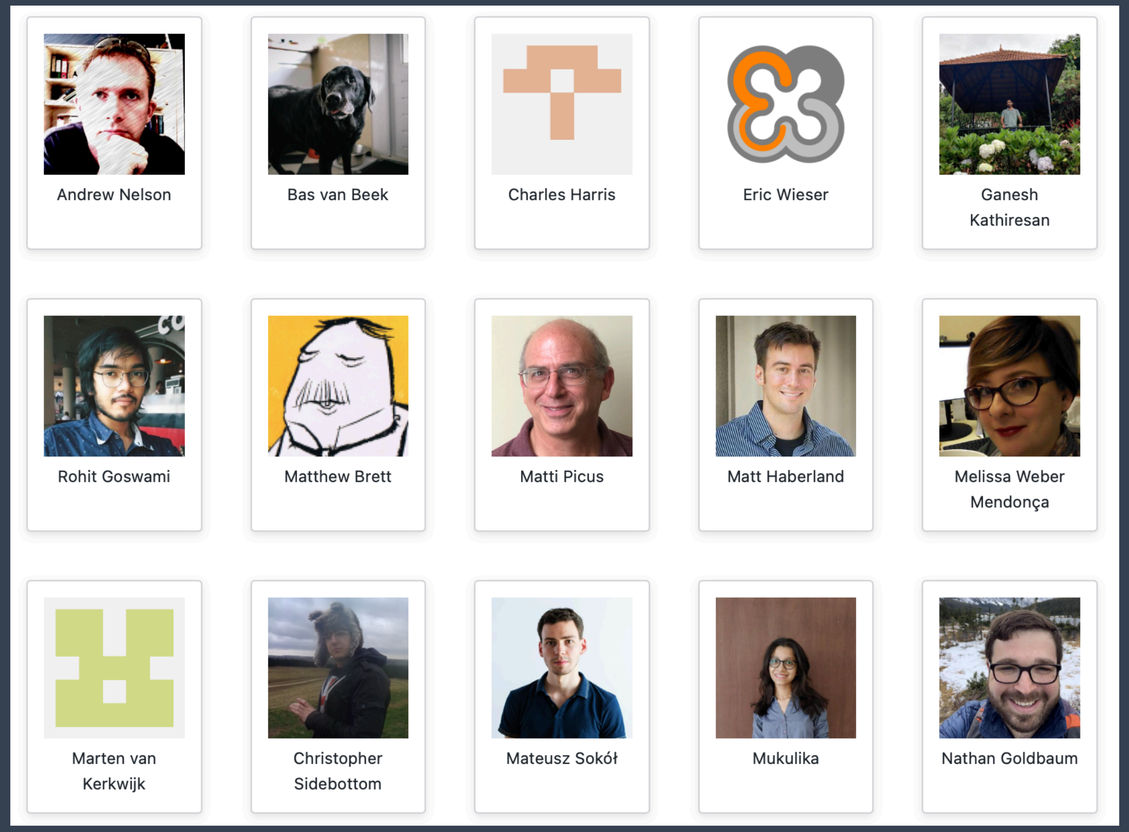
Debugging tips

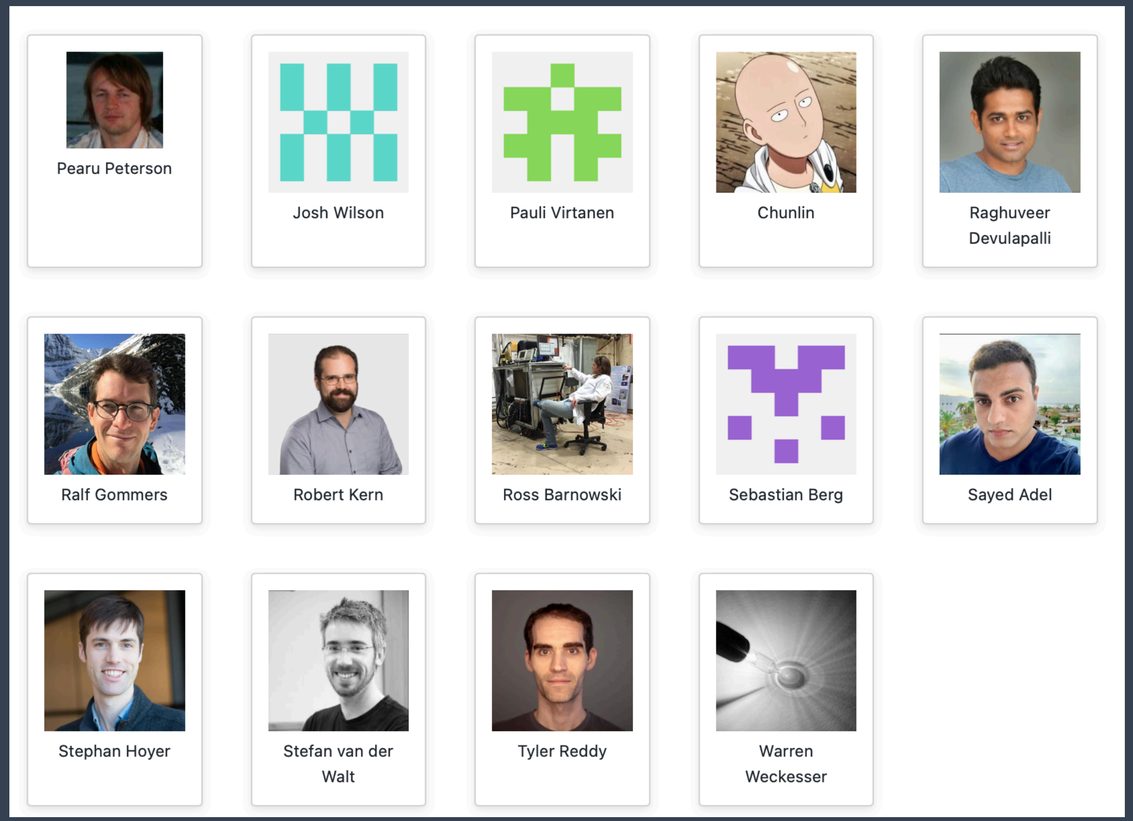




https://wizardzines.com/zines/debugging-guide/

Community of Maintainers





Becoming a maintainer

- Review code from other developers
  - In most projects, anyone can review code!
  - Do a code review for every PR you send in.
- Triage issues and answer questions
- Fix problems that are blocking others
- Become a known face and name
- Take risks and make mistakes.
- Eventually, your inability to merge code will cause inefficiency solved by giving you a commit bit.

A new NumPy String DType

A new NumPy String DType

```
>>> arr = np.array(
...     ["this is a very long string: 😎", "short string"],
...     dtype=StringDType())

>>> arr
array(['this is a very long string: 😎', 'short string'],
      dtype=StringDType())
```

A new NumPy String DType

```
>>> arr = np.array(
...     ["this is a very long string: 😎", "short string"],
...     dtype=StringDType())

>>> arr
array(['this is a very long string: 😎', 'short string'],
      dtype=StringDType())

>>> np.strings.str_len(arr)
array([29, 12])
```

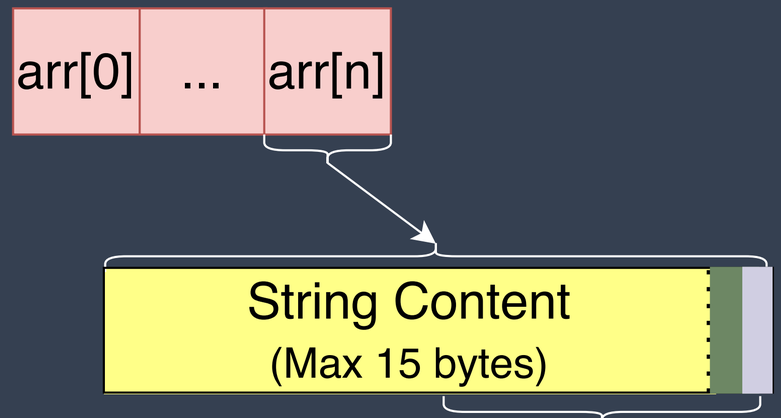
A new NumPy String DType

```
>>> arr = np.array(
...     ["this is a very long string: 😎", "short string"],
...     dtype=StringDType())

>>> arr
array(['this is a very long string: 😎', 'short string'],
      dtype=StringDType())

>>> np.strings.str_len(arr)
array([29, 12])

>>> isinstance(np.strings.str_len, np.ufunc)
True
```

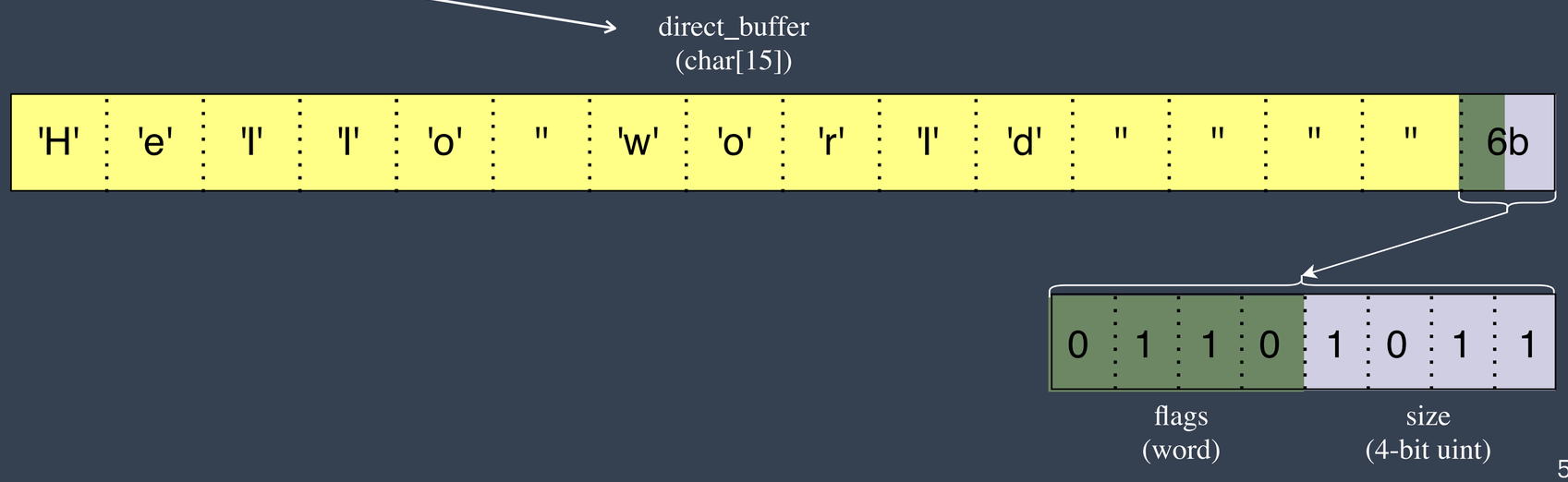
A new NumPy String DType

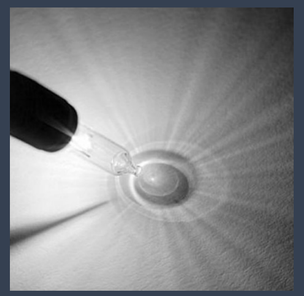
```
>>> dt = StringDType(na_object=np.nan)
>>> arr = np.array(["hello", nan, "world"], dtype=dt)
>>> arr[1]
nan
>>> np.isnan(arr[1])
True
>>> np.isnan(arr)
array([False,  True, False])
```

Short String Optimization




```
>>> arr = np.array(
    ['entry',
     ...
     'numpy is a very cool library',
     'hello world'],
    dtype=np.dtypes.StringDType())
```



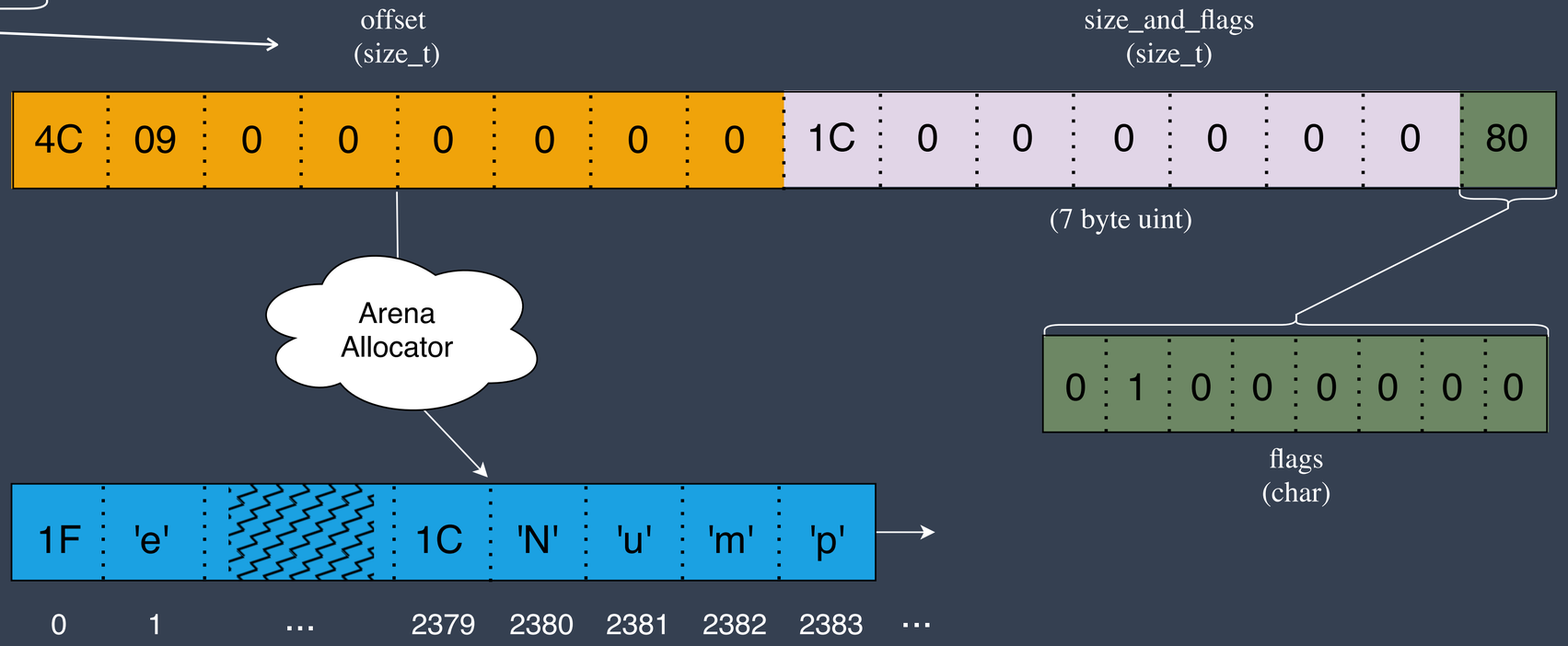


Warren Weckesser

Arena Allocator

```
>>> arr = np.array(
    ['entry the first',
     ...
     'numpy is a very cool library',
     'hello world'],
    dtype=np.dtypes.StringDType())
```

Marten van Kerkwijk



My NumPy year

- Big projects are hard, but you can do them!
- Scientific Python projects are welcoming.
- NumPy 2.0 has lots of cool stuff and is the product of lots of hard work.