

## Script Draft #3

To see with comments on Google Drive, check out:

<https://docs.google.com/document/d/1AzK9Q1vCPjcK6j9pMQTOM6K-zMiZf5FU/edit?usp=sharing&oid=110556856523158639329&rtpof=true&sd=true>

Reviewed with Mukulika and Ross at Script Review, Jul 12, 2023

Reviewed with Mukulika and Jules on NumPy Docs Meeting, Jul 17, 2023

Shared with NumPy Community Meeting, Jul 19, 2023

- Overall message: NumPy is driven by people who use it: in this case, scientists!
- Characters
  - 3 students: -> 2 students? To make space for other characters?
    - 3 stooges energy
    - A: Most informed (spirit of OS), potential user-contributor “let’s do that!”
    - B: Least informed (let’s just get this done), ordinary user, “why would you do that?!”
    - C: Straight man (no strong preference), eager to please “um, what is that, exactly?”
  - Professor
    - ~~Starts students journey by making NumPy used for class~~
    - ~~Later revealed to be part of NumPy core team~~
  - PR maker (Ganesh)
    - Seen through screen (open source contributors across the world)
  - Maintainer (Sebastian)
    - Seen through screen (open source contributors across the world)
- Format:
  - **Narrative with informational diagrams throughout**
- Target audience:
  - **Strongest appeal to academia** probably: professors, scientists, students
  - will this resonate to a general audience? I think the story will get them started but understandable if there’s some fall-off when things get detailed
  - Industry: ?
- Will not:
  - Explain in too much detail how to set up environment, how Git works
  - what open source is
  - How numpy differs from other projects
- Goals:
  - the **early contributor journey**
  - NumPy’s relationship to open science

- illustrate to those who've been around for a while what it's like to be a newcomer
- importance of **guiding figure in entering open source**
- **want OS people to hand to their friends to answer the 'what do you do' question**
- Informative sections/themes
  - Very brief introduction to **Open Science**
  - How **NPZ archive** works and it's usability to scientists
  - **How usability problems affect scientists**
  - **How scientists can solve usability problems**
  - How website and **documentation guide newcomers**
  - Process of writing an issue for NumPy
  - Relationship between issues and PRs: how one leads to another
  - Very brief introduction to Git for PRs: cloning, branching
  - **Contributor workflow** and role of maintainer
  - **Demystify and humanize: Who is the core team of NumPy?**
  - **How PRs are reviewed and new features weighed in NumPy, an old codebase with downstream projects**, thus the focus on backward compatibility
  - When PRs are merged and release cycle
  - Seeing your feature implemented!
  - How credit is given
  - The **importance of user-contributor in NumPy and Open Science**
  - How to **continue to be involved in the NumPy community and it's various channels**
- **Comments and Questions**
  - Is this a story that will continue with same characters?
  - Difference between big picture contributor pathways and smaller scope story, early contributor experience
  - OK to no go into detail about NPZ example
  - Understandable to scope smaller
  - Great to have high-level goals: eg target audience, aim to please everyone will not please anyone
  - Solid foundation with reasonable scope: expanding can get too much
  - Coming at complete opposite approach: start with narrative, highlight how a community driven open source looks like
    - eg this is the contributor workflow diagram vs virtue of story itself where students try to solve problems themselves

## Opening scene

- Purpose: Establish setting, characters and goal of story
- Setting: Opening shot to library, University of Monty

- Three students settling down to table, carrying books, coffee, etc
- Students chat about previous class
  - Happy we finished the lab section
  - mention professor, let's move to using NumPy to look at Cell imaging data, - Actions while chatting: Opening laptop, drinking coffee
- Opening lines
  - Thanks for coming here guys, called meeting after class, want your help in running NumPy, never used it before, looks intimidating, Hope I don't mess up (not confident)
  - No problem, happy to help (narrative parallel of supporting each other like NumPy community supporting each other), I've played around with in my Python class (Comp Sci class? Is there a more specific class)

Script

B: Whew, I'm so happy we're done with the lab class.

A: Buuuut we're not done yet.

B: We need to look at the cell image data, right? With NumPy?

C: Yes... and I have no idea how to use NumPy. So thanks for meeting after class to help me.

A: No problem. we're happy to help. I've played around with NumPy in a Computer Science class.

### Rising Action: Discover the problem

- Purpose: Running through scientific process for both reader and newbie to NumPy character
- Setting: Library
- Diagram explaining process
  - Well, you remember what we did in the lab right?
  - single cell microscope data stored as numpy arrays
  - Need advice on this section
- Visuals: Terminal
  - Ok, let's start with opening the terminal
  - Make a .py file?
  - Open editor
    - import numpy as np
    - data = np.load, achive npz, downloaded the dataset
- Hitting the problem
  - Open NPZ archive
  - Named arrays inside archive, couldn't find out where the names where
  - 3rd person, looking at... worksheet? The docs have a files attribute that tells you the name
  - why can't you see the names as is? why won't it work like a dictionary?
  - We can just follow the worksheet, whatever

- No, I don't like that... that's a usability shortcoming (or another phrase, that sounds more informed)
- We can go forward with this archive contains arrays, access them, get this done quick

Script:

A: You remember what we did in the lab, right?

C: Yup, we finished staining the cells. So now all the cell image information is bundled up into a zipped file. So we use NumPy to extract the file.

A: Yup, and the image itself is organized as arrays.

B: Arrays? How are images a bunch of numbers? Can't we just open the image... as an image? See it with our eyes!

A: Well, in microbiology research, we're shifting through lots and lots of images of cells. And with NumPy's arrays (narrays), we can do image clean-up and find patterns much faster than manually going through images one by one.

B: Hmph. That makes sense, I guess.

C: OK! So how do we get started?

A: Let's start by making a .py file.

In the file, I'll start with importing NumPy. Our professor gave us our cell images as an .npz file, which I have downloaded here. So I'll define the data variable as `np.load('cell_data.npz')`

A: And this is our opened .npz file!

C: Great! But... this is a lot of arrays. How are we supposed to know which one is ours?

A: You're right. Don't tell me I have to manually check for the filename one by one....

A: OK, so I figured it out.

A: For each array, you have to check for the 'filename' attribute.

B: What the-?!. Why can't you see the filename right there? You said NumPy is supposed to be faster! Better than manual!

A: I mean, I expected that KeysView would tell me the filename. But in reality, it doesn't. I can work around it, since I know how to inspect objects.

B: So you need to work around its wonkiness!

C: Uh oh.... I know 'array'. But I don't know about all this 'key' stuff!

C: But, hey, hey it's alright, as long as it works!

B: I don't think we should accept 'as long as it works'. It makes our lives harder!

A: You're right. It is hard. But we can change this!

## Rising Action: Confusion, Deciding Next Steps, Disagreement

- Purpose: Showing the difficult transition from finding a problem to taking action
- Setting: Library
  - I think this is worth bringing up...
  - Um, to who? The professor?
  - No, not just the professor. We can actually bring this up as an issue.

- Brief description of open source: its like
  - Diagram: what numpy means to scientists, students, labs, etc
- All this talk, but I've never actually interacted with the NumPy community, heh... just attended a talk about open science
- Man, I just want to get this over with vs let's take a bigger action
- Stare off!
- I mean, our professor likes NumPy, i'm sure he would say this is relevant haha
- OK, let's try it out, since we got time and all.
- Hey I've never used Github before, Oh I have an account but only uploaded my code, not repo, I've made forks but not contributed....
- So we're lost then....

Open source, open science

A: NumPy is open source software. Anyone can contribute and change the codebase. Even as grad students, we can make suggestions, such as making it easier to open .npz files. And then other people can build off that suggestion and make that feature a reality!

In that way, we can contribute to open science! Since NumPy is a tool used in microbiology, we are building the tools that make science research more collaborative and transparent! Isn't that amazing?!

B: Buuuuut what are we actually supposed to do?

A: Umm... I think we're supposed to "make an issue"? On the Github forums. Saying we have a problem. And suggest a solution...

A: Ok, so I never contributed or anything yet, but I learnt a lot at that Open Science lecture last semester!

C: I have a Github account! We can try it with mine!

A: Great! We can try looking at NumPy.org to get started.

C: Well, I'm looking at NumPy.org. And umm...

A: Let's click over here! (pointing at contribute)

B: This looks complicated. A mailing list, Slack, community calls? Too much!

A: No, no, no, I'm pretty sure we can start with Github. Let's click here.

A: There we go! Now we're on NumPy's Github page, with a list of issues.

A: Let's create a New Issue here. What we want is a Feature Request

## Taking Action, Making the Issue

- Purpose: Show experience, how do people find out more about open source, using only documentation? No in-person next to them
- Setting: Library
- Let's start by looking at the NumPy website!

- Look up NumPy.com, show scrolling experience, keyword extraction, huh! Oh! What?
  - Zoom in making a issue, if you see a byg
- OK, so we need to use one account
- Oh gosh, do I really make a issue?
  - Go through options, clicking, writing
- One person on Github writing issue, one person open terminal describing issue, last person staring at them OK, submit! Off into the ether! Hmm, did anything happen? Well I have notifications open, we'll hear back
- Do we need to complete the assingment now?
- I guess.. lol no

Let's write an issue. We start by describing the problem... and a possible solution.  
(image of sending a letter to a forum bulletin board)

Hi, we're microbiology students using NumPy to look at our cell images.

#### PROBLEM

We found it difficult to quickly see the filename in an npz file.

Example code:

```
>>> data = np.load("example.npz")
```

I expected that KeysView would tell me the filename. But it doesn't.

```
>>> list(data.keys())
['x', 'y']
```

I can work around it, since I know how to inspect objects. But my classmate doesn't.

#### POSSIBLE SOLUTION

Could the filenames be easier to see? Maybe with array.\_repr?

A: I don't know to write the code for this...

C: But hopefully someone will?

B: (presses submit button) Done! Off into the ether!

B: (closes C's laptop abruptly)

B: After all that work, I need a break.

A: But we didn't finish our assignment! We can still use NumPy, it's just ... inconvenient.

B: (doesn't care)

C: Well I have notifications open, we'll hear back.

### Next Action #1: What is the contribution flow? Making a PR and Role of Maintainer

- Purpose: Show understanding of open source growing, not just magically completed, introduce another character

- Setting: a few hours later at canteen? silly student stuff, eating 36 chicken nuggets. moved away from library
- oh someone responded!
- “This also bothers me! I will Write implementation attempt”
- yay! But who is this person? Do we need to do anything?
- no one directly tells them yes or or no until someone else comments
- No, I think they’ll make a PR... for us?
- Wait, how does this all work? what is this person doing?
- Diagram of clarifying the character’s confusion about forking branching, submitting a PR
- Scroll down more, see comment of maintainer say yes, would be good +1 emoji
  - Why is there another person?
  - What’s a maintainer?
  - Diagram Role of maintainers, helpful (not want to use word gatekeeper... moderate? give direction?)
  - Why doesn’t maintainer implement this themselves?
  - Point is that different people work together
- Well, we’ll hear about it i guess
- 

(imagery of community bulletin board)

C: Oh, someone responded!

B: Aghh. (has to stop breaktime). Fine, let’s take a look.

A: Wow, a maintainer commented!

Maintainer: +1. Thanks for the suggestion on improving the user experience.

A: And someone else...!

PR Maker: This bothers me too! This could be implemented by including explicit info about the keys in the NpzFile repr. I’ll make a pull request.

C: Who are these people? What are they saying? Do we need to do something now?!

A: The first person is a NumPy maintainer. This person leads NumPy, deciding what features to add and coordinating people to make all this happen.

The second person seems to be a regular contributor. Who wants to help solve the problem by making a pull request!

C: A pull request?

A: A contributor can add to the code with a pull request. There’s a bunch of other stuff, like push, clone, branching...

A: But don’t worry, we’re not doing any of that now, since we’re not touching the code.

B: Why doesn't the maintainer implement this themselves? Like what does +1 even mean?

A: Well, the maintainer doesn’t do everything... NumPy is such a big project that it needs a lot of different people! That’s what makes open source collaborative!

C: (scrolls further down) Wait, it looks like this person already made a PR!  
C: (opens PR)  
C: (clicks on Files Changed)  
C: Even if we're not touching the code, you can see what is being changed.  
B: (stares at before and after) I don't get it...???  
C: I wonder what the PR maker is doing now... They probably know everything...

### Next Action #2: A PR is made

- Purpose: transition to PR maker
- Setting: Next week/class/lecture, gathering before class
  - Email notification on phone
  - Group gathers around phone
  - A PR that mentions our issue!
  - Hey, it's the same person from earlier! Guess they actually did it, huh
  - I want to check it out!
  - Look at code, I don't understand a lot, all these separate files, how do they hook together and change each other... oh this is a lot
  - More importantly, who is this person?

### Next Action #3: Responding to a review

- Purpose: Demystify and humanize contributors
  - no direct communication between PR maker to students, which is often the reality
- Setting: 9000 miles away, where PR maker is
  - asynchronous and distributed nature
  - Thinking of code PR submitted
  - Proud, nervous, don't psyche yourself out
  - This is a **relatively small fix**
  - I **checked the issue tracker**, someone brought up same issue 10 years ago
  - Didn't **mail mailing** list yet, seems like a smaller fix- or am I just trying to get myself the yes ahead?
  - Check student Github profile? See major at university? Did similar but haven't used NPZ in years, but remembers bugging me too
  - See comment pop up from maintainer
  - what does your comment mean?
    - Show text of comment
    - Dialogue overlaid
    - Explanation about **Backwards compatibility**? "I wish this function accepted these arguments, unpacked tuples", "We're not likely to accept it, propose it to the mailing list" "Here's the reason why, not because you're idea, we might



have wished we designed this way, snapshot to grad students code, if change than breaks”

- Many people not learn about backwards compatibility formally in school, absorb via osmosis, how ties to decision making
- See comment about “implement a `__repr__` in the direction...”, students wouldn’t know what that means but PR maker know
- What does the PR submitter do?
  - Read the comment from maintainer
  - Show thinking process
  - **Oh yeah I talked to (maintainer) recently in community call**
  - Make changes
  - Respond to comment

PRM is looking at the screen. On the screen is the PR he’s created and a comment from the maintainer with feedback.

They look worried, thinking really hard,

PRM: Oh, I really don’t know anything..

No, no you do know. And I’m happy to get a review on my PR.

What he’s saying makes sense. I just overlooked it earlier. I’m 80% there and we’re just figuring out the rest of the 20%.

These students... It’s been years since I was in a lab, but I remember working around this .npz problem.

How many hours could I have saved myself if this feature was added years ago?

Why didn’t I submit an issue back then?

Ah, I was probably too stressed about my thesis! And the time is now.

I wonder what Maintainer is doing now... he probably knows everything.

## Next Action #4: Final review, merging and release cycle

- Purpose: Demystify and humanize contributors
  - Show maintainers are not scary all knowing creatures
  - reliant on others on the same way
- Setting: 4500 miles away
- Merge at end by maintainer (need to mention release cycle?)
- I’m glad (PR maker) is submitting this
- Students
- Final reviewing process
- Merge
- Will be seen at next release cycle
- Write nice comment back
  - hope it works out! next release cycle

M settles into a chair and desktop with a cup of coffee..

M: Time for a final review.

I can see PRM commented back and added some changes.

Thanks! Other ... @student, you should see the change in the next release cycle, happening in 2 months!

Press merge button

Does a stretch, smiles now that its done

I hope that the students are doing good. It's been years since I taught students. So I forgot about all these smaller usability problems. Haha, I guess I don't know that much anymore.

### Closing Action: PR merged, seeing the change they made in the world

- Purpose: Recognition
  - 1 months later: Students gather again at library, talking about mid terms done
  - Oh, hey, check out this email!
  - new microscope data, that was our idea, version release,
  - type in file name, says 'cell'! Success! High five!
    - Woah, this makes it easier to ... do our work...
  - PR maker has in release mailing list release
    - Oh hey, it's that guy again! The PR maker!
  - Students, we have credit in Github history, by making this issue known

C: Wooah, hey guys, check out this email!

C: I subscribed to the NumPy mailing list to see when the next release would happen.

C: Now our requested feature is here, in this release!

What, for real?

The guy who made our feature real, he's right here!

Woah!

And that's the PR they made!

That's so cool! This is what makes open source transparent!

B: This is a nice email and all, but I need to actually see this feature work.

B; (opens up laptop)

Opens .py file (same example as last time)

Runs KeysView

OH SHI-! IT ACTUALLY WORKS (everyone screaming, blur)

### Resolution, teaser for future actions in open source

- Usability, hope it makes better, different ways people contribute
  - Let's go on community meeting next week, saw it on the mailing list
  - I feel proud of us, not just eat 36 chicken nuggets
  - Who know a little action could do so much
  - Spirit of open source, open science

Woah, OK. I get it now. It does feel cool to know that you've made a change, even in a small way.

And other scientists get to...

B: Slack, mailing list, different avenues. Doesn't too much now.

## Credits and Ending Notes

- Credit specific people and actual PRs
- Make QR code for the links?
- Note to new to open source: real story! You could too!
  - Links
    - Issue Ross made: <https://github.com/numpy/numpy/issues/23319>
    - PR Ganesh implemented: <https://github.com/numpy/numpy/pull/23357>
    - Sebastian and Ross review: <https://github.com/numpy/numpy/pull/23357#pullrequestreview-1336728619>
- Worksheet on your own open source journey?

## Comments and Questions

- Demonstrates but not over explains
- Documentation has not told story yet, in dry text, contributor guide, how it's supposed to work 'please comment on mailing list' but in practice, people may not read in full,
- Everyone's impression of 'how does NumPy get developed?', demystify
- Concept of authority figure: Maybe dropping the Professor entirely?
  - Ross motivation: if not professor, "how do raise this issue", rely on themselves, no appeal to higher authority for first idea, less top down, don't need PhD already
  - Constrict the scopes: more focused on open source part, professor reinforces the academic angle, "who do we tell?!", find the issue tracker
- Suggestion: Remove professor, 9000 miles away, PR maker, no direct communication between PR maker to students, which is often the reality. See through screen and Github, know it's a real person, no one directly tells them yes or no until someone else comments. Reflects asynchronous and distributed nature.
  - See the email notification: students response of 'OMG what now?' again
  - If remove professor, can be a more general scenario, eg hobbyists
- When students are making the issue: sort issue tracker to see if anyone has made an issue
  - IRL: npz archive issue from 10 years ago,
    - Maybe for narrative, no previous issue shows up

- Opportunity to highlight that best practice
- How much information it ends up being?
- 3 factors:
  - scope
  - how much would these characters actually know
  - best practice
- maybe someone comments later, “can you show code more specific”
- Lead to question about ‘ask the mailing list first’
  - Answer: For bigger feature request that may break backwards compatibility, should ask mailing list, even if not aware it could.
  - In reality of asking mailing list, more stringent side, more about mailing list, could lead to another rabbit hole
  - What usually happens: people make issue, then move bigger conversation to mailing list
- Tendency for documentation to overexplain,
- Can use comic narrative to open conversation about templates as possibly unifying, or to ask mailing list
- Next steps: Answer on document, Slack, meet at next NumPy Docs