

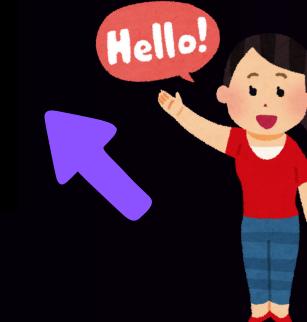
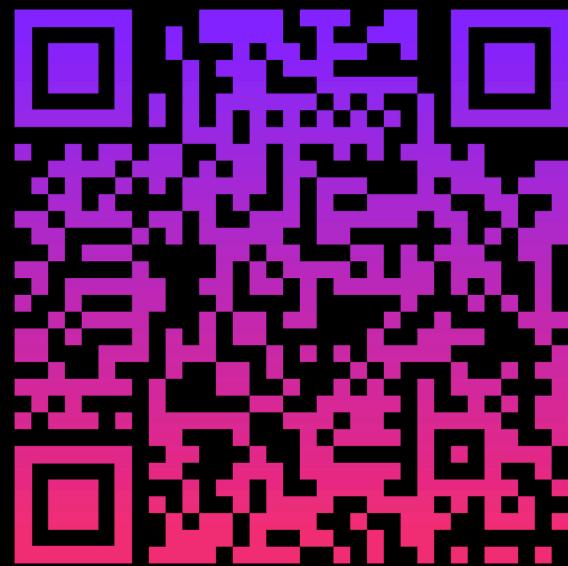
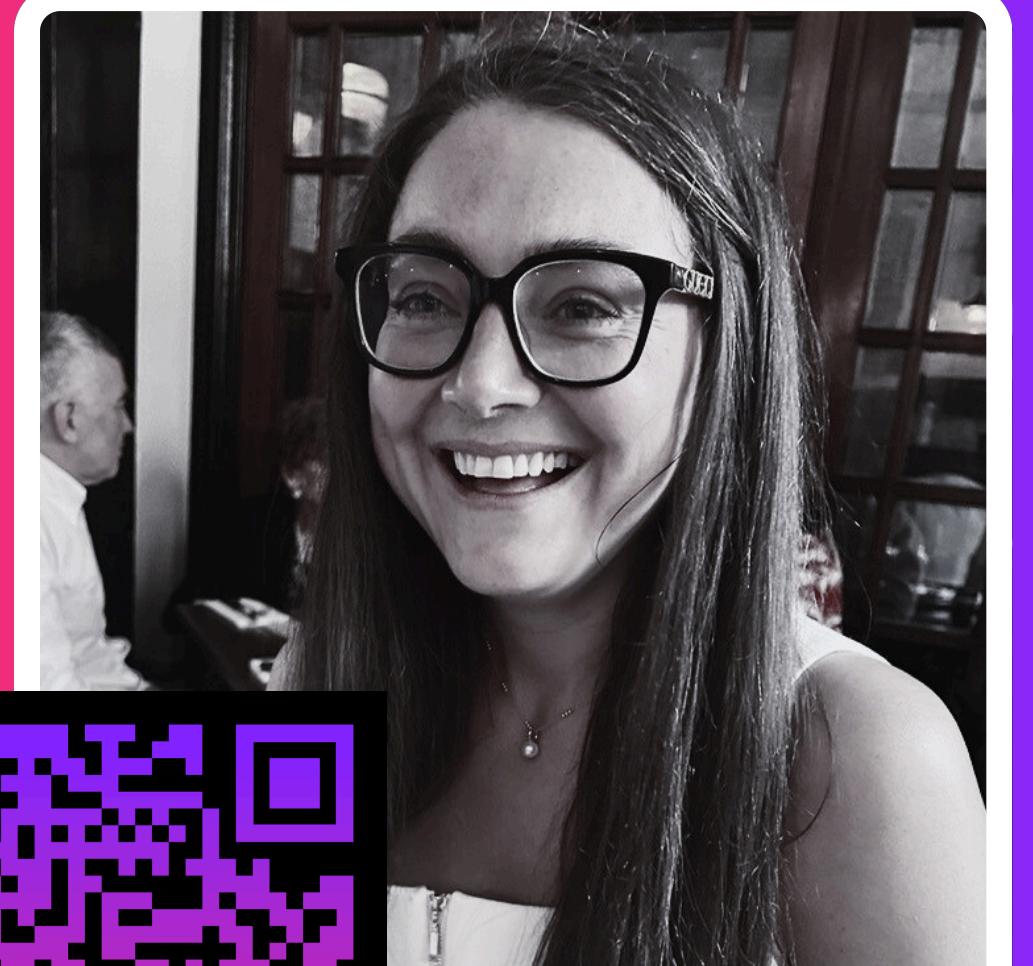


21 September 2025

A junior's perspective: why doing difficult things is good for you and good for your team

Presented by: **Katie Bickford**

Introduction



- Junior Software Engineer at the Bennett Institute for Applied Data Science at the University of Oxford
- Career switcher
 - 10 years working in international medical communications & medical education
 - Northcoders bootcamp, Jan-Apr 2024

The ticket

- Enhance observability for the OpenCodelists service
 - OpenTelemetry – instrument code, collect, & export data
 - Honeycomb – analyse, visualise & query data
- Add usernames to telemetry spans to help us better support our users

A screenshot of a GitHub issue card for ticket #2255. The card has a pink header bar. The title is "Enhance OTel spans with an identifier for logged-in users #2255". Below the title are three buttons: "Closed" (purple), "Task" (yellow), and "#2320" (blue). The main content area starts with a user profile picture of mikerkelly and the text "mikerkelly opened on Dec 17, 2024 · edited by mikerkelly". To the right are "Edits", "Member", and a "..." button. The first section is titled "Why are we doing this?" with the text: "Honeycomb allows us to view OTel data exposed by the site. This allows us to monitor performance. This will be enhanced if we can view traces and spans by who is making them. This may help us identify users facing particular issues, for example." The second section is titled "How will we know when it's done?" with the following bullet points:

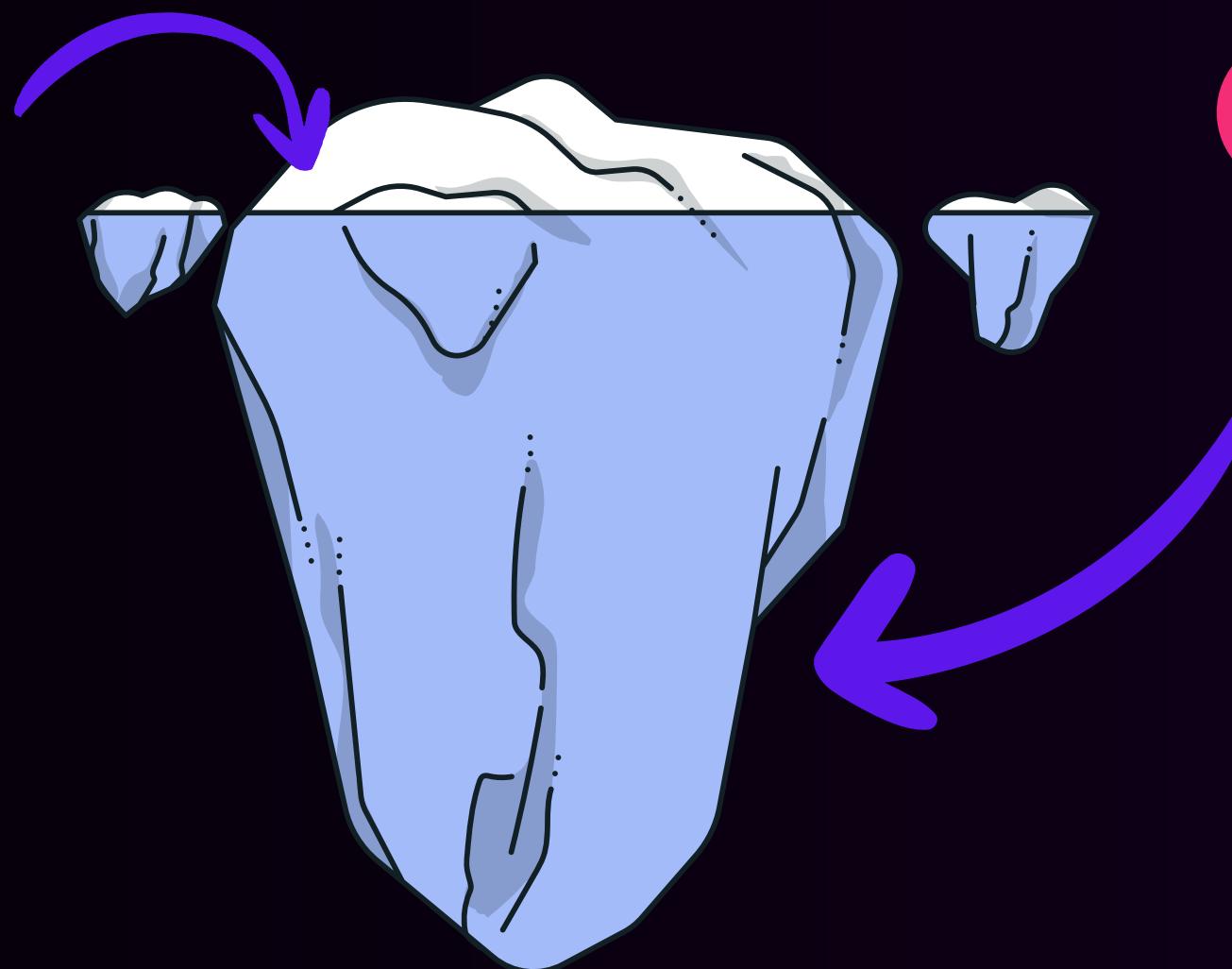
- OpenCodelists Honeycomb queries can use logged-in user as a search term.
- We have better developer documentation of how we use OTel, Honeycomb, and how to develop against them.

At the bottom left is a link "Defining delivery tasks guidance". At the bottom right are buttons for "Create sub-issue" and a smiley face icon. On the right side of the card, there are sidebar sections for "Assignees" (KatieB5), "Labels" (initiative:onboarding), "Type" (Task), "Projects" (Team REX, status Done), and "Milestone" (No milestone).

Expectation vs reality

What I thought I'd learn

- What Honeycomb is
- What OTEL is
- What observability is
- How we use Honeycomb and OTEL in OpenCodelists (OCL)



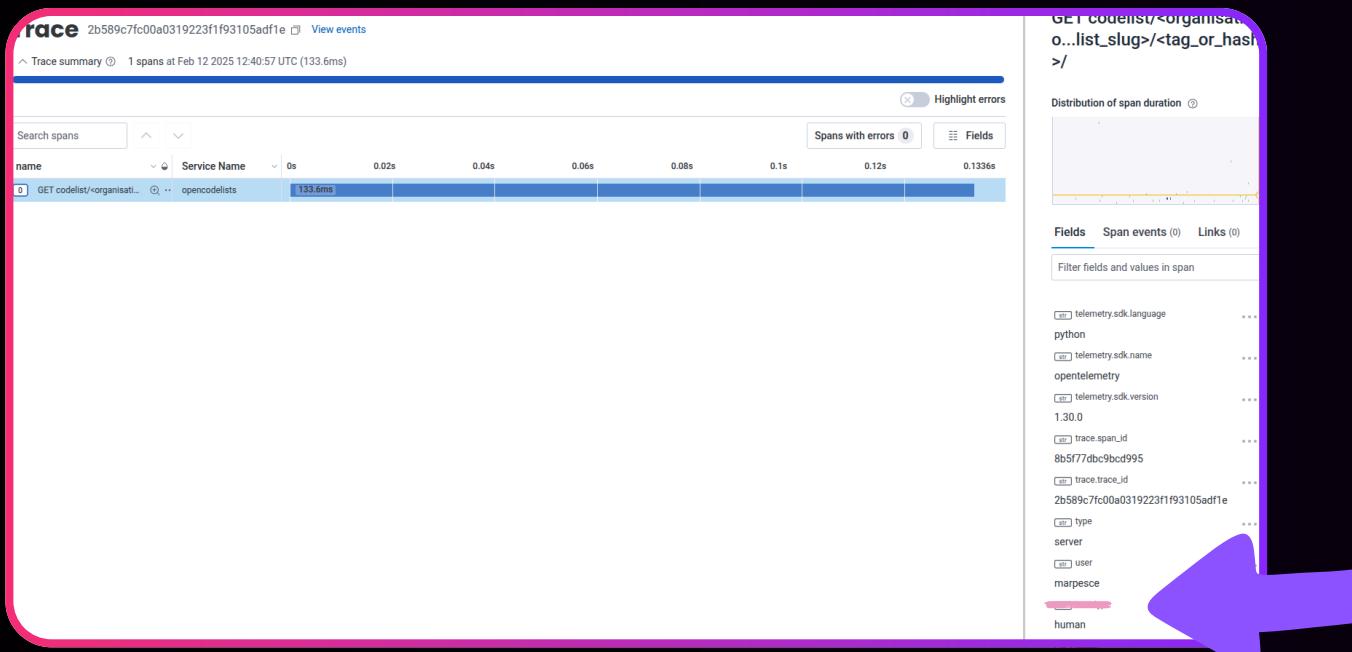
Additional things I learnt

- OCL structure
- Environment variables, bash and just
- Django middleware processing for the request-response cycle
- Team dynamics, conventions and work processes
- Hypothesis-driven ways to work
- How to debug things I've never come across before
- Command line Git and commit messages
- Writing a PR description that doesn't suck
- Pytest fixtures
- Writing documentation
- People have opinions on everything, there is rarely a 100% right or 100% wrong answer

The outcome

- We can now query our telemetry data by logged in users 🎉
- If logged in users make requests, we can see who they are and help them if we need to 🎉

A screenshot of a query interface titled "Query in". The WHERE clause contains the condition "user = marpesce". The visualization is set to COUNT. The ORDER BY clause shows "COUNT desc". A purple arrow points from the "user = marpesce" entry in the WHERE clause to the "user = marpesce" entry in the screenshot below.



Commits 4

Files changed 5

+145 -19



Learnings for other juniors



Image: from
Merve Taner's Medium article, published on May 28, 2017, at:
<https://medium.com/@merve.taner/how-to-standout-as-a-junior-developer-to-get-your-first-developer-job-3e975315055d>

1. Ask for help early and often



- Don't wait until everything's on fire
- There is value in struggling through problems, but there's a limit to this
- Time box it e.g. 1-hour

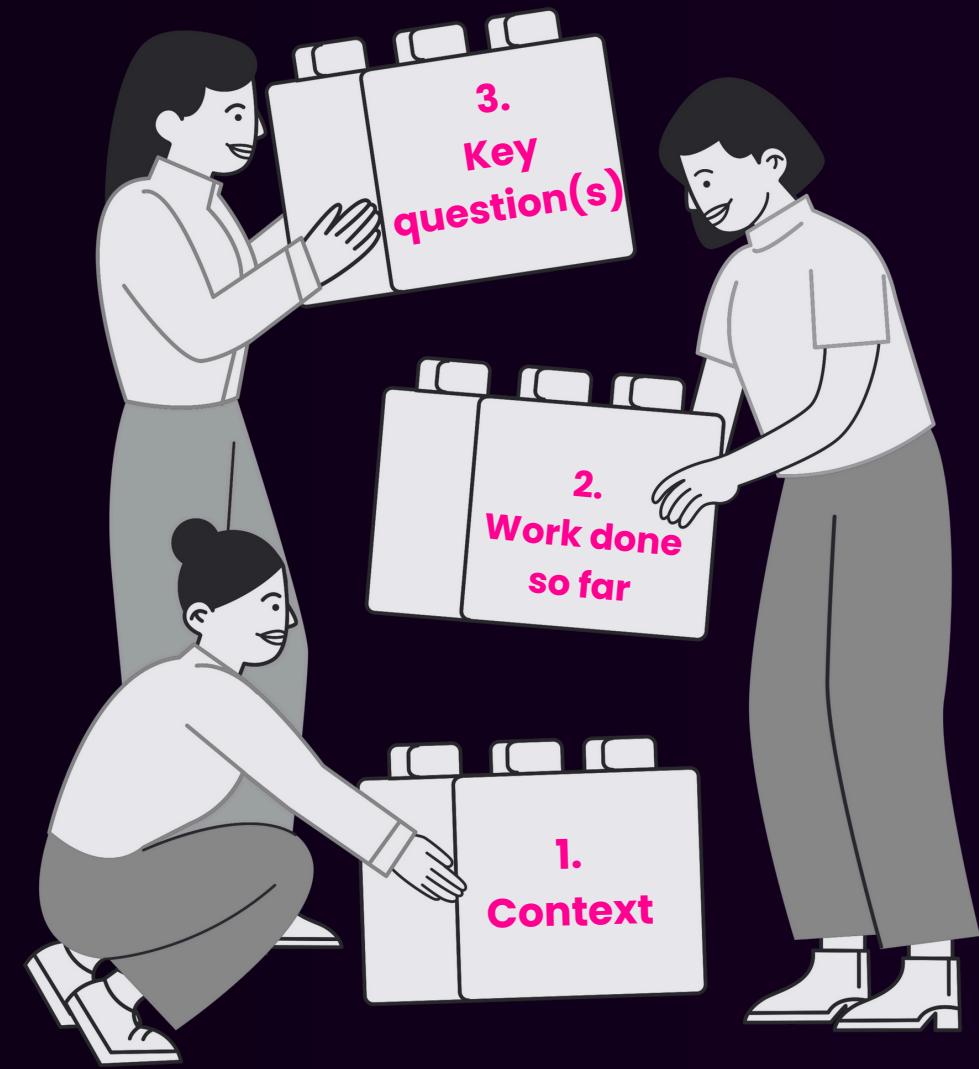


Image: KC Green's web comic strip "On Fire,"

2. Communicate problems effectively



- Give context, but keep it focused
- What have you tried so far (+ why), what happened when you tried that
- Be specific about what you'd like help with



3. Pairing doesn't have to be synchronous



- Track progress in shared documents
- Draft PR reviews
- <Slack> messages; direct or in your team channel if you have one (bonus points)

What's
'pairing',
Katie?

A screenshot of a Wikipedia article titled "Pair programming". The title is highlighted with a purple arrow. The page includes a navigation bar with "Article" and "Talk" tabs, a summary from Wikipedia, and a detailed description of what pair programming is, mentioning the driver, observer/navigator roles, and frequent role switching.



4. You're benefitting your team



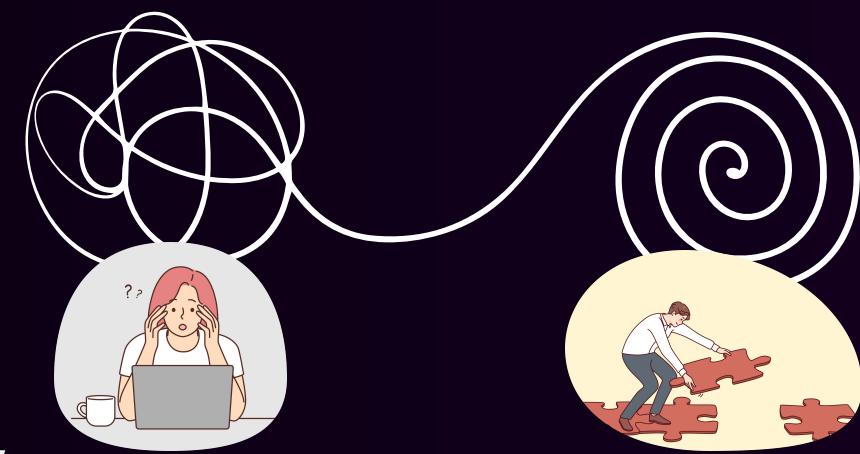
- Others get a chance to learn something new or strengthen their own understanding
- Questions spark key discussions
- Lead to improved documentation or processes



5. It really does get easier



- You build base knowledge
- You get better at debugging and articulating problems
- You become more confident tackling challenges



“

“That which we persist in doing becomes easier for us to do; not that the nature of the thing itself is changed, but that our power to do is increased”

— Ralph Waldo Emerson

”

PyCon UK 2025

Summary of learnings for other juniors

- 1. Ask for help early and often**
- 2. Communicate problems effectively**
- 3. Working together doesn't have to be synchronous**
- 4. You are benefitting your whole team**
- 5. It really does get easier**



PyCon UK 2025

Learnings for Seniors

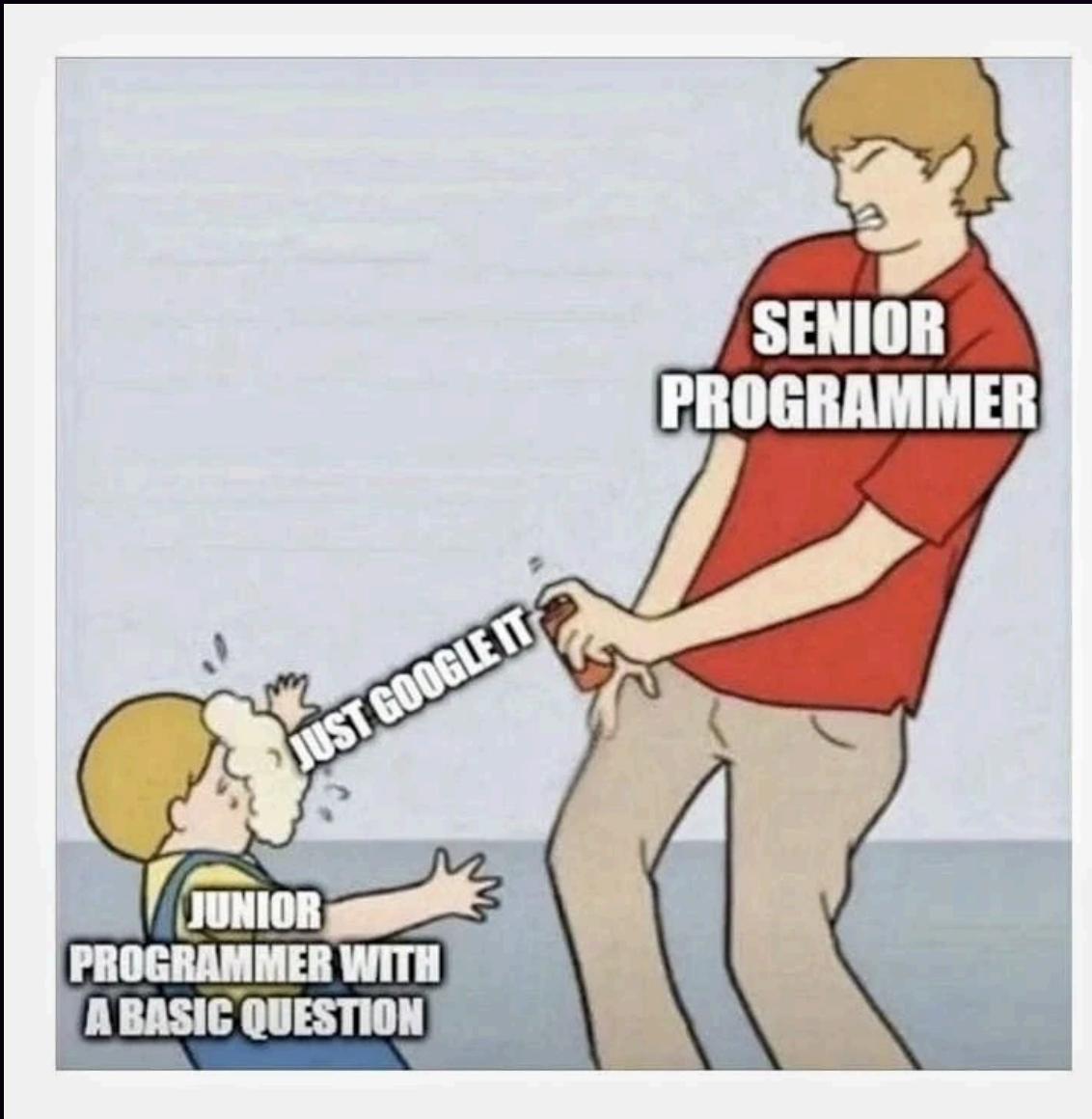


Image: from Thodoris Kouleris @tkouleris, shared on Jul 21, at:
<https://app.daily.dev/posts/just-google-it-efyvyfzvd>"

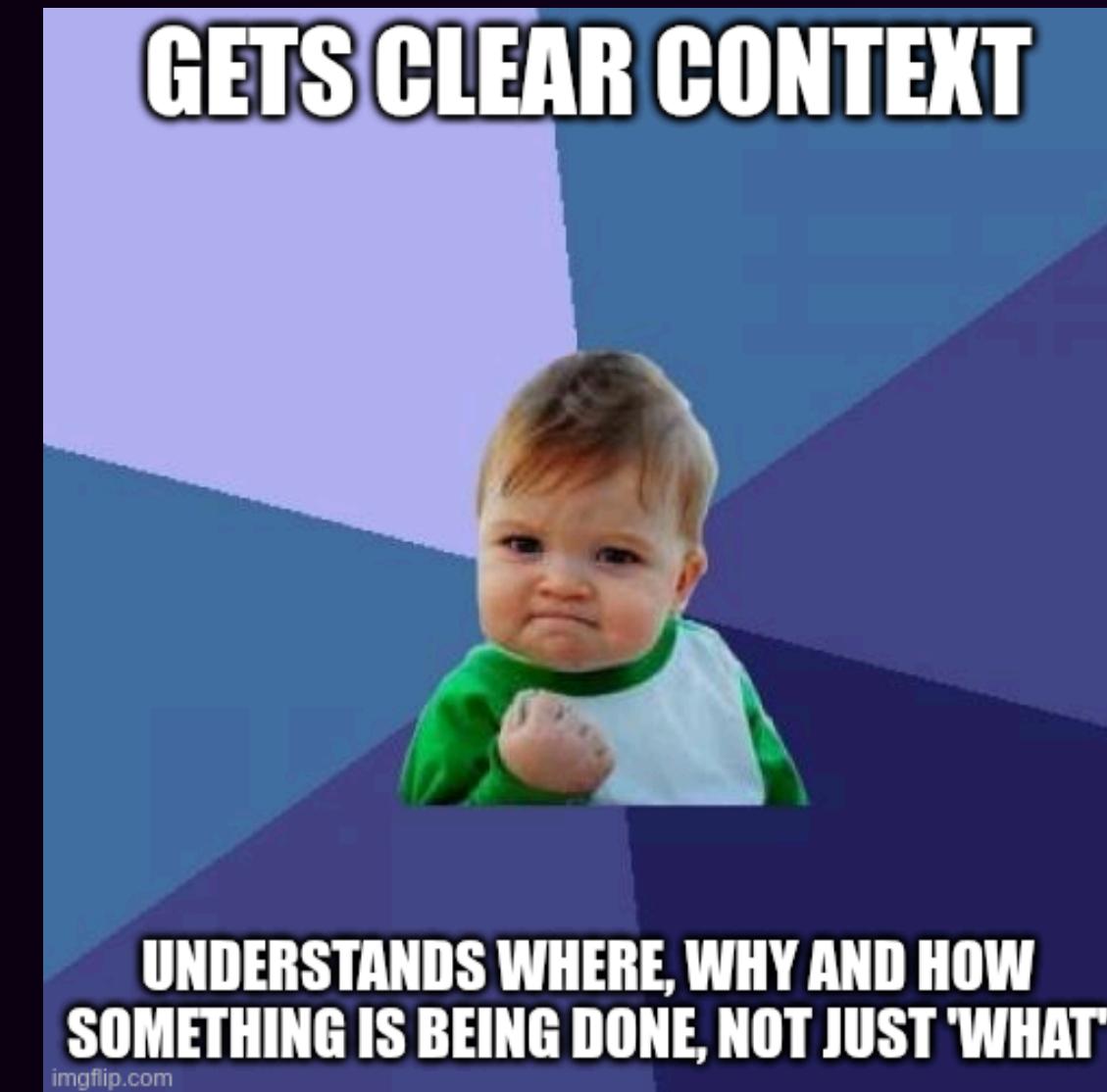
1. Invest time



- *Validate plans*
- *Explain frameworks*
- *Talk through example implementations*
- *It pays dividends!*

2. Provide context

- *Don't just say "it's this", or "Google it"*
- *Explain why and how*



3. Role model that it's OK to not know and forget

- *Admit when you don't know something*
- *Reassure that it's OK to forget something*
- *Help find the answer together*

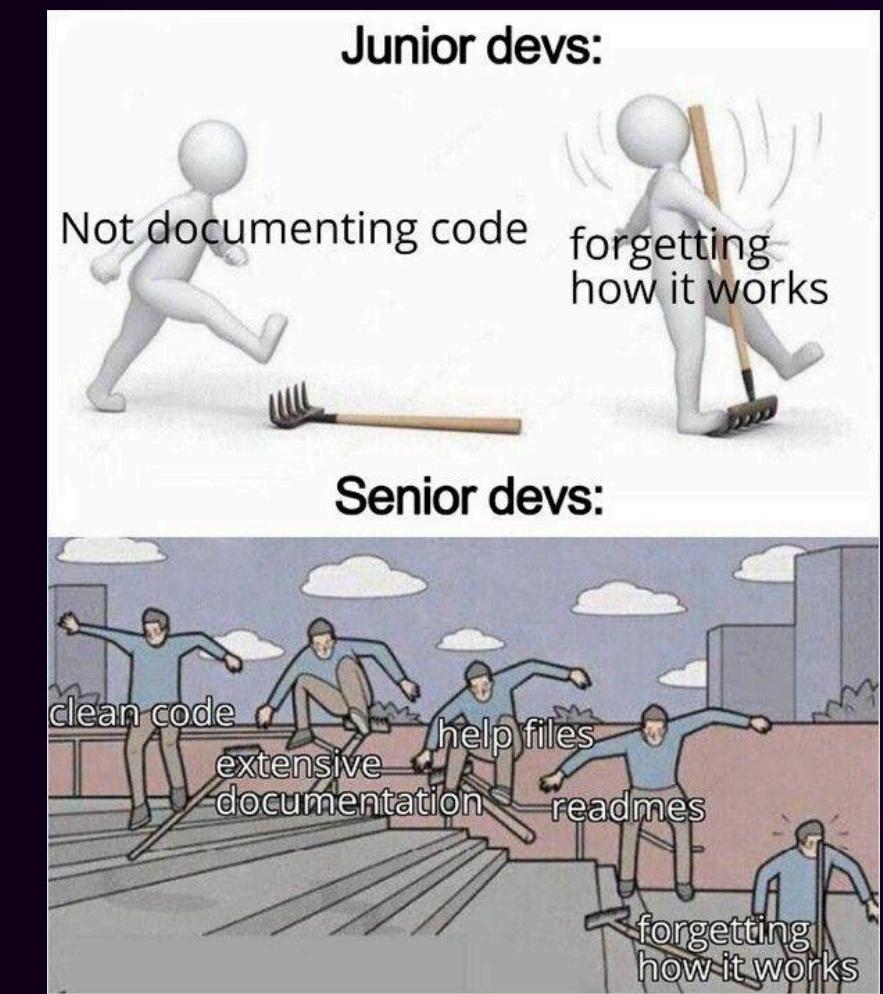


Image: from <https://programmerhumor.io/programming-memes/the-documentation-paradox-wp6>

4. Give feedback

- *Specific*
- *Constructive*
- *With real-life examples of behaviour if possible*

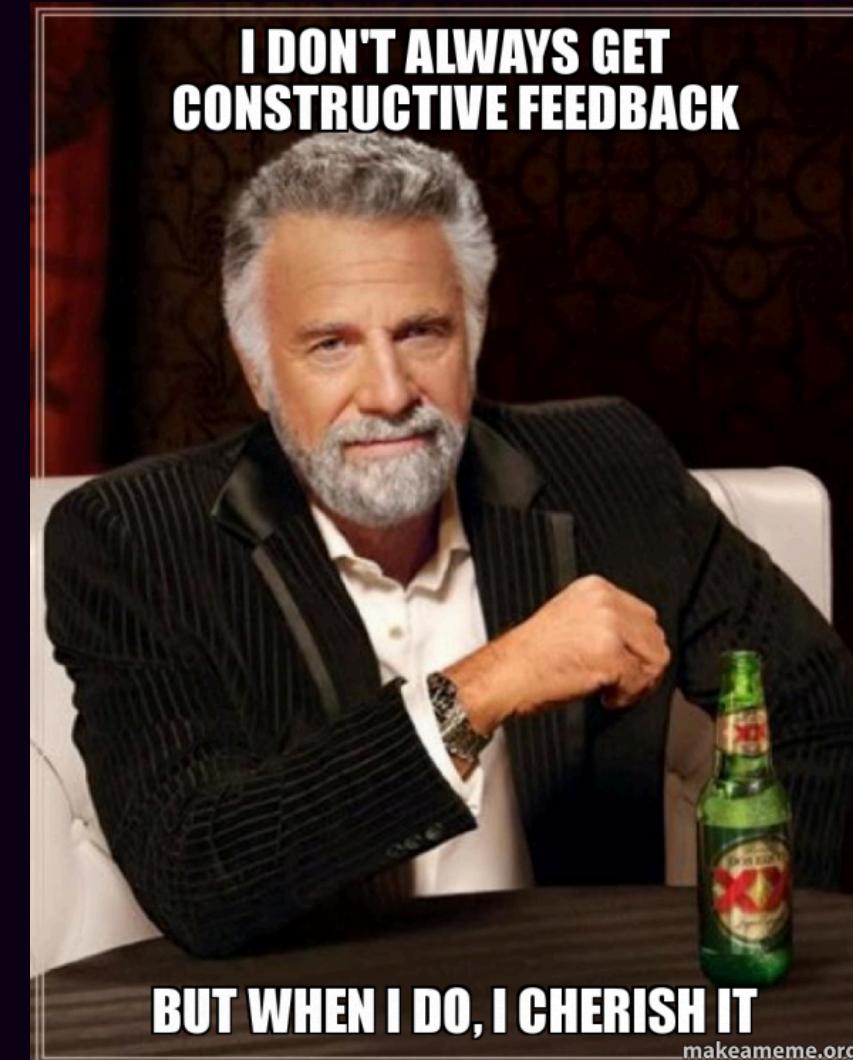
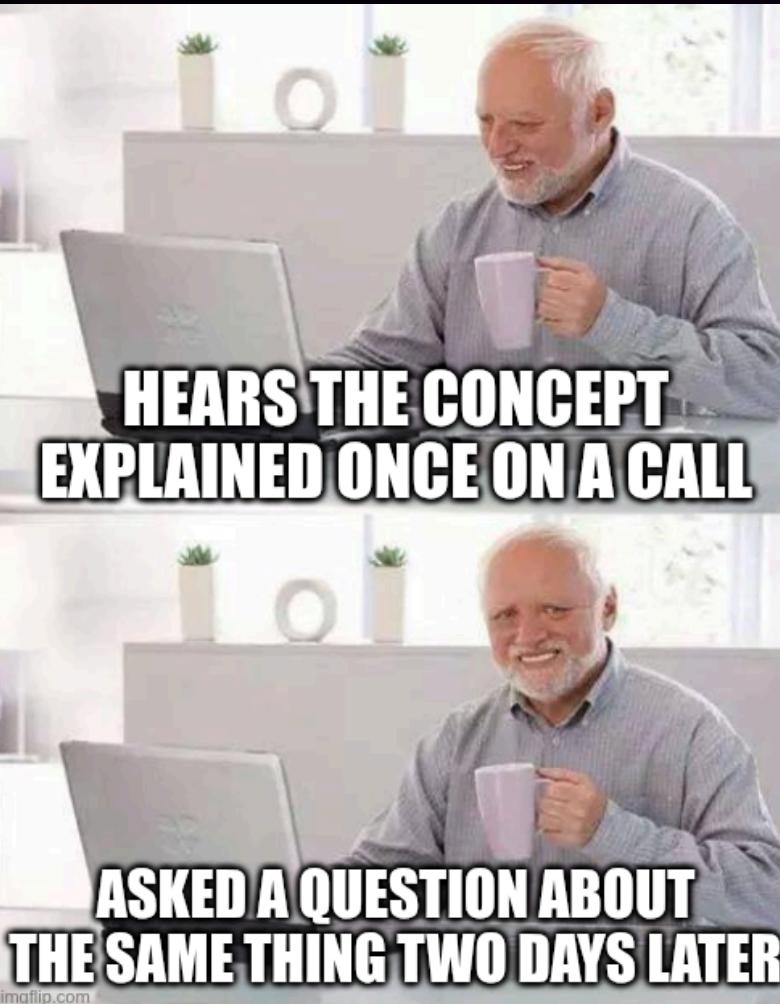


Image: from <https://makeameme.org/meme/i-dont-always-42b78db401>

5.Signpost to helpful resources



- *Reflection and retrospective learning*
- *Consolidate learning at own pace*
- *Benefit both parties!*
- *Cater to different learning styles*

Image: made by Katie Bickford, on
<https://imgflip.com/memegenerator/Hide-the-Pain-Harold>

6. Provide encouragement and reassurance



Image: made by Katie Bickford, on
<https://imgflip.com/memegenerator/83059216/Motivational-Hedgehog-is-Motivational>

- *Recognise effort, process and dedication*
- *Provide encouragement & reassurance*
- *Highlight future value of work being done*

Summary of learnings for seniors

- 1. Invest time**
- 2. Provide context**
- 3. Role model that it's OK to not know and to forget**
- 4. Give feedback**
- 5. Signpost to helpful resources**
- 6. Give encouragement and reassurance**



PyCon UK 2025

Closing remarks

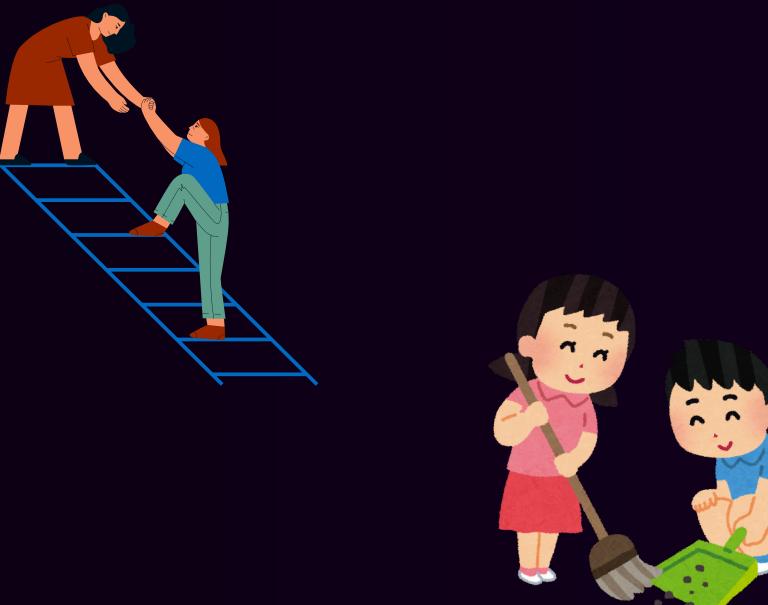
Juniors

Do the difficult things



Seniors

Take the time to help your juniors



Thank you

for listening!

