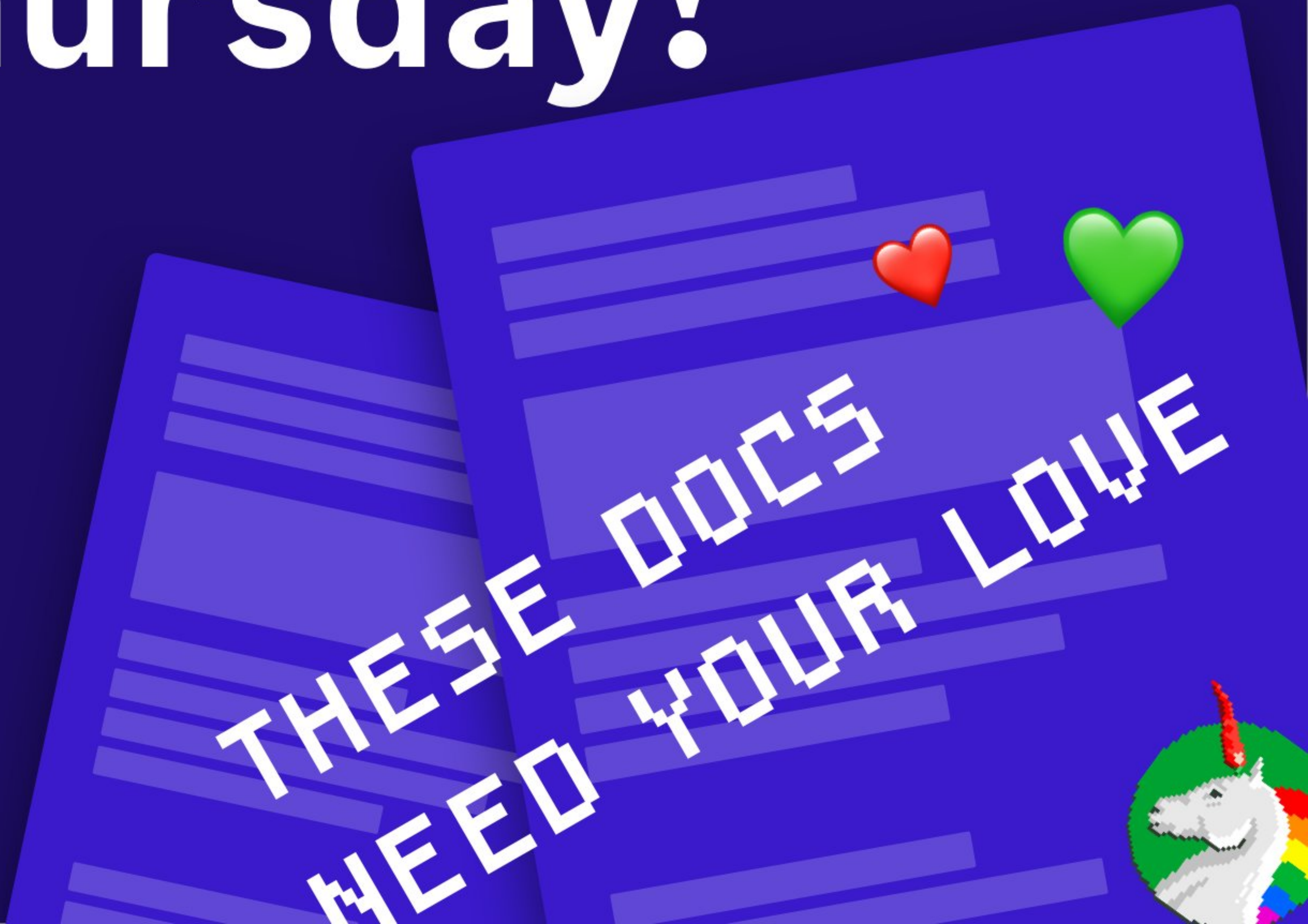
Spice goes

DocsThursday!

21st February

HKI-Workshop

17–20



Tool vs. **Code**

inspired by Carolyn Stransky talk

UX --> **CX**

The **same techniques apply** for internal API docs, a public commercial product, or a side project of yours

Paraphrased from Ben Hall, The Art of Documentation

How to get started?

1. Your own open source project
2. An open source project of a buddy
3. A project that you love but could use some love
4. A hobby or personal interest based project
5. One of the public Futurice projects on github
6. One from our list of projects that could use some love

Good starting points

1. Github Readme
2. Github Wiki (optional MVP)
3. Static Site Generator

source: Beyond the Readme, Rand McKinney

A good readme is table stakes

– Rand McKinney

source: Beyond the Readme, Rand McKinney

> The readme is equally as important as a products home page
>
> –Ben Hall, The Art of Documentation

source: The Art of Documentation, Ben Hall

You have 5 minutes to hook people

- Will come confused
- Try to understand what problem it solves
- Want to quickly experiment with it
- Question: how do you show them the value

source: The Art of Documentation, Ben Hall

Common mistakes

- 17% forget to include project name
- 31% don't describe the project
- 46% missing project location
- Way too much information, but the wrong kind

source: Readable Readme, Beck

The Confidence building readme

1. Identify the project
2. Evaluate
3. Use the project (once)
4. Engage with the project

source: Readable Readme, Beck

Readme Content

1. Getting Started Tutorial
2. Installation
3. Summary (why use this, what problem does it solve?)
4. Related resources
5. Link to Contributors & License file (choose a license)
6. List of Maitainers, Core Team, Contributors

source: loosely based on material from Beyond the Readme, Rand McKinney

Quick note about design - Excitement generators

They can be **lovely touches of thoughtfulness** in the design, but also things like the **language** the product uses and the **personality** it has.

source: https://uxplanet.org/getting-real-about-delightful-design-24af65ebbe93

Next steps

1. Find a project
2. Find a starting point
3. #writethedocs

Resources

Beyond the README: Creating Effective Documentation for Your Project by Rand McKinney, IBM (link) 21m video

With 25 years of experience writting developer documentation, McKinney provides motivation, strategies, and practical applications for writting developer documentation. He introduces the 2 audiences for developer documentation: contributors and users with an emphasis on the latter.

Loopback Readme Guidelines (link)

Altough this might be a bit to verbose for starter documenation, as it perhaps focuses on contributors, it has some great advice about what should be in a readme file and how it should be formatted.

A Look Into Static Site Generators For Open Source Docs by Carolyn Stransky link (24m video) - A review of 46 polled open source projects, from small home-grown to very large projects. Some key takeaways are most projects are concerned with conributor experinence, the community of the static site gen, and automation. Interesting was that language was not a preference but rather having a less steep learning curve was key. Most projects interviewed cited that they were happiest when using a variety of tools that fit their specific needs.

The Art of Documentation and Readme.md for Open Source Projects - Ben Hall, Katacoda
(35m video)

Write The Docs Podcasts, Confrences, Slack Community of Techinical Writers. A huge amount of resources.

build a website with github - (video for talk 1h28m)[https://www.youtube.com/watch?v=812E14gFgb4&t=106s]

Write the Readable README by Daniel D. Beck (23m video) - In this video Beck did a survey of over 200 public opensource readme files from projects with over 10k stars on github, projects he was familair with, and some closed source readmes. Bias is towards established popular products.

README Checklist by Daniel D. Beck - A series of the 4 ways to build confidence in your users from a readme, described in the talk Write the Readible Readme, that you can easily go through to evaluate your current readme and where it could use some love.

Tools

docosaurus.io - Created by the Facebook Opensource Team to create a simple way to document projects.
gatsby - if you are react
slate - for apis
jekyll - free on github, complicated and yml
readthedocs.org / sphynx-doc.org - long learning curve

tocdoc - generates a Table Of Contents for markdown files automagically. Important flag: --github

allcontributors - a tool that will automatically add all of the contributors to your project readme file docz - great project for documenting react projects with mdx. Also a great example of good documentation.

read the docs + sphinx - a free but add supported way to automatically build and host your docs. You can use standard markdown files in a docs directory and you will need to use python to configure initilaly.