



WhatsApp Group



• GIT & BEYOND •

Contributing to Open Source

Navigating Repos & Collaborating on Projects

Abhijith R • [__](#)





Session Overview



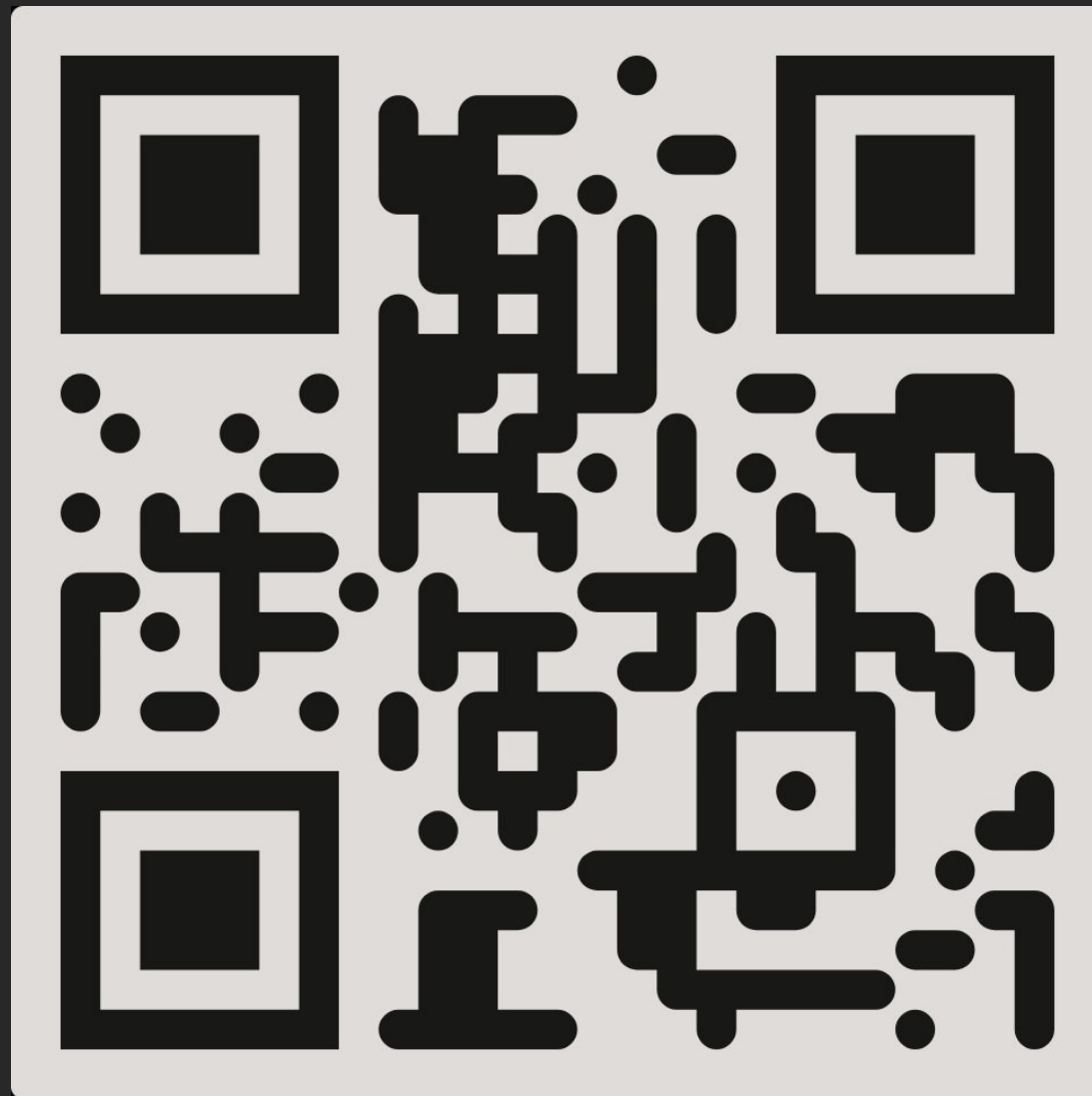
1 · Navigate

- Open any unfamiliar repo
- know what it does,
- whether you're allowed to use or modify it, and
- Its rules for contributing.



2 · Collaborate

- Understand the workflow
- git push:
- opening a pull request,
- getting reviewed by a stranger, and
- getting merged.



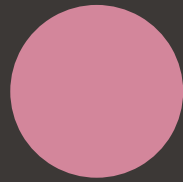
`blogs.jithurx.dev`



You Already Use Open Source



Code editor



Web browser



Language
runtime

Every one of those is open source. **These applications are built through contribution and collaboration...**



What "Open Source" Actually Means

Source code
is available

+

A license granting
use, study, modify
& redistribute

=

Open Source



No LICENSE file = all rights reserved by default

More about "Open Source License"



Permissive Licenses

Permissive licenses allow you to modify and integrate the code into proprietary, closed-source commercial products.

- MIT License
- Apache License 2.0
- BSD Licenses

Copyleft Licenses

Copyleft licenses enforce a "share-alike" rule. If you modify and distribute the software, your changes must also be released as open-source under the exact same license

- GNU General Public License (GPL)
- GNU Lesser General Public License
- GNU Affero GPL
- Mozilla Public License (MPL)



Navigating an Open Source Repository

README .md

The front door — what the project does and how to run it.

LICENSE

The legal terms for using/modifying/redistributing the code.

CONTRIBUTING .md

The project's own rule-book for submitting changes.

CODE_OF_CONDUCT .md

Community behavior expectations.

Issues

The public to-do list / bug tracker. Find work here.

Pull Requests

Proposed changes awaiting review.



Finding Something to Work On

Use GitHub filters to find beginner-friendly work:

```
is:open is:issue label:"good first issue"  
is:open is:issue label:"help wanted"
```

good first issue

Small, well-scoped, safe for a first-timer

help wanted

Maintainer explicitly wants outside help

documentation

No app code needed — lowest barrier to entry

bug

Something's broken, needs a reproducible fix



The Collaboration Workflow

- Find a project & read README/CONTRIBUTING
- **Fork the repo & Clone your fork locally**
- Create a Branch
- Make changes -> Commit -> Push
- **Open a Pull Request (PR)**
- Automated Checks (CI) run
- Maintainer reviews & comments
- Update PR with new commits if needed
- **PR gets approved and MERGED**



Before You Write Code

- ☐ Read the README
- ☐ Read CONTRIBUTING.md
- ☐ Set up and run the project locally
- ☐ Pick ONE issue
- ☐ Comment on it first (claim it), if the project expects that



Open a Real Pull Request

```
# 1. Fork the repo on GitHub (click "Fork", top right)

# 2. Clone YOUR fork
git clone https://github.com/<your-username>/tasklite.git
cd tasklite

# 3. Create a branch
git checkout -b fix-<short-description>

# 4. Make your change, then:
git add .
git commit -m "Fix: <short description>"

# 5. Push your branch
git push origin fix-<short-description>

# 6. On GitHub: click "Compare & pull request" → submit
```



Collaboration Etiquette

Search First

Search existing issues before opening a new one to avoid duplicates.

Be Specific

Write clear, specific descriptions for your issues and pull requests.

Accept Feedback

Take review feedback as mentorship, not criticism. "Changes requested" is normal.

Start Small

A merged typo fix.



Homework & Resources

HOMework

Open one real pull request, anywhere, by next week.
Doesn't matter how small .

Where to find projects:



goodfirstissue.dev

Beginner-friendly issues across many projects



up-for-grabs.net

Projects actively welcoming new contributors



codetriage.com

Issues emailed to you over time — builds a habit



[freeCodeCamp](https://freeCodeCamp.org)

Large, beginner-tolerant JS/web project

Thank You

