

# HOW TO PARTICIPATE IN FREE/LIBRE OPEN SOURCE SOFTWARE PROJECTS

---

Dr. Kuljit Kaur Chahal

Department of Computer Science

Guru Nanak Dev University, Amritsar, India

Eric S. Raymond

“Every good work of software starts by scratching a developer's personal itch.”

# How does a FLOSS project start?

- A free software project emerges
  - When there is a need to solve a **problem ??**
  - With the ability to find a solution.
    - At least the knowledge to start
    - Richard Stallman GNU
  - When a simple, well designed partial solution exists in a form
    - To be shared to draw the attention of others
      - who have the same problem and
      - who may be interested in the solution.

I am Richard Stallman, inventor of the original much-imitated EMACS editor, now at the Artificial Intelligence Lab at MIT. I have worked extensively on compilers, editors, debuggers, command interpreters, the Incompatible Timesharing System and the Lisp Machine operating system.

# How does a FLOSS project start?

- Small problems => unexpected innovations
  - Linux => GNU/Linux

"I'm doing a (free) operating system (just a hobby, won't be big and professional like gnu)

- took years to exert its transforming effect.
  - Python project started in 1990
- When the users of a product participate as co-developers
  - high speed
  - Good quality
  - new features
- unexpected innovations come after attacking small problems

# How does a FLOSS project start?

- Initial source code
  - becomes the seed
- A community
  - Attracted by the code itself
    - Simple, well designed, well written
  - Attracted by the authors
  - Attracted by the problem domain
  - Attracted by the coding platform
  - Attracted by the hosting platform
- Policies and rules
  - Code of conduct
  - Contribution guidelines
- The hosting platform
  - a source code repository (version control system),
  - public forum facilities, and
  - project web pages.

# FLOSS users or developers

- The FLOSS paradigm revolves around
  - how code is distributed and consumed.
    - licenses
  - how code is produced.
    - Collaborative/participatory
      - Commons based
      - Foundation backed
      - Corporate Sponsored

# Developer Motivation

- Intrinsic Motivation
  - Coding is fun.
- Extrinsic Motivation
  - Money?
    - not the main driver.
    - open source substitutes cooperation for financial transactions
  - Reputation
  - Job prospects
- Cooperation
  - Psychological, social needs

Q: Are you convinced?

Ans: Yes, no



# Management and Coordination

- Self coordination
  - 80% of activities by a small number of individuals
- core-periphery structure
  - Highly technical and Managerial tasks by core
  - Less technical and Supporting tasks e.g. bug fixing by periphery
- Community dynamics
  - Developer turnover
  - Developer progression

# The onion model

- A road map for developers' progression in open source projects

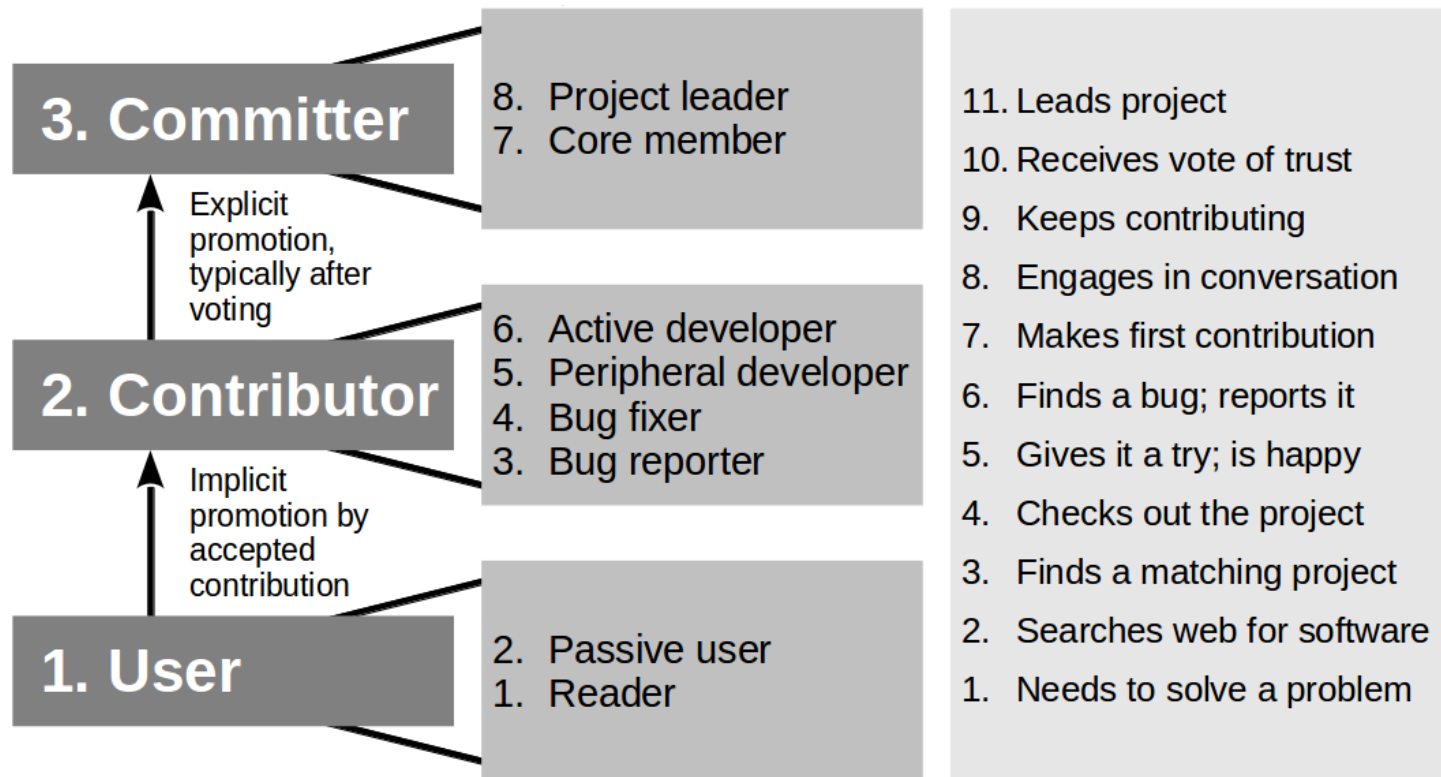


Fig. source Dirk Riehle. "How Open Source is Changing the Software Developer's Career." IEEE Computer vol. 48, no. 5 (May 2015). Page 51-57..

# Who is what in a FLOSS community?

- User - who directly use the code
- Author - project creator who creates the initial code
- Maintainer - project curator who changes/updates the code
- Committer -
- Contributor, Developer - Writing code, documentation
  - *Active contributors* (also called “regular contributors” or “long-term contributors”)
  - casual contributors or “one-time contributors,

# Community Dynamics:

- **User** may leave if they don't get what they are looking for
- **Author** may also leave at some point of team
  - Handover the project to someone
    - Interested
    - user
- **Developers** join a project due to motivation
  - when individual motivation declines members might leave, but they are replaced by new, eager faces.
- **Maintainers**
  - They might cite feelings of obligation, community, or helping others as reasons for looking after the project.
  - For solo maintainers, however, when their motivation declines the decision to leave can have serious consequences for the project.
  - After a maintainer leaves, another developer might step up.
  - A lesser-known developer may be eager for an opportunity to maintain a popular project.
  - maintainers may eventually step down, find a replacement, or disappear altogether.

**Join and Leave!!!**

# A good FLOSS product has .....

- the conditions of innovation:
  - open standards and protocols
  - to connect other software
- good documentation,
- a well-defined extension mechanism
  - that allows people to "roll their own"
- licenses to allow others to extend without getting permission.
- Last but not the least, an active community

# Past success stories

- According to Linus Torvalds,
  - Linux succeeded partly due to good design principles which paved the way for easy extensions in the future.
- Similarly, Larry Wall tells
  - Perl's feature-set evolved naturally as per needs of its users.
- A modular architecture
  - allows users to extend the system's functionality without having to change existing core functionality.
  - Which further allows the project to scale with its community and also allows the authors to retain control over the core product.

# What can you contribute?

- No contribution is small!!
- Great things happen with small contributions from people with different skills
- Coding skills
  - Fixing bugs
  - New features/enhancement
  - Code improvements
- Language skills
  - Documentation
  - Fixing typos
  - Translation in different languages
- Writing tutorials, how-to-pages, newsletters

# Open source is not only about software

- Several types of projects
  - Education
    - Curriculum
    - Books
    - Tutorials
    - Courses
  - Research
    - <https://github.com/jeanqasaur/women-in-programming-languages-research>
  - Lists
    - <https://github.com/public-apis/public-apis>
    - <https://github.com/sindresorhus/awesome>
    - <https://github.com/h5bp/Front-end-Developer-Interview-Questions>



# Contributing to an OSS project

- Technical Contributions
  - code
  - Documentation
  - Discussions
  - Reviews
  - issues
- Non technical contributions
  - like organizing events
  - providing financial support

- Developers use version control to track and synchronize their changes,
  - multiple people often strangers
  - Sharing work artefacts
  - different time zones
- A version control system, such as Git, helps developers
  - Track changes
  - Resolve any differences
  - Coordinate tasks.

# Using GitHub as the platform

- Lowers barriers to contribute
- on GitHub user interface every project has the same structure e.g. code, issues, pull requests, wiki
  - regardless of its language or function.
- Each project has a landing page in the form of a README file.
- A Code of Conduct policy to govern community behaviour.

Do you have GitHub account?

Ans: Yes, No

# What does a FLOSS project look like

- On the Github platform

The screenshot shows the GitHub repository page for 'sindresorhus/awesome'. The repository is public and has 190k stars, 22.9k forks, and 7.3k watchers. The main branch is 'main' with 2 branches and 0 tags. The repository contains a list of files and folders, including .github, media, .editorconfig, .gitattributes, awesome.md, code-of-conduct.md, contributing.md, create-list.md, and license. The repository is described as 'Meta tweaks' and has 1,030 commits. The 'About' section lists interesting topics: lists, awesome, unicorns, resources, and awesome-list. The 'Sponsor this project' section is also visible.

sindresorhus / awesome Public

Sponsor Watch 7.3k Fork 22.9k Star 190k

< Code Issues 21 Pull requests 20 Actions Security Insights

main 2 branches 0 tags

Go to file Add file Code

sindresorhus Meta tweaks 75ba1c4 2 days ago 1,030 commits

.github	Remove macOS Command Line (#2051)	6 months ago
media	Add social preview to the repo	2 years ago
.editorconfig	add .editorconfig	6 years ago
.gitattributes	Prevent the lint script from affecting repo language stats	6 months ago
awesome.md	Fix a link	13 months ago
code-of-conduct.md	Use HTTPS links	2 years ago
contributing.md	Fix typo in contributing.md (#1423)	3 years ago
create-list.md	Improve the list guidelines	3 years ago
license	Use HTTPS links	2 years ago

About

Awesome lists about all kinds of interesting topics

lists awesome unicorns resources

awesome-list

Readme CC0-1.0 License Code of conduct 190k stars 7.3k watching 22.9k forks

Sponsor this project

Lets walk around

<https://github.com/sindresorhus/awesome>

# Technical aspects of contributing

- Choose a project
  - License
  - Active or dead
    - date of the project's last commit
    - Number of open issues
    - Number of open pull requests
  - Stable or not
  - Code of conduct
  - Community
- Understand the project/codebase
- Choose a task
  - Create an issue
  - Look for open issues
- Set up the environment
- Follow the style/guidelines
- Make contributions

✓ 3c5343a yesterday 🕒 4,458 commits

🕒 Issues 3 🔗 Pull requests 10

## About

A collective list of free APIs

api lists open-source list  
development public resources dataset  
free software apis public-api  
public-apis

📖 Readme

📜 MIT License

☆ 181k stars

👁 3.5k watching

🔗 20.9k forks

# Social aspects of Contributing

- Time
- Interest
- Skills
- Patience
- Respect for different cultures
- Respect for others time

Let's be nice to each other!

# Choosing a task

- Understand the project
  - Spend time and effort
  - Identify the task
- online resources for beginners\*
  - <https://up-for-grabs.net/>
  - <https://goodfirstissue.dev/>
  - <https://www.firsttimersonly.com/>
  - <https://github.com/MunGell/awesome-for-beginners>

\*<https://dev.to/pragativerma18/how-to-pick-an-open-source-project-to-contribute-to-4kak>



# Contributor Guidelines

Front-end-Developer-Interview-Questions/CONTRIBUTING.md at main · h5bp/Front-end-Developer-Interview-Questions

---

## Contributing to Front-end Developer Interview Questions

Please take a moment to review this document in order to make the contribution process easy and effective for everyone involved.

Following these guidelines helps to communicate that you respect the time of the developers managing and developing this open source project. In return, they should reciprocate that respect in addressing your issue or assessing patches and features.

## Using the issue tracker

The [issue tracker](#) is the preferred channel for spelling mistakes, errors or any general feedback. Please respect the following restrictions:

- Please **do not** derail or troll issues. Keep the discussion on topic and respect the opinions of others.
- Please **do not** open issues or pull requests that involve including **answers** to any of the questions.

## Pull requests

Please adhere to the coding conventions used throughout the project (spelling, indentation, punctuation etc.).

Adhering to the following process is the best way to get your work included in the project:

**IMPORTANT:** By submitting patches, you agree to allow the project owners to license your work under the terms of the [MIT License](#).

# Follow a Coding/Writing Style

- Style guide
  - to keep code/content clean and uniform
  - Easy to read and understand.
- Code style as per the preferences of the project's author.

Has consistent formatting and proper spelling/grammar.

- The link and description are separated by a dash.

Example: - [AVA](...) - JavaScript test runner.

- The description starts with an uppercase character and ends with a period.
- Consistent and correct naming. For example, `Node.js` , not `NodeJS` or `node.js` .

# Follow a Coding/Writing Style

By submitting this pull request I confirm I've read and complied with the below requirements 🙌

Please read it multiple times. I spent a lot of time on these guidelines and most people miss a lot.

## Requirements for your pull request

---

- Don't waste my time. Do a good job, adhere to all the guidelines, and be responsive.
- You have to review at least 2 other [open pull requests](#). Try to prioritize unreviewed PRs, but you can also add more comments to reviewed PRs. Go through the below list when reviewing. This requirement is meant to help make the Awesome project self-sustaining. Comment here which PRs you reviewed. You're expected to put a good effort into this and to be thorough. Look at previous PR reviews for inspiration. **Just commenting "looks good" or simply marking the pull request as approved does not count!** You have to actually point out mistakes or improvement suggestions.
- You have read and understood the [instructions for creating a list](#).
- This pull request has a title in the format `Add Name of List`.
  - ✓ `Add Swift`
  - ✓ `Add Software Architecture`
  - ✗ `Update readme.md`
- The heading title of your list should be in [title case](#) format: `# Awesome Name of List`.
  - ✓ `# Awesome Swift`
  - ✓ `# Awesome Web Typography`
  - ✗ `# awesome-swift`
  - ✗ `# AwesomeSwift`

What looks better?  
Gian or GIAN

# Contributor Covenant Code of Conduct

---

## Our Pledge

---

In the interest of fostering an open and welcoming environment, we as contributors and maintainers pledge to making participation in our project and our community a harassment-free experience for everyone, regardless of age, body size, disability, ethnicity, gender identity and expression, level of experience, nationality, personal appearance, race, religion, or sexual identity and orientation.

## Our Standards

---

Examples of behavior that contributes to creating a positive environment include:

- Using welcoming and inclusive language
- Being respectful of differing viewpoints and experiences
- Gracefully accepting constructive criticism
- Focusing on what is best for the community
- Showing empathy towards other community members



Positive

Examples of unacceptable behavior by participants include:

- The use of sexualized language or imagery and unwelcome sexual attention or advances
- Trolling, insulting/derogatory comments, and personal or political attacks
- Public or private harassment
- Publishing others' private information, such as a physical or electronic address, without explicit permission
- Other conduct which could reasonably be considered inappropriate in a professional setting



Negative

<https://github.com/sindresorhus/awesome/blob/main/code-of-conduct.md>

## Our Responsibilities

---

Project maintainers are responsible for clarifying the standards of acceptable behavior and are expected to take appropriate and fair corrective action in response to any instances of unacceptable behavior.

Project maintainers have the right and responsibility to remove, edit, or reject comments, commits, code, wiki edits, issues, and other contributions that are not aligned to this Code of Conduct, or to ban temporarily or permanently any contributor for other behaviors that they deem inappropriate, threatening, offensive, or harmful.

## Scope

---

This Code of Conduct applies both within project spaces and in public spaces when an individual is representing the project or its community. Examples of representing a project or community include using an official project e-mail address, posting via an official social media account, or acting as an appointed representative at an online or offline event. Representation of a project may be further defined and clarified by project maintainers.

## Enforcement

---

Instances of abusive, harassing, or otherwise unacceptable behavior may be reported by contacting the project team at [sindresorhus@gmail.com](mailto:sindresorhus@gmail.com). All complaints will be reviewed and investigated and will result in a response that is deemed necessary and appropriate to the circumstances. The project team is obligated to maintain confidentiality with regard to the reporter of an incident. Further details of specific enforcement policies may be posted separately.

Project maintainers who do not follow or enforce the Code of Conduct in good faith may face temporary or permanent repercussions as determined by other members of the project's leadership.

# A checklist before you contribute\*

- When you've found a project you'd like to contribute to, do a quick scan to make sure that the project is suitable for accepting contributions. Otherwise, your hard work may never get a response.
- Here's a handy checklist to evaluate whether a project is good for new contributors.
- **1. Meets the definition of open source**
- Does it have a license?
  - Usually, there is a file called LICENSE in the root of the repository.

- code-of-conduct.md
- contributing.md
- create-list.md
- license
- pull\_request\_template.md
- readme.md



main ▾

awesome / license



sindresorhus/awesome is licensed under the  
**Creative Commons Zero v1.0 Universal**

The Creative Commons CC0 Public Domain Dedication waives copyright interest in a work you've created and dedicates it to the world-wide public domain. Use CC0 to opt out of copyright entirely and ensure your work has the widest reach. As with the Unlicense and typical software licenses, CC0 disclaims warranties. CC0 is very similar to the Unlicense.

## Permissions

- ✓ Commercial use
- ✓ Modification
- ✓ Distribution
- ✓ Private use

## Limitations

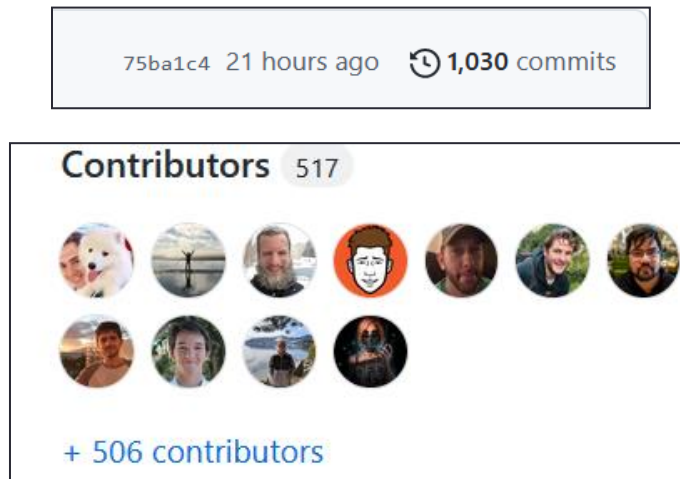
- ✗ Liability
- ✗ Trademark use
- ✗ Patent use
- ✗ Warranty

\*<https://opensource.guide/how-to-contribute/>

# A checklist before you contribute

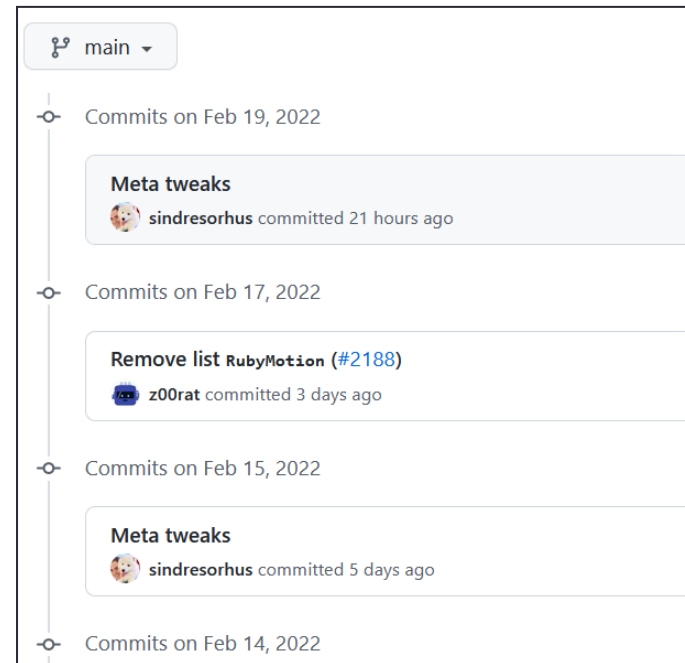
## 2. Project actively accepts contributions

- Look at the commit activity on the main branch. On GitHub, you can see this information on a repository's homepage.
- 2.1 When was the latest commit?
- 2.2 How many contributors does the project have?
- 2.3 How often do people commit? (On GitHub, you can find this by clicking "Commits" in the top bar.)



Lets walk around

<https://github.com/sindresorhus/awesome>



# A checklist before you contribute

- Next, look at the project's issues.
- 2.4 How many open issues are there?
- 2.5 Do maintainers respond quickly to issues when they are opened?
- 2.6 Is there active discussion on the issues?
- 2.7 Are the issues recent?
- 2.8 Are issues getting closed? (On GitHub, click the "closed" tab on the Issues page to see closed issues.)

The screenshot shows the GitHub interface for the repository `sindresorhus/awesome`. The repository is public and has 7.3k watches, 22.9k forks, and 190k stars. The 'Issues' tab is selected, showing 21 open issues. A search filter `is:issue is:open` is applied. The first issue is titled 'Pull request template: badge placement rule contradicts the linter', opened on 15 Nov 2021 by ZoomRmc, with 7 comments. The interface includes navigation links for Code, Issues, Pull requests, Actions, Security, and Insights, as well as buttons for Sponsor, Watch, Fork, Star, and a New issue button.

`sindresorhus/awesome` Public

Sponsor Watch 7.3k Fork 22.9k Star 190k

<> Code Issues 21 Pull requests 20 Actions Security Insights

Filters  Labels 8 Milestones 0 New issue

21 Open ✓ 276 Closed

Author Label Projects Milestones Assignee Sort

**Pull request template: badge placement rule contradicts the linter** 7

#2097 opened on 15 Nov 2021 by ZoomRmc



# A checklist before you contribute

- Now do the same for the project's pull requests.
- 2.9 How many open pull requests are there?
- 2.10 Do maintainers respond quickly to pull requests when they are opened?
- 2.11 Is there active discussion on the pull requests?
- 2.12 Are the pull requests recent?
- 2.13 How recently were any pull requests merged? (On GitHub, click the "closed" tab on the Pull Requests page to see closed PRs.)

sindresorhus / awesome Public Sponsor Watch 7.3k Fork 22.9k Star 190k

> Code Issues 21 **Pull requests 20** Actions Security Insights

Filters  Labels 8 Milestones 0 New pull request

	Author	Label	Projects	Milestones	Reviews	Assignee	Sort
<b>20 Open</b> ✓ 1,732 Closed							
<b>add Photonics</b> ✓ #2189 opened 3 days ago by joamatab							3
<b>Add Malware Persistence</b> ✗ #2184 opened 5 days ago by Karneades							6
<b>Add Open Source Games</b> ✓ #2171 opened 15 days ago by michelpereira							8

# A checklist before you contribute

- 3. Project is welcoming
- A project that is friendly and welcoming signals that they will be receptive to new contributors.
- 3.1 Do the maintainers respond helpfully to questions in issues?
- 3.2 Are people friendly in the issues, discussion forum, and chat (for example, IRC or Slack)?
- 3.3 Do pull requests get reviewed?
- 3.4 Do maintainers thank people for their contributions?

Hello @michelpereira 🍌

I spotted a couple things in this PR that do not r

This pull request has a title in the format Add

Yours should be named "Add Games".

Includes a succinct description of the project/

You should basically include a description of what is the theme of the list (open-source games) and not just a descriptive of wh  
in it (curated list of x.)

The repo should have awesome-list & awesome as GitHub topics. I encourage you to add more relevant topics.

Just update your repository.

Has a Table of Contents section. [...] Must not feature Contributing or Footnotes sections.

Remove the *Contribute* link from your TOC.

OpenNotrium - Open source version of Notrium, a top-down survival game.

This should probably go in the `Awesome Game Remakes` awesome list.

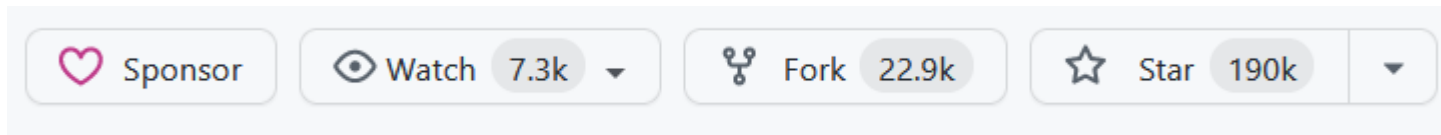
As suggested by @sindresorhus, the whole page might be worth a re-read. Some things have different spelling, eg. `JS` and `JavaScript`, which is not ideal for searching.

Thanks for making an Awesome list! 🍌

It looks like you didn't read the [guidelines](#) closely enough. I noticed multiple things that are not followed. Try going through the list point for point to ensure you follow it. I spent a lot of time creating the guidelines so I wouldn't have to comment on common mistakes, and rather spend my time improving Awesome.

# Don't waste your time!!!

- if you want to make a substantial contribution, open an issue to ask before working on it.
- It's helpful to watch the project for a while (on GitHub, you can click “Watch” to be notified of all conversations), and get to know community members, before doing work that might not get accepted.



# Use Templates and checklists

- Templates for issues and pull requests are like contact forms for customer support:
  - they help “prescreen” user requests and filter out noise.
- Maintainers sometimes use checklists
  - to make sure that contributors have read and agreed to all their conditions before submitting a pull request.

# On to the GitHub platform

- Basic Developer Workflow
- GitHub provides features for the basic developer workflow
  - Social network for programmers
  - Issue trackers to raise errors/problems/concerns/requests
  - pull requests to submit solutions/patches
- A clean way to submit, review, and merge proposed contributions.
- GitHub's user interface is easy and intuitive.
  - Compared to the old setup of downloading tarballs to get code and surfing the mailing lists to discuss the issues