

Hi 

It feels awesome to be here

Stefan Fejes

Front End Engineer

20 years old 🎂

I do JavaScript 🧑

and open source ✨

<https://stefanfejes.com>



Building an open source
community from the ground up

[30-seconds/30-seconds-of-code](#)

A curated collection of useful JavaScript snippets that you can understand in 30 seconds or less.

 JavaScript  50.5k  5.7k

[30-seconds/30-seconds-of-interviews](#)

A curated collection of common interview questions to help you prepare for your next interview.

 JavaScript  7.3k  564

[30-seconds/30-seconds-of-css](#)

A curated collection of useful CSS snippets you can understand in 30 seconds or less.

 13.1k  825

[30-seconds/30-seconds-of-react](#)

Curated collection of useful React snippets that you can understand in 30 seconds or less.

 JavaScript  2.5k  207

[30-seconds/30-seconds-of-python](#)

A curated collection of useful Python snippets that you can understand in 30 seconds or less.

 Python  5.3k  668

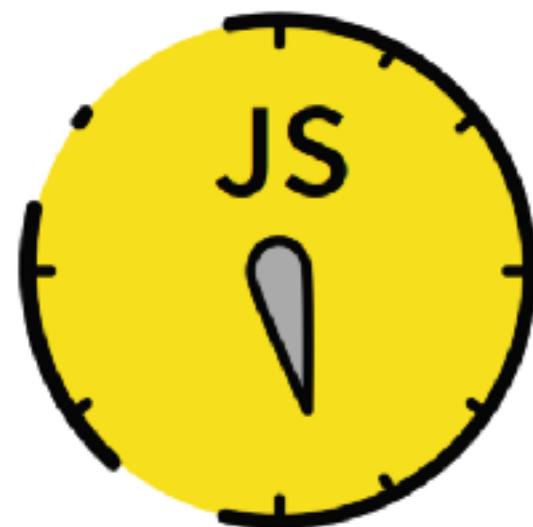
[accessibility-guide](#)

A curated collection of web accessibility tips, tricks, and best practices

 JavaScript  412  22

<https://github.com/fejes713>

@fejes713



30 seconds of code

30secondsofcode.org

Watch ▾

1.7k

Unstar

50.5k

Fork

5.7k

*“Curated collection of useful
JavaScript snippets that you
can understand in 30
seconds or less.”*

?

@fejes713

The internet will make those bad words go away



Essential

Googling the Error Message

O RLY?

*The Practical Developer
@ThePracticalDev*

@fejes713

Cutting corners to meet arbitrary management deadlines



Essential

Copying and Pasting from Stack Overflow

O'REILLY®

The Practical Developer
@ThePracticalDev

@fejes713

Software can be chaotic, but we make it work



Expert

Trying Stuff Until it Works

O RLY?

The Practical Developer
@ThePracticalDev

@fejes713



@fejes713

30 seconds of code

← Back to Search



debounce

FUNCTION INTERMEDIATE

Creates a debounced function that delays invoking the provided function until at least `ms` milliseconds have elapsed since the last time it was invoked.

Each time the debounced function is invoked, clear the current pending timeout with `clearTimeout()` and use `setTimeout()` to create a new timeout that delays invoking the function until at least `ms` milliseconds has elapsed. Use `Function.prototype.apply()` to apply the `this` context to the function and provide the necessary arguments. Omit the second argument, `ms`, to set the timeout at a default of 0 ms.



```
const debounce = (fn, ms = 0) => {
  let timeoutId;
  return function(...args) {
    clearTimeout(timeoutId);
    timeoutId = setTimeout(() => fn.apply(this, args), ms);
  };
};
```

[+ EXAMPLES]

30 seconds of code

← Back to Search



debounce

FUNCTION INTERMEDIATE

Creates a debounced function that delays invoking the provided function until at least `ms` milliseconds have elapsed since the last time it was invoked.

Each time the debounced function is invoked, clear the current pending timeout with `clearTimeout()` and use `setTimeout()` to create a new timeout that delays invoking the function until at least `ms` milliseconds has elapsed. Use `Function.prototype.apply()` to apply the `this` context to the function and provide the necessary arguments. Omit the second argument, `ms`, to set the timeout at a default of 0 ms.



```
const debounce = (fn, ms = 0) => {
  let timeoutId;
  return function(...args) {
    clearTimeout(timeoutId);
    timeoutId = setTimeout(() => fn.apply(this, args), ms);
  };
};
```

EXAMPLES

```
window.addEventListener(
  'resize',
  debounce(() => {
    console.log(window.innerWidth);
    console.log(window.innerHeight);
  }, 250)
); // Will log the window dimensions at most every 250ms
```

in numbers:

300+ snippets made by 180+ contributors

5000 commits, 800 merged PRs and 200 solved issues

6th fastest growing open source repo of 2018

an idea that started series of educational projects

So... How it all started?



What was your dream job
when you were a child?



This was my dream spaceship when I was a kid. 🚀

@fejes713

I liked the fact that you could do
something that will be
appreciated by the community.

... my first days in tech were
not that great

How I used to cheat in high school



friend's code:

```
return a + b;
```



my code:

```
return x + y;
```

I was just changing variable names



The test in which I got caught:



friend's code:

```
static void swap(int a, int b, ref char[] arr)
{
    char trt = arr[a];
    arr[a] = arr[b];
    arr[b] = trt;
}
```



my code:

```
static void swap(int x, int y, ref char[] arr)
{
    char trt = arr[x];
    arr[x] = arr[y];
    arr[y] = trt;
}
```

The test in which I got caught:



friend's code:

```
static void swap(int a, int b, ref char[] arr)
{
    char trt = arr[a];
    arr[a] = arr[b];
    arr[b] = trt;
}
```



my code:

```
static void swap(int x, int y, ref char[] arr)
{
    char trt = arr[x];
    arr[x] = arr[y];
    arr[y] = trt;
}
```

first commit - 12/12/2017

Add reverseString #12

Merged Chalarangelo merged 1 commit into 30-seconds:master from unknown repository on Dec 12, 2017

Conversation 1

fejes713 commented 4 minutes ago
No description

Add reverseString #12

Chalarangelo commented 4 minutes ago
Chalarangelo left a comment

Thanks! I will merge it. I am not sure if I will merge snippet files, now.

1 heart

Chalarangelo bookmarked 4 minutes ago

Created confirm ending of the string and reverse string #14

Closed 2 participants

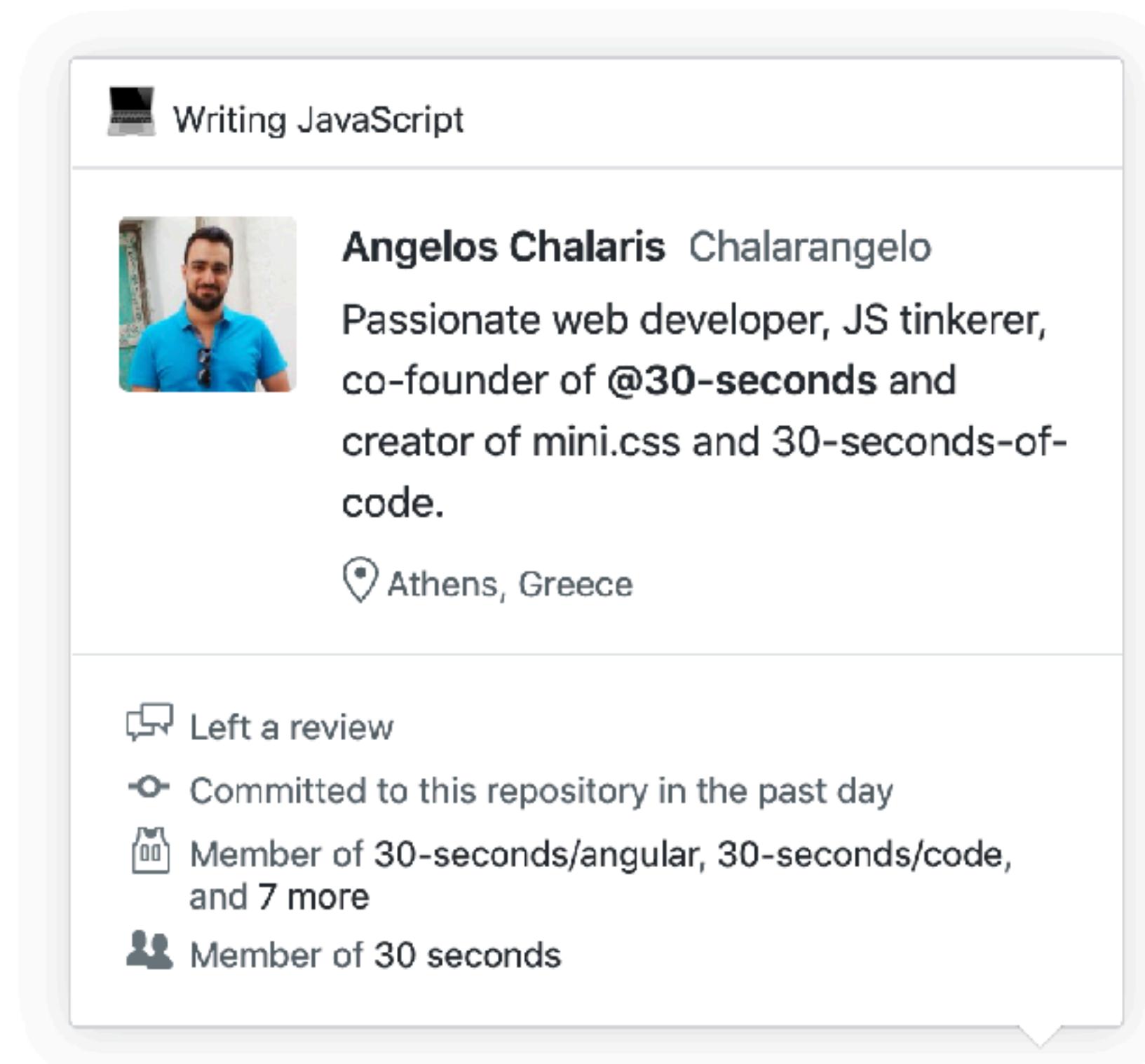
fejes713 commented 4 minutes ago

@Chalarangelo the good old times 😺

Lock conversation

@fejes713

Thanks to @Chalarangelo's fast and
supportive response, that was the day I said
I'd like to contribute to and maintain
30secondsofcode.org



Thanks to **@Chalarangelo's** fast and supportive response, that was the day I said I'd like to contribute to and maintain *30secondsofcode.org*

takeaway #1:

Let every idea, no matter how
crazy show its true potential.

Finding an idea:

open source existing personal projects

open source a part of your work project

start something new 

... next few weeks were chaos



E118

Open source is a team effort

Finding a team:

It's okay to start alone (and form a team later on)

Talk to friends and colleagues about your idea

Like somebody else's idea? Join them.



Search or jump to...

/ Pull requests Issues Marketplace Explore



fejes713 / github-universe-talk Private

Unwatch 4 Star 0 Fork 0

Code

Issues 2

Pull requests 0

Projects 0

Wiki

Security

Insights

Settings

Stefan's talk about 30 seconds of code movement for GitHub Universe 2019

Edit

Manage topics

3 commits

1 branch

0 releases

2 contributors

Branch: master

New pull request

Create new file

Upload files

Find file

Clone or download



fejes713 Merge pull request #3 from fejes713/badges

Latest commit 13da023 20 days ago



README.md

Add badges

20 days ago



README.md



github-universe-talk

build passing test passing

Stefan's talk about 30 seconds of code movement for GitHub Universe 2019



takeaway #2:

Having great people around
you is essential

“The team was always very helpful and encouraging which made the atmosphere of the repo very welcoming and I felt great to be a part of it” - @atomiks

“The team was always very helpful and encouraging which made the atmosphere of the repo very welcoming and I felt great to be a part of it” - @atomiks

“I felt great to be a part of it”

takeaway #3:

Build an amazing developer
experience around the project



- Anyone can have
a secret formula.

Developer experience checklist

- quality codebase
- document your code
- simplify the onboarding experience
- welcome first-time contributors

quality codebase:

Stick to a coding style guide,
be consistent

quality codebase:

Organize everything and
remove clutter

quality codebase:

Use code linters and
formatters on git-hooks

Developer experience checklist

- quality codebase
- document your code
- simplify the onboarding experience
- welcome first-time contributors

document your code:

Traditional advice applies
here, and a bit more than that

Good commits 101

a guide to hidden documentation



@fejes713

good commits 101:

Tell stories through your
commits

git commit -m “lalalalala”

don't do this 🚫

short one line title

**Longer description of what the change does
(if the title isn't enough).**

**An explanation of why the change is being
made.**

**Perhaps a discussion of context and/or
alternatives that were considered**

short one line title

**Longer description of what the change does
(if the title isn't enough).**

**An explanation of why the change is being
made.**

**Perhaps a discussion of context and/or
alternatives that were considered**

short one line title

**Longer description of what the change does
(if the title isn't enough).**

**An explanation of why the change is being
made.**

**Perhaps a discussion of context and/or
alternatives that were considered**

short one line title

**Longer description of what the change does
(if the title isn't enough).**

**An explanation of why the change is being
made.**

**Perhaps a discussion of context and/or
alternatives that were considered**

```
git log -grep "403"
```

A screenshot of the Visual Studio Code interface. The title bar reads "COMMIT_EDITMSG — 30-seconds-of-code". The left sidebar shows a file tree with the following structure:

- OPEN EDITORS 1 UNSAVED
- .git > COMMIT_EDITMSG
- 30-SECONDS-OF-CODE
 - > .github
 - > .travis
 - > assets
 - > cist
 - > glossary
 - > scripts
 - > snippet_data
 - > snippets
 - > snippets_archive
- src
 - docs
 - > components
 - pages
 - JS 404.js
 - JS about.js
 - JS archive.js
 - JS glossary.js
 - JS index.js M
 - JS list.js
 - JS search.js
 - > state
 - > styles
 - > templates
 - > util
 - > static-parts
 - > static
 - > test
 - > vscode_snippets
 - _headers
 - .eslintrc.json
 - .gitignore
 - .mdlrc
 - > OUTLINE
 - > NPM SCRIPTS

The main editor area displays the following commit message template:

```
1
2
3 # Please enter the commit message for your changes. Lines starting
4 # with '#' will be ignored, and an empty message aborts the commit.
5 #
6 # On branch master
7 # Your branch is up to date with 'origin/master'.
8 #
9 # Changes to be committed:
10 # modified:   src/docs/pages/index.js
11 #
12
```

At the bottom, the status bar shows: Ln 1, Col 1 Spaces: 2 UTF-8 LF Git Commit Message Formatting: X 1

@fejes713

references:

<https://blog.mocoson.co.uk/talks/2015/01/12/telling-stories-through-your-commits/>

<https://github.com/k88hudson/git-flight-rules>

Developer experience checklist

- quality codebase
- document your code
- simplify the onboarding experience
- welcome first-time contributors

simplify the onboarding experience:

**Make README file appealing - it's
the first thing somebody sees
when they enter your project**

README.md content

The screenshot shows the contents of a README.md file. At the top, there's a yellow circular icon with 'JS' and a cursor icon inside. To its right, the text '30 seconds of code' is displayed. Below this, the title '30 seconds of code' is repeated. A horizontal line follows. Underneath, there's a row of GitHub-style status badges: License (CC0-1.0), downloads (23k), npm (v1.2.3), vulnerabilities (0), build (passing), awesome, product Hunt (vote), code style (semistandard), PRs (welcome), and welcome. A note below the badges states: 'Curated collection of useful JavaScript snippets that you can understand in 30 seconds or less.' A bulleted list provides instructions for using the snippets. At the bottom, there's a 'Related projects' section with links to other repositories.

- Use `ctrl + F` or `command + F` to search for a snippet.
- Contributions welcome, please read the [contribution guide](#).
- Snippets are written in ES6, use the [Babel transpiler](#) to ensure backwards-compatibility.
- You can import these snippets into VSCode, by following the instructions found [here](#).
- You can search, view and copy these snippets from a terminal, using the CLI application from [this repo](#).
- If you want to follow 30-seconds-of-code on social media, you can find us on [Facebook](#), [Instagram](#) and [Twitter](#).

Related projects

- [30 Seconds of CSS](#)
- [30 Seconds of Interviews](#)
- [30 Seconds of React](#)
- [30 Seconds of Python](#)
- [30 Seconds of PHP](#)
- [30 Seconds of Knowledge](#)

- 1.name and catchy description
- 2.nice graphics
- 3.quick start guide
- 4.links to docs
- 5.contributing guidelines
- 6.code of conduct
- 7.licence

simplify the onboarding experience:

Write good
CONTRIBUTING.md

what, where, how, who

The screenshot shows a GitHub README file with the following content:

95 lines (86 sloc) | 8.44 KB Raw Blame History

CONTRIBUTION GUIDELINES

30 seconds of code is a community effort, so feel free to contribute in any way you can. Every contribution helps!

Here's what you can do to help:

- Submit [pull requests](#) with snippets and tests that you have created (see below for guidelines).
- [Open issues](#) for things you want to see added or modified.
- Be part of the discussion by helping out with [existing issues](#) or talking on our [gitter channel](#).
- Tag uncategorized snippets, update snippet tags to better categorize snippets.
- Fix typos in existing snippets, improve snippet descriptions and explanations or provide better examples.
- Write tests for existing snippets (see below for guidelines).

SNIPPET SUBMISSION AND PULL REQUEST GUIDELINES

- DO NOT MODIFY ANY OF THE README.md FILES!** Make changes to individual snippet files. **Travis CI** will automatically build the `README.md` files when your pull request is merged.
- Snippet filenames** must correspond to the title of the snippet. For example, if your snippet is titled `awesomeSnippet`, the filename should be `awesomeSnippet.md`.
 - Use `camelCase`, not `kebab-case` or `snake_case`.
 - Avoid capitalization of words, except if the whole word is capitalized (e.g. `URL` should be capitalized in the filename and the snippet title).
- Snippet metadata** must be included in all snippets in the form of frontmatter.
 - All snippets must contain a title.
 - All snippets must contain tags, prefixed with `tags:` and separated by commas (optional spaces in-between).
 - Make sure the first tag in your snippet's tags is one of the main categories, as seen in the `README.md` file or the website.
 - Snippet tags must include a difficulty setting (`beginner`, `intermediate` or `advanced`), preferably at the end of the list.
- Snippet titles** should be the same as the name of the function that is present in the snippet.
 - All snippet titles must be prefixed with `title:` and be at the very first line of your snippet's frontmatter.

@fejes713

The screenshot shows the GitHub interface for creating a new issue. At the top, there are navigation links: Code, Issues (675), Pull requests (131), Projects (0), Insights, and Settings. The 'Issues' link is highlighted with an orange bar.

The main title is 'Issue: Bug report'. Below it, a sub-instruction says: 'Create a report to help us improve. If this doesn't look right, choose a different type.' A GitHub logo icon is present.

The form includes fields for 'Title' (with a placeholder 'What happened?'), 'Write' (selected tab), 'Preview', and a rich text editor toolbar with icons for bold, italic, quote, code, and other styling options. Below the toolbar is a text area for 'Leave a comment' and a file attachment section with the instruction: 'Attach files by dragging & dropping, selecting them, or pasting from the clipboard.'

At the bottom left is a note: 'Styling with Markdown is supported', and at the bottom right is a green 'Submit new issue' button.

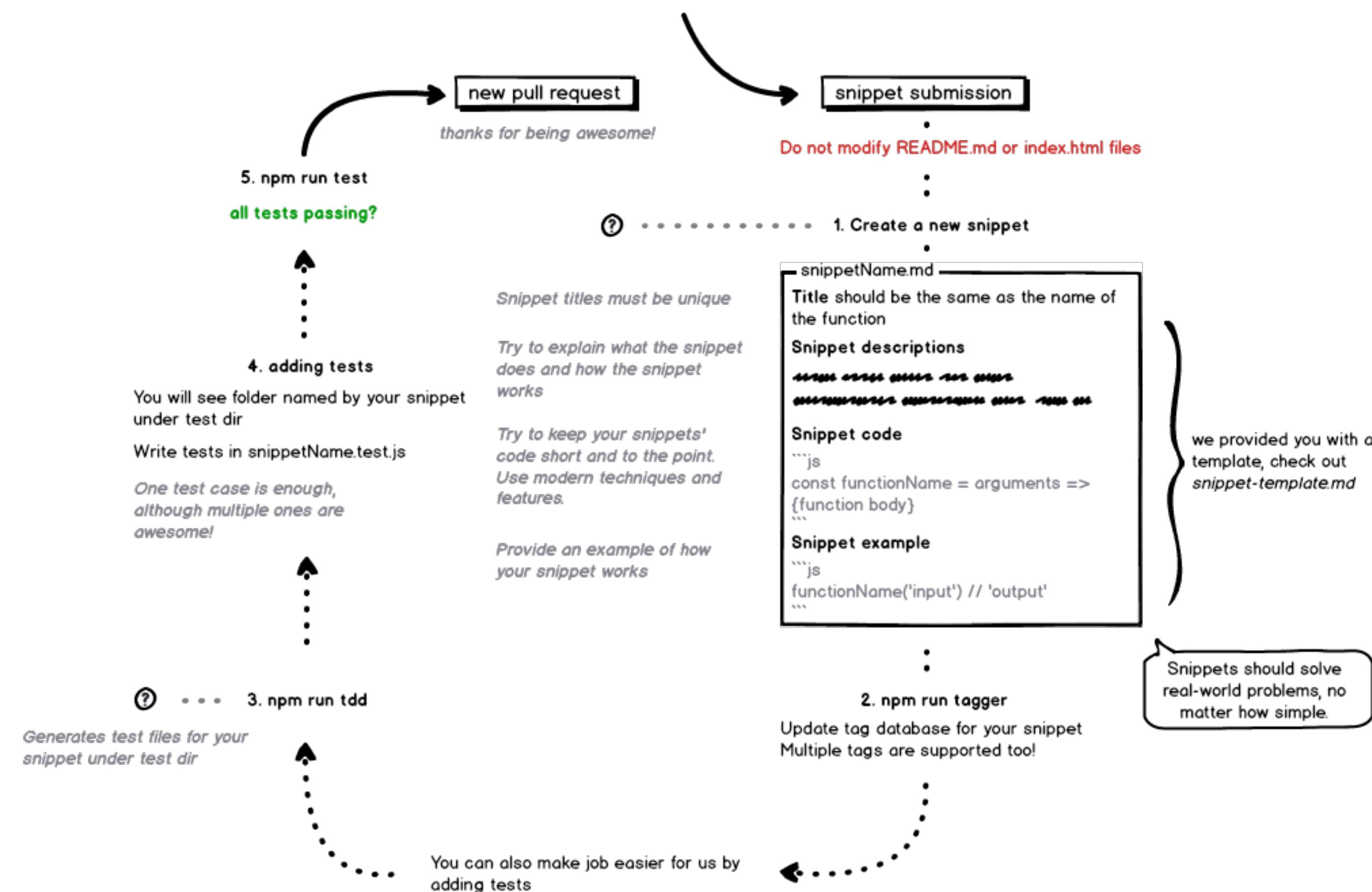
A modal window titled 'None yet' is open on the right, featuring a yellow hand icon and the message: 'It looks like this is your first time opening an issue in this project! Be sure to review the [contributing guidelines](#) and [code of conduct](#).'. Below this, another section says: 'Looking for help? Check out the project's instructions for getting support.'

source: help.github.com/en/articles/setting-guidelines-for-repository-contributors

simplify the onboarding experience:

Explain like I'm 5

Contribution guidelines



source: <https://github.com/30-seconds/30-seconds-of-code>



Changes approved

1 approving review [Learn more.](#)

[Show all reviewers](#)



All checks have passed

4 successful checks

[Show all checks](#)



This branch has no conflicts with the base branch

Merging can be performed automatically.

[Merge pull request](#)



You can also [open this in GitHub Desktop](#) or view [command line instructions](#).

@fejes713

Developer experience checklist

- quality codebase
- document your code
- simplify the onboarding experience
- welcome first-time contributors

With over 40.000.000 users,
GitHub is the right place to start
building the community

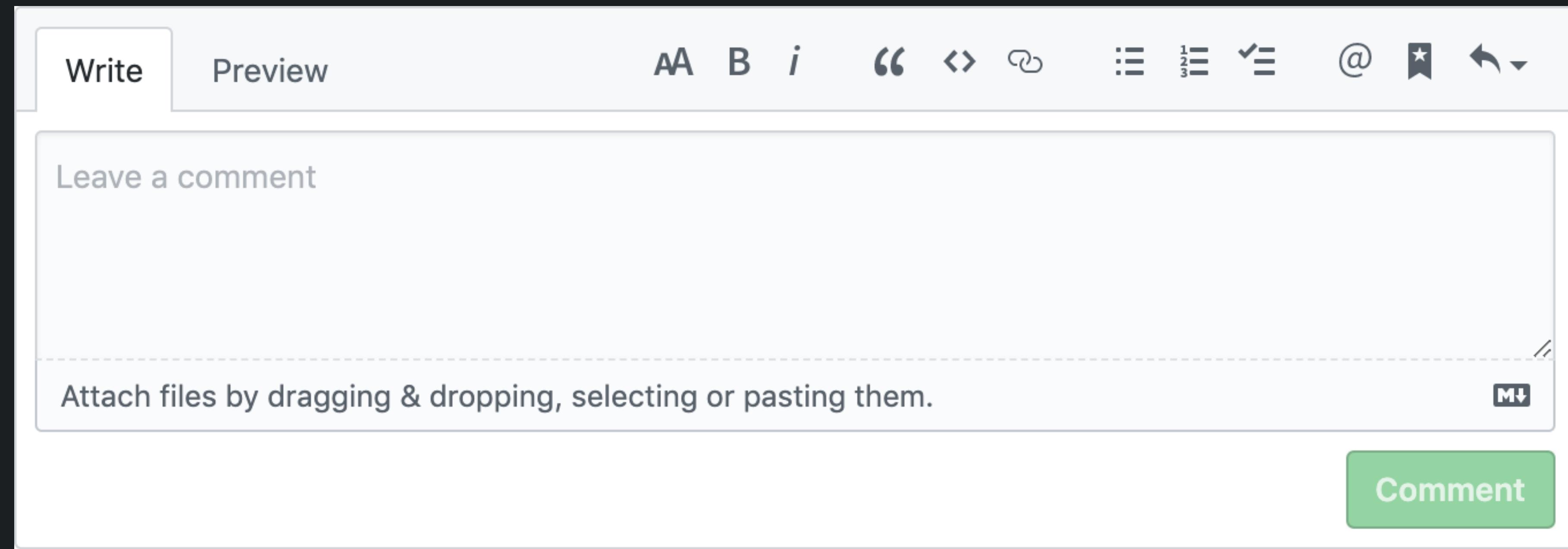


“If I make 8 pull requests, do I get 2 free shirts?” 😅

A screenshot of a GitHub pull request comparison interface. At the top, there are dropdown menus for 'base repository' (set to '30-seconds/30-seconds-of...'), 'base' (set to 'master'), 'head repository' (set to 'fejes713/30-seconds-of-code'), and 'compare' (set to 'master'). Below these, a green checkmark icon indicates that the branches are 'Able to merge'. A button labeled 'Create pull request' is visible, along with a note to 'Discuss and review the changes in this comparison with others.' and a help icon.

1. Find motivation
2. Figure out the ways to contribute good first issue
3. Fork the repository
4. Make changes
5. Open a Pull Request

It takes just a bit of negativity
to remove motivation



@fejes713

... be nice to everybody.



fejes713 approved these changes 17 days ago

[View changes](#)

fejes713 left a comment

Member + 😊 ...

Hey @ananyaneogi 🙌 PR Looks good to me 😊

Thank you so much for adding this question ❤️ This is definitely a useful one for sure!

The plan is to collect a few more questions and then introduce the accessibility filter on our website.

@fejes713



ananyaneogi commented 17 days ago

Author

Contributor

+ 😊 ...

Thanks! @fejes713 😊 I will try to add more questions under the accessibility tag!



1



1

@fejes713

**When communicating on a project, don't
use complicated language for people
who didn't grow up speaking English**

takeaway #4:

Treat contributors like guests.
Make them feel like they're home.

developer experience checklist

- quality codebase
- document your code
- simplify the onboarding experience
- welcome first-time contributors

COLLECTION

Learn to Code

Resources to help people learn to code

[Suggest edits](#)



[30-seconds / 30-seconds-of-code](#)

[★ Unstar](#)

A curated collection of useful JavaScript snippets that you can understand in 30 seconds or less.

★ 50335

⌚ 5729

● JavaScript

@fejes713



30 Seconds of CSS

A curated collection of useful CSS snippets you can understand in 30 seconds or less.

css.30secondsofcode.org

@fejes713

How do you compare two objects in JavaScript?

[Hide answer ^](#)

Even though two different objects can have the same properties with equal values, they are not considered equal when compared using `==` or `===`. This is because they are being compared by their reference (location in memory), unlike primitive values which are compared by value.

In order to test if two objects are equal in structure, a helper function is required. It will iterate through the own properties of each object to test if they have the same values, including nested objects. Optionally, the prototypes of the objects may also be tested for equivalence by passing `true` as the 3rd argument.

Note: this technique does not attempt to test equivalence of data structures other than plain objects, arrays, functions, dates and primitive values.

```
function isDeepEqual(obj1, obj2, testPrototypes = false) {
  if (obj1 === obj2) {
    return true
  }

  if (typeof obj1 === "function" && typeof obj2 === "function") {
    return obj1.toString() === obj2.toString()
  }

  if (obj1 instanceof Date && obj2 instanceof Date) {
    return obj1.getTime() === obj2.getTime()
  }
}
```



React

Ticker

Renders a ticker component.

- The ticker state is initially set to zero
- When the Tick! button is clicked, timer is incremented periodically at the given interval
- When the Reset button is clicked, the value of the timer is set to zero and the setInterval is cleared
- The setInterval is cleared once the desired time is reached
- time and interval are the required props

```
class Ticker extends Component {
  constructor(props) {
    super(props);
    this.state = {ticker: 0}
    this.interval = null
  }

  tick = () => {
    this.reset()
    this.interval = setInterval(() => {
      this.setState({ticker: this.state.ticker + 1})
    }, props.interval)
  }

  reset = () => {
    this.setState({ticker: 0})
  }
}
```

30secondsofknowledge.com

@fejes713

takeaway #5:

Don't be anxious about
numbers

⬇ Pinned Tweet



Ujjwal Sharma
@ryzokuken

▼

When a subway station is a better programmer than you are...



10:17 AM · Aug 17, 2018 · Twitter for Android

5.8K Retweets 22.2K Likes

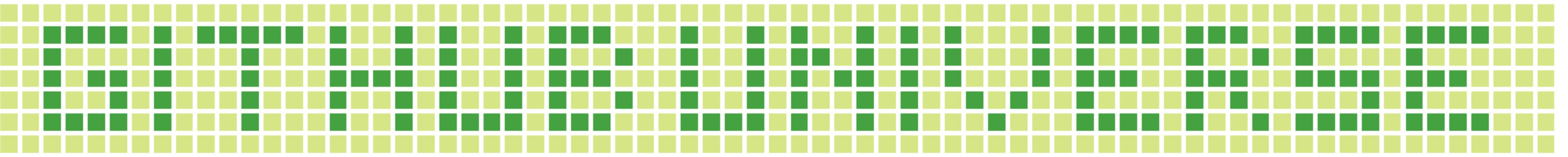


@fejes713

The very first Christmas after product launch:



I made 83 commits for 83 unit tests 😅



@fejes713

📌 Pinned Tweet



Jason Lengstorf but spoopier 🐶

@jlengstorf



This is a periodic reminder: open source maintainers are doing their best to keep up with a firehose of questions, bugs, imperfect docs, and so many other tasks it's easy to take for granted. Please thank and be kind to OSS maintainers. Usually they only hear the bad feedback.

4:35 PM · Feb 23, 2019 from [Portland, OR](#) · Twitter for Android

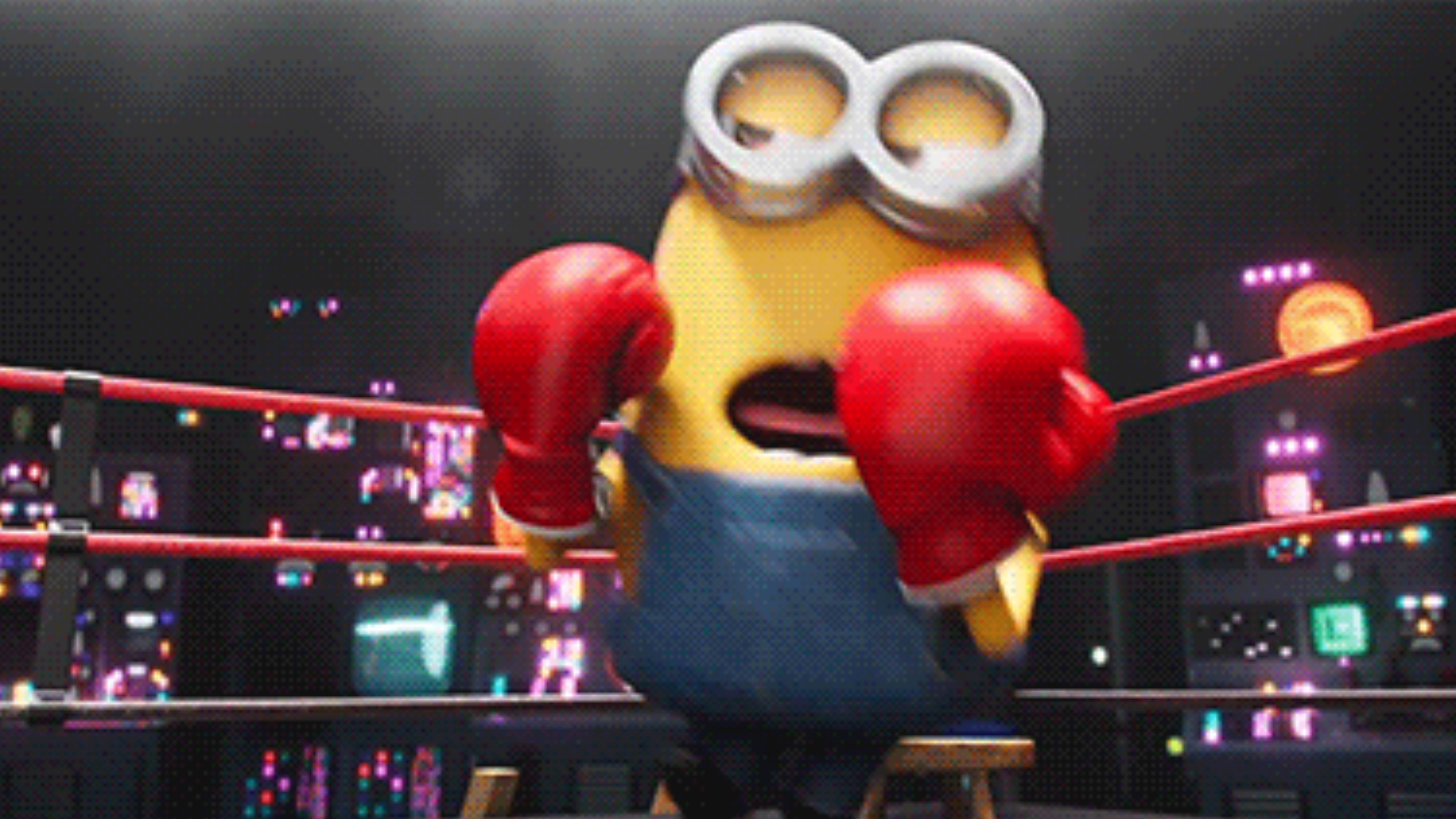
501 Retweets 1.8K Likes



@fejes713

takeaway #6:

Haters are always louder



Feedback on 30secondsofinterviews.org

A: “*Yeah, good luck finding a developer who can answer that without some major googling and debugging. I interviewed ~50 front end developers in NYC, and I can guarantee not one of them would answer that. I wouldn't answer that. Calling that question an "intermediate" is insanity.*”

B: “*Coming out of a bootcamp in NYC, we could all answer that.*”



contributor

twitter: @fejes713

mail: ns.fejes.stefan@gmail.com

web: stefanfejes.com

Thank you