Package Management Learnings from Homebrew

Mike McQuaid

Mike McQuaid

https://mikemcquaid.com

Why?

brew install
works: no-one cares

brew install breaks: we're fucked

my toilet works: no-one cares

my toilet breaks: shit everywhere

Act 1

5.0.0

"Homebrew is slow"

"Bundler is slow"

"Cargo is fast"

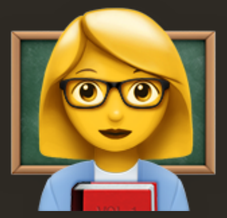
"uv is fast"

"Bundler is fast"

"$brew in Rust is fast"

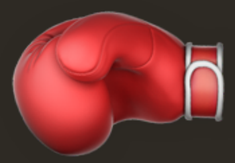
`HOMEBREW_DOWNLOAD_CONCURRENCY=auto`

`HOMEBREW_USE_INTERNAL_API=1`



brew prof

ghcr.io



bash

"Homebrew is fast(er)"

Act 2

Support

"Homebrew compiles"

"Homebrew extracts"

Homebrew on macOS x86_64

Homebrew on macOS x86_64
Homebrew on Linux x86_64

Homebrew on macOS x86_64
Homebrew on Linux x86_64
Homebrew on macOS arm64
Homebrew on Linux arm64

Homebrew on macOS x86_64
Homebrew on Linux x86_64
Homebrew on macOS arm64
Homebrew on Linux arm64
Homebrew on …?

Homebrew on macOS 26
Homebrew on macOS 15
Homebrew on macOS 14

Homebrew on macOS 26 x86_64
Homebrew on macOS 26 arm64
Homebrew on macOS 15 x86_64
Homebrew on macOS 15 arm64
Homebrew on macOS 14 x86_64
Homebrew on macOS 14 arm64

Homebrew on macOS 26 x86_64
Homebrew on macOS 26 arm64
Homebrew on macOS 15 x86_64
Homebrew on macOS 15 arm64
Homebrew on macOS 14 x86_64
Homebrew on macOS 14 arm64
Homebrew on Linux x86_64
Homebrew on Linux arm64

"Homebrew is slow"

Homebrew on macOS 26 arm64
Homebrew on macOS 15 arm64
Homebrew on macOS 14 x86_64
Homebrew on macOS 14 arm64
Homebrew on Linux x86_64
Homebrew on Linux arm64

26-09

Homebrew on macOS 27 arm64
Homebrew on macOS 26 arm64
Homebrew on macOS 15 arm64
Homebrew on Linux x86_64
Homebrew on Linux arm64

"Homebrew hates me"

"Homebrew cannot support me"

📲 `--no-quarantine`

⚠️ `odeprecated`

⛔ `odisabled`

🔥 deleted

Platform Support

Support for different platforms ("targets") are organized into three tiers, each with a different set of guarantees. For more information on the policies for targets at each tier, see the Target Tier Policy.

Targets are identified by their "target triple" which is the string to inform the compiler what kind of output should be produced.

Component availability is tracked here.

Tier 1 with Host Tools

Tier 1 targets can be thought of as "guaranteed to work". The Rust project builds official binary releases for each tier 1 target, and automated testing ensures that each tier 1 target builds and passes tests after each change.

Tier 1 targets with host tools additionally support running tools like `rustc` and `cargo` natively on the target, and automated testing ensures that tests pass for the host tools as well. This allows the target to be used as a development platform, not just a compilation target. For the full requirements, see Tier 1 with Host Tools in the Target Tier Policy.

All tier 1 targets with host tools support the full standard library.

https://doc.rust-lang.org/rustc/platform-support.html

Homebrew Documentation

Search ⌘ K

Support Tiers

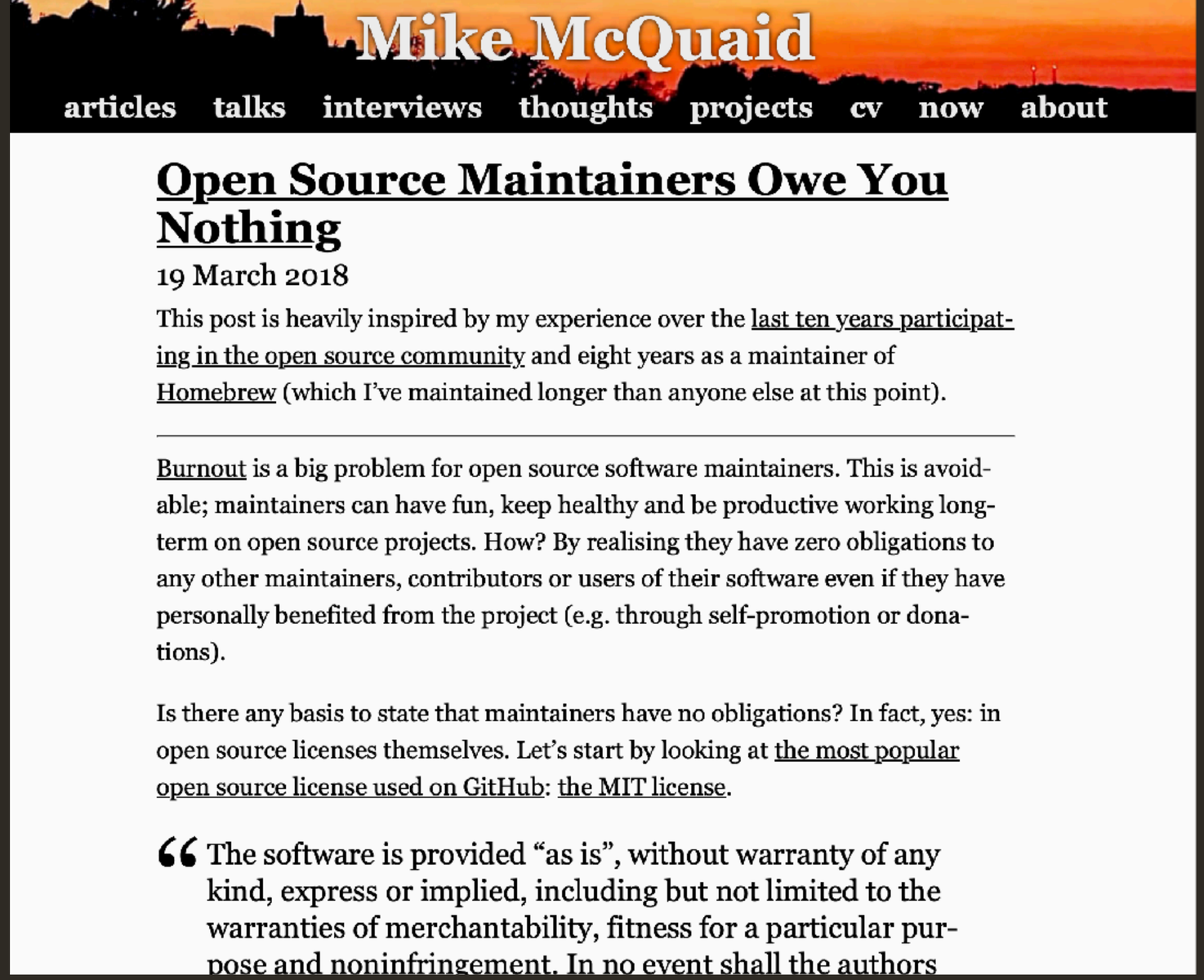
Homebrew defines three support tiers to help users understand how well Homebrew is expected to work on different systems.

These tiers describe the level of compatibility, automation coverage, and community support that the project actively maintains. They also set expectations for how we handle issues, pull requests, and regressions.

Tier 1

A Tier 1 configuration is considered fully supported. These configurations receive the highest level of CI coverage and are prioritized during issue review and formula development.

https://docs.brew.sh/Support-Tiers



https://mmq.lol/nothing

Act 3

Money

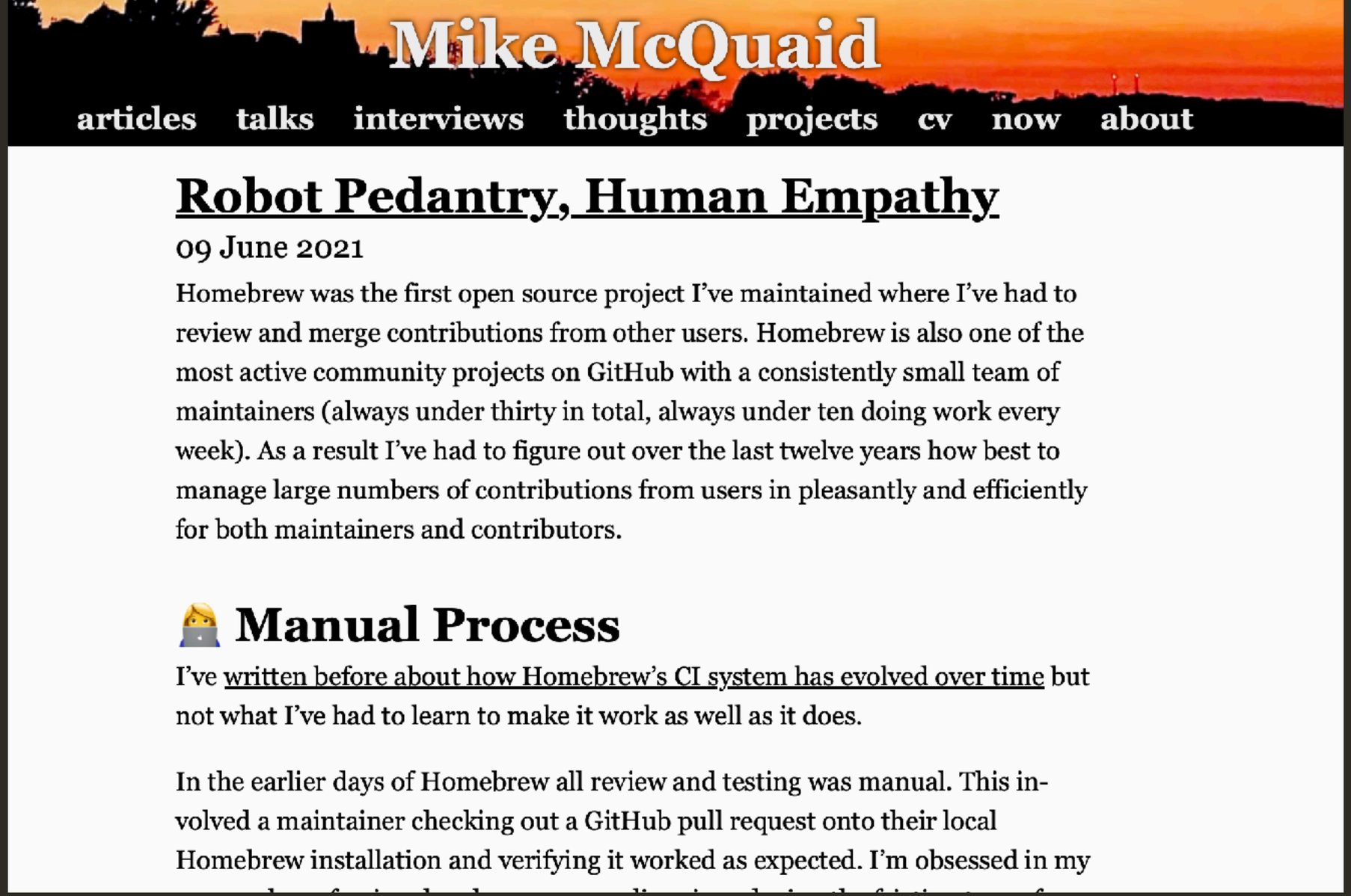
~~Money~~

Code

Code

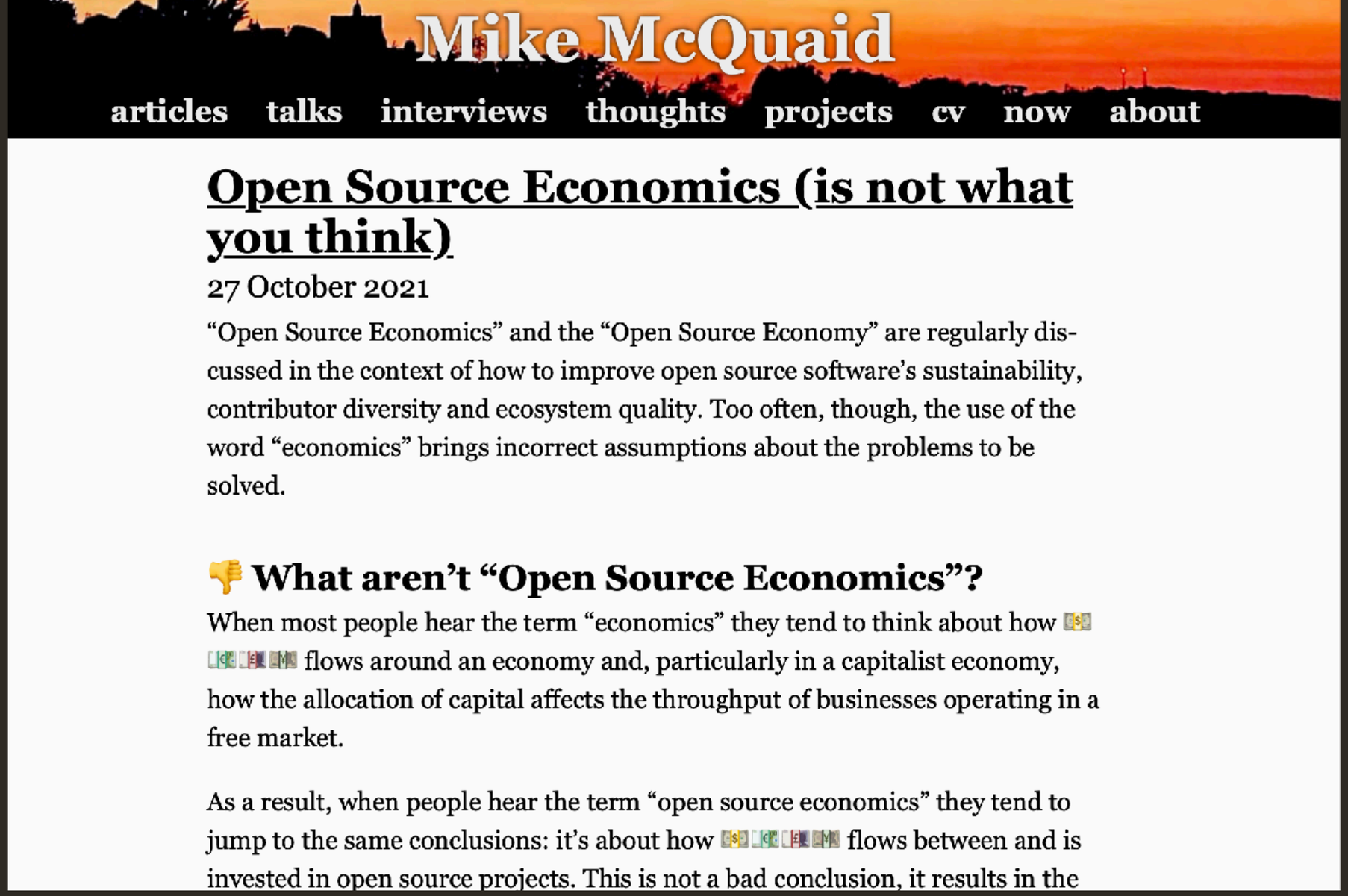
Humans

self promotion



https://mmq.lol/robot

https://mmq.lol/funnel



https://mmq.lol/economics

Mike McQuaid

articles talks interviews thoughts projects cv now about

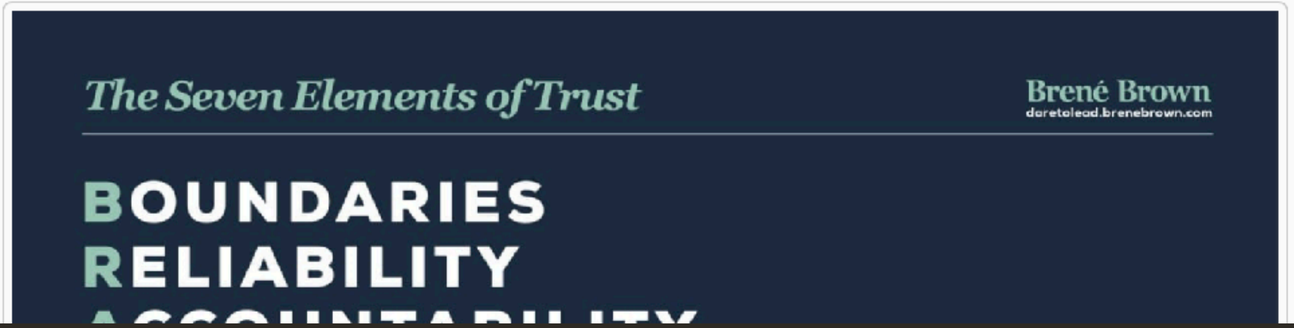
Saying No

20 January 2022

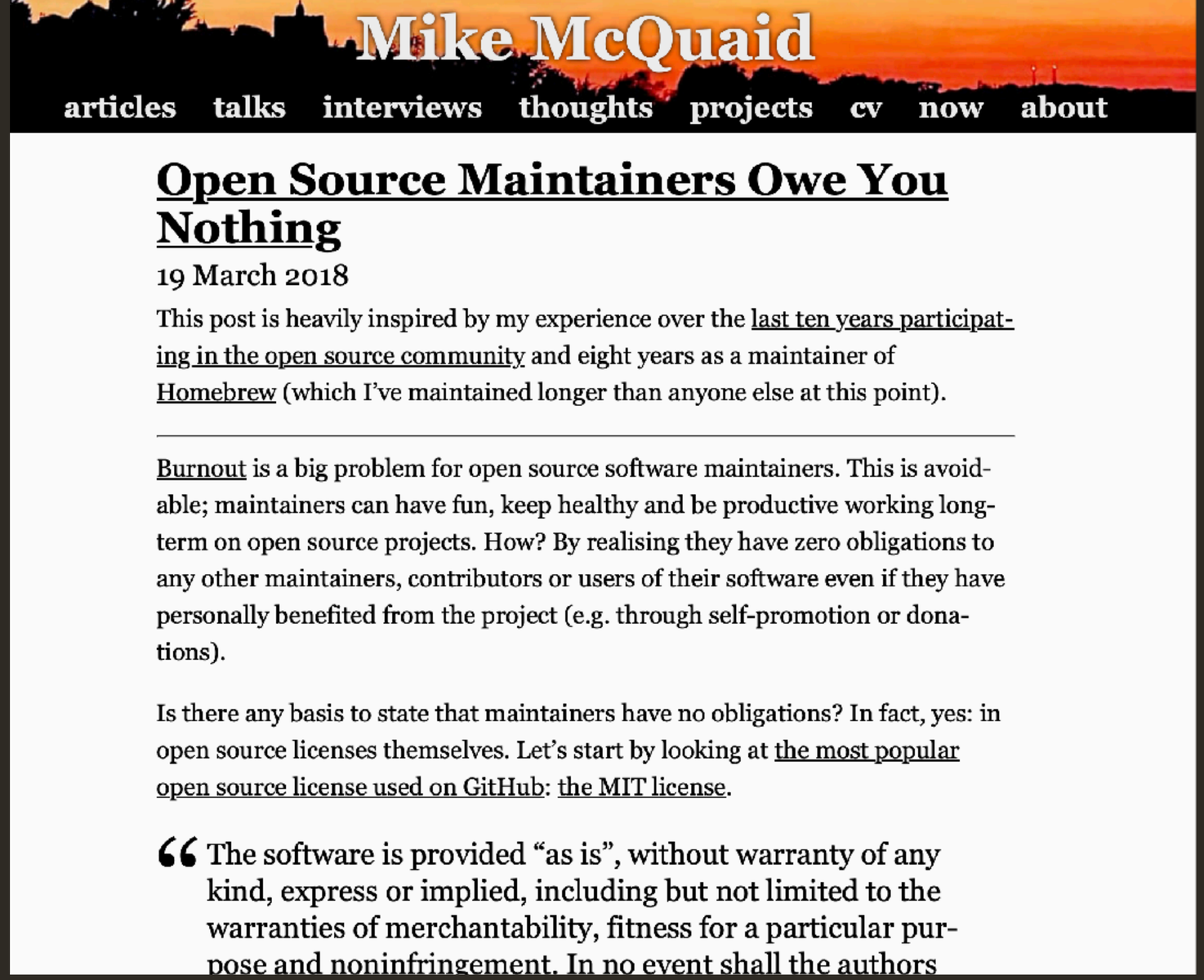
I've developed a bit of a reputation inside and outside work as being pretty good with "saying no" and setting boundaries. In this post, I will help you do so, too.

Why say "no"?

I've been consuming a lot of Brené Brown's podcasts and read one of her books recently, and they've got me thinking about what's at the root of being able to say "no" regularly and well.



https://mmq.lol/no



https://mmq.lol/nothing

Acts

1. Default to fast
2. Publish support reality
3. Optimise for maintainers

You!

 Make something faster

 1 sentence "supported"

 Set maintainer boundary

Questions?

mike@brew.sh
https://mikemcquaid.com