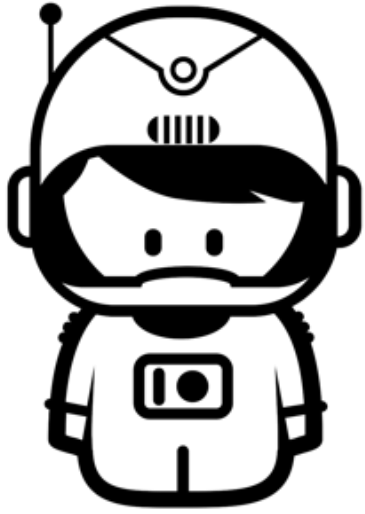
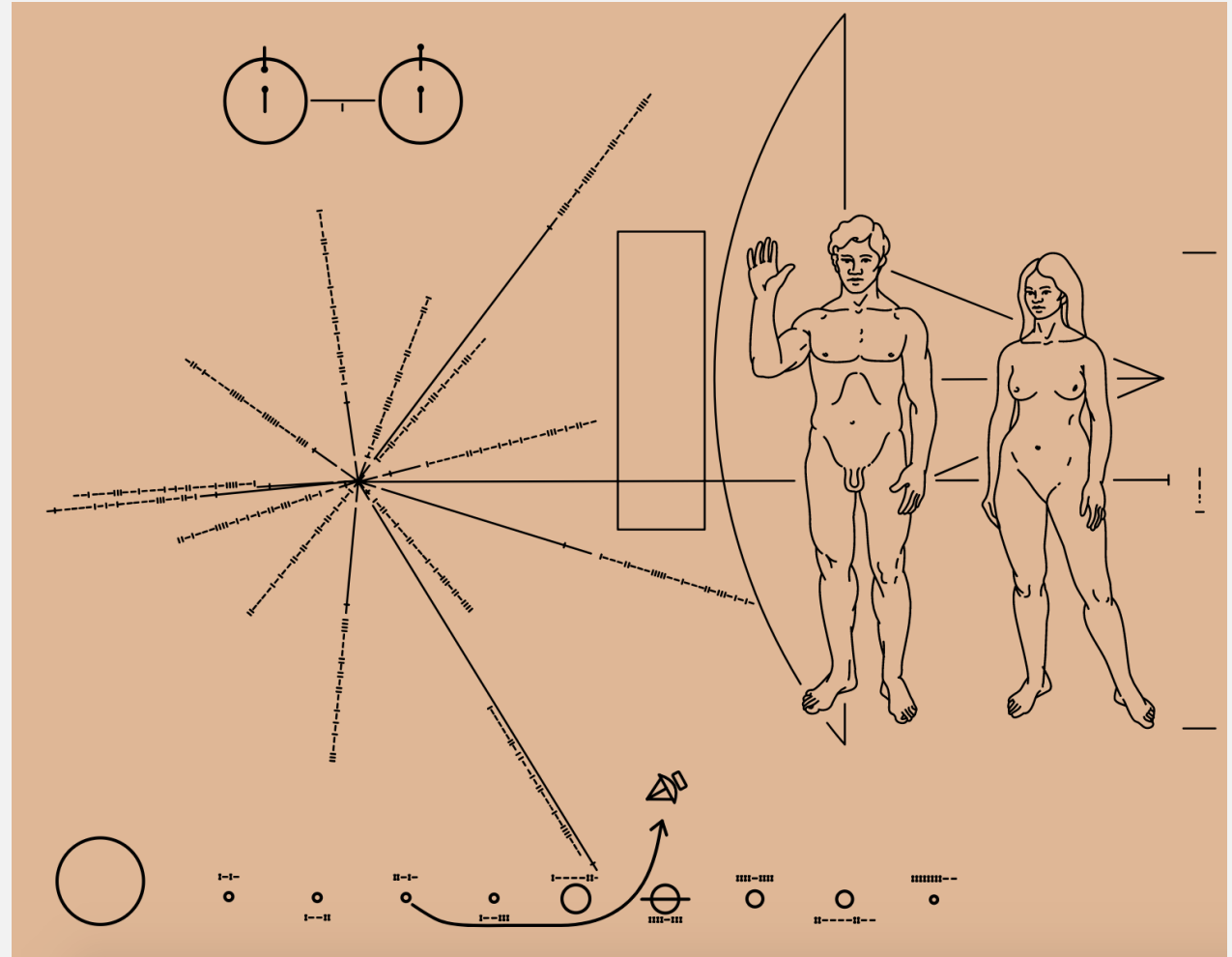


Watch this space...

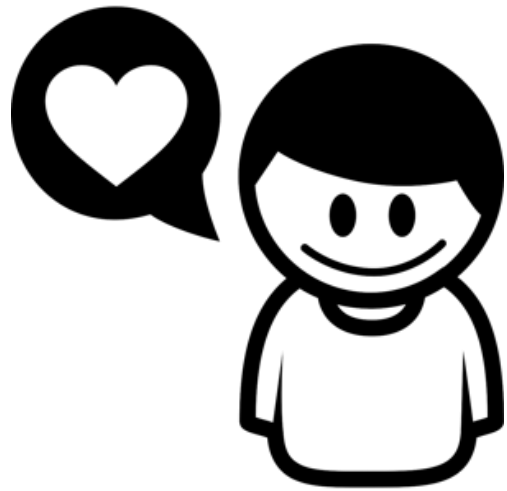




Welcome, makers
of our space!



https://en.wikipedia.org/wiki/Pioneer_plaque



Coding should be
for everyone...



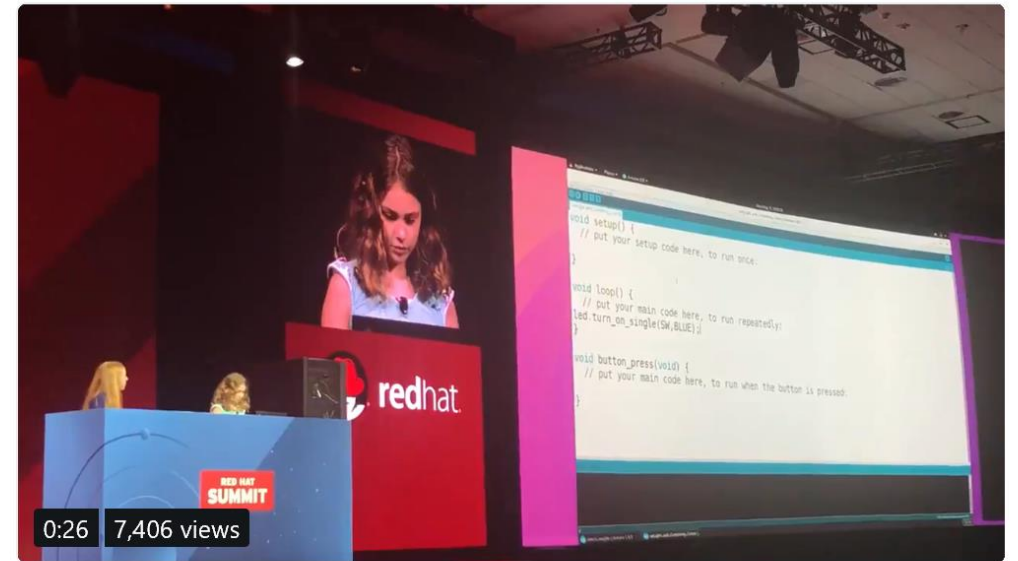
Stephanie Wonderlick

@stephstad

Follow



On her 11th birthday, this amazing girl is coding on stage in front of 7500+ people at Red Hat Summit. THIS is the future of **#tech**. **#rhsummit** **#girlsintech** **#womwnintech** **#jewelbots**

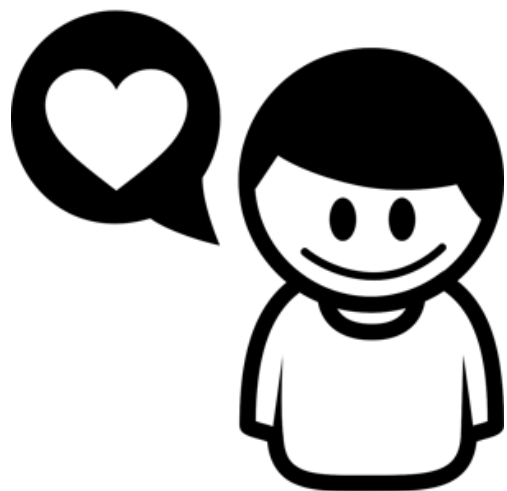


11:15 PM - 9 May 2018

161 Retweets 413 Likes



twitter.com/stephstad/status/994340133733253120



Coding can be a
second chance...



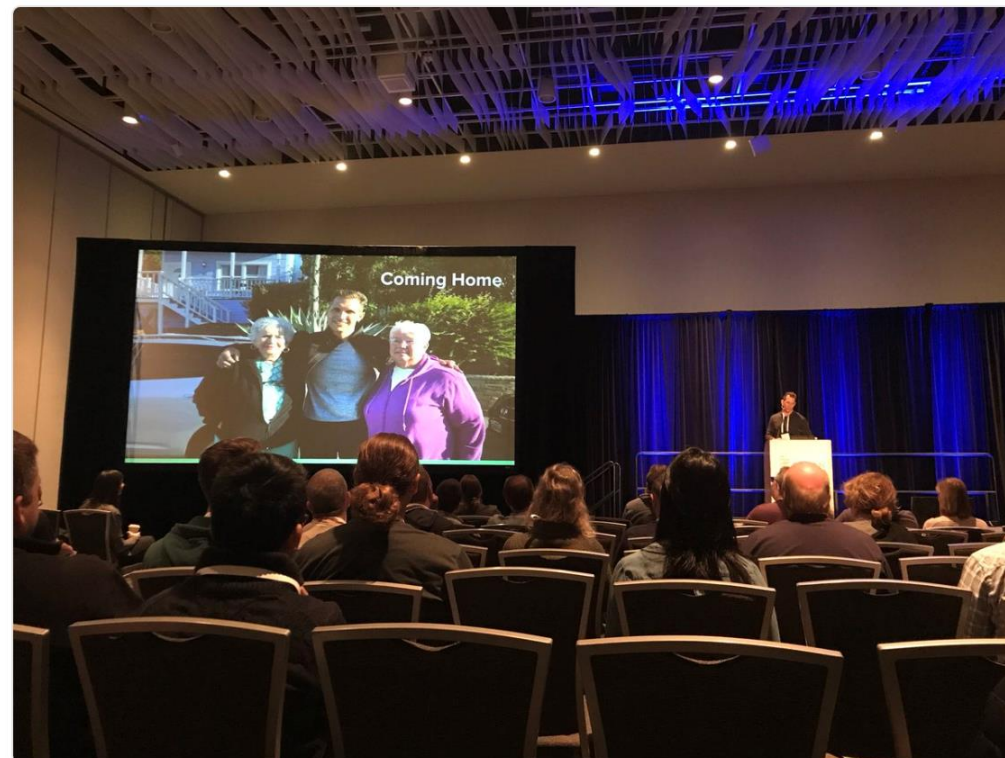
Robert Ward

@JBalloonist

Follow



It's never too late to learn how to code. Chris Schumacher spent 17 years in prison and is now a software engineer after learning to code while serving time [#PyCon2018](#)



4:09 PM - 11 May 2018

twitter.com/JBalloonist/status/994957763154411520



What's wrong with
coding?



nothing
null
undefined

///

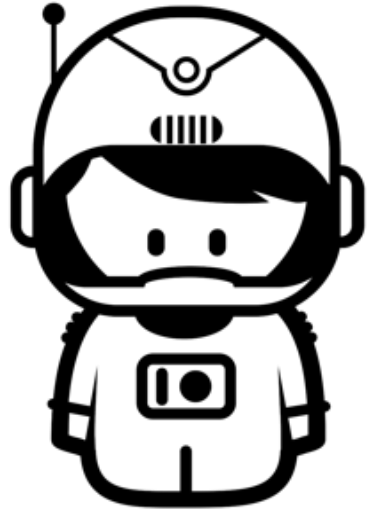
[]
false
NaN



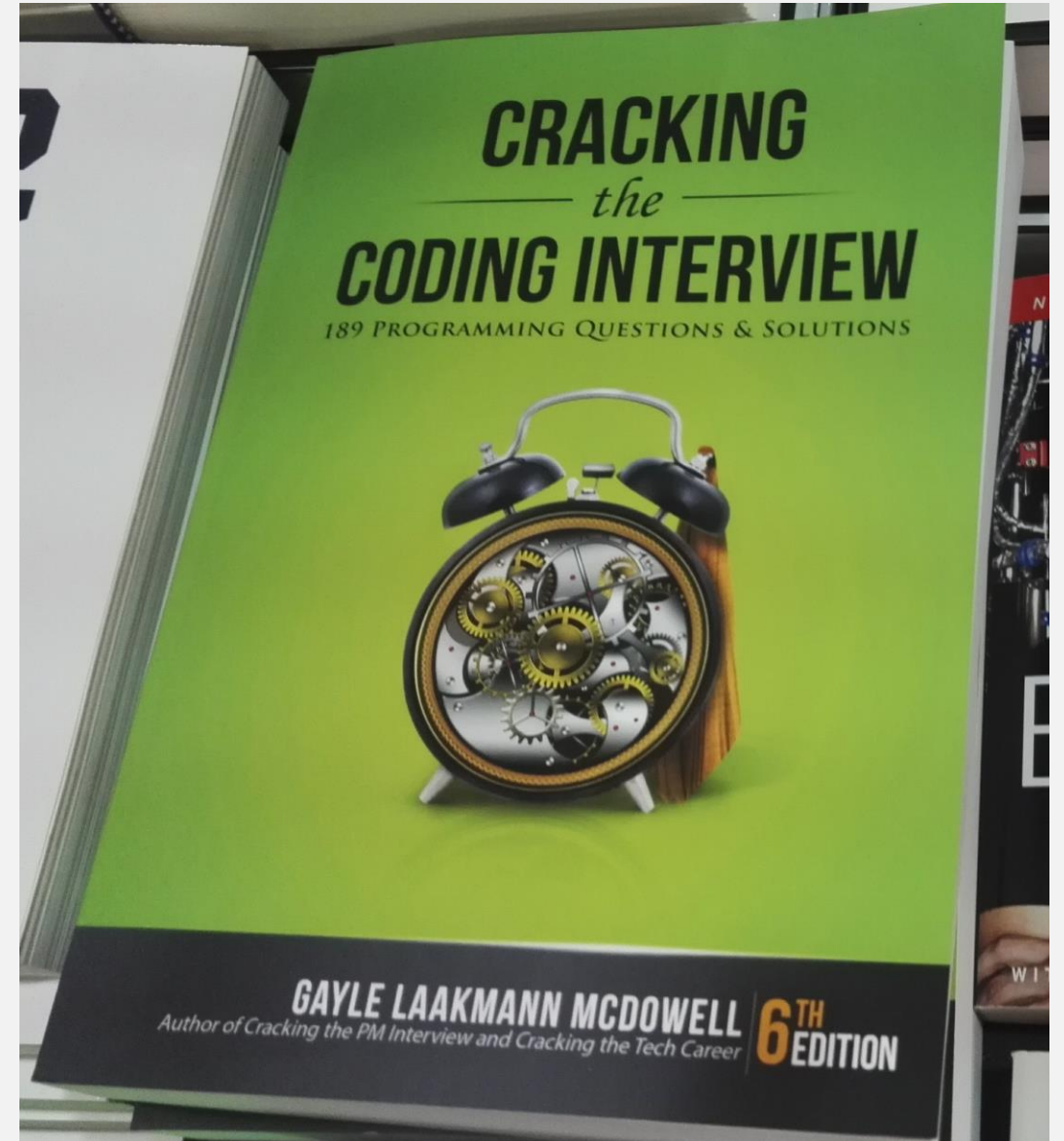
I am worried about
the **perception** of
coding...

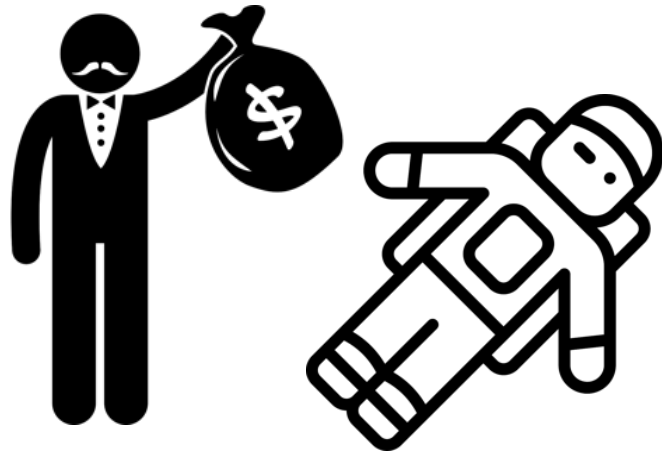


- **Huge misconception:** everybody who learns to code will have a great, fulfilling job.
- **Danger:** people who just learned to code will immediately build products without understanding what they do.
- **Hype isn't good:** we are professionals, not miracle workers.




The "coder" of
today is the
astronaut of the 50s





An astronaut with
job security and no
physical demands...



Matt Rickard

@mattrickard

Follow



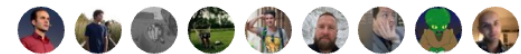
our github bot just got a job offer

to minikube-bot ▾

We found you on GitHub and we believe you might be interested in our Go Developer opening.

10:01 PM - 13 Dec 2017

939 Retweets 2,380 Likes





If we look closely,
things look bad..



The software did exactly what it was told to do. The reason it failed is that it was told to do the wrong thing.

knife. The man fled.

<https://www.theatlantic.com/technology/archive/2017/09/saving-the-world-from-code/540393/>



We start to realise
the mess we created



Software disenchantment



Translations: [French](#) [Italian](#) [Portuguese](#) [Russian](#) [Spanish](#)

I've been programming for 15 years now. Recently our industry's lack of care for efficiency, simplicity, and excellence started really getting to me, to the point of me getting depressed by my own career and the IT in general.

Nikita Prokopov

<http://tonsky.me/blog/disenchantment/>



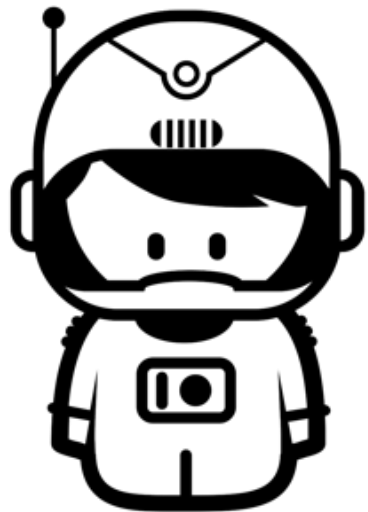
Code as craft (or
even art) is 3.62%
of what we do...





We need to
improve the quality
of day-to-day code



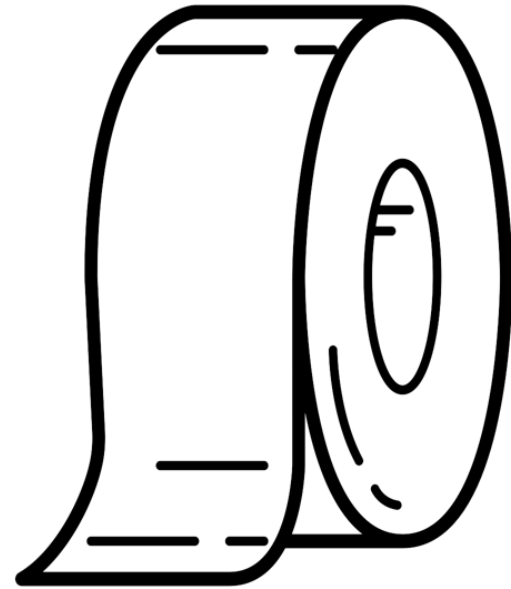
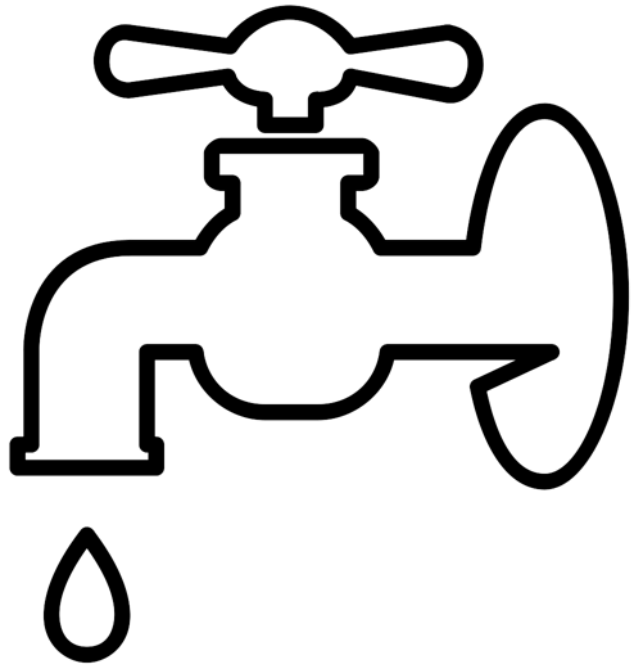


It started simple and
future-proof...



warnerbros.com/archive/spacejam/movie/jam.htm

Openness and resilience is both gift and curse...



Web tech was terrible and we hacked our way through



ಠ_ಠ(ಁಁ)ಠ_ಠ

The need to be
forgiving...

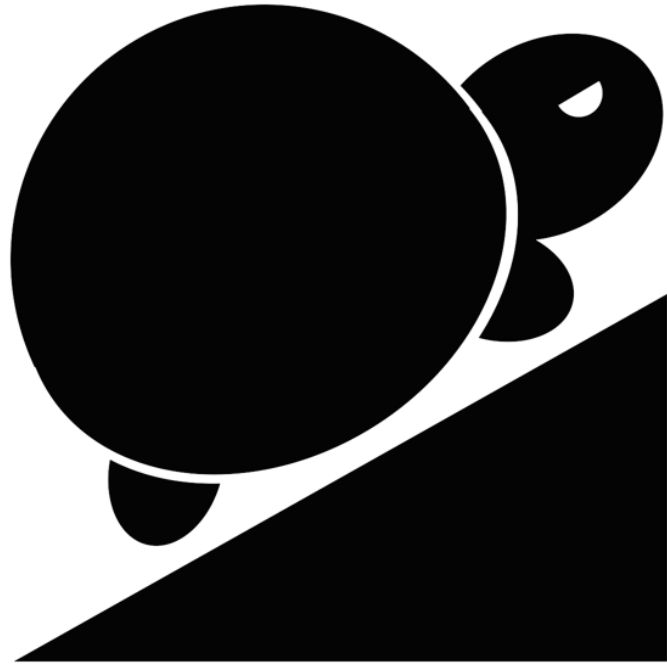


- Anyone is allowed to write code for the web – or use whatever they want to create that code.
- Users shouldn't suffer from developer errors
- Legacy products will never get fixed
- You can't break the web

Years of technical debt...



The web of today is slow and annoying...



Which is odd, seeing that
most users are on mobiles



With unreliable connections.





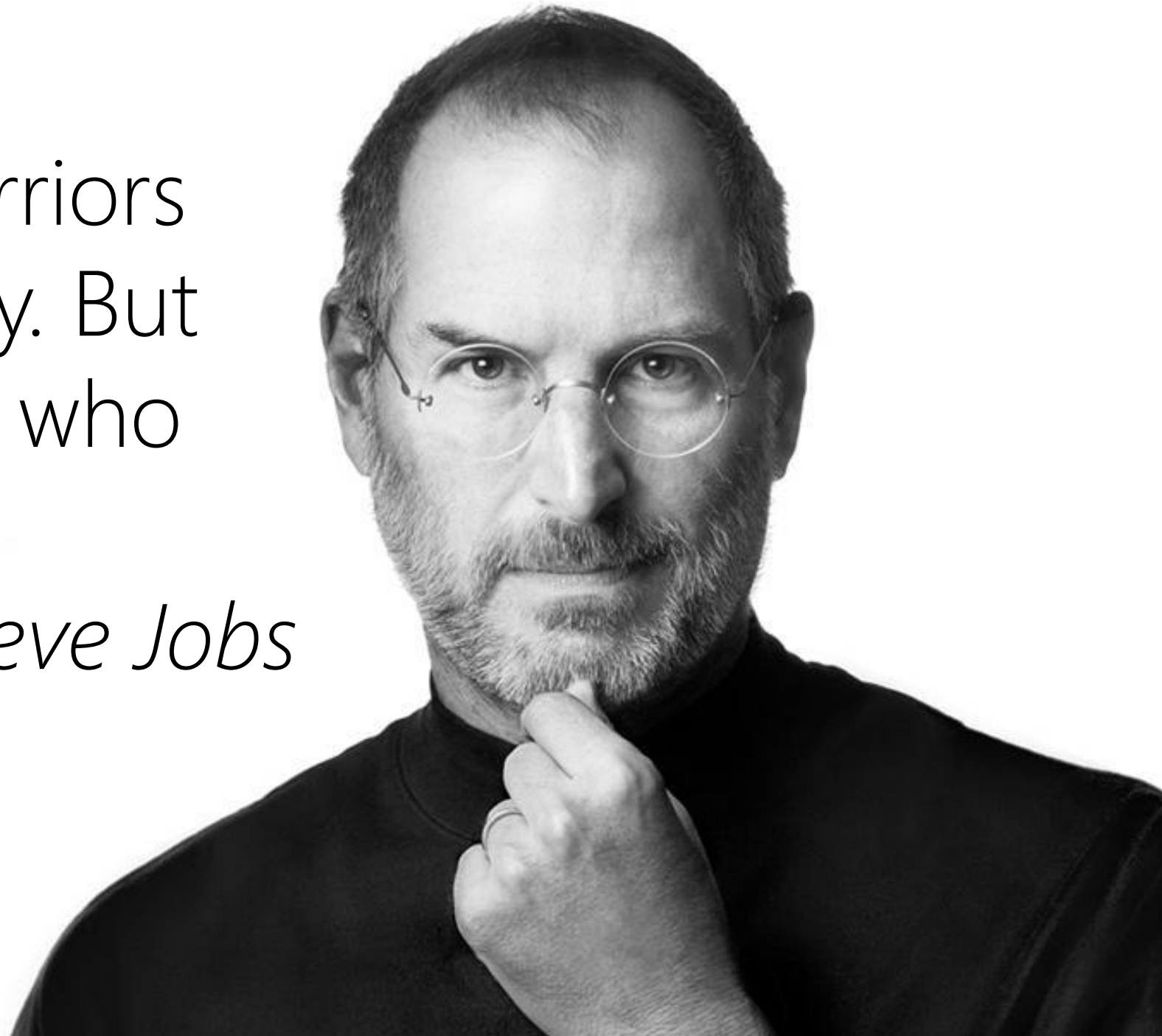
It is up to us to clean
this mess up.



- Developing software products is a **service** to our users
- We all profited from years of “let’s see what works”
- It is time to be **more responsible** as creators of software products
- And that’s not being a **Rockstar, Ninja** or **Code Warrior**.

It may be the warriors
who get the glory. But
it's the engineers who
build societies.

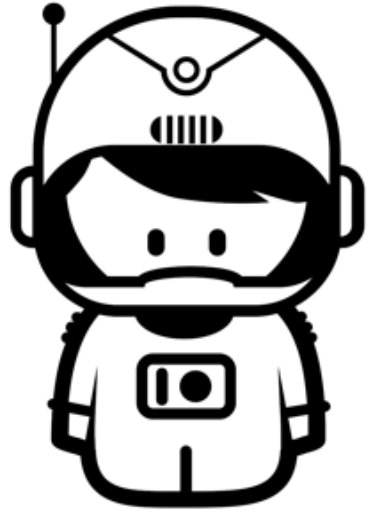
– *Steve Jobs*



It may be the warriors
who get the glory. But
it's the engineers who
build societies.

– *B'elanna Torres*

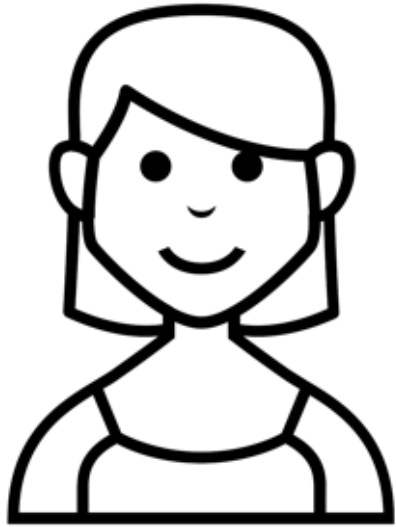




Astronauts are
users empowered
by engineers



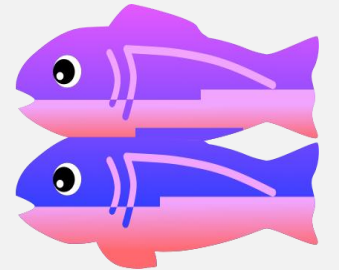
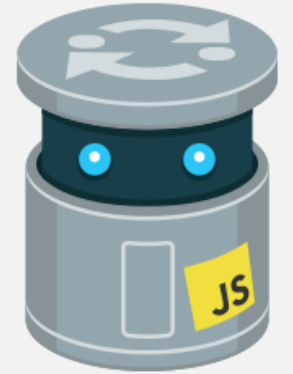
Director of Apollo Flight Computer
Programming Margaret Hamilton
<https://github.com/chrislgarry/Apollo-11>



It was never easier
to get started...



moz://a





Soon you realise
that you need to
know a lot...



- Performance
 - Accessibility
 - Interoperability
 - Security
 - Maintainability
-
- Browser Quirks
 - Developer toolchains
 - Browser developer tools
 - Abstractions
 - CLI
 - Editors



Where to learn all that?



- Standards documentation?
- Browser vendor docs?
- Conferences?
- Books?
- Workshops?
- Developer tools?
- News channels?

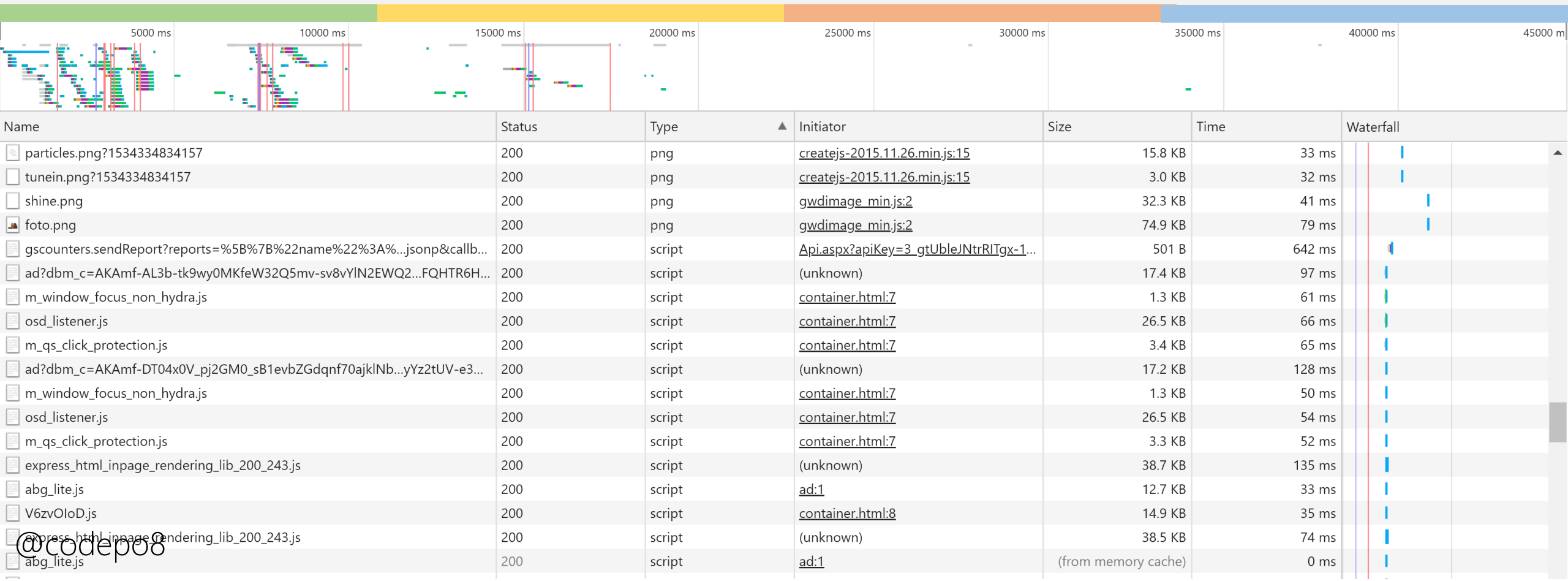


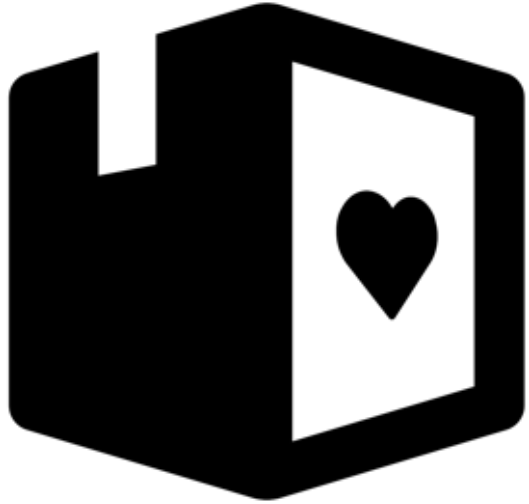
Learning obstacles



- Resources with upvoting options favour the how over the why
- We're not paid to learn, we're paid to deliver
- Peer pressure makes developers who don't know things afraid to admit it
- There's always a "good enough" way to create a lot in a short amount of time.

Tooling and insights into how our code works is great





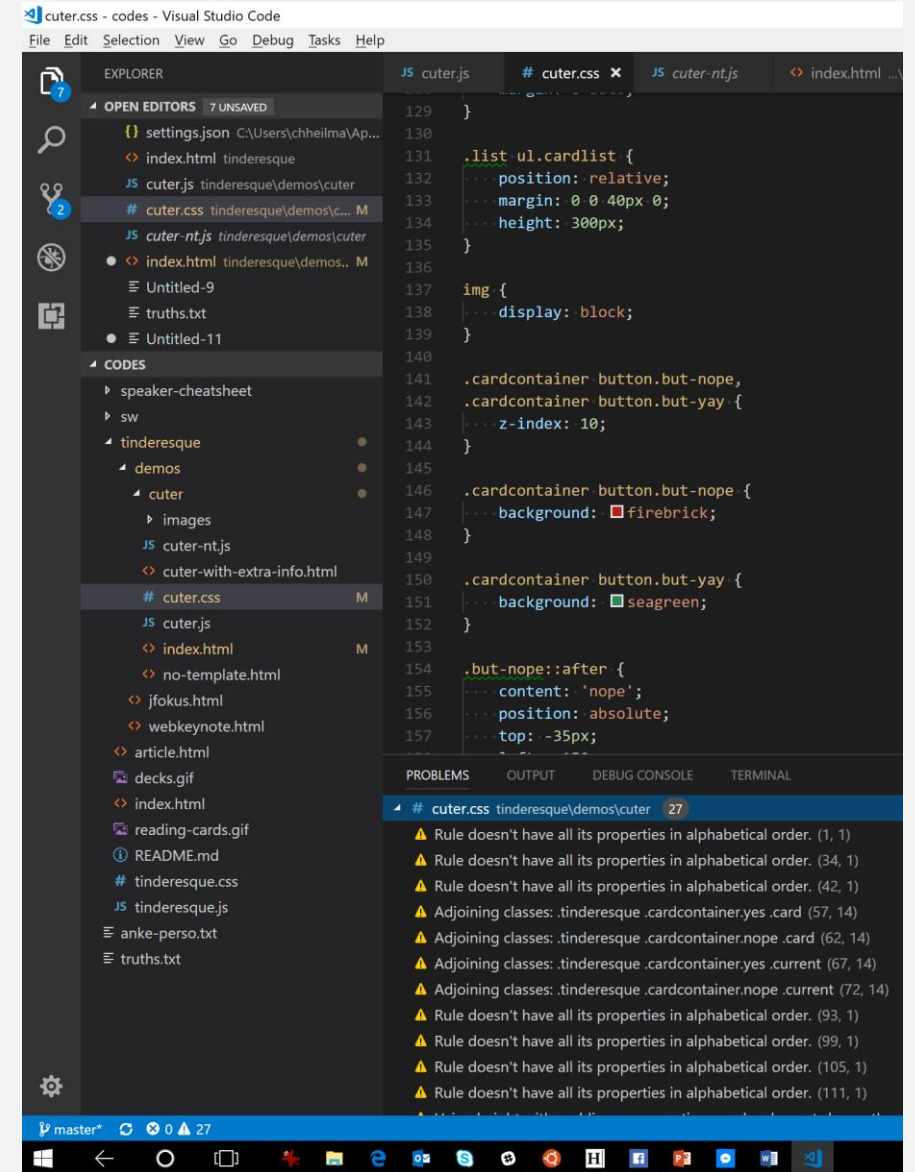
Customisable best practices in context



- Having information isn't enough when people don't go there
- We need to prevent mistakes before they happen
- We need to allow people to customize these experiences. An intranet site needs different settings than a marketing site
- So, let's target editors and build processes



Linting and inline insights in editors




code.visualstudio.com

webhint.io



A JS Foundation Project



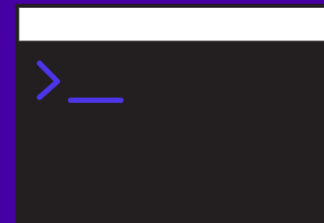
 webhint [SCANNER](#) [DOCUMENTATION](#) [ABOUT](#)

Search Documentation

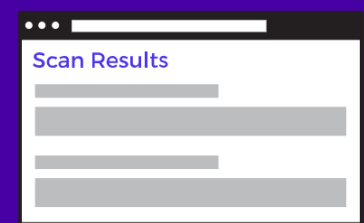


Use webhint to improve your website

webhint is a linting tool that will help you with your site's accessibility, speed, security and more, by checking your code for best practices and common errors. Use the online scanner or the CLI to start checking your site for errors.



GET STARTED WITH THE CLI



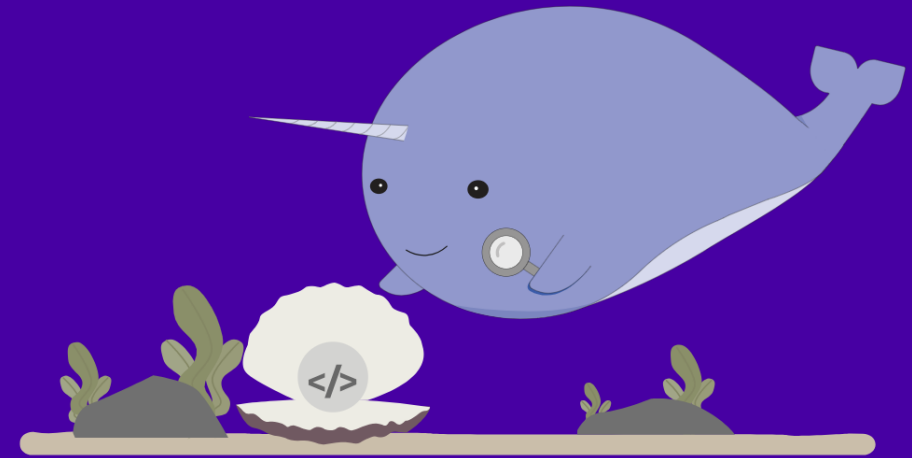
TRY THE ONLINE SCANNER

Enter a URL...

Use the online scanner

Not sure what you want to configure the CLI to check? Use the online scanner for a pre-configured set of rule checks to cover the basics.

RUN SCAN



Get a report on issues...

SCANNED URL: <http://example.com>

Finished

WARNINGS

1

ERRORS

9

SCAN TIME

00:14


VERSION

3.0.1

SCAN CONFIGURATION

[View JSON file](#)

PERMALINK

<https://webhint.io/scanner/bc3ec732-afde-4cde-b762-ee8ba6102288> 

ACCESSIBILITY



1 ERROR
0 WARNINGS

INTEROPERABILITY



4 ERRORS
0 WARNINGS

PERFORMANCE



2 ERRORS
0 WARNINGS

PWA



0 ERRORS
1 WARNING

SECURITY



2 ERRORS
0 WARNINGS

Get explanations why these are issues & how to fix them

Interoperability

- CLOSE ALL

content-type: 2 errors

DOCUMENTATION

CLOSE DETAILS

✗ ERROR

'content-type' header charset value should be 'utf-8'.

<http://example.com/>

✗ ERROR

'content-type' header media type value should be 'image/x-icon', not 'text/html'.

<http://example.com/favicon.ico>

meta-charset-utf-8: 2 errors

DOCUMENTATION

CLOSE DETAILS

✗ ERROR


'charset' meta element should be the first thing in '<head>'.


<http://example.com/:4:5>

```
<meta charset="utf-8">
```


webhint.io

Get in-depth explanation on the issues...

A JS Foundation Project


webhint

SCANNER DOCUMENTATION ▾ ABOUT ▾

Search Documentation 

Home \ Documentation \ User guide \ Require `Content-Type` HTTP response header with appropriate value

Require `Content-Type` HTTP Response Header With Appropriate Value

 Edit | [Signal Issue](#)

CONCEPTS >

FURTHER CONFIGURATION >

HINTS ▾

[AMP HTML Validator](#)

[Accessibility assessment with aXe](#)

[Avoid HTTP redirects in requests](#)

[Avoid exceeding CSS stylesheet limits](#)

[Babel configuration hint set](#)

[Babel configuration is valid](#)

[Check `devtool` property in webpack configuration](#)

[Check for broken links](#)

[Check if the TypeScript target is appropriate](#)

[Check if webpack configuration file](#)

`content-type` warns against not serving resources with the `Content-Type` HTTP response header with a value containing the appropriate media type and charset for the response.

Why is this important?

Even though browsers sometimes [ignore](#) the value of the `Content-Type` header and try to [sniff the content](#), it's indicated to always send the appropriate media type and charset for the response as, among other:

- resources served with the wrong media type may be blocked (see also: [X-Content-Type-Options hint](#)), or the official media type may be required
- not sending the appropriate `charset`, where appropriate, may prevent things from being rendered correctly thus creating a bad user experience (see also: [meta-charset-utf-8 hint](#))

What does the hint check?



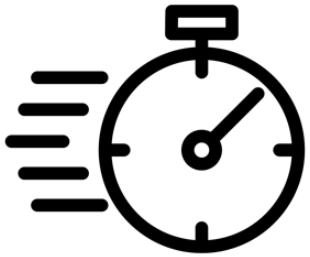
Hinting during deployment



- A web interface to get reports doesn't scale to production
- Why not install hinting at as part of your build/deployment process?
- That way you can also test the setup of your bundler/preprocessor to avoid huge release sizes

```
$ hint example.com
```

Out of the box hints...



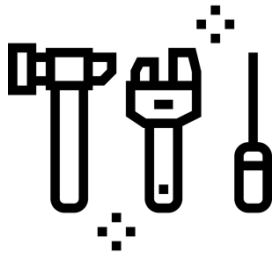
Performance



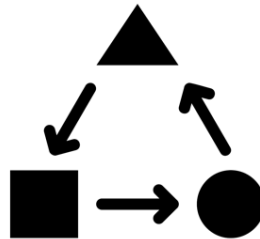
Accessibility



PWA
Readiness



Development
environment

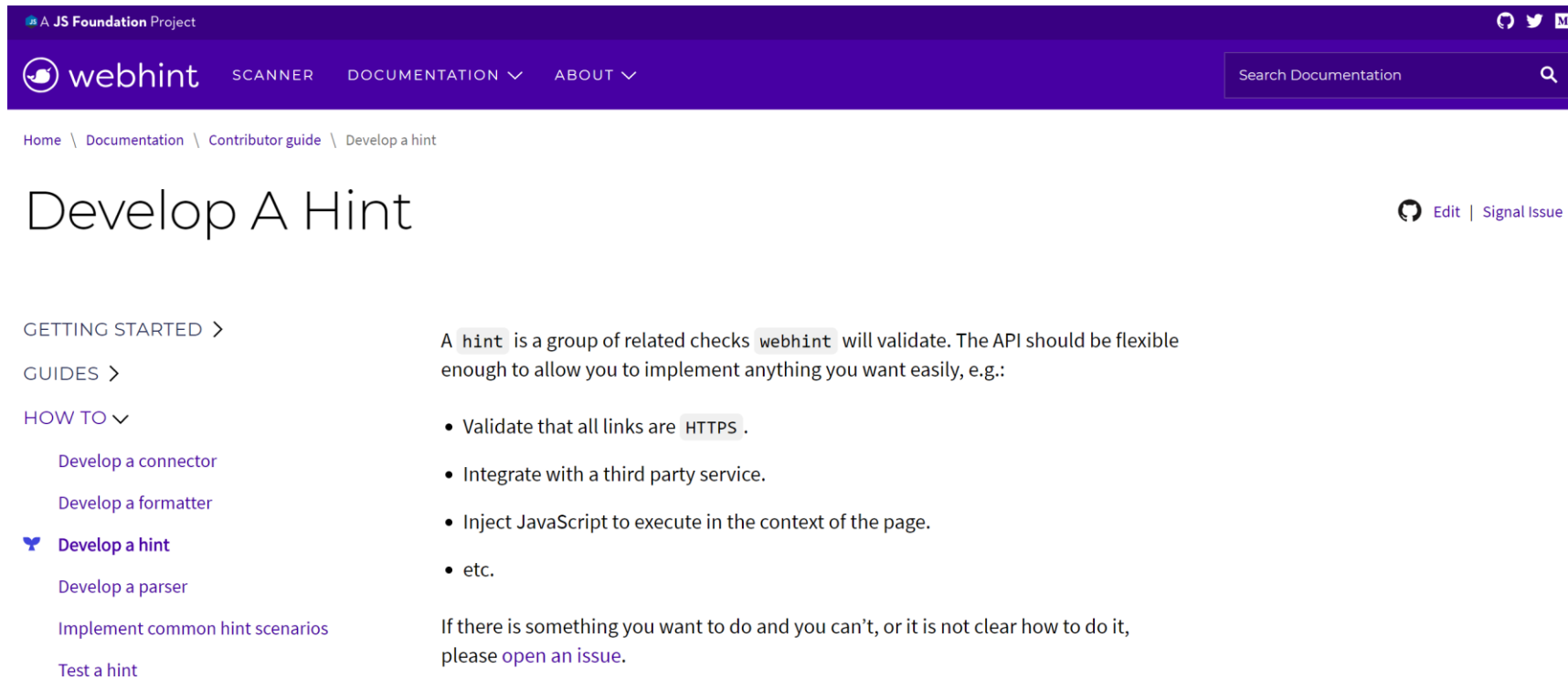


Interoperability



Security

Customising hints...



The screenshot shows the webhint documentation website. The header is purple with the webhint logo, navigation links (SCANNER, DOCUMENTATION, ABOUT), and a search bar. The breadcrumb trail is 'Home \ Documentation \ Contributor guide \ Develop a hint'. The main heading is 'Develop A Hint' with 'Edit' and 'Signal Issue' links. The left sidebar contains a list of links: 'GETTING STARTED >', 'GUIDES >', 'HOW TO >', 'Develop a connector', 'Develop a formatter', 'Develop a hint' (highlighted with a blue icon), 'Develop a parser', 'Implement common hint scenarios', and 'Test a hint'. The main content area explains that a hint is a group of related checks and lists three bullet points: 'Validate that all links are HTTPS', 'Integrate with a third party service', and 'Inject JavaScript to execute in the context of the page.' It also includes a note about opening an issue for unclear requirements.

A JS Foundation Project

webhint SCANNER DOCUMENTATION ▾ ABOUT ▾

Search Documentation 🔍

Home \ Documentation \ Contributor guide \ Develop a hint

Develop A Hint

[Edit](#) | [Signal Issue](#)

GETTING STARTED >

GUIDES >

HOW TO ▾

- Develop a connector
- Develop a formatter
- Develop a hint**
- Develop a parser
- Implement common hint scenarios
- Test a hint

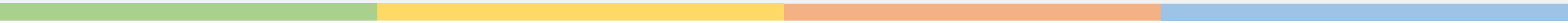
A **hint** is a group of related checks webhint will validate. The API should be flexible enough to allow you to implement anything you want easily, e.g.:

- Validate that all links are **HTTPS**.
- Integrate with a third party service.
- Inject JavaScript to execute in the context of the page.
- etc.

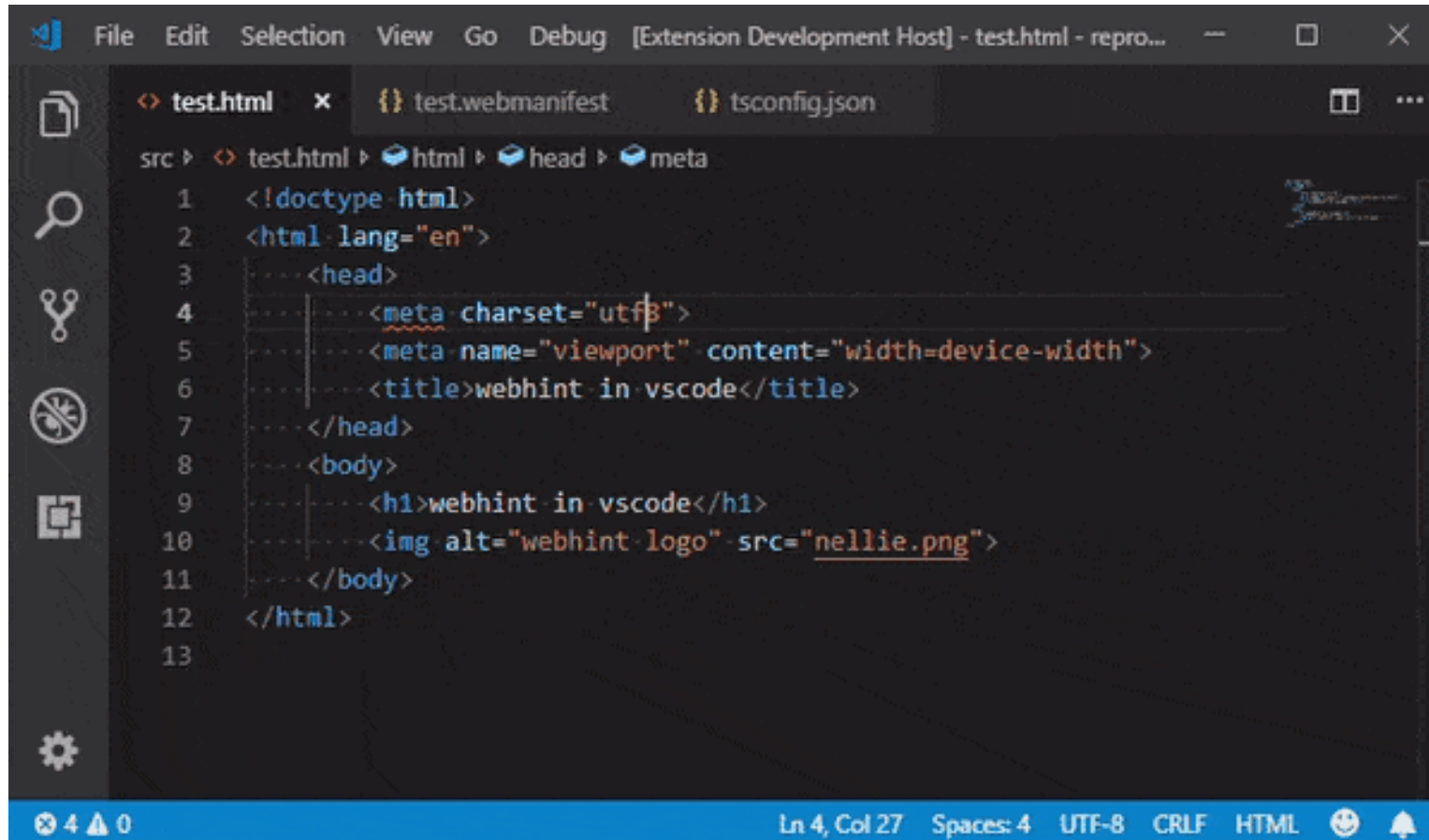
If there is something you want to do and you can't, or it is not clear how to do it, please [open an issue](#).



How about
preventing people
to make errors?



webhint @code extension

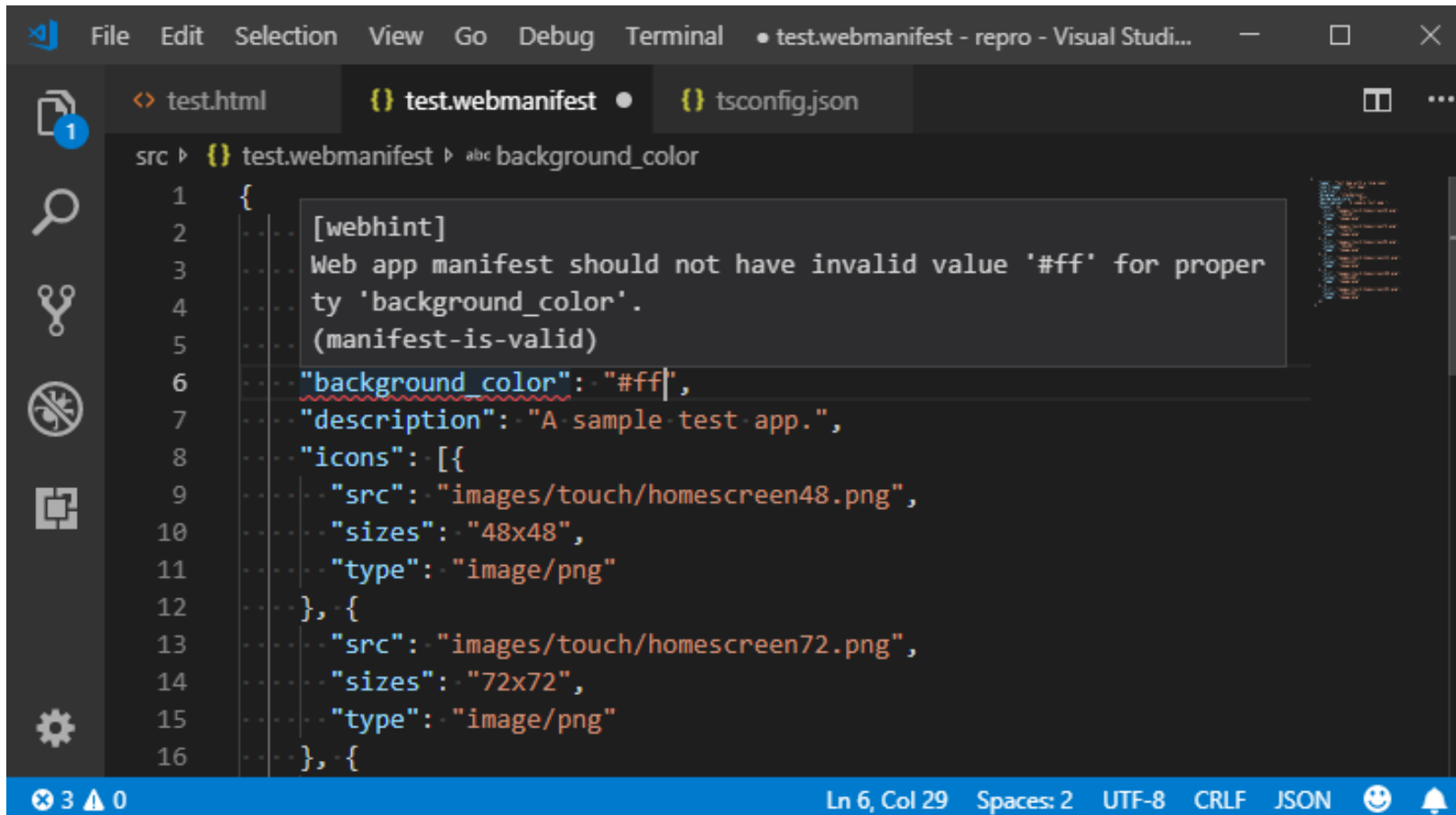


```
File Edit Selection View Go Debug [Extension Development Host] - test.html - repro...
test.html x test.webmanifest tsconfig.json
src > <> test.html > html > head > meta
1 <!doctype html>
2 <html lang="en">
3 <head>
4 <meta charset="utf8">
5 <meta name="viewport" content="width=device-width">
6 <title>webhint in vscode</title>
7 </head>
8 <body>
9 <h1>webhint in vscode</h1>
10 
11 </body>
12 </html>
13
```

Ln 4, Col 27 Spaces: 4 UTF-8 CRLF HTML

<https://medium.com/webhint/getting-hints-from-visual-studio-code-69118e48de1b>

Create valid app manifests...



The screenshot shows the Visual Studio Code editor interface. The top menu bar includes File, Edit, Selection, View, Go, Debug, and Terminal. The title bar indicates the active file is 'test.webmanifest' in a project named 'repro'. The editor has three tabs: 'test.html', 'test.webmanifest', and 'tsconfig.json'. The 'test.webmanifest' tab is active, showing a JSON file with the following content:

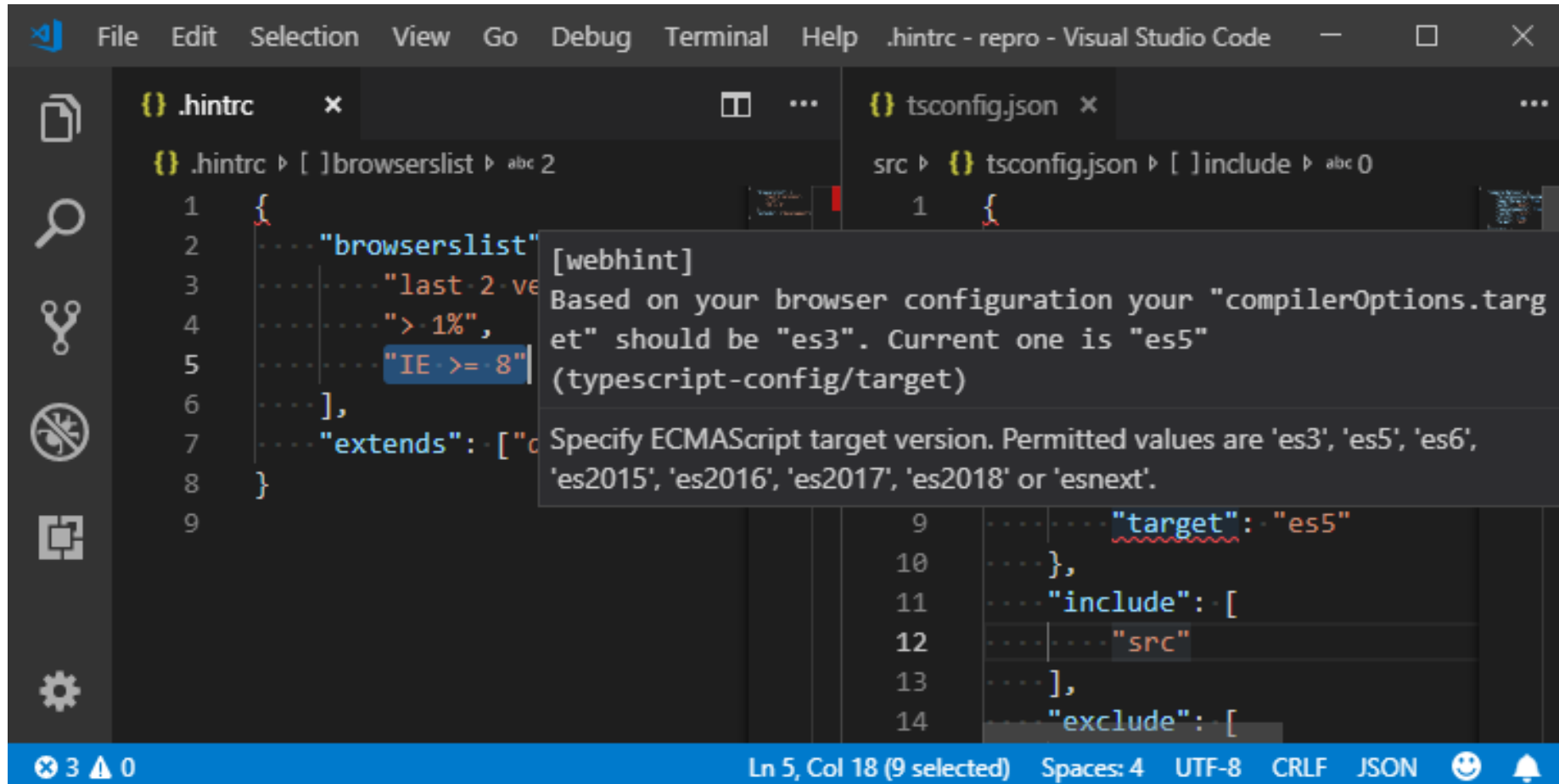
```
1 {
2   [webhint]
3   Web app manifest should not have invalid value '#ff' for proper
4   ty 'background_color'.
5   (manifest-is-valid)
6   "background_color": "#ff",
7   "description": "A sample test app.",
8   "icons": [{
9     "src": "images/touch/homescreen48.png",
10    "sizes": "48x48",
11    "type": "image/png"
12  }, {
13    "src": "images/touch/homescreen72.png",
14    "sizes": "72x72",
15    "type": "image/png"
16  }, {
```

A red squiggly line underlines the value '#ff' on line 6. A tooltip is displayed over this line, containing the following text:

```
[webhint]
Web app manifest should not have invalid value '#ff' for proper
ty 'background_color'.
(manifest-is-valid)
```

The status bar at the bottom shows 3 errors and 0 warnings. The current cursor position is Ln 6, Col 29. The encoding is UTF-8, line endings are CRLF, and the file type is JSON.

Fixing your tooling setup...



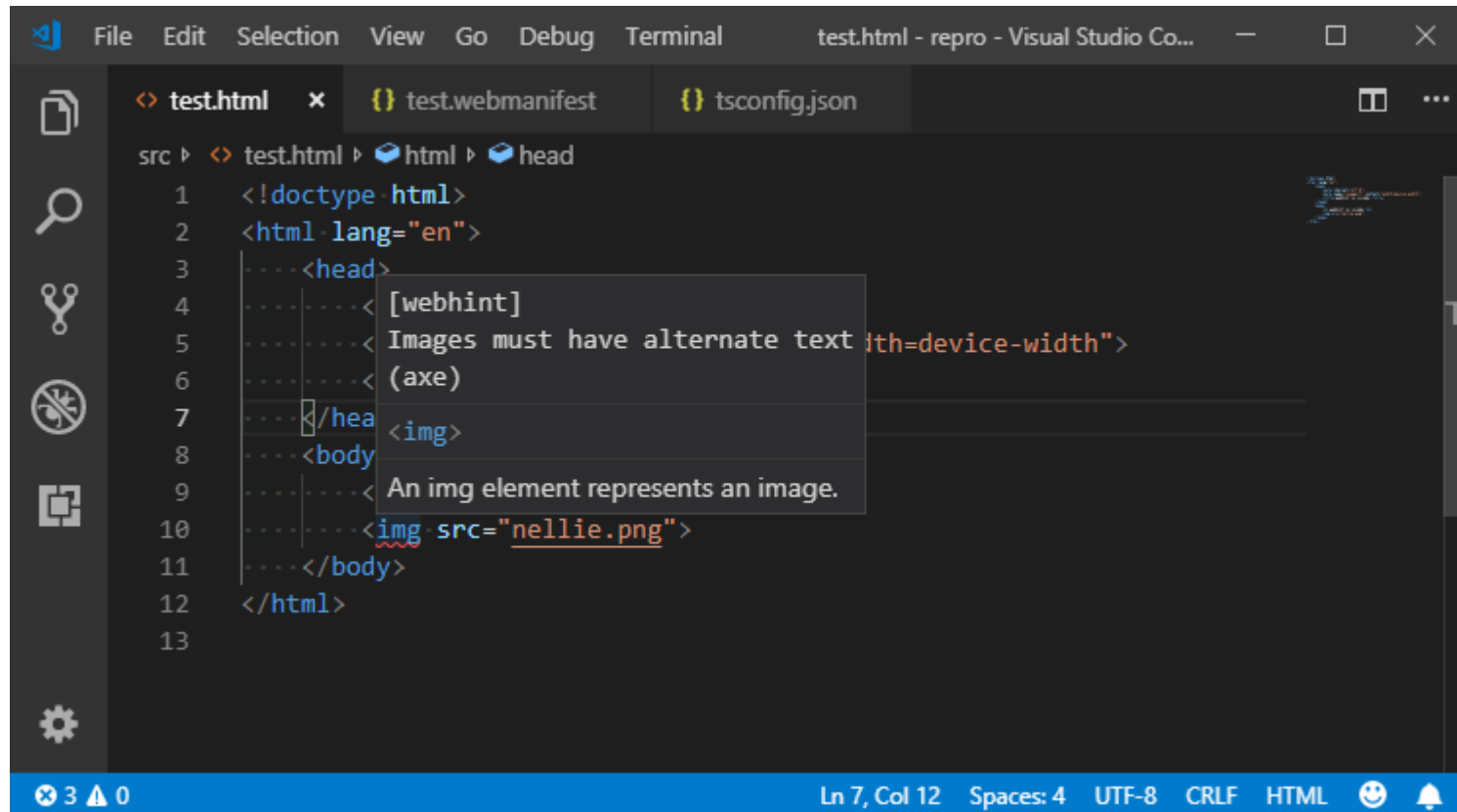
The screenshot shows the Visual Studio Code interface with two files open: `.hintrc` and `tsconfig.json`. The `.hintrc` file contains a configuration for browserslist, and the `tsconfig.json` file contains a configuration for TypeScript. A webhint error message is displayed over the `tsconfig.json` file, indicating that the `compilerOptions.target` should be `"es3"` based on the browser configuration in `.hintrc`.

```
.hintrc
1 {
2   "browserslist": [
3     "last 2 versions",
4     "> 1%",
5     "IE >= 8"
6   ],
7   "extends": [
8     "..."
9   ]
10 }

tsconfig.json
1 {
2   "compilerOptions": {
3     "target": "es5"
4   },
5   "include": [
6     "src"
7   ],
8   "exclude": [
9     "..."
10  ]
11 }
```

[webhint]
Based on your browser configuration your "compilerOptions.target" should be "es3". Current one is "es5"
(typescript-config/target)
Specify ECMAScript target version. Permitted values are 'es3', 'es5', 'es6', 'es2015', 'es2016', 'es2017', 'es2018' or 'esnext'.

Are you creating inaccessible content?



The screenshot shows the Visual Studio Code editor interface. The file explorer on the left shows a project structure with 'test.html', 'test.webmanifest', and 'tsconfig.json'. The main editor area displays the content of 'test.html'. The code is as follows:

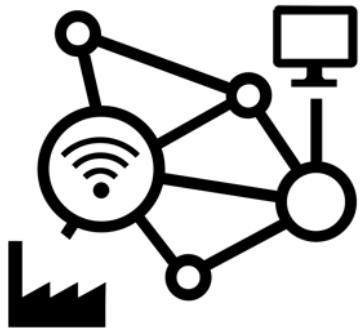
```
1 <!doctype html>
2 <html lang="en">
3   <head>
4     <[webhint]
5     <Images must have alternate text with=device-width">
6     <(axe)
7   </head>
8   <body>
9     <img>
10    
11  </body>
12 </html>
13
```

A tooltip is visible over the code, displaying the following messages:

- [webhint]
- Images must have alternate text with=device-width">
- (axe)
- An img element represents an image.

The status bar at the bottom indicates 'Ln 7, Col 12', 'Spaces: 4', 'UTF-8', 'CRLF', 'HTML', and a notification icon.

We're working right now for a much different web than we used to...



And the web is under attack...





The open web is
under attack...



Parliament adopts its position on digital copyright rules

Press Releases [PLENARY SESSION](#) 12-09-2018 - 13:27

- Tech giants must pay for work of artists and journalists which they use
- Small and micro platforms excluded from directive's scope
- Hyperlinks, "accompanied by "individual words" can be shared freely
- Journalists must get a share of any copyright-related remuneration obtained by their publishing house



Copyright rules need updating for the digital age© AP Images/European Union-EP

<http://www.europarl.europa.eu/news/en/press-room/20180906IPR12103/parliament-adopts-its-position-on-digital-copyright-rules>



The open web is under attack...



Federal Communications Commission

Browse by CATEGORY

Browse by BUREAUS & OFFICES

Search

About the FCC | Proceedings & Actions | Licensing & Databases | Reports & Research | News & Events | For Consumers

Home / EDOCS / Commission Documents /

FCC Takes Action to Restore Internet Freedom

Full Title: FCC Acts To Restore Internet Freedom
Document Type(s): News Release
Bureau(s): Wireline Competition, Media Relations, Wireless Telecommunications

Description:
Reverses Title II Framework, Increases Transparency to Protect Consumers, Spur Investment, Innovation, and Competition

Related Document(s):
Declaratory Ruling - [FCC Releases Restoring Internet Freedom Order](#)
Order - [FCC Releases Restoring Internet Freedom Order](#)
Report and Order - [FCC Releases Restoring Internet Freedom Order](#)

Document Dates
Released On: Dec 14, 2017
Adopted On: Dec 14, 2017
Issued On: Dec 14, 2017

Tags:
[Contributions - Universal Service](#)

Files

- News Release: [Docx](#) [Pdf](#) [Txt](#)
- Pai Statement: [Docx](#) [Pdf](#) [Txt](#)
- Rosenworcel Statement: [Docx](#) [Pdf](#) [Txt](#)
- Carr Statement: [Docx](#) [Pdf](#) [Txt](#)
- O'Reilly Statement: [Docx](#) [Pdf](#) [Txt](#)
- Clyburn Statement: [Docx](#) [Pdf](#) [Txt](#)

<https://www.fcc.gov/document/fcc-takes-action-restore-internet-freedom>



The open web is
under attack...



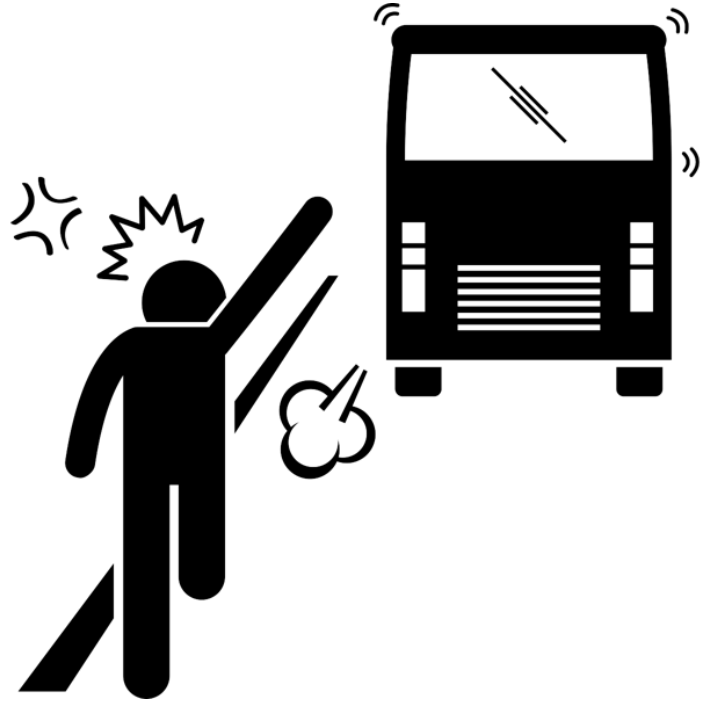
The Global Internet Is Being Attacked by Sharks, Google Confirms

By Will Oremus



Sharks' attraction to undersea fiber-optic cables has been well-documented over the years.

Screenshot / YouTube



Are we too late?



The Switch

Former Google chief predicts the Internet will split by 2028: A Chinese Web and an American one



<https://www.washingtonpost.com/technology/2018/09/21/former-google-chief-predicts-internet-will-split-by-chinese-web-an-american-one/>



Our accumulated insights are a great starting point...



- We accumulated a lot of great knowledge on **how to do web the right way**
- Instead of re-iterating this knowledge and hope people get it, let's use this **as a base to start from**
- Our **technology stack** has evolved from a text editor and a terrible browser to amazing, open tools.
- Let's introduce new people of the web to them early on - **as part of the development setup.**



It's cool to be an
engineer – NASA has
them!



- Nothing should stop you from building wild and cool things
- Thinking about code as a career, maybe it is time to concentrate on making what we do more reliable
- If we stop writing bloated and short-lived solutions that can be maintained we free ourselves to explore the universe...

Thanks!

Chris Heilmann
@codepo8

