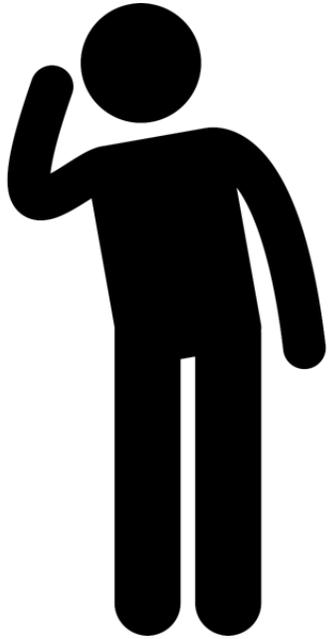


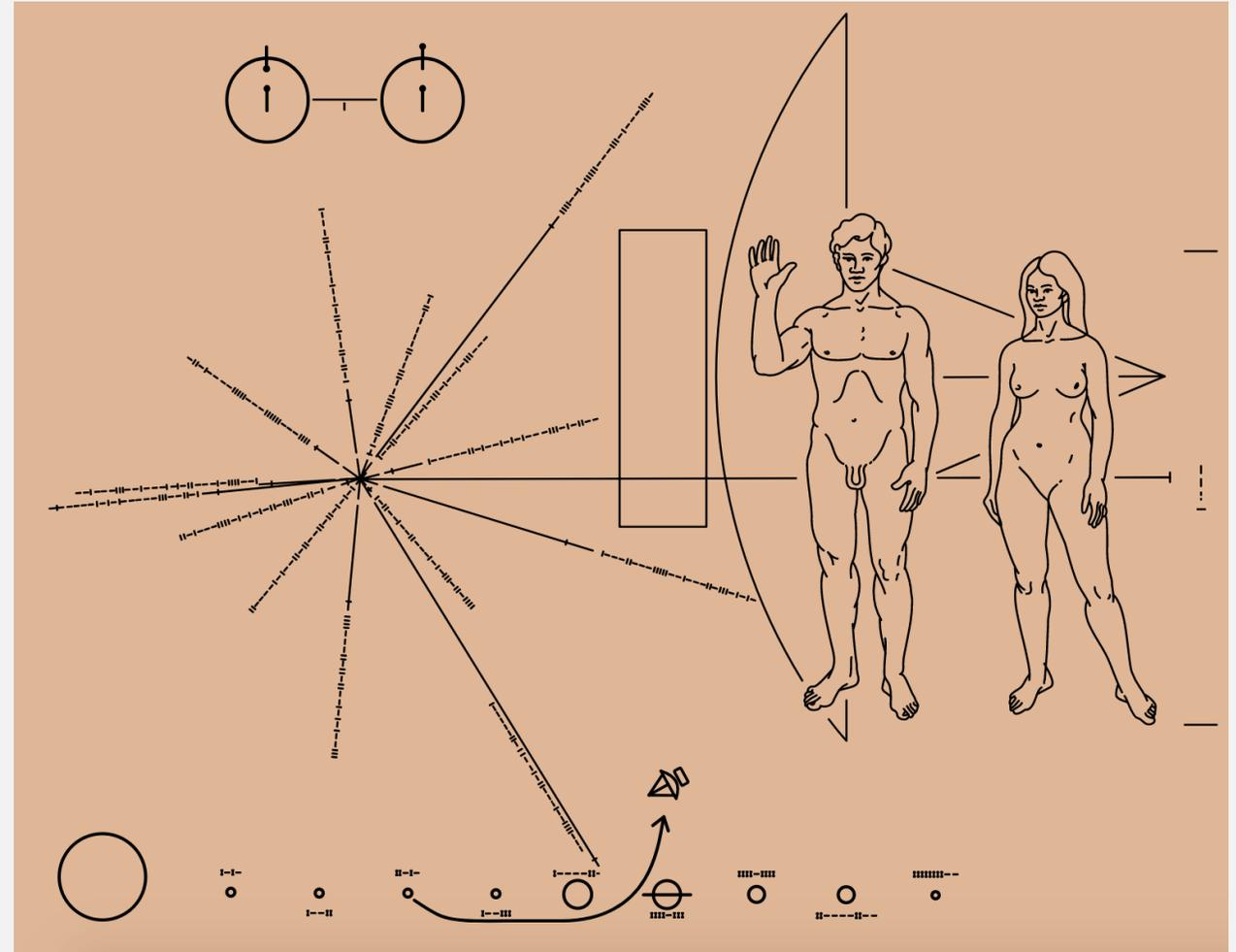
Watch this space...



Chris Heilmann (@codepo8) October 2018



It is nice to be recognised...



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



It is tougher to live up to an ideal...



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

CRACKING the CODING INTERVIEW
BY PROGRAMMING QUESTIONS & SOLUTIONS
GAYLE LAAKMANN MCDOWELL 6TH EDITION

#WeAreDevs
Sacrificing the Golden Calf of coding

Sacrificing the golden calf of coding

<https://www.youtube.com/watch?v=iPfMatOgRBg>



We're past the
software honeymoon



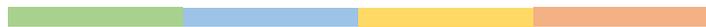
The software did exactly what it was told to do. The reason it failed is that it was told to do the wrong thing.

knife. The man fled.

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



Even we're becoming
disenchanted...



@codepo8

Software disenchantment

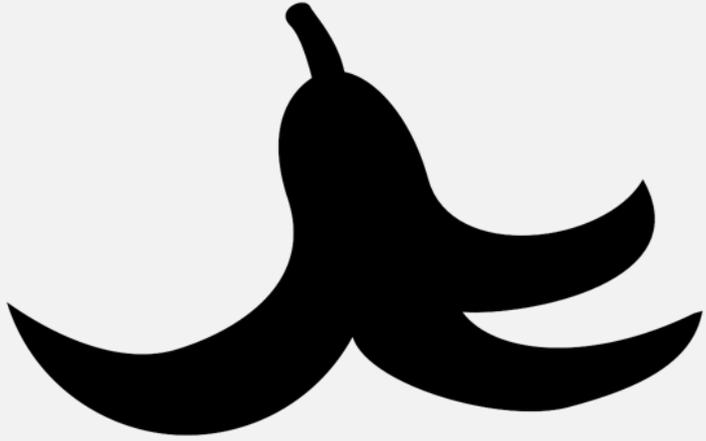
Niktonsky
and
the SOFTWARE
Disenchantment

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

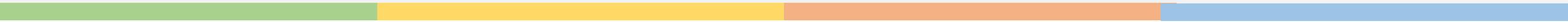
I've been programming for 15 years now. Recently our industry's lack of care for efficiency, simplicity, and excellence started really getting to me, to the point of me getting depressed by my own career and the IT in general.

Nikita Prokopov

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

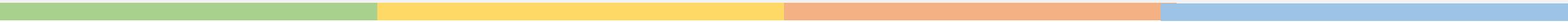


What happened?
Where did we go wrong?



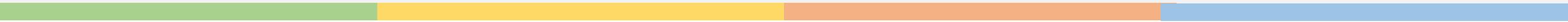


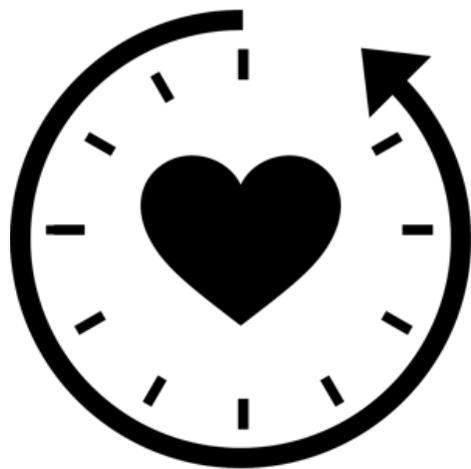
We lost interest in
owning our
environment





Specifically, the
open, world-wide
web...



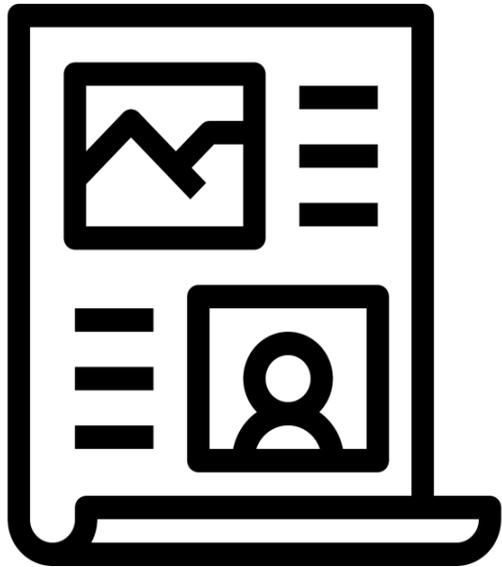


Let's go back a few years...

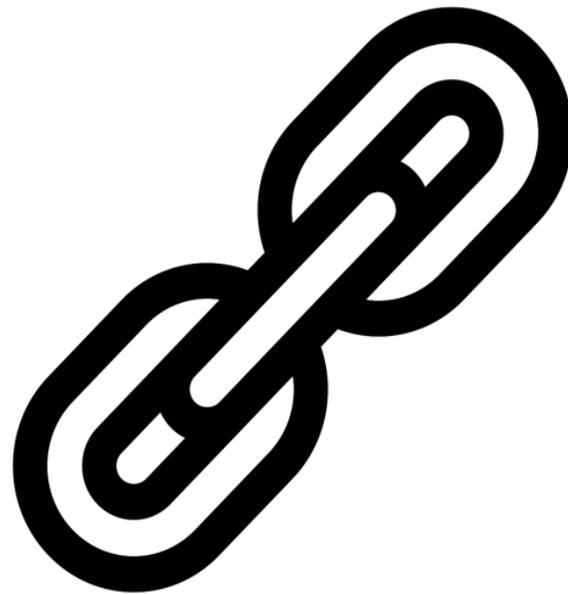


warnerbros.com/archive/spacejam/movie/jam.htm

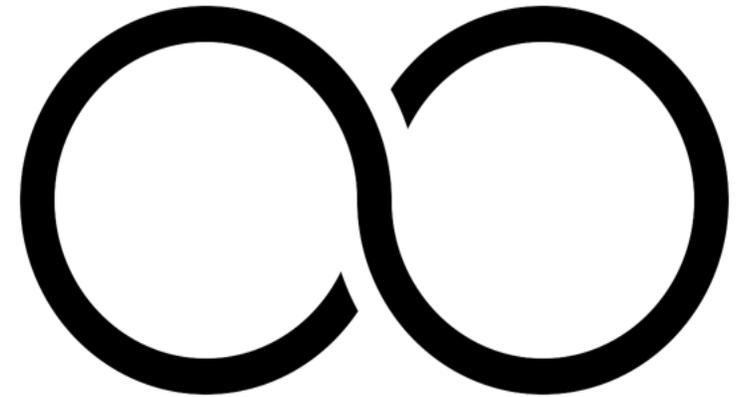
Merits of the open web...



Content is king

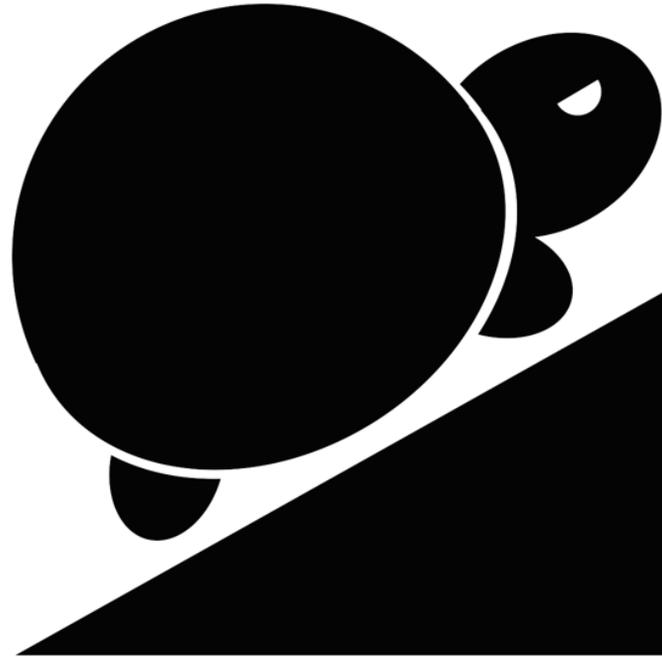


Decentralisation and
interchange



Longevity and
maintainability

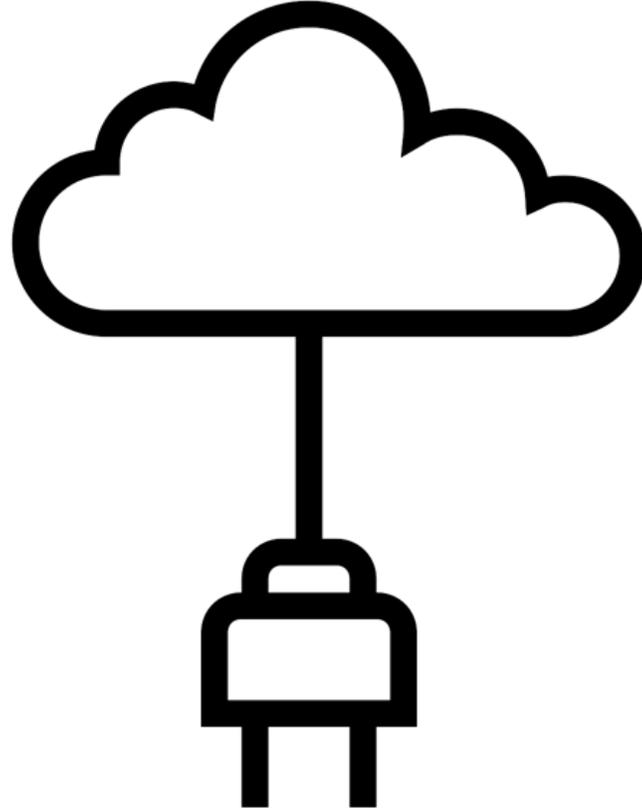
The web has become slow
and annoying...



Whereas access to the web
is on mobile devices...



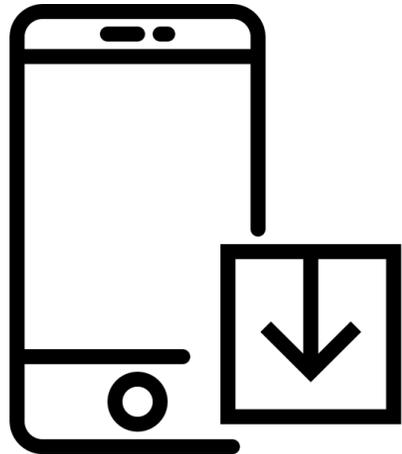
And unreliable connections.



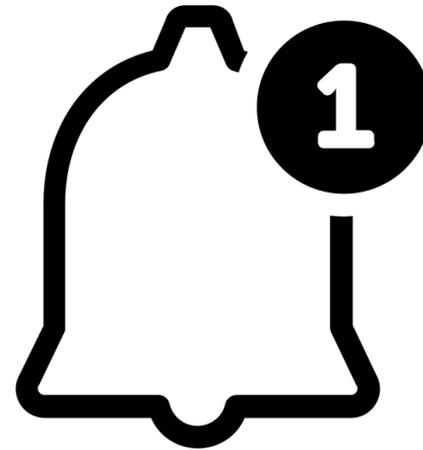
Instead of reaching web content, we get pestered...



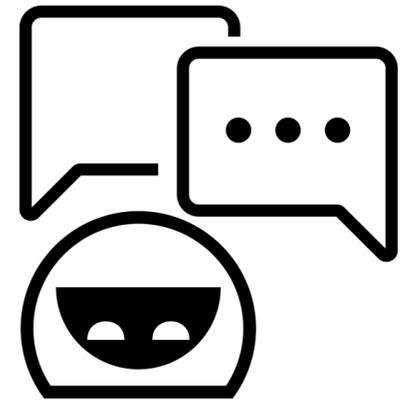
Get our
newsletter!



Download the app!

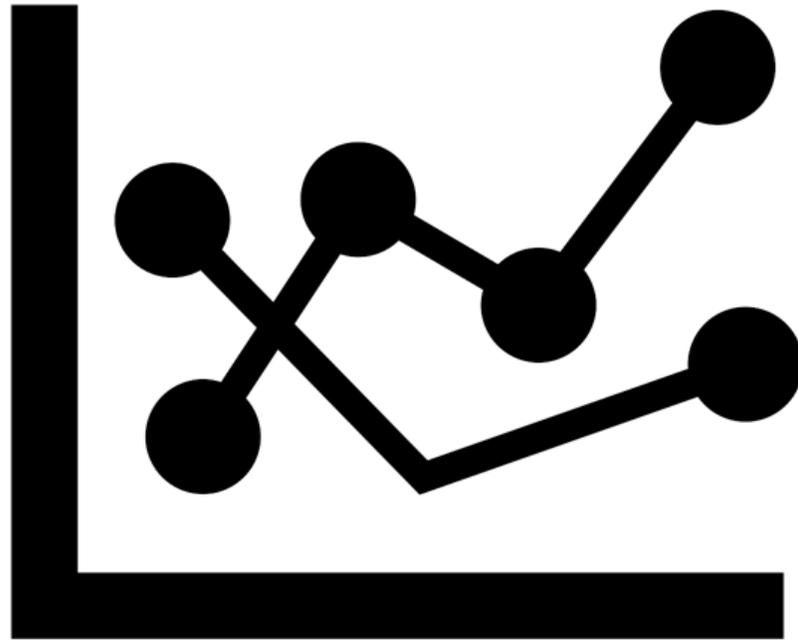


Allow notifications!



Add our chatbot!

Tracking our users is more important than content...

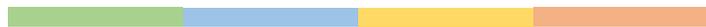


Even the plumbing of the web is under attack...





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

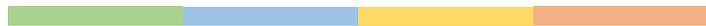


Copyright rules need updating for the digital age © AP Images/European Union-EP

<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'Rielly 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...



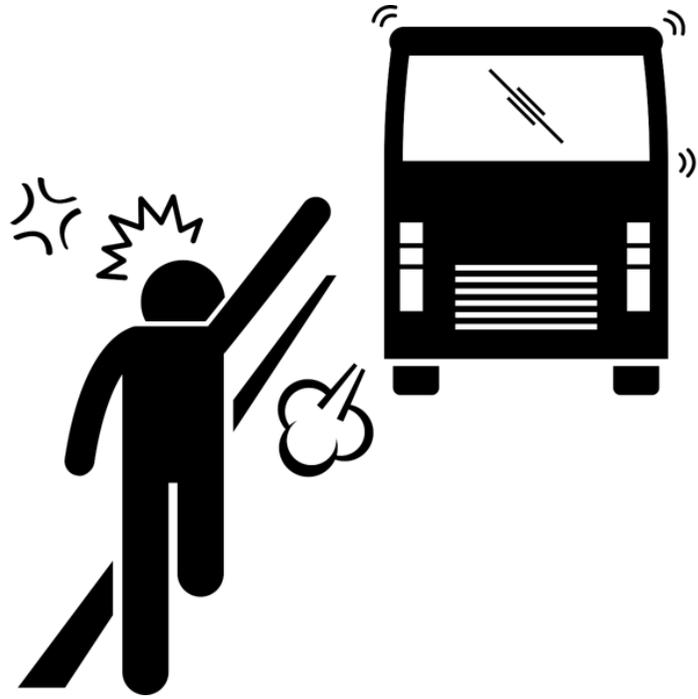
The Global Internet Is Being Attacked by Sharks, Google Confirms

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



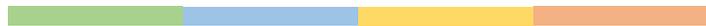
<https://www.washingtonpost.com/technology/2018/09/21/former-google-chief-predicts-internet-will-split-by-chinese-web-an-american-one/>

There are some interesting attempts to re-boot the web





dat and beaker browser



A distributed data community

Dat is a nonprofit-backed data sharing protocol for applications of the future.

[Install Now](#)

Dat Base
DAT FOR RESEARCHERS
Open Data: archive, catalogue, and share data pipelines.

Dat Protocol
BUILD CUSTOM DAT APPS
Live Syncing: easy file transfers, distributed databases, p2p streams.

Decentralized Web
PEER-TO-PEER WEBSITES
Beaker Browser: browse, create, and host websites over Dat.

Beaker [About](#) [Documentation](#) [Install Beaker](#)

JULY 31 Decentralized Web Summit, San Francisco

A browser for the next-generation Web

Beaker is an experimental browser for exploring and building the peer-to-peer Web.

[Install Beaker](#) [Take a tour](#)

File Name	Size	Last Modified
./	60.47 kB	
fonts		Apr 23, 2016, 2:55pm
images		Apr 23, 2016, 2:55pm
.datignore	10 B	May 8, 2016, 12:58pm
dat-links.js	138 B	Apr 23, 2016, 2:52pm
dat.json	111 B	Jul 6, 2016, 4:10pm
index.html	4.47 kB	Apr 23, 2016, 3:04pm

<https://beakerbrowser.com/>
<https://datproject.org/>



Solid / Inrupt



Chris Heilmann

@codepo8



Tim Berners-Lee unveils Solid, an open source project to decentralize the web and give users control of their data, and Inrupt, a startup to guide the project (Tim Berners-Lee)



One Small Step for the Web... – Tim Berners-Lee – Medium

I've always believed the web is for everyone. That's why I and others fight fiercely to protect it.

<https://www.fastcompany.com/90243936/exclusive-tim-berners-lee-tells-us-his-radical-new-plan-to-upend-the-world-wide-web>

https://medium.com/@timberners_lee/one-small-step-for-the-web-87f92217d085

@codepo8

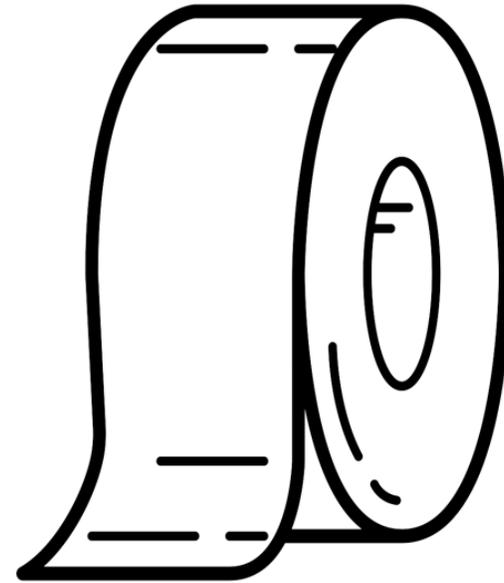
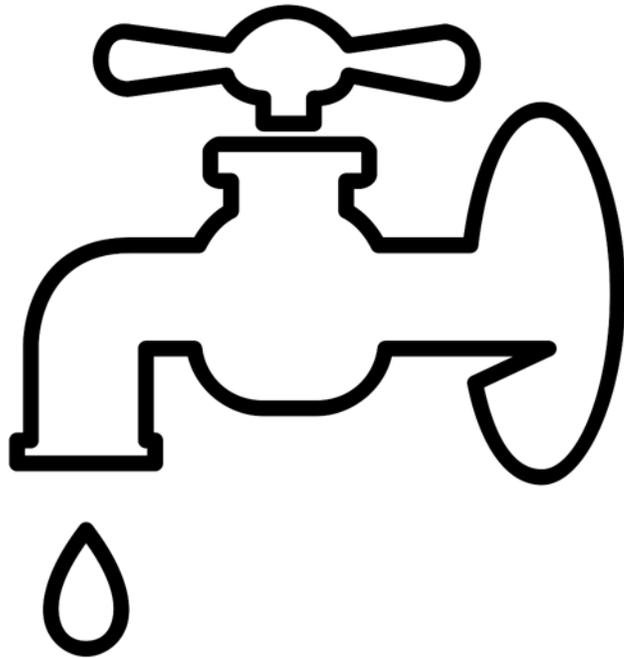
But is it really that bad
already? Can't we fix this?



I think we should.

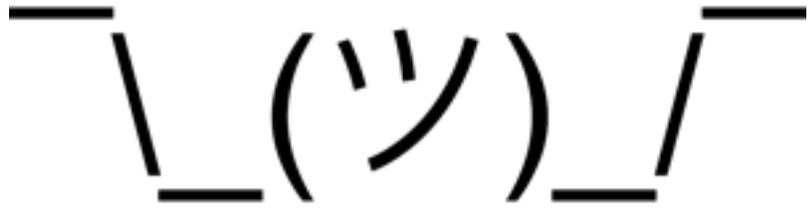


Openness and resilience is both gift and curse...



Web tech was terrible and
we hacked our way through





The need to be
forgiving...

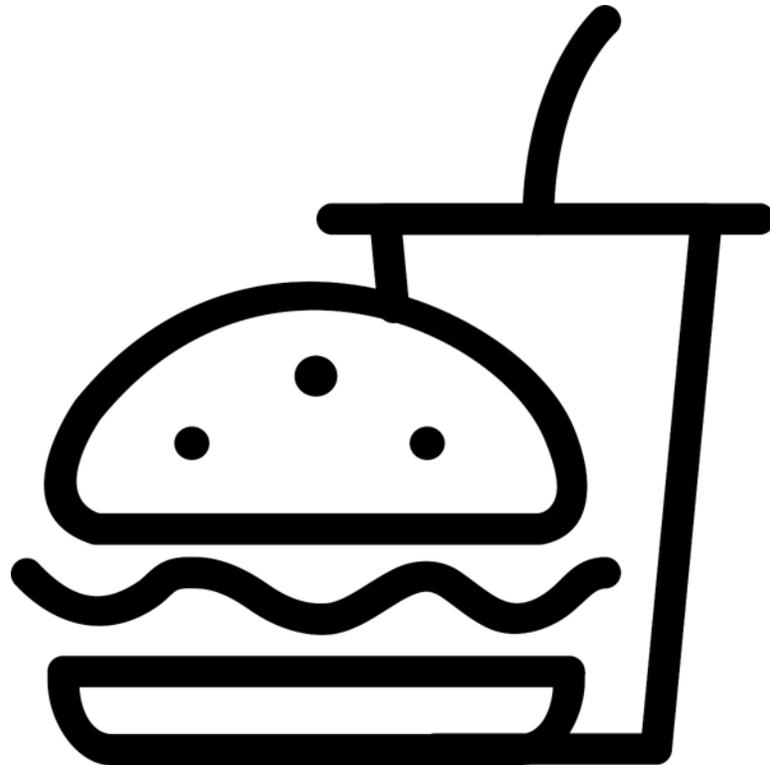


- Anyone is allowed to write code for the web – or use whatever they want to create that code.
- Whatever goes wrong, it is not the fault of the developer
- Users shouldn't suffer from developer errors
- Legacy products will never get fixed
- You can't break the web

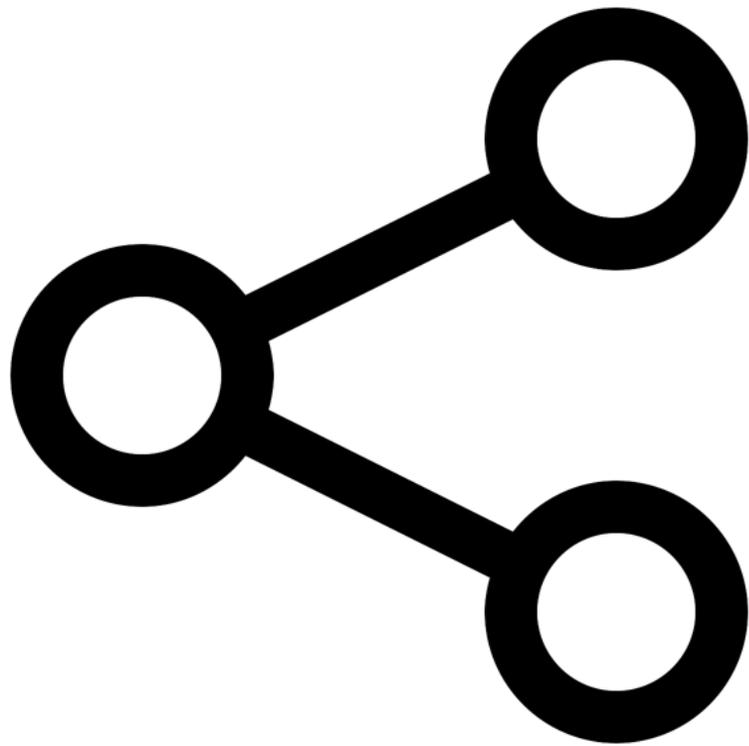
Years of technical debt...



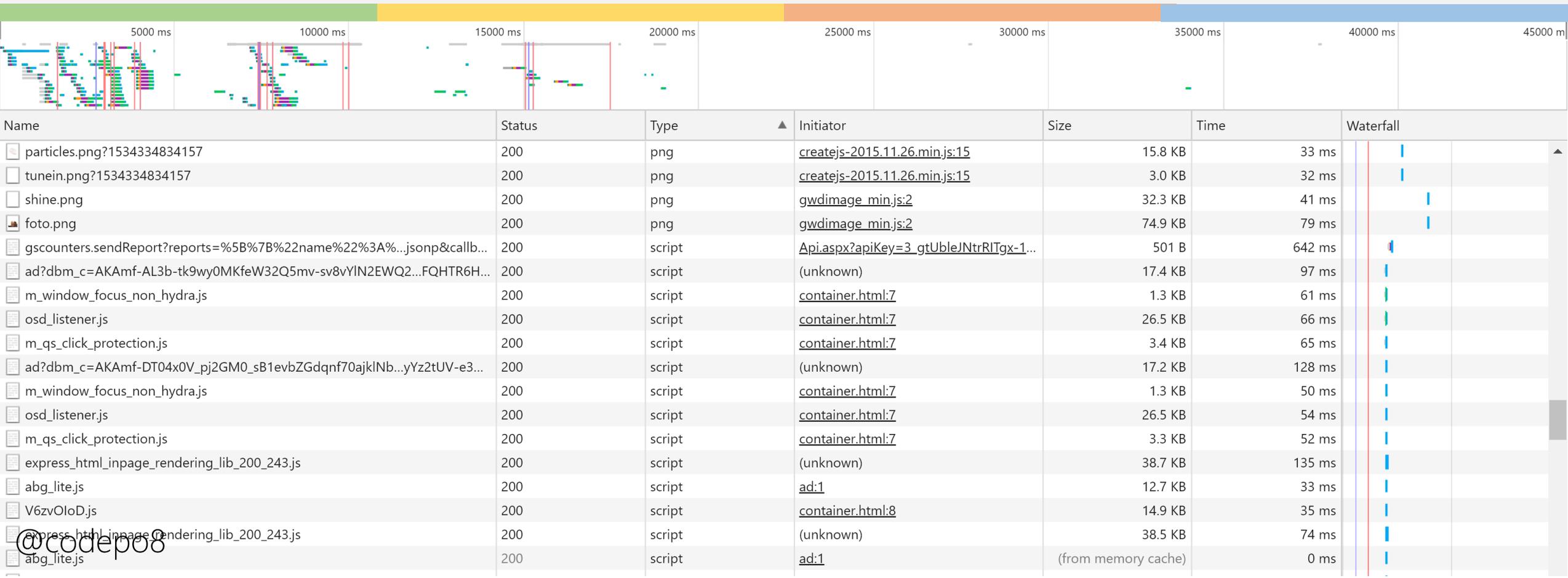
Quick, clever solutions beat understanding our job



Using shared code and building on it makes us fast

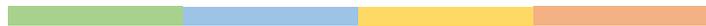


Fully supporting old code comes at a high price...





The mess we
created...



- The web has become a render target amongst many others
- Releasing often and developer convenience trumps semantics and performance of the final product
- Web sites are slow and huge because we work on fast computers and good connections
- Being a web expert is less exciting than being a full stack developer

It is not all broken...



Let's take a look at the great things we have...



Excellent resources and communication channels...

MDN web docs
Technologies ▾ References & Guides ▾ Feedback ▾
Sign in

Resources for developers, by developers.

Search the docs

Web Technologies → Learn web development → Developer Tools →

moz://a
Proudly non-profit.
Donate before Dec 31.
Donate Now →

Learn the best of web development
Get the latest and greatest from MDN delivered straight to your inbox.
you@example.com
Sign up now →

developer.mozilla.org

Fetch LS Global 78.35% + 0.06% = 78.41%

A modern replacement for XMLHttpRequest.

Current Agent	IE	Edge	Firefox	Chrome	Safari	IOS Safari	Opera Mini	Chrome for Android	UC Browser for Android	Samsung Internet
52		52	49			10.2				4
55		55	60			10.3				5
56		15	61		10.1					
57	11	16	57	62	11	11.1	all	62	11.4	6.3
58		17	58	63	TP					
59			59	64						
60			60	65						

Notes Known issues (0) Resources (7) Feedback

- Specification [fetch.spec.whatwg.org]
- Demo [addyosmani.com]
- Polyfill [github.com]
- IE platform status: Shipped [status.modern.ie]
- Chrome platform status: Enabled by default [chromestatus.com]
- Firefox platform status: 39 [platform.status.mozilla.org]
- WebKit platform status: Supported [webkit.org]

caniuse.com

FRONT-END checklist

The Front-End Checklist Application is perfect for modern websites and meticulous developers!

Project Name: [] HEAD | HTML | WEBFONTS | 0/47 ✓ high priority
Page title or URL: [] CSS | JAVASCRIPT | IMAGES | 0/25 ✓ medium priority
Developer's name or team: [] ACCESSIBILITY | PERFORMANCE | 0/16 ✓ low priority

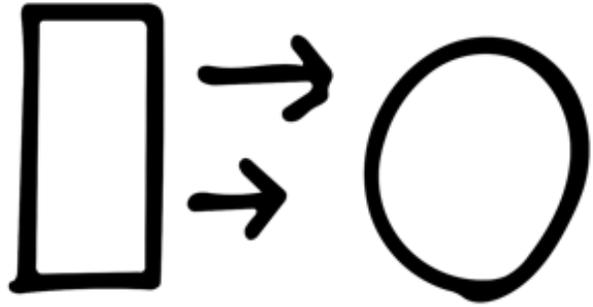
Generate report

Start new checklist

Supporter Airbrake.io Full-stack error tracking & analytics for JS developers. Try it Free!

Filter by tags
ALL BEST PRACTICES SEO TESTING CSS JAVASCRIPT PERFORMANCE META TAG HTML TAG CSS PERFORMANCE SOCIAL META HTML SECURITY IMAGES REQUESTS ACCESSIBILITY WEBFONT

frontendchecklist.io



Browsers are more
than content display
tools...



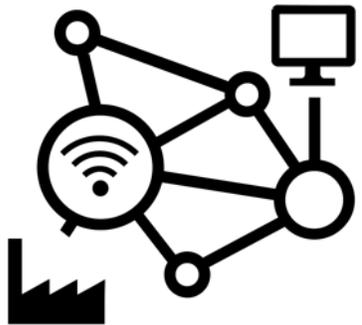
- Show web and multimedia content fast and reliably
- Access the web regardless of ability
- Keep users safe by blocking unsafe content and patch attack vectors
- Remember users' history and access credentials – or remember nothing
- Allow for detailed customisation (including extensibility)
- Sync content across a range of devices
- Include developer tools



Browsers as a
platform cater to
different audiences

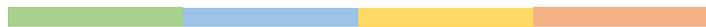
- Consumers of the web
- Creators of web content
- Web Developers
- Extension Developers
- App Developers
- IT Departments...

A changed, ubiquitous web.





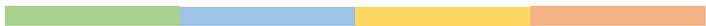
Web Developers
need to know a lot...



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



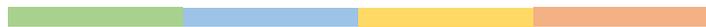
Where to learn
all that?



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



Learning obstacles



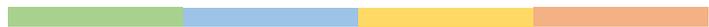
- Resources with upvoting options favour the how over the why
- We're not paid to learn, we're paid to deliver
- Peer pressure makes developers who don't know things afraid to admit it
- There's always a "good enough" way to create a lot in a short amount of time.

Issues with current tools...

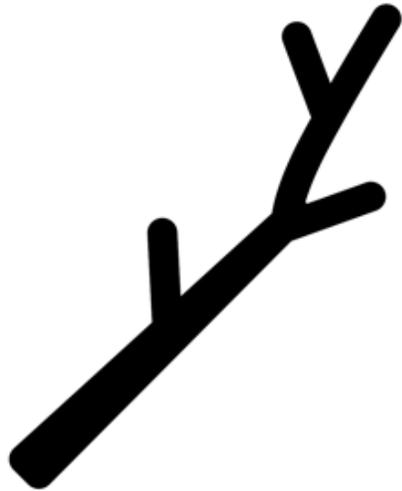




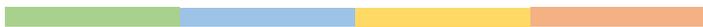
Too many
promises...



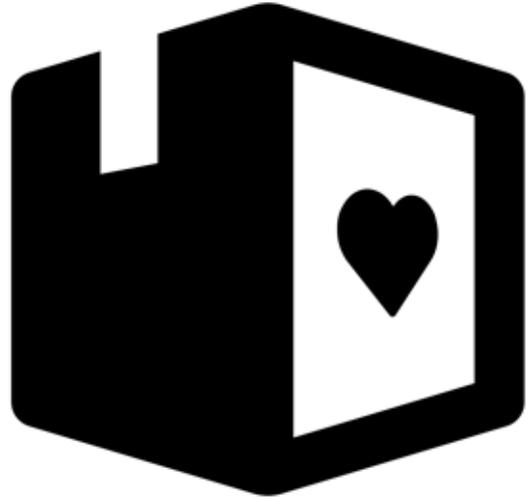
- “This tool makes \$x easy. No need to look it up.”
- “\$HUGE_CORP_APP uses this configuration and toolchain – if it works for their needs, it should rock for you, too”
- “Using this configuration optimises for the *modern* web”



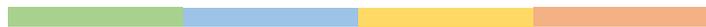
Too many rules and barriers



- Opinionated tools demand you to fix things that don't apply to your product
- Dependencies and complex toolchains can be daunting
- "Best practices" often don't get updates, and are contextual
- Having to follow strict rules without understanding them is not fun



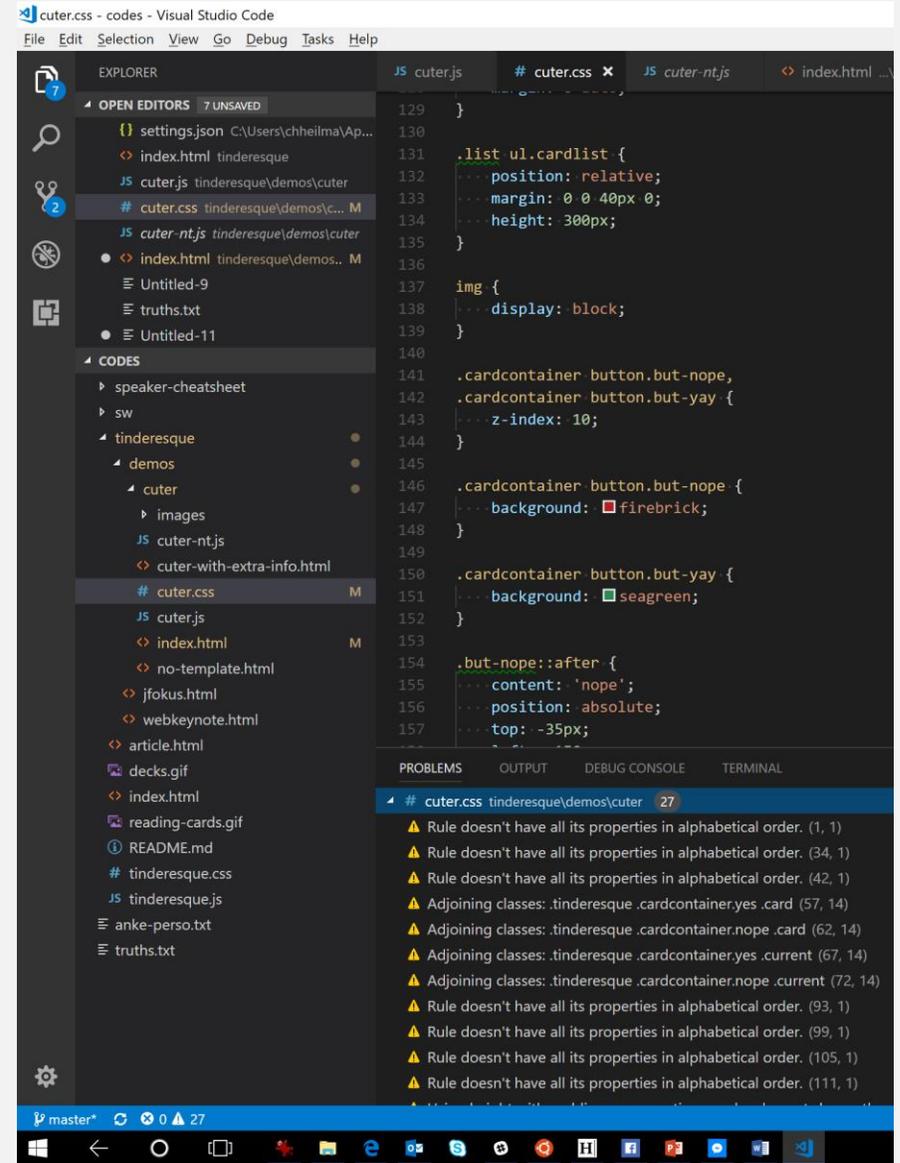
Customisable best practices in context



- 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



webhint.io



A JS Foundation Project



SCANNER

DOCUMENTATION ▾

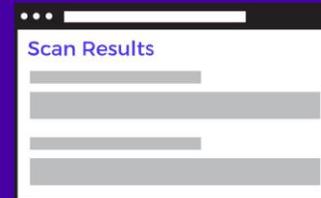
ABOUT ▾

Search Documentation

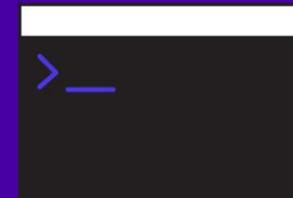


Use webhint to improve your website

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.



TRY THE ONLINE SCANNER



GET STARTED WITH THE CLI

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...

[Home](#) \ [Scanner](#) \ [Results](#)

SCANNING 100% SCAN TIME: 00:20

HINTS URL: <http://example.com> DATE: 2018-10-25 10:48

8 YOUR SCAN RESULT LINK: <https://webhint.io/scanner/15d800c2-7dd2-4891-95d4-ac5ae4d602c7>

webhint version: 3.4.11 Configuration JSON

Hints

Accessibility  + expand all

+ axe: 1 hints

Interoperability  + expand all

+ meta-charset-utf-8: 2 hints

PWA  + expand all

 ACCESSIBILITY	HINTS 1	PASSED 0/1
 INTEROPERABILITY	HINTS 2	PASSED 8/9
 PWA	HINTS 1	PASSED 4/5
 PERFORMANCE	HINTS 2	PASSED 7/8
 SECURITY	HINTS 2	PASSED 7/9

Get explanations why these are issues & how to fix them

Interoperability

 close all

meta-charset-utf-8: 2 hints

Why is this important 

How to fix this 

hint #1: 'charset' meta element should be the first thing in '<head>'. 

<http://example.com/>

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

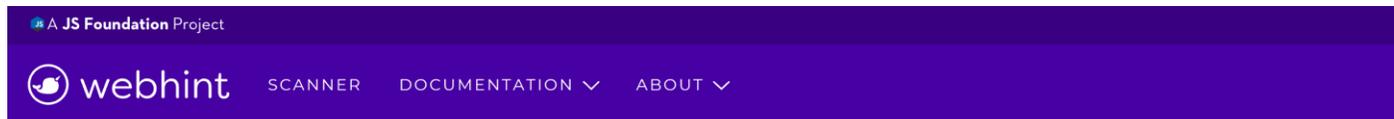
hint #2: 'charset' meta element is not needed as one was already specified. 

<http://example.com/>

```
<meta http-equiv="Content-type" content="text/html; charset=utf-8">
```

 ACCESSIBILITY	HINTS 1	PASSED 0/1
 INTEROPERABILITY	HINTS 2	PASSED 8/9
 PWA	HINTS 1	PASSED 4/5
 PERFORMANCE	HINTS 2	PASSED 7/8
 SECURITY	HINTS 2	PASSED 7/9

Get in-depth explanation on the issues...



Home \ Documentation \ User guide \ Require charset meta tag with the value of `utf-8`

Require Charset Meta Tag With The Value Of `Utf-8`

CONCEPTS >

CONFIGURING WEBHINT >

DEVELOPMENT FLOW
INTEGRATION >

HINTS

[Nu HTML Test](#)

[AMP HTML Validator](#)

[Performance budget](#)

[Avoid HTTP redirects in requests](#)

[Accessibility assessment with aXe](#)

[Require `Strict-Transport-Security` response header](#)

`meta-charset-utf-8` checks if the page explicitly declares the character encoding as `utf-8` using a meta tag early in the document.

Why is this important? [🔗](#)

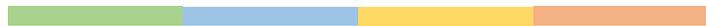
The character encoding should be specified for every HTML page, either by using the charset parameter on the `Content-Type` HTTP response header (e.g.: `Content-Type: text/html; charset=utf-8`) and/or using the charset meta tag in the file.

Sending the `Content-Type` HTTP header is in general ok, but it's usually a good idea to also add the charset meta tag because:

- Server configurations might change (or servers might not send the charset parameter on the `Content-Type` HTTP response header).



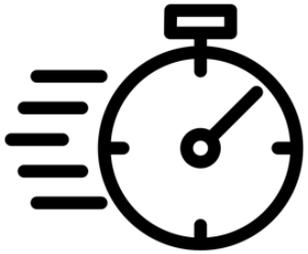
Hinting during deployment



- Web resources are fun but not always accessible or applicable to your product
- 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...



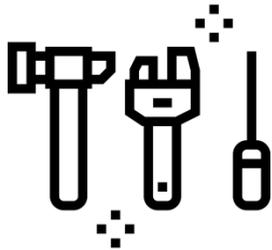
Performance



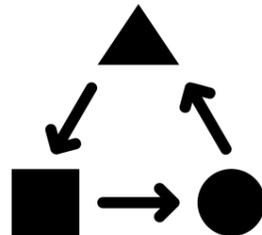
Accessibility



PWA
Readiness



Development
environment

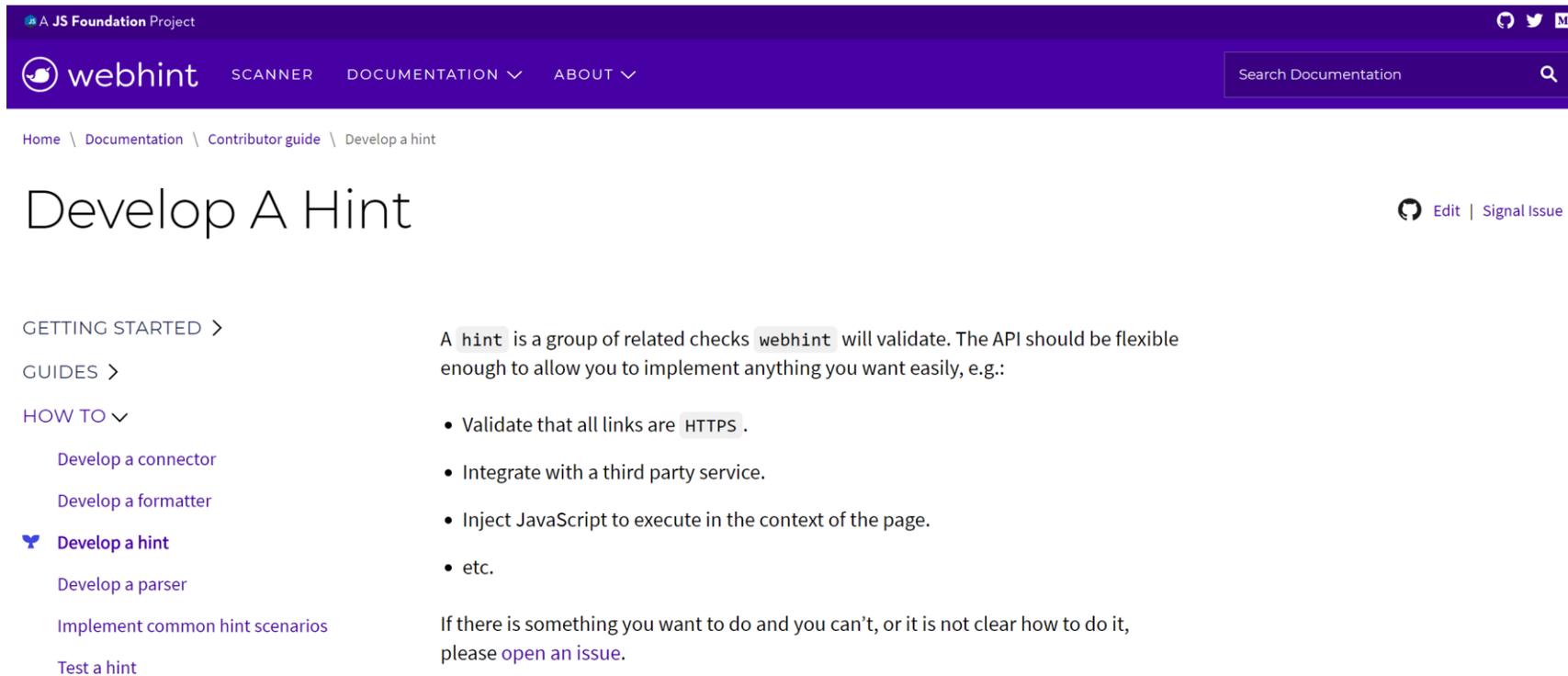


Interoperability



Security

Customising hints...



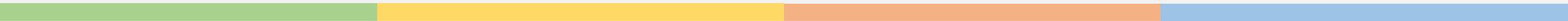
The screenshot shows the webhint documentation page for 'Develop A Hint'. The page has a purple header with the webhint logo and navigation links: SCANNER, DOCUMENTATION (with a dropdown arrow), and ABOUT (with a dropdown arrow). A search bar for 'Search Documentation' is located on the right side of the header. Below the header, a breadcrumb trail reads: Home \ Documentation \ Contributor guide \ Develop a hint. The main heading is 'Develop A Hint', with 'Edit' and 'Signal Issue' links on the right. A left sidebar contains a table of contents with sections: GETTING STARTED >, GUIDES >, HOW TO (with a dropdown arrow), and 'Develop a hint' (which is highlighted with a blue icon). Under 'HOW TO', there are links for 'Develop a connector', 'Develop a formatter', 'Develop a hint', '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 webhint will validate. It states that 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.

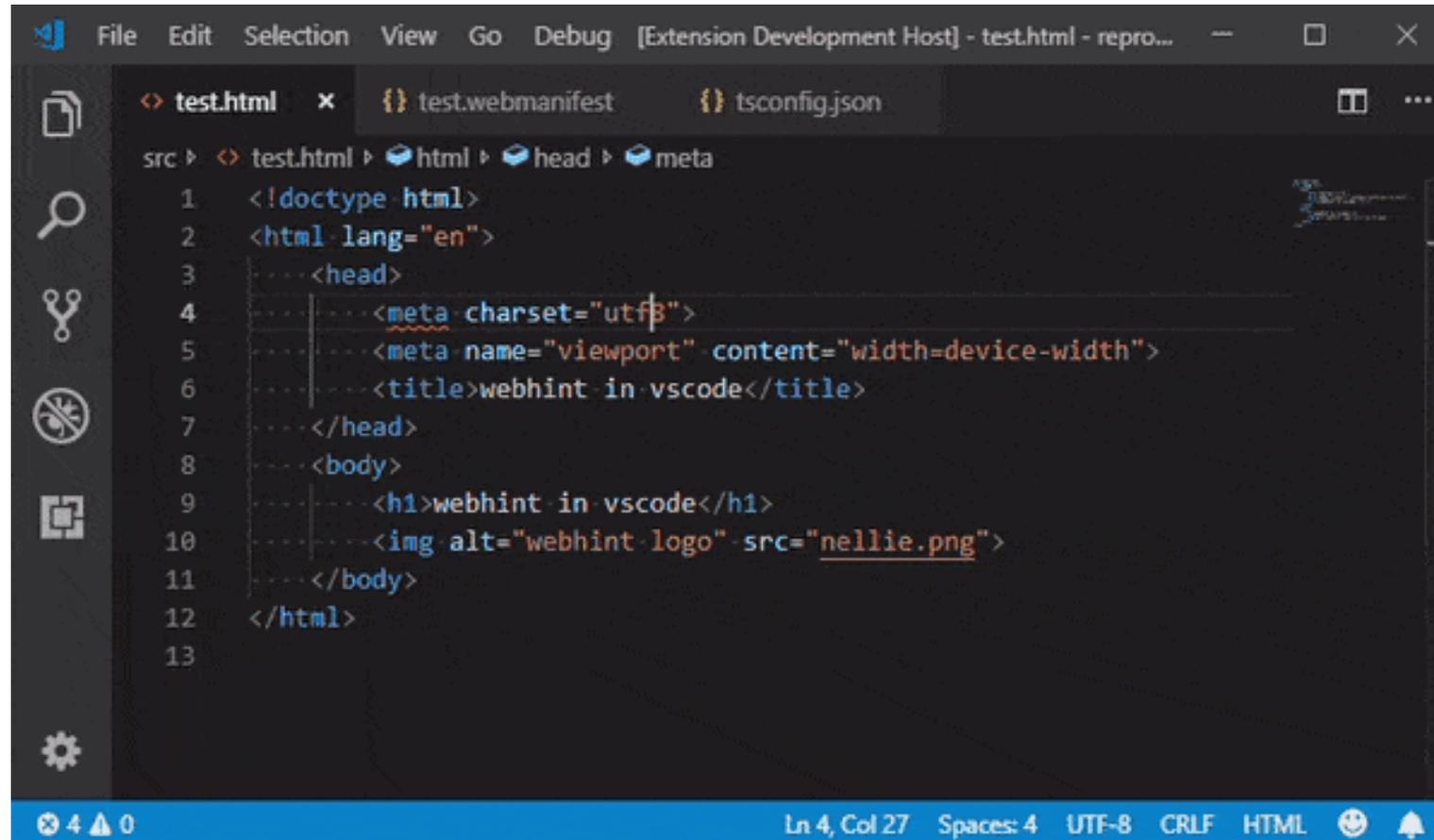
It concludes by saying: 'If there is something you want to do and you can't, or it is not clear how to do it, please [open an issue](#).'



How about
preventing people
to make errors?



webhint @code extension



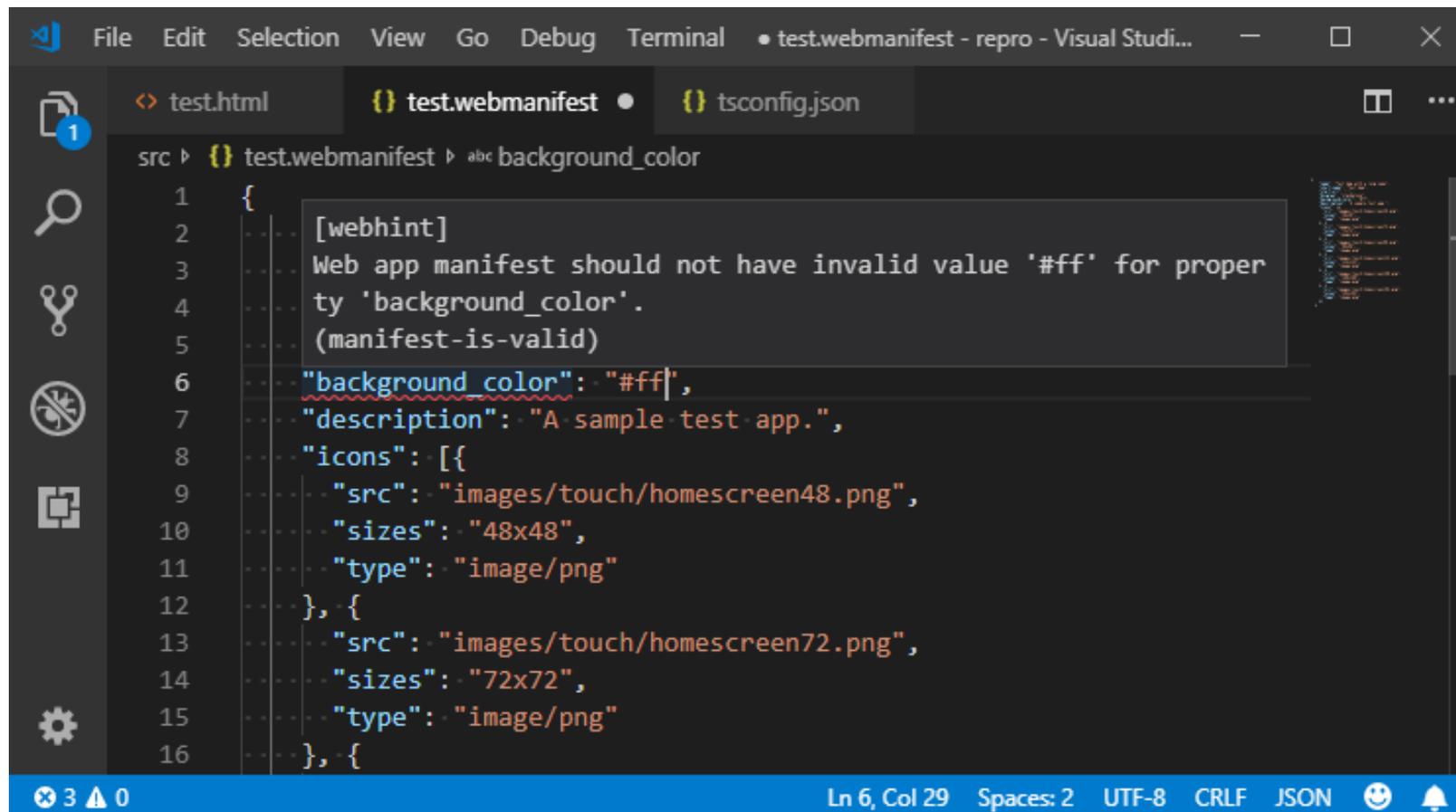
The screenshot shows the Visual Studio Code editor interface. The top menu bar includes File, Edit, Selection, View, Go, and Debug. The title bar indicates the current file is 'test.html' in an 'Extension Development Host'. The editor has three tabs open: 'test.html', 'test.webmanifest', and 'tsconfig.json'. The 'test.html' tab is active, showing the following HTML code:

```
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
```

The status bar at the bottom shows 'Ln 4, Col 27', 'Spaces: 4', 'UTF-8', 'CRLF', 'HTML', and a notification bell icon.

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

Create valid app manifests...



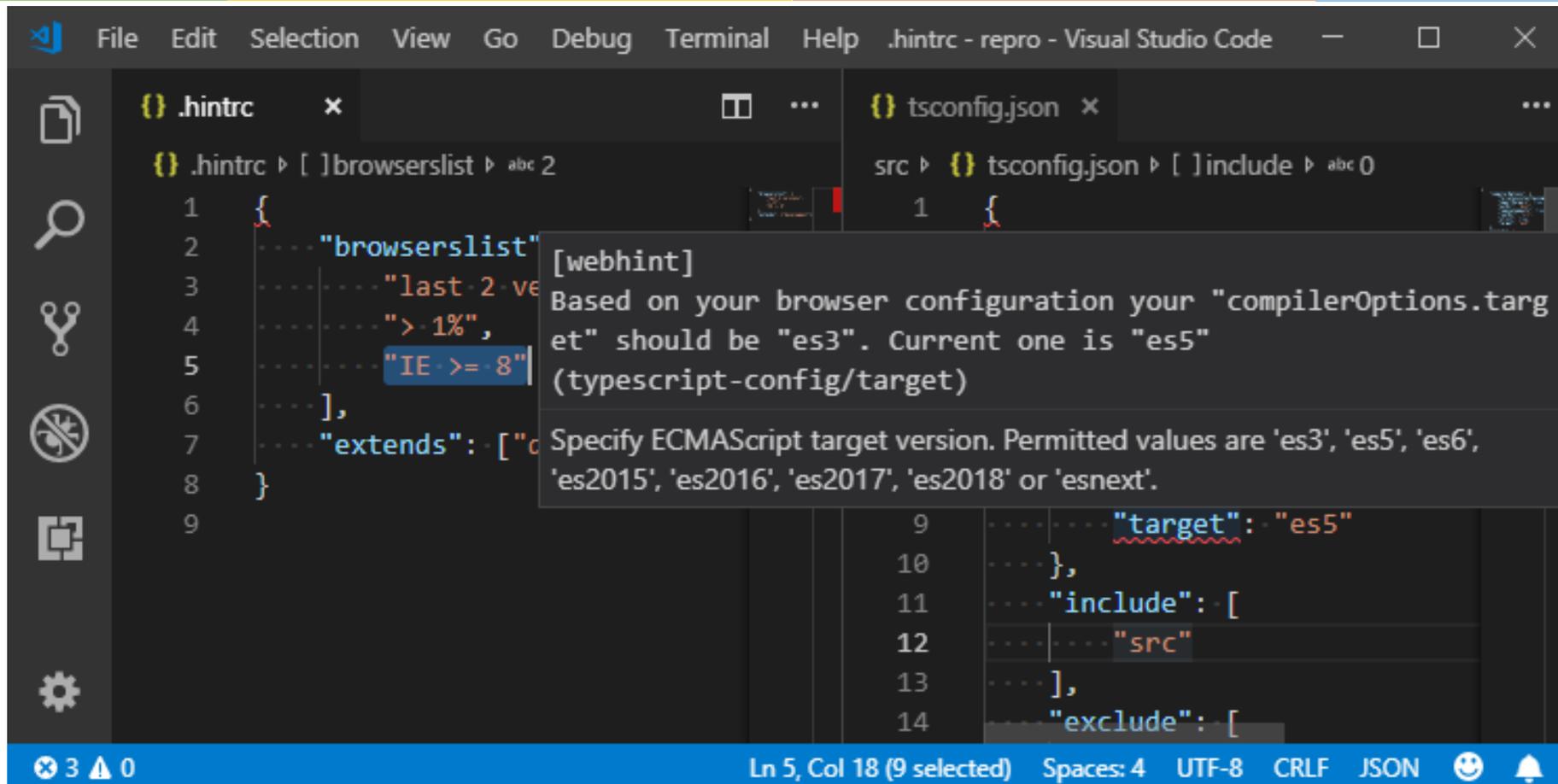
The screenshot shows the Visual Studio Code editor with a TypeScript file named `test.webmanifest` open. The file contains a JSON object representing a web app manifest. A red squiggly line underlines the value `"#ff"` for the `background_color` property. A tooltip is displayed over this error, containing the following text:

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

```
1 {
2   [webhint]
3   Web app manifest should not have invalid value '#ff' for property 'background_color'.
4   (manifest-is-valid)
5   "background_color": "#ff",
6   "description": "A sample test app.",
7   "icons": [{
8     "src": "images/touch/homescreen48.png",
9     "sizes": "48x48",
10    "type": "image/png"
11  }, {
12    "src": "images/touch/homescreen72.png",
13    "sizes": "72x72",
14    "type": "image/png"
15  }, {
16    "src": "images/touch/homescreen96.png",
17    "sizes": "96x96",
18    "type": "image/png"
19  }, {
20    "src": "images/touch/homescreen144.png",
21    "sizes": "144x144",
22    "type": "image/png"
23  }, {
24    "src": "images/touch/homescreen192.png",
25    "sizes": "192x192",
26    "type": "image/png"
27  }],
28 }
```

The status bar at the bottom of the editor shows 3 errors and 0 warnings. The current cursor position is at line 6, column 29. The file encoding is UTF-8, line endings are CRLF, and the language 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, including `"IE >= 8"`. The `tsconfig.json` file has a `"target": "es5"` property. A webhint tooltip is displayed over the `tsconfig.json` file, providing a warning and a suggestion.

```
File Edit Selection View Go Debug Terminal Help .hintrc - repro - Visual Studio Code
```

```
.hintrc x
```

```
tsconfig.json x
```

```
src tsconfig.json include abc 0
```

```
1 {
```

```
2   "browserslist": [
```

```
3     "last 2 versions",
```

```
4     "> 1%",
```

```
5     "IE >= 8"
```

```
6   ],
```

```
7   "extends": [
```

```
8 ]
```

```
9
```

```
1 {
```

```
9   "target": "es5"
```

```
10 },
```

```
11 "include": [
```

```
12   "src"
```

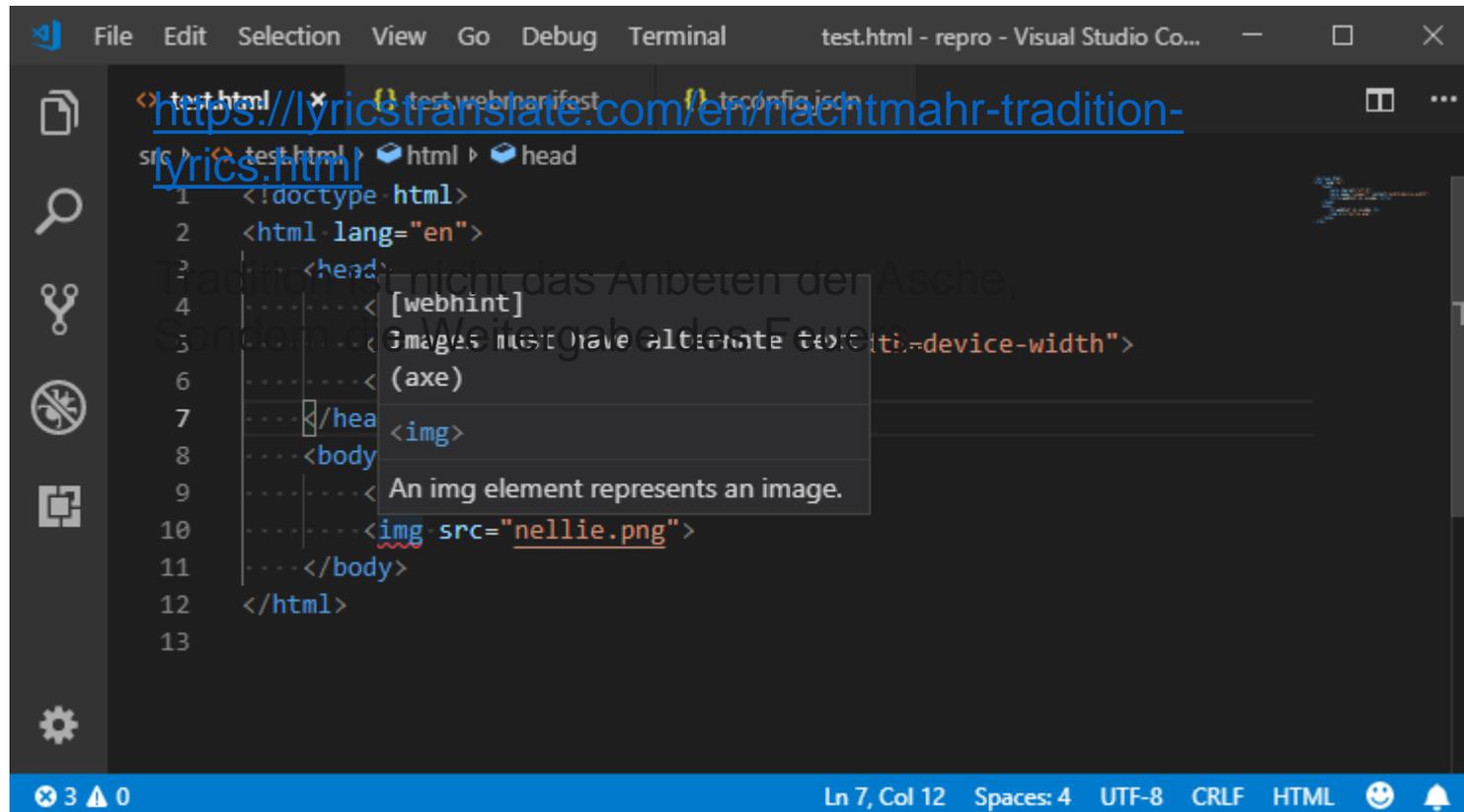
```
13 ],
```

```
14 "exclude": [
```

```
Ln 5, Col 18 (9 selected) Spaces: 4 UTF-8 CRLF JSON
```

[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?



```
test.html x test.webmanifest testconfig.json
https://lyricstranslate.com/en/nachtmahr-tradition-lyrics.html
1 <!doctype html>
2 <html lang="en">
3 <head>
4 [webhint]
5 
6 (axe)
7 </head>
8 <body>
9 
10 </body>
11 </html>
12
13
```

Ln 7, Col 12 Spaces: 4 UTF-8 CRLF HTML



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.

Thanks!

Chris Heilmann
@codepo8

