

Hinting at a better web...





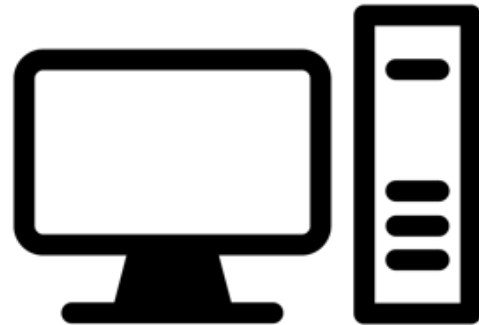
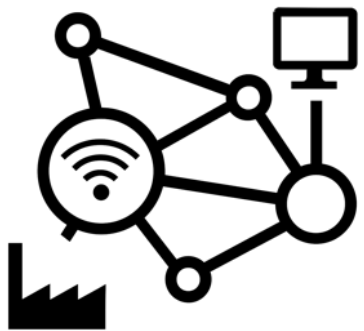
What is the purpose of a browser?

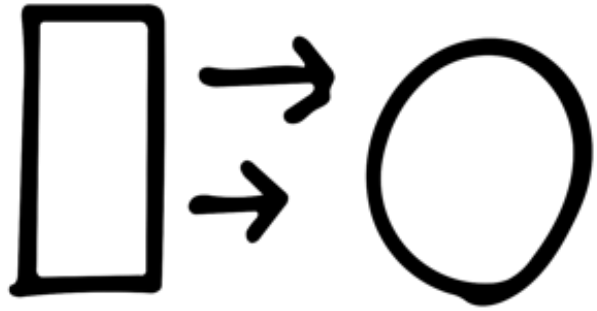


- Render web content
- Show multimedia content
- Follow and inspire web standards



A changed, ubiquitous web.





The job of a browser
changed drastically...

- Show web and multimedia content fast and reliably
- Give access to 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



Audiences with
different needs...



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



Web Developers
need to know a lot...



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



We are a troubled
lot...



- We enjoy a lot of **privilege**, but we also feel a lot of **pressure**.
- We're in **high demand**, but the people who want us have **no idea what we do**.
- We're better off than ever in terms of tooling, technology and standards support – yet we feel that if we don't innovate, *someone* will eat our lunch.

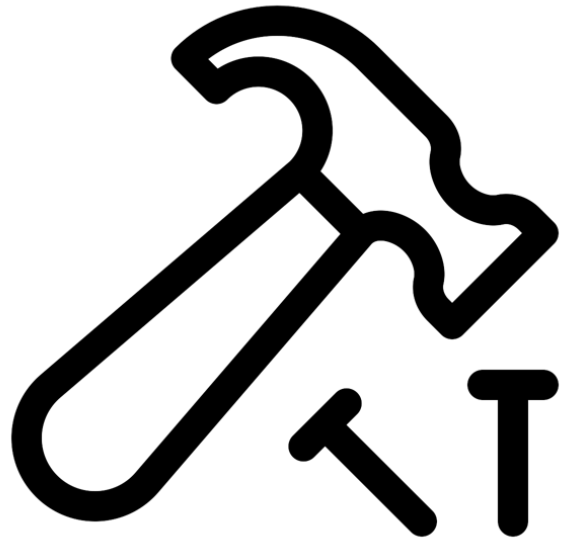


"JavaScript fatigue"



“ JavaScript fatigue: the inability to keep up with the latest tools, the fear of becoming obsolete, the constant change of the ecosystem, and the overwhelming choice.

- Kim Maida/Auth0



“Tools can fix that!”



- Best practice setups
- Boilerplates
- Build tools with automated optimisation
- Replacing JavaScript and the DOM with something better
- Debugging tools
- Generators



“Full-stack anxiety”



“ There are now **more tools and plugins** developed to help aid site creation than there ever has been before.

- Paul Spencer

<https://humaan.com/blog/the-web-isnt-what-it-used-to-be/>

<https://medium.com/@FirstMark/are-you-suffering-from-full-stack-anxiety-a315325dd0f9>



“Full-stack anxiety”



“ For graduates starting out these ‘helpful’ tools and plugins are a double edged sword. Job applications rapidly cite these technologies as prerequisites for winning a position.

- Paul Spencer

<https://humaan.com/blog/the-web-isnt-what-it-used-to-be/>

<https://medium.com/@FirstMark/are-you-suffering-from-full-stack-anxiety-a315325dd0f9>



“Full-stack anxiety”

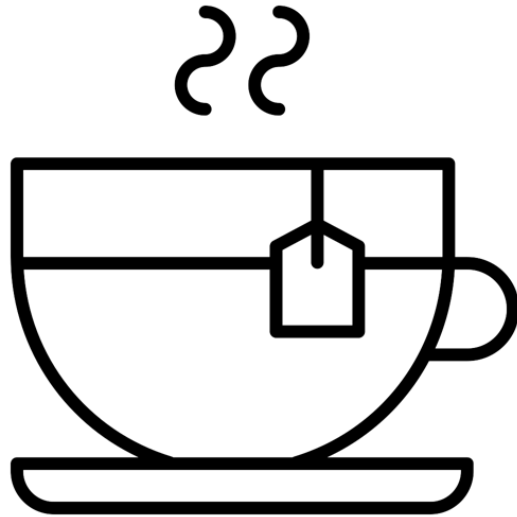


“ As a result, we see more “full stack anxiety”, where things are advancing too quickly and the amount of learning required for a fresh graduate is completely overwhelming.

- Paul Spencer

<https://humaan.com/blog/the-web-isnt-what-it-used-to-be/>

<https://medium.com/@FirstMark/are-you-suffering-from-full-stack-anxiety-a315325dd0f9>



Wanted: T people



“

Companies have been attracted to T-shaped persons, where the vertical bar of the T represents expertise in a single field and depth of related skills, and the horizontal bar is the ability to collaborate across disciplines and apply knowledge in fields outside one's expertise.

- Joel Califa

<https://humaan.com/blog/the-web-isnt-what-it-used-to-be/>

<https://medium.com/@FirstMark/are-you-suffering-from-full-stack-anxiety-a315325dd0f9>



More wanted: **W**
people



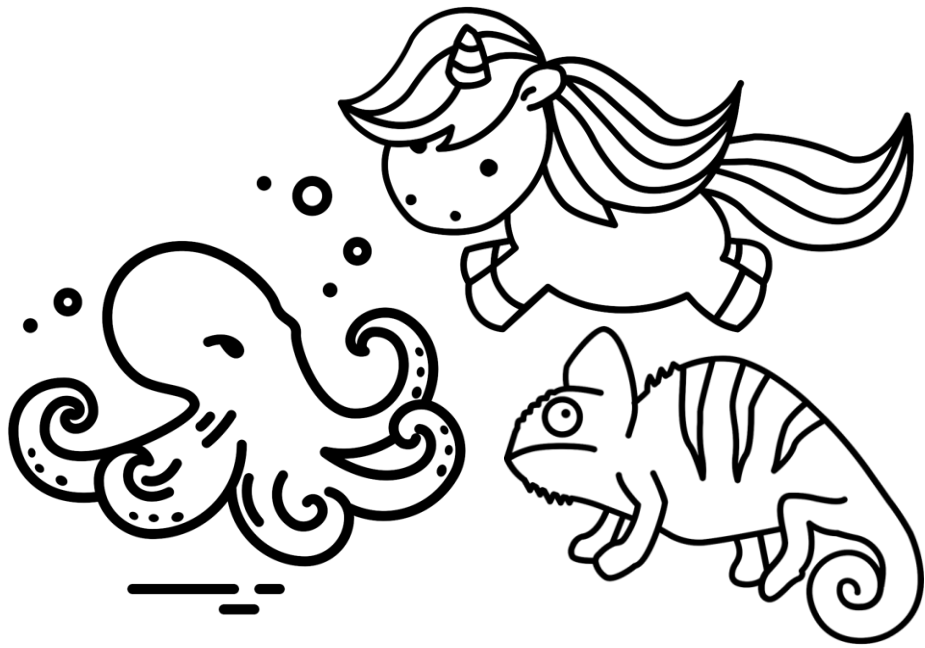
“

The new, hip thing is a W-shaped designer. Hiring managers are looking for designers who are experts in not one, but two fields (say both visual design and experience design), while maintaining a foundation in the rest (coding, lettering, etc.).

- Joel Califa

<https://humaan.com/blog/the-web-isnt-what-it-used-to-be/>

<https://medium.com/@FirstMark/are-you-suffering-from-full-stack-anxiety-a315325dd0f9>

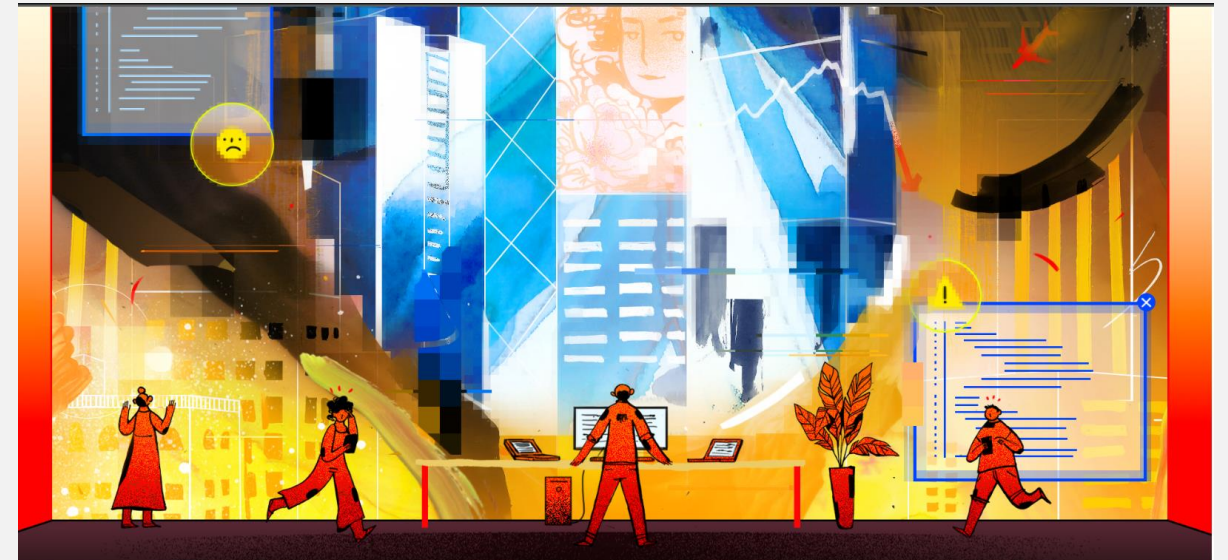


Really wanted: A-Z
people, a.k.a.:
"Full Stack Devs"

- Know everything without your company having to give you time to learn it
- Be young and independent, so you can enjoy all the overtime you have to put in to reach arbitrary deadlines
- Have years of experience especially applying brand new technologies



All in all, the state of software and the web isn't surprising.



Lynn Scurfield

The Coming Software Apocalypse

A small group of programmers wants to change how we code—before catastrophe strikes.

THERE WERE six hours during the night of April 10, 2014, when the entire population of Washington State had no 911 service. People who called for help got a busy signal. One Seattle woman dialed 911 at least 37 times while a stranger was trying to break into her house. When he finally crawled into her living room through a window, she picked up a kitchen knife. The man fled.

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



But, why is that?



We don't lack resources:

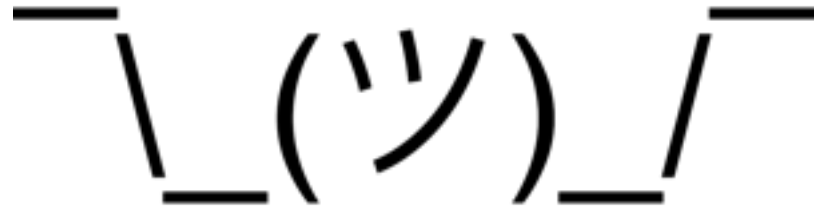
- **Platform:** Excellent browsers, mature OSes
- **Information:** MDN, caniuse.com
- **Tools:** developer tools, highly customisable editors, collaboration suites
- **Abstractions:** React, Vue, Angular...
- **Hands-on training:** video courses, training courses on GitHub, workshops



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.



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 the browser's fault
- Users shouldn't suffer from developer errors
- Legacy products will never get fixed
- You can't break the web



The mess that
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



The mess that created...



You Retweeted

 **Dylan Beattie**  @dylanbeattie · Feb 20

```
$ create-react-app my-react-app  
$ cd my-react-app  
$ cloc .
```

Before you even open a text editor, your project contains 1.5 million lines of code. Most of it contributed by volunteers and enthusiasts. No formal review or release process.

This is 'normal' in modern web development.

Language	files	blank	comment	code
JavaScript	12193	204003	223944	1127134
Markdown	1755	71061	2	183288
JSON	1767	523	0	132948
TypeScript	1089	4499	24967	28653
HTML	55	108	30	8198
C/C++ Header	21	1148	347	5959
YAML	162	146	135	5121
XML	12	241	13	2271
CoffeeScript	27	591	51	1513
CSS	23	76	24	1412
Bourne Shell	8	173	129	902
make	23	209	197	670
C++	10	100	54	474
Windows Module Definition	5	88	0	442
Python	1	81	175	355
D	5	0	0	71
Handlebars	2	10	0	54
Bourne Again Shell	2	11	1	43
TOML	1	2	0	39
Lisp	1	0	0	6
DOS Batch	1	0	0	2
SUM:	17163	283070	250069	1499555



The mess that
created...



🔄 You Retweeted


 **Katie Hempenius** @katiehempenius · Feb 17
"27% of sites are 90-99% third party content"

By @paulcalvano w/ @HTTPArchive data

... of 3rd Party Content Per Site - HTTP Archive, Au

... ty content

**38% of sites have
between 75% and 99%
3rd party content**



What is the distribution of 1st party vs 3rd party res...

The last time we looked at this query, we had 470K sites in the HTTP Archive. Now that we're upwards of 1.3 million, I was wondering whether this changed much. S...

discuss.httparchive.org

<https://discuss.httparchive.org/t/what-is-the-distribution-of-1st-party-vs-3rd-party-resources/100/13>

¯_(ツ)_/¯

Basic things get
forgotten...



```
#banner {  
  position: absolute;  
  top: 0;  
  text-align: center;  
  padding: 5px;  
  pointer-events: none;  
  transition: 300ms;  
}  
  
#banner span {  
  background: rgba(0,0,0,.7);  
  padding: 5px;  
  font-size: 7vh;  
  color: white;  
}  
  
#banner span:empty {  
  display: none;  
}
```

┌ (ツ) ─┐

Basic things get forgotten...



```
#banner {  
  position: absolute;  
  top: 0;  
  text-align: center;
```



[Redacted Name] · Feb 12
if `var` is empty, the <p> will not be "empty"

```
<p>
```

```
  {{ var }}
```

```
<p>
```

```
  color: white;
```

```
}
```

```
#banner span:empty {
```

```
  display: none;
```

```
}
```



Even better – we
concentrate on the
wrong things...



Lynn

@chordbug

Follow

🌶️ HOT JAVASCRIPT TIP: 🌶️

to increment some counter on the page,

```
node.innerText += 1
```

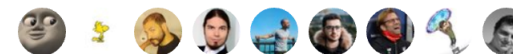
doesn't work (0 → 01 → 011 → ...), but

```
node.innerText -= -1
```

works fine (0 → 1 → 2 → ...)

4:36 PM - 5 Feb 2019

1,735 Retweets 4,865 Likes



165 1.7K 4.9K



Even better – we
concentrate on the
wrong things...



to increment some counter on the page,

```
node.innerText += 1
```

doesn't work (0 → 01 → 011 → ...), but

```
node.innerText -= -1
```

works fine (0 → 1 → 2 → ...)



Lynn @chordbug · Feb 5

(because of this coercion behavior. of course. of course)

```
> "5" - 1
```

```
<< 4
```

```
> "5" + 1
```

```
<< "51"
```





Even better – we
concentrate on the
wrong things...



Josh SOON **Screaming. Forever.** @YamiNoSenshi · Feb 5

Replying to @chordbug

There's strong typing, weak typing, and whatever the hell this is.



Siobhan Beeman @ProfBeeman · Feb 5

Passive-aggressive typing.





Even better – we
concentrate on the
wrong things...



Lynn
@chordbug

Follow

🌶️ HOT JAVASCRIPT TIP: 🌶️

to increment some counter on the page,

```
node.innerText += 1
```

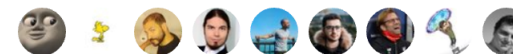
doesn't work (0 → 01 → 011 → ...), but

```
node.innerText -= -1
```

works fine (0 → 1 → 2 → ...)

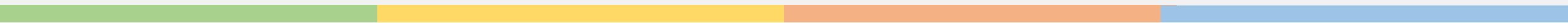
4:36 PM - 5 Feb 2019

1,735 Retweets 4,865 Likes

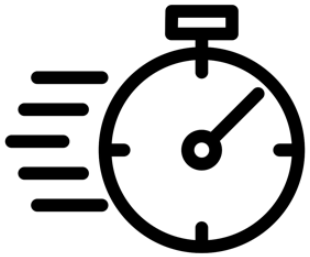


165 1.7K 4.9K

\$ cd ..



Things that matter and you should know...



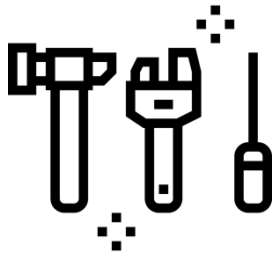
Performance



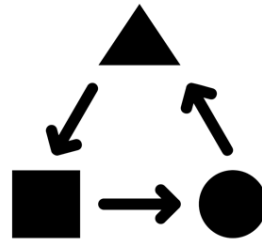
Accessibility



PWA
Readiness



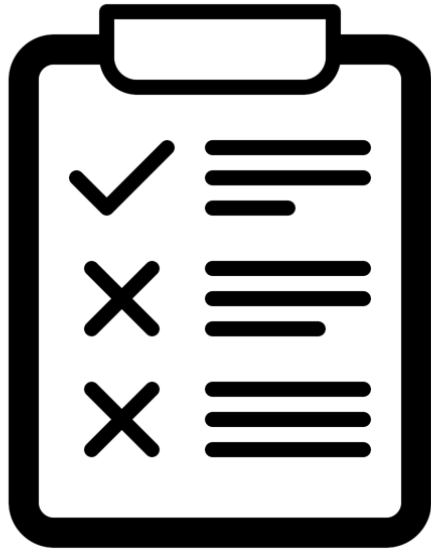
Development
environment



Interoperability



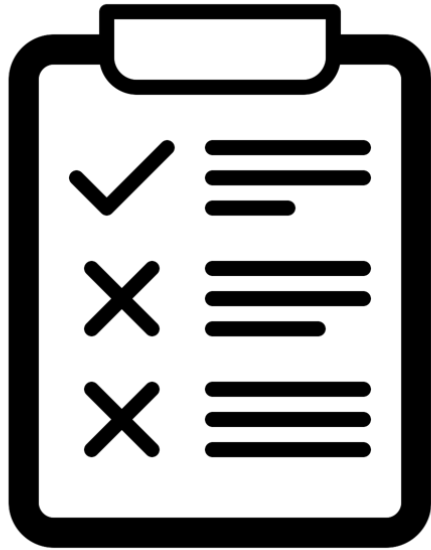
Security



There are tools and services for this...



- **Axe-Core** for accessibility testing
<https://www.deque.com/axe/>
- **Snyk** for security testing
<https://snyk.io/>
- **SSL scanner**
<https://www.ssllabs.com/ssltest/>
- In-browser devtools audit with Lighthouse and (<https://web.dev>)
- **webhint.io** for an all-in-one solution



There are tools and services for this...



Audits


Identify and fix common problems that affect your site's performance, accessibility, and user experience. [Learn more](#)

 Device

- Mobile
- Desktop

Audits

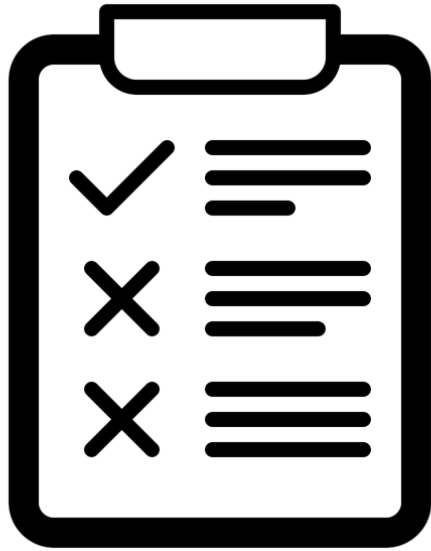
- Performance
- Progressive Web App
- Best practices
- Accessibility
- SEO

 Throttling

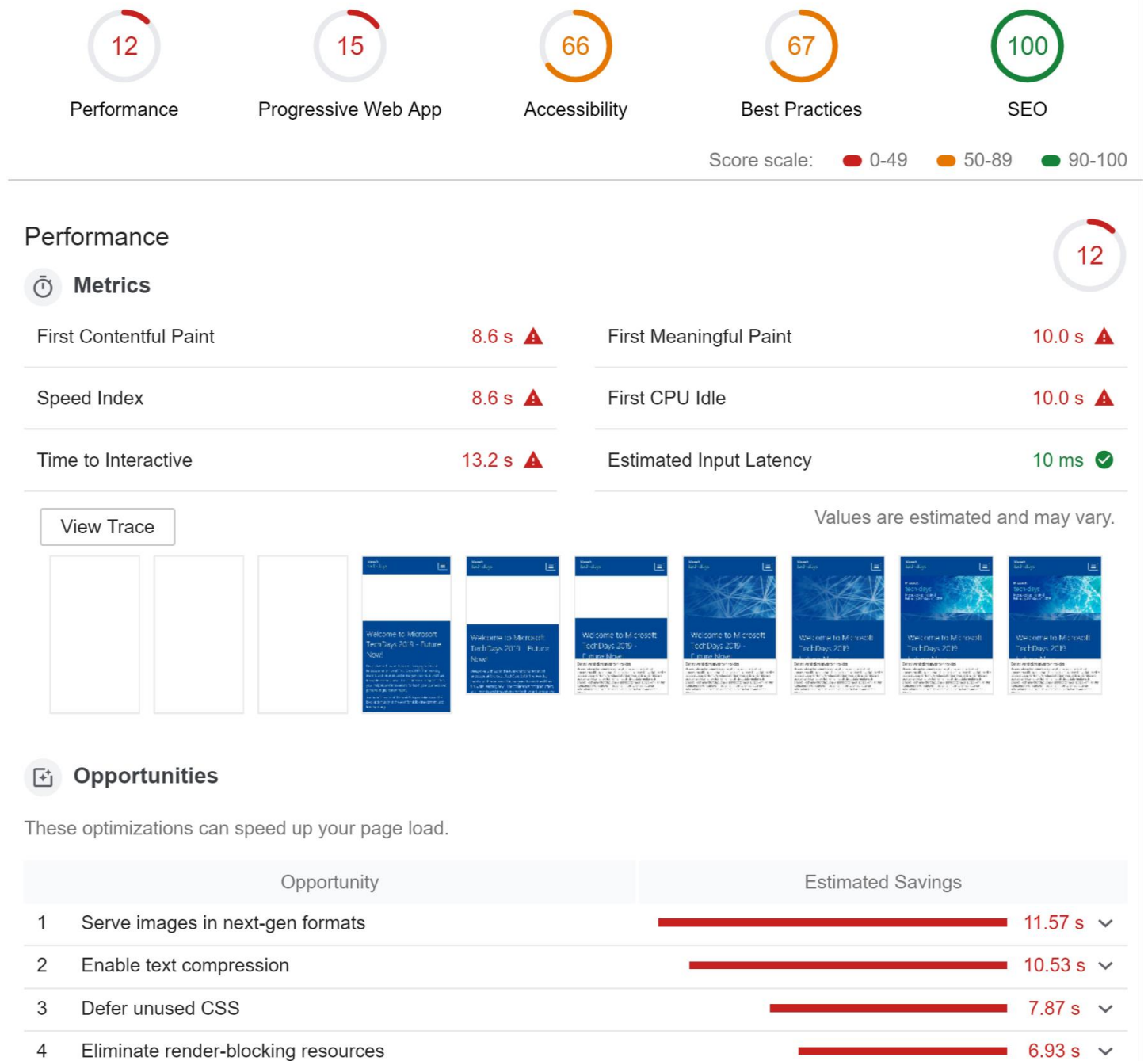
- Simulated Fast 3G, 4x CPU Slowdown
- Applied Fast 3G, 4x CPU Slowdown
- No throttling

Clear storage

Run audits



Auditing report

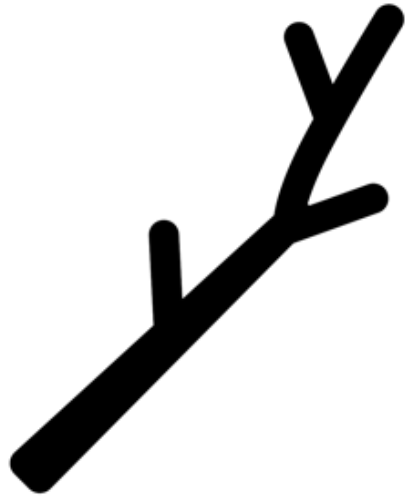




Too many
promises...



- “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



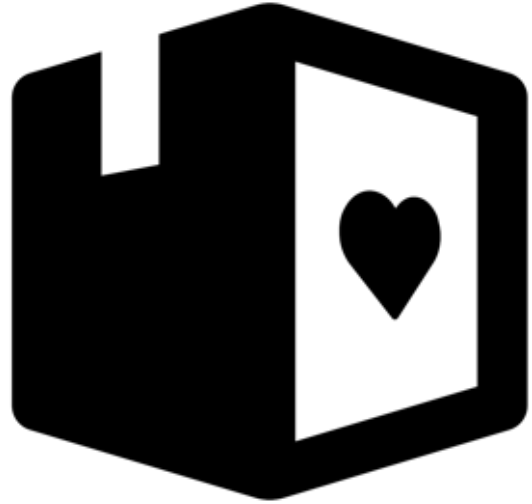
- Opinionated tools demand you to fix things that don't apply to your product
- Dependencies and complex toolchains can be daunting
- "Best practices" often don't get updates, and are contextual
- Having to follow strict rules without understanding them is not fun



What is "best practice"?



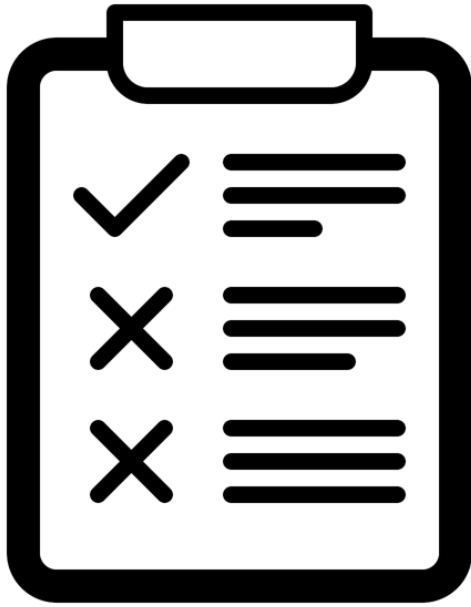
- What is considered a best practice is dependent on the goal
- The problem is that the business goals of the product don't necessarily mesh with our goals as developers
- Even worse – often what we defined as a best practice years ago isn't quite as important to new developers
- Often we only learn their value once we messed up



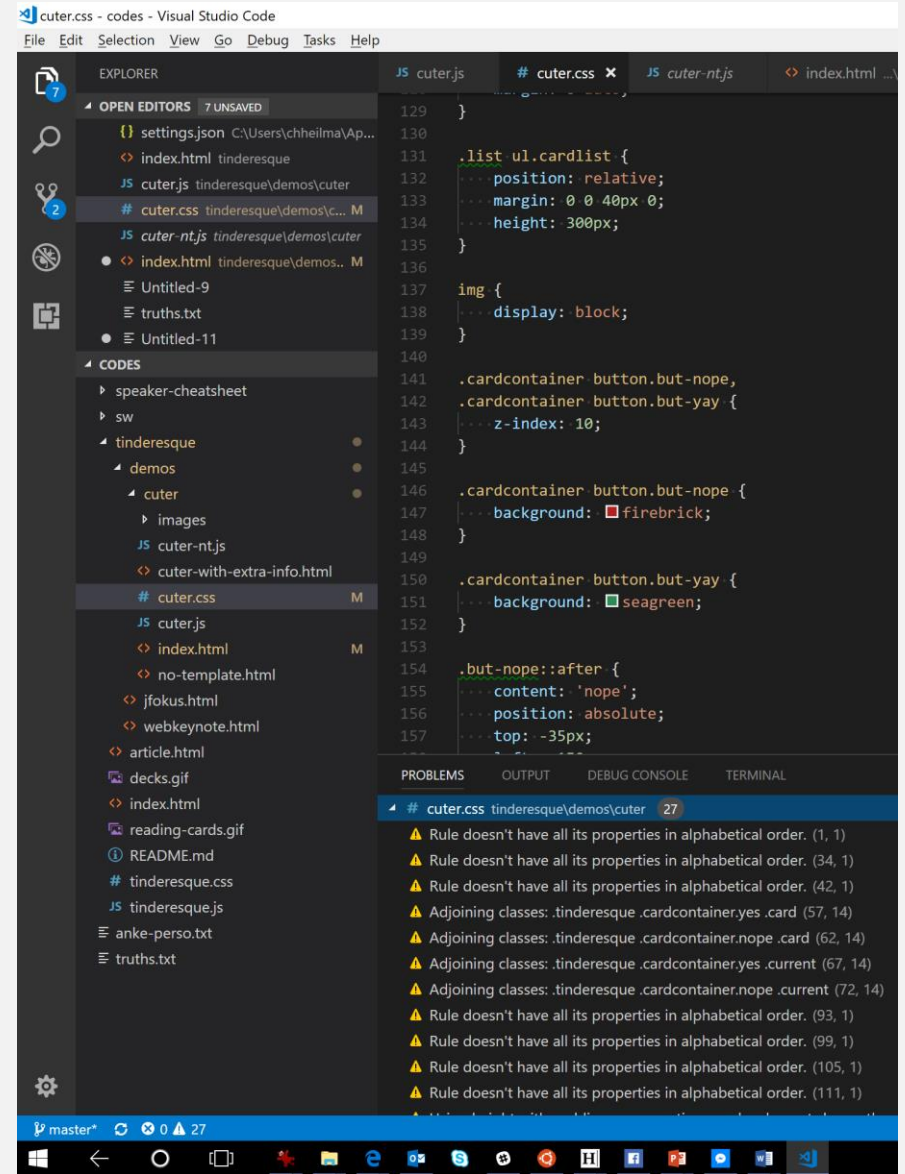
Customisable best practices in context



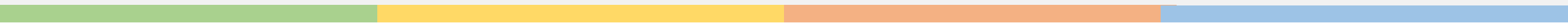
- Having information isn't enough when people don't go there
- We need to prevent mistakes before they happen
- We need to allow people to customize these experiences. An intranet site needs different settings than a marketing site
- So, let's target editors and build processes



Linting and inline insights in editors



\$ cd ..





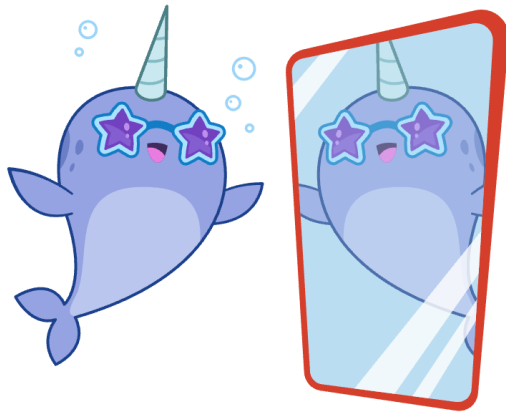
webhint.io

<https://github.com/webhintio>

<https://webhint.io>

<https://www.npmjs.com/package/hint>

Webhint is open, and needs your help...



FULLY CUSTOMIZABLE

Every site is different. webhint adapts its feedback when you give it more information: [ignore 3rd-party code](#), [prioritize *your* users' browsers](#), and [control the results](#) with minimal setup.

[USER GUIDE](#)



CREATE YOUR OWN HINTS

With the help of our [contributor guide](#), you can [create new hints](#) to suit *your* needs. You can help webhint help even more people like you by contributing your hints back!

[HINT DOCUMENTATION](#)



COMMUNITY DRIVEN

webhint welcomes anyone who wants to make the web a better place. Testing, [filing issues and feature requests](#), [contributing code](#), and improving the documentation are just the start!

[WEBHINT GITHUB](#)

SCANNING 100%

SCAN TIME: 01:54

HINTS

URL: <http://techdaysfi.com/>

DATE: 2019-02-25 11:19

362

YOUR SCAN RESULT LINK: <https://webhint.io/scanner/690eaa75-06b4-4393-8c99-8e76bbc80bf4>

webhint version: 4.3.1 Configuration JSON

Security

expand all

- disown-opener: 6 hints
- no-disallowed-headers: 3 hints
- no-protocol-relative-urls: 4 hints
- sri: 24 hints
- strict-transport-security: 57 hints
- validate-set-cookie-header: 5 hints
- x-content-type-options: 43 hints
- no-vulnerable-javascript-libraries: 1 hints

COMPATIBILITY	HINTS 68	PASSED 6/7
PWA	HINTS 1	PASSED 3/4
PERFORMANCE	HINTS 138	PASSED 1/7
PITFALLS	HINTS 0	PASSED 0/0
SECURITY	HINTS 144	PASSED 0/9

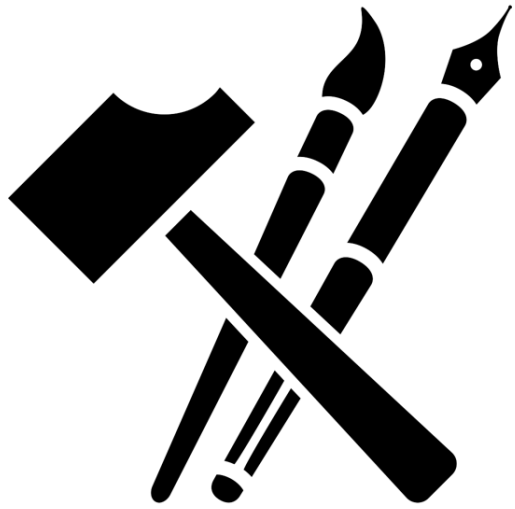


Adding experience
to build processes...



- A web interface or browser dev tools to get reports doesn't scale to production
- Why not add best practices as part of your build/deployment process?
- That way you can also test the setup of your bundler/preprocessor to avoid huge release sizes

```
> npx hint https://example.com  
└─ https://example.com/ downloaded
```



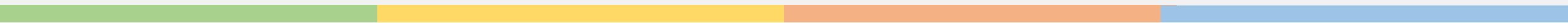
Customise the tests
to your needs...



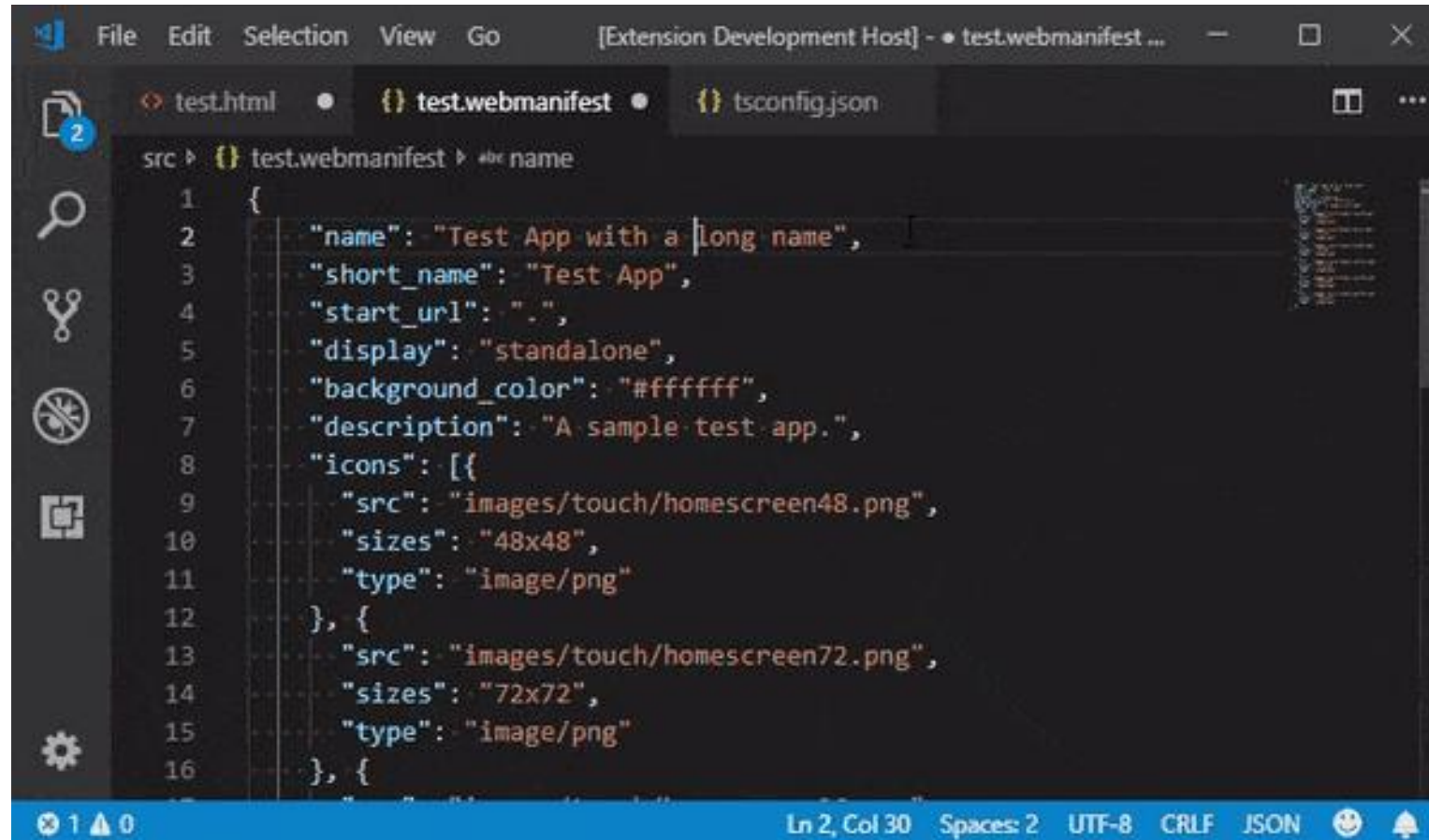
- Once you know what your product's needs are, you can turn different tests on and off
- You can also write your own tests to become part of the scan – like “make sure every footer has a picture of a leprechaun” – just one example
- You can create these with a cli wizard:

```
$ npm create hintrc
```

\$ cd ..



webhint @code extension



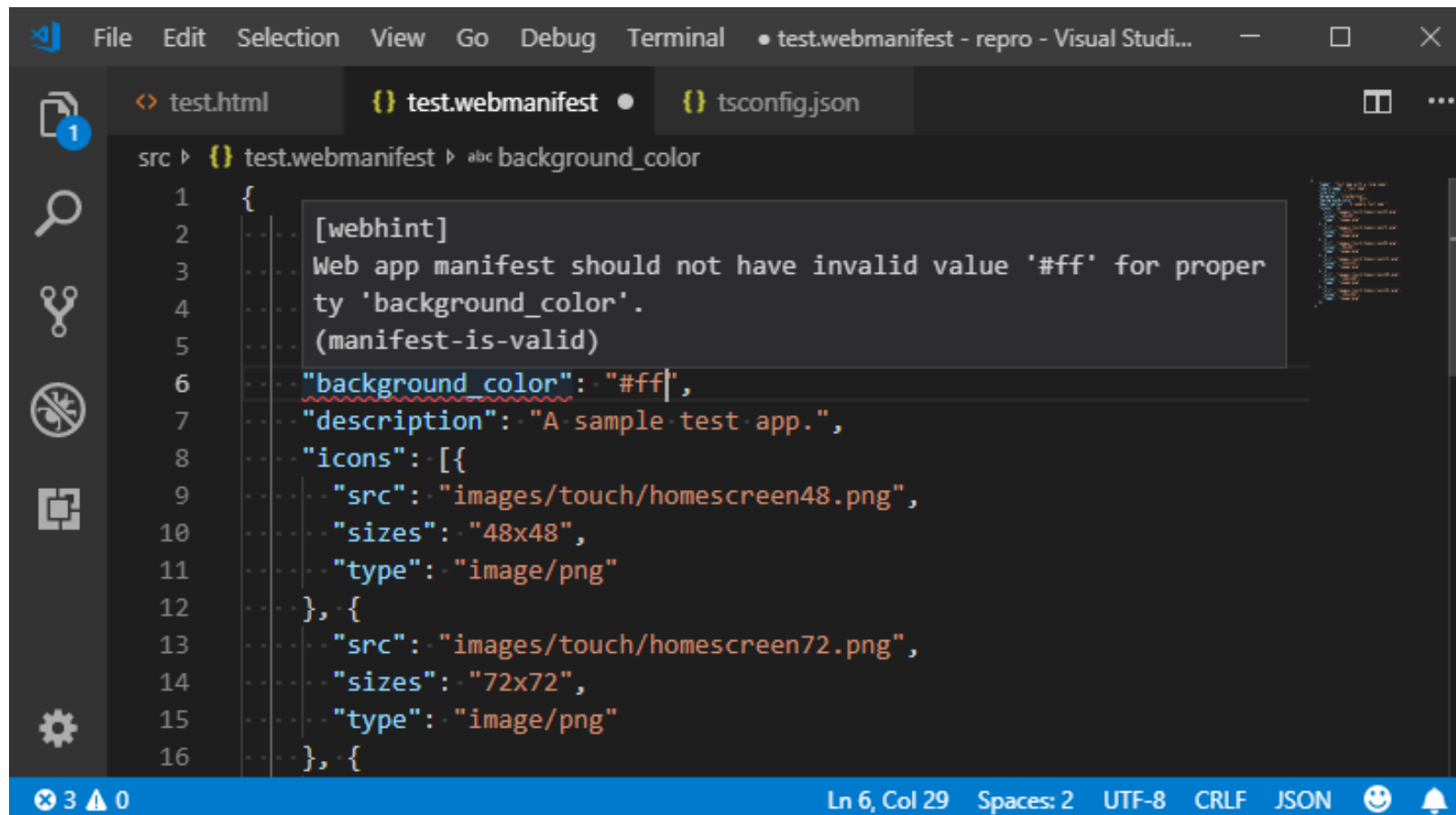
The screenshot shows the Visual Studio Code editor interface. The active file is `test.webmanifest`, which is a JSON file. The editor displays the following JSON content:

```
1 {
2   "name": "Test App with a long name",
3   "short_name": "Test App",
4   "start_url": ".",
5   "display": "standalone",
6   "background_color": "#ffffff",
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  }, {
```

The status bar at the bottom indicates the current position is `Ln 2, Col 30`, with `Spaces: 2`, `UTF-8`, `CRLF`, and `JSON` encoding.

<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 active file is `test.webmanifest` located in the `src` directory. The code defines a web manifest with the following structure:

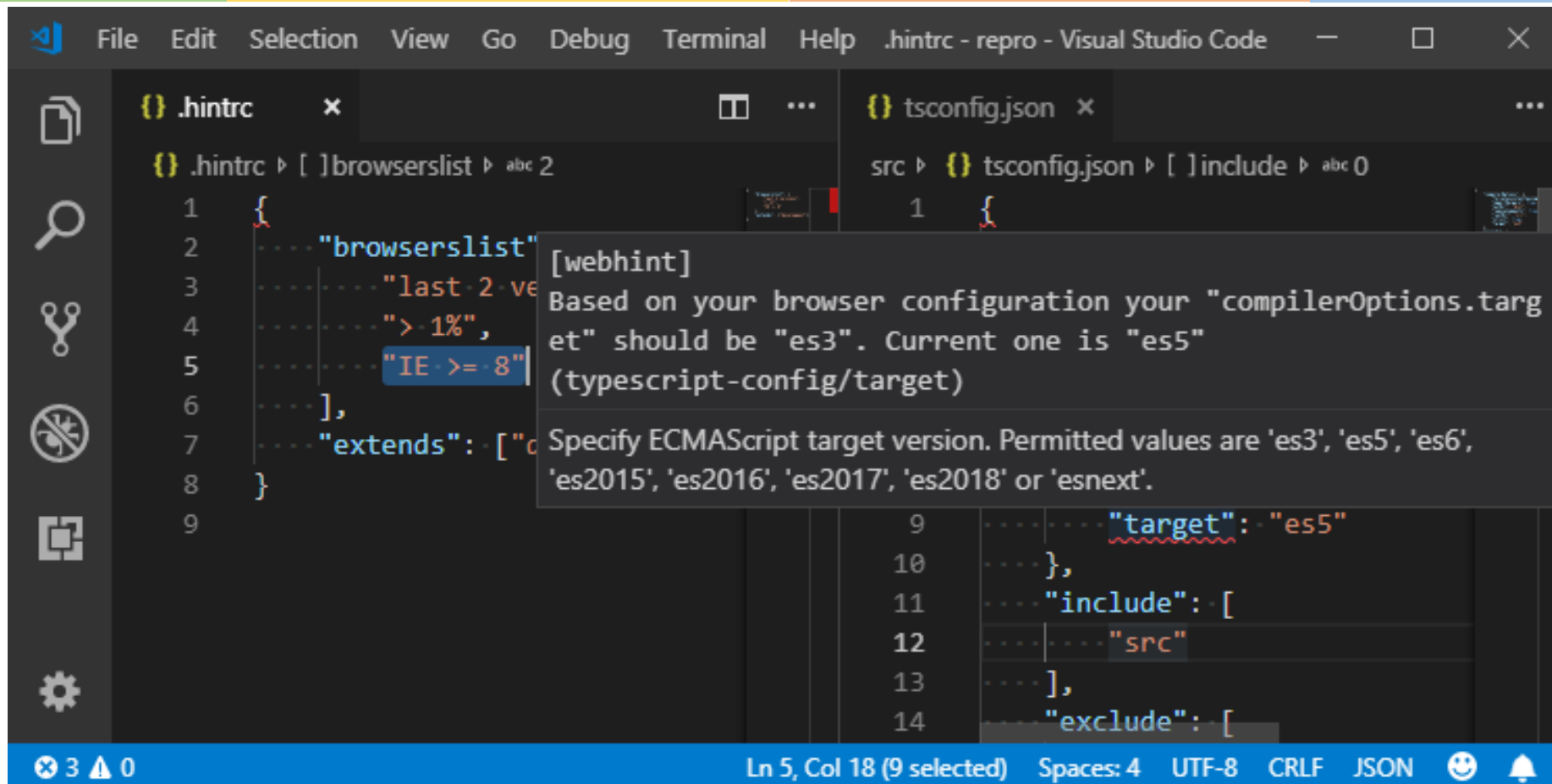
```
1 {
2   [webhint]
3   Web app manifest should not have invalid value '#ff' for proper
4   ty 'background_color'.
5   (manifest-is-valid)
6   "background_color": "#ff",
7   "description": "A sample test app.",
8   "icons": [{
9     "src": "images/touch/homescreen48.png",
10    "sizes": "48x48",
11    "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 error, 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 of the editor shows the current cursor position as `Ln 6, Col 29`, along with other settings like `Spaces: 2`, `UTF-8`, `CRLF`, and `JSON`.

Fixing your tooling setup...



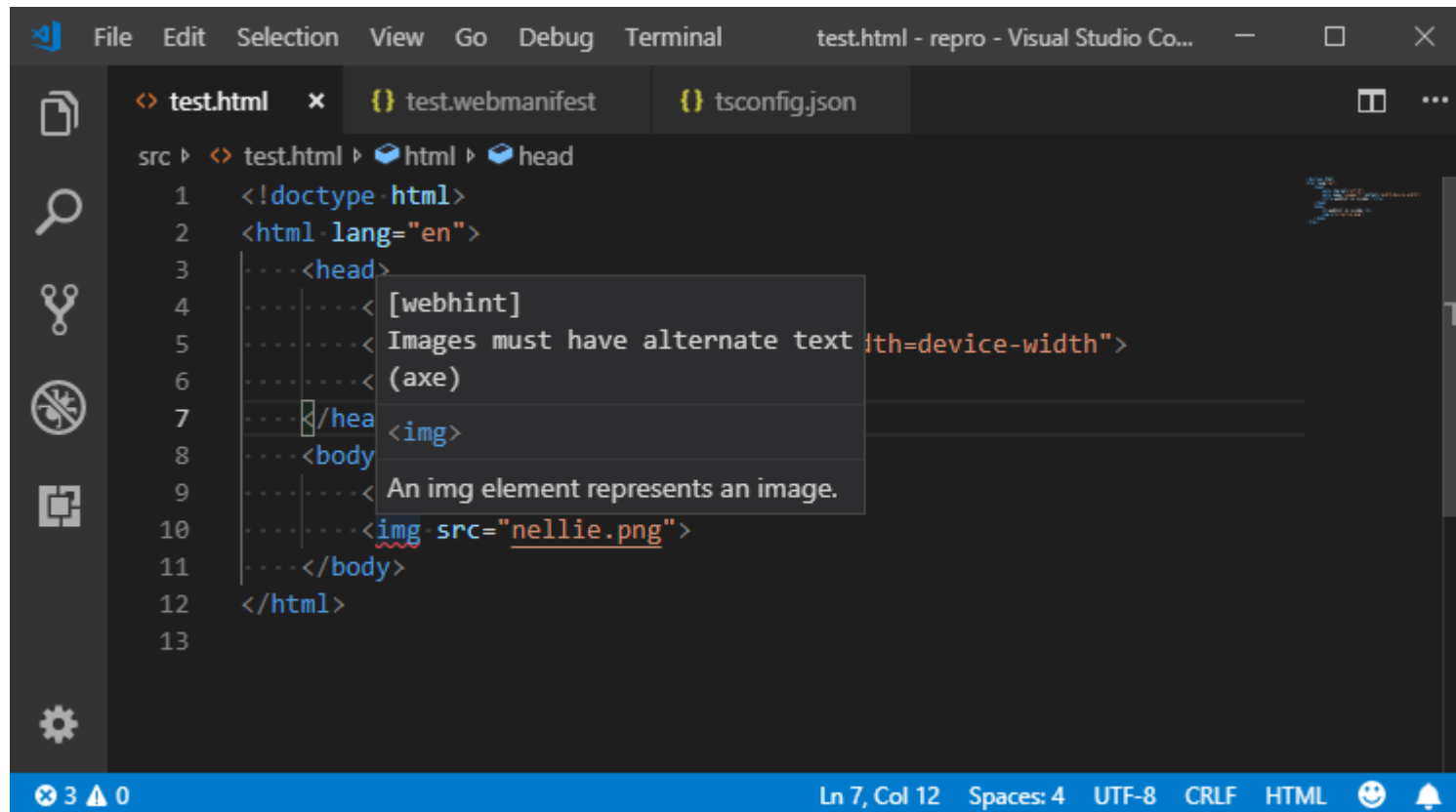
The screenshot shows the Visual Studio Code interface with two files open: `.hintrc` and `tsconfig.json`. The `.hintrc` file contains a browser configuration for Internet Explorer 8. The `tsconfig.json` file has a `target` property set to `"es5"`. A webhint tooltip is displayed over the `target` property, providing the following information:

```
[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'.
```

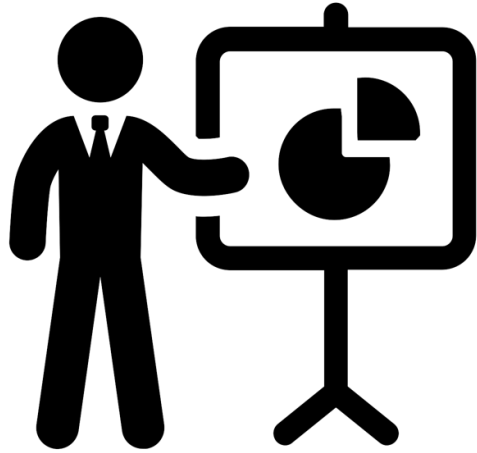
The status bar at the bottom indicates the current cursor position is at line 5, column 18, with 9 characters selected. The file encoding is UTF-8, line endings are CRLF, and the file type is JSON.

Are you creating inaccessible content?



```
File Edit Selection View Go Debug Terminal test.html - repro - Visual Studio Co...
test.html x test.webmanifest tsconfig.json
src > test.html > html > head
1 <!doctype html>
2 <html lang="en">
3   <head>
4     [webhint]
5     Images must have alternate text (axe)
6     <img alt="device-width">
7   </head>
8   <body>
9     
10    An img element represents an image.
11  </body>
12 </html>
13
```

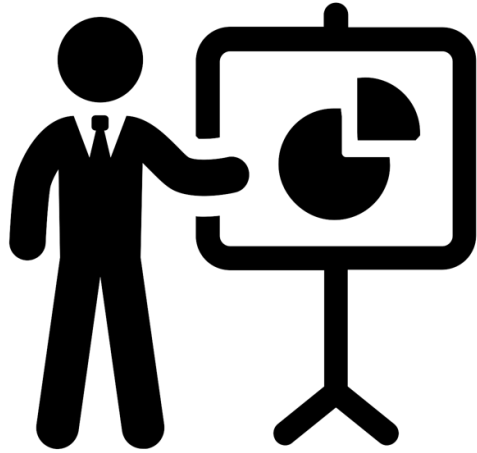
Ln 7, Col 12 Spaces: 4 UTF-8 CRLF HTML



In summary...



- We accumulated a lot of knowledge over the years what software should do for users and how to make sure it does
- We suck at making sure busy and new developers learn about it
- We know, however, that people are happy to use open editors, and node as a platform
- Instead of whining that people don't follow best practices, let's make them part of the build process



In summary...

- We're past a stage where developers have the chance to learn the trade before applying it
- Our editors are much more than dumb text writing tools and not as overwhelming as huge IDEs.
- We tend to build products using a toolchain we customise to the need of the product
- Both are an opportunity to instill a sense of ownership to developers and to learn about our mistakes before our users suffer from them.

Thanks!

Chris Heilmann
@codepo8

