



JavaScript
fwdays

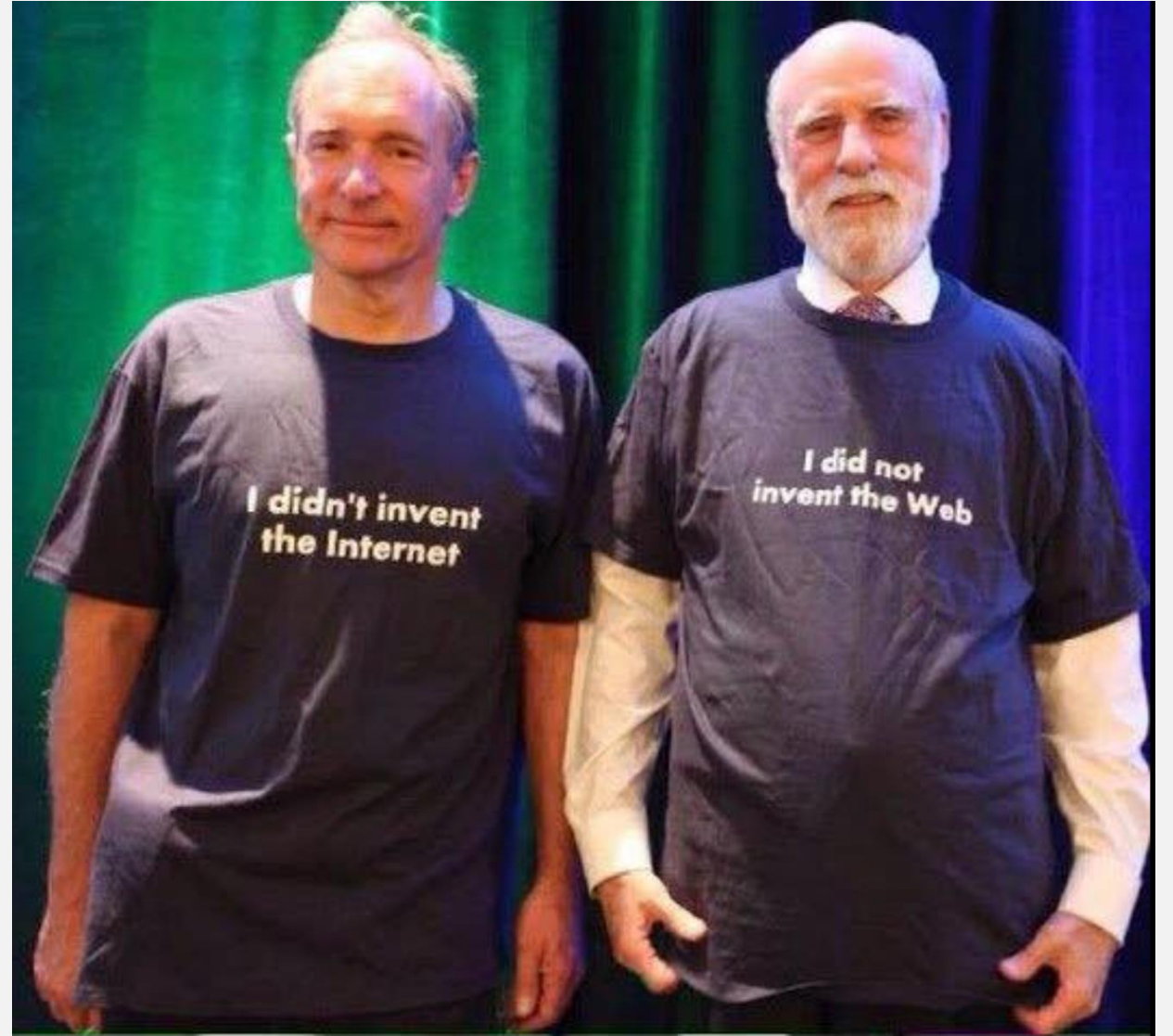
Hinting at a better web

Chris Heilmann

Microsoft



Happy 30th birthday,
World Wide Web!

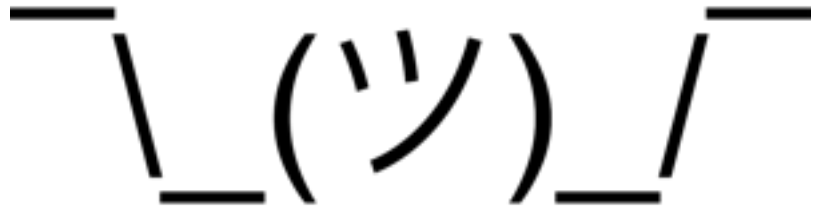




Happy 30th birthday,
World Wide Web!



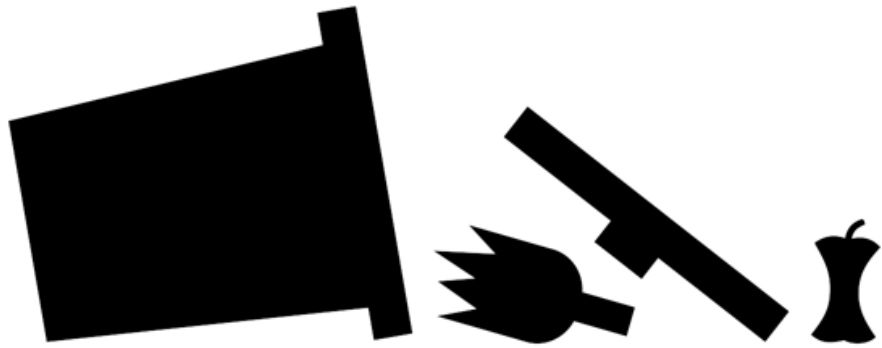
- Excellent publishing medium
- Low barrier to entry
- Dumb, sturdy technologies and ideas - reliability beats finesse



The need to be
forgiving...



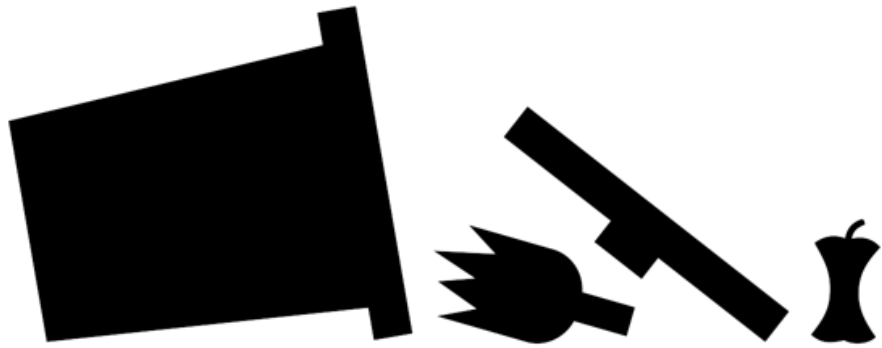
- Anyone is allowed to write code for the web – or use whatever they want to create that code.
- Users shouldn't suffer from developer errors; user agents need to be forgiving (RIP XHTML).
- What you put on the web will not be consumed the way you intended it to – your products need to be flexible.



World Wide Mess



- This freedom meant people took liberties...
- As things “just work”, we spent years adding suboptimal products to the web.
- These legacy products are likely to never get fixed – maintenance is boring.
- Browsers are big and slow because they need to show the web of yesterday, today and tomorrow.



World Wide Mess



Katie Hempenius

@katiehempenius

Following

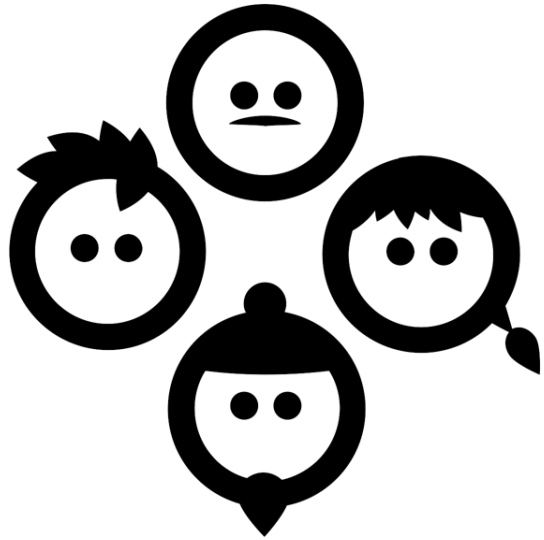


Page Weight Percentiles: overall & by resource type

If one of these categories is super high for a site, it's often an indicator that there's "low hanging fruit" performance wins to be had by optimizing that resource type.

(Data source: [@HTTPArchive](#), Desktop Oct'18 data)

	Overall	Images	JS	Font	CSS	HTML
p90	5.8 MB	4 MB	1.2 MB	290 KB	200 KB	130 KB
p75	3 MB	1.8 MB	780 KB	170 KB	100 KB	60 KB
p50	1.5 MB	650 KB	420 KB	100 KB	50 KB	30 KB
p25	700 KB	180 KB	200 KB	50 KB	20 KB	10 KB
p10	280 KB	40 KB	90 KB	30 KB	10 KB	10 KB



The web is made
for humans...

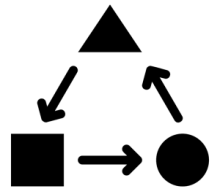


- Control of content is in the hand of the consumer
- We build things for people to consume in a simple fashion
- We build things for people to do tasks with – our job is to make that easy
- Humans are not in a binary state, we change over time and have different needs

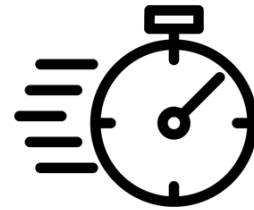
On the web, we should care about...



Accessibility



Interoperability

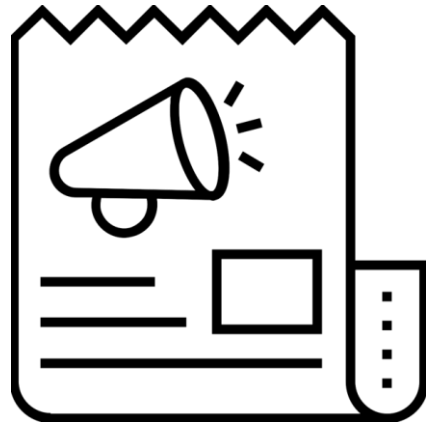


Performance

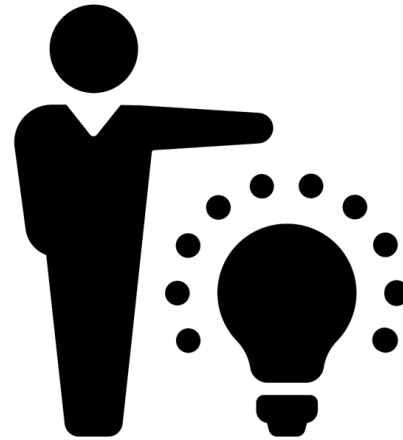


Security

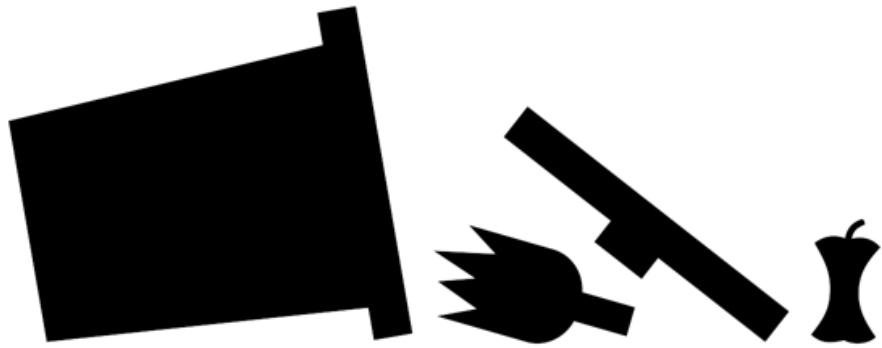
What we do care about is...



Time to market



Constant innovation



The web is an
accessibility mess...



The WebAIM Million

An accessibility analysis of the top 1,000,000 home pages

[Home](#) > [WebAIM Projects](#) > The WebAIM Million

Introduction

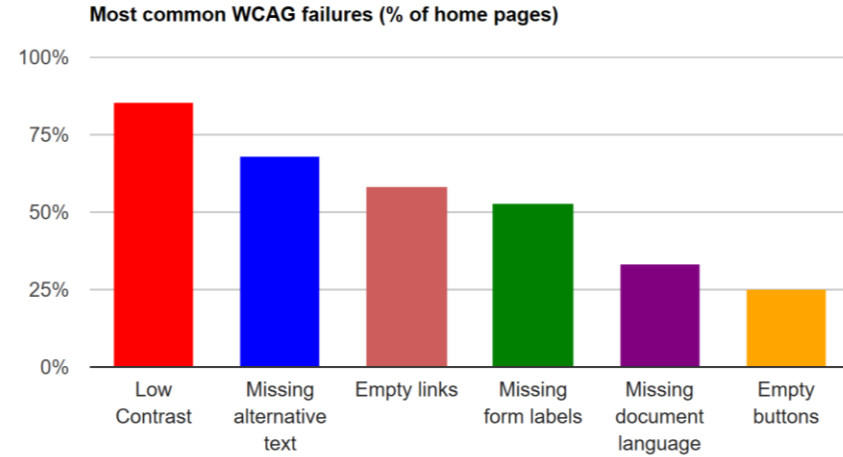
In February 2019, WebAIM conducted an evaluation of the home pages for the top 1,000,000 web sites using the [WAVE stand-alone API](#) (with additional tools to collect site technology parameters). While this research focuses only on automatically detectable issues, the results paint a rather dismal picture of the current state of web accessibility.

<https://majestic.com/reports/majestic-million>

<https://webaim.org/projects/million/>



Rampant accessibility failures...



Most common types of WCAG 2 failures

WCAG Failure Type	# of home pages	% of home pages
Low contrast text	852,868	85.3%
Missing alternative text for images	679,964	68%
Empty links	581,408	58.1%
Missing form input labels	528,482	52.8%
Missing document language	329,612	33.1%
Empty buttons	250,367	25%

<https://majestic.com/reports/majestic-million>

<https://webaim.org/projects/million/>



Problems caused by framework usage



JavaScript Frameworks

Framework	# (and %) of home pages	Avg. # of errors	% decrease/increase of errors
MooTools	27,033 (2.7%)	52.3	-12.6%
TweenMax	25,802 (2.6%)	58.3	-2.3%
Mustache	11,803 (1.2%)	65.5	9.9%
Knockout.js	7,046 (0.7%)	65.6	10.0%
React	31,960 (3.2%)	65.5	10.1%
Backbone.js	22,678 (2.3%)	69.3	16.6%
Angular	14,939 (1.5%)	70.9	19.2%
RequireJS	21,168 (2.1%)	71.1	19.7%
Handlebars	15,307 (1.5%)	73.5	23.6%
Vue.js	6,639 (0.7%)	73.7	23.7%
Prototype	14,376 (1.4%)	74.2	24.8%

<https://majestic.com/reports/majestic-million>

<https://webaim.org/projects/million/>



Problems caused by library usage



JavaScript Libraries

Library	# (and %) of home pages	Avg. # of errors	% decrease/increase of errors
Zepto	7,708 (0.8%)	44.0	-26.4%
YUI	20,136 (2.0%)	52.7	-11.9%
jQuery Migrate	313,391 (31.3%)	61.7	5.1%
Modernizr	197,260 (19.7%)	63.2	7.5%
Flickity	8,598 (0.9%)	64.9	8.8%
SoundManager	5,671 (0.6%)	64.9	8.8%
prettyPhoto	32,684 (3.3%)	65.4	10.0%
Lightbox	53,171 (5.3%)	65.8	10.9%
Dojo	7,111 (0.7%)	66.4	11.5%
Hammer.js	22,961 (2.3%)	67.4	13.3%
Lodash	17,219 (1.7%)	68.2	14.7%
Select2	22,194 (2.2%)	74.0	24.7%
jQueryUI	187,057 (18.7%)	71.4	25.3%
Fancybox	50,554 (5.1%)	73.8	25.3%
Script.aculo.us	8,825 (0.9%)	75.0	25.9%
Underscore.js	27,457 (2.7%)	74.7	26.1%
Moment.js	35,028 (3.5%)	75.6	28.0%
Slick	39,866 (4.0%)	79.8	35.7%
Fingerprintjs	8,864 (0.9%)	81.8	37.6%
jQuery	774,058 (77.4%)	64.0	43.3%
Lazy.js	15,884 (1.6%)	192.8	235.3%

<https://majestic.com/reports/majestic-million>

<https://webaim.org/projects/million/>



Problems caused by library usage



“ The vast majority of the top one million home pages utilize jQuery. **Home pages with jQuery averaged 19.2 more errors than those without jQuery.** The presence of jQuery corresponds with nearly **15 million detected errors**, or over **25%** of all of the accessibility errors we detected.

<https://majestic.com/reports/majestic-million>

<https://webaim.org/projects/million/>

¯_(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] · 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
```

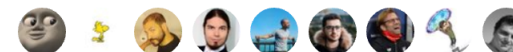
doesn't work (0 → 01 → 011 → ...), but

```
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"
```



8



77



423





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
```

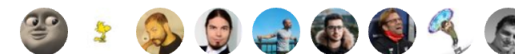
doesn't work (0 → 01 → 011 → ...), but

```
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

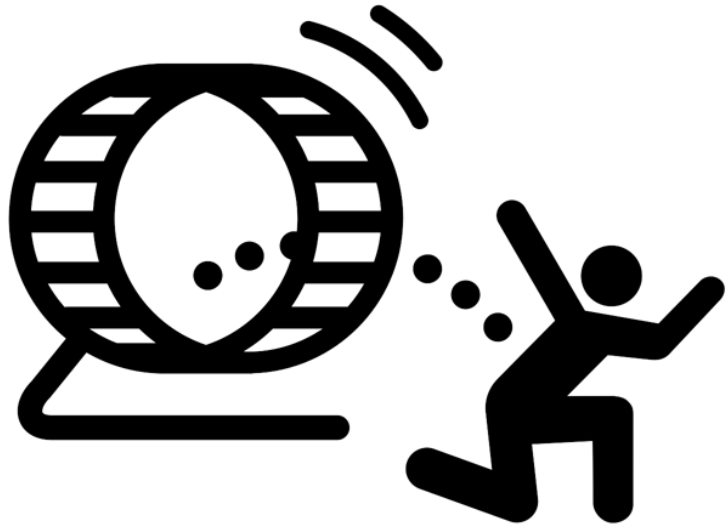




And we keep busy...



- It is easy to feel out of the loop
- Often we pad our products with solutions because we want to be seen using them – as they are “professional”
- It doesn’t feel enough to build a sturdy product – it needs to be clever
- We overcomplicate our job requirements and block prospective new talent

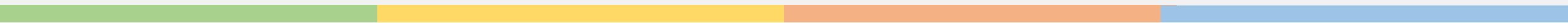


Let's take a breath
and think web...

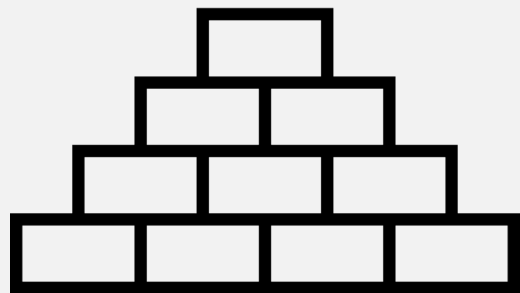


- It is not about our convenience as developers or our business plans
- It is about creating content and tools for human consumption
- The good news is that in our current environment, we have a great platform for that

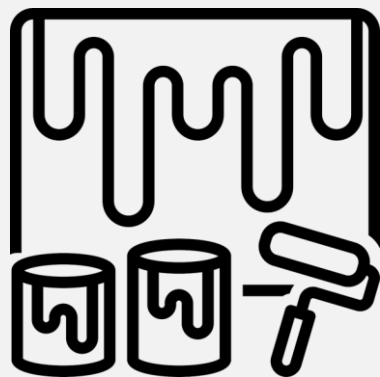
\$ cd ..



HTML

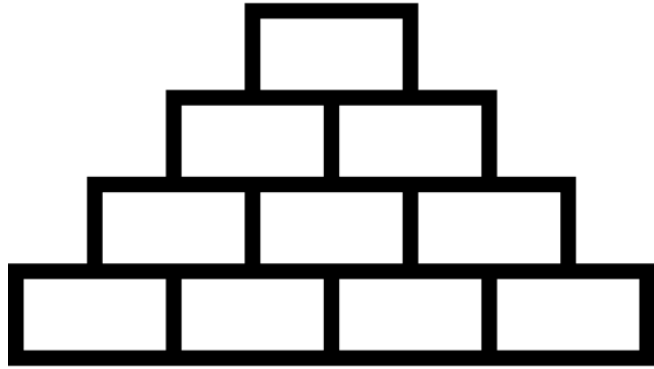


CSS



JavaScript

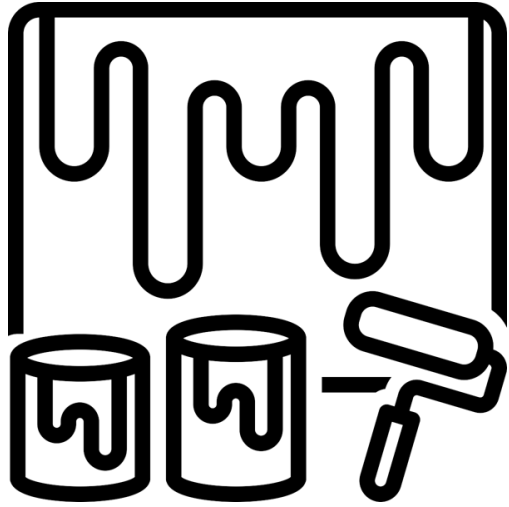




HTML



- Is the foundation of any web interface
- HTML elements come with lots of free accessibility and interaction features
- Static HTML can be cached and compresses down to almost nothing
- You can't go wrong by betting on HTML these days



CSS



- Allows you to style elements and give them a certain look and feel
- Is a fully-fledged layout engine for complex interfaces with unknown content and reading direction (Flexbox / Grid)
- Is an animation/transition engine that allows you to create performant animations that the browser controls for you
- Has variables and conditions
- Can react to changes in the environment.

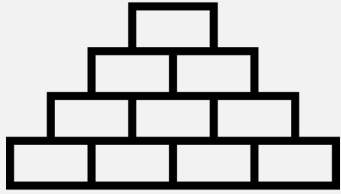


JavaScript

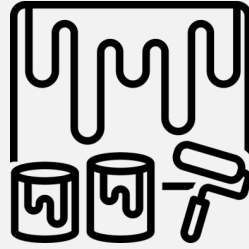


- Allows you to do everything, even creating the other two technologies
- Can be used on the server, on the client and in applications – each of which follows different rules
- Puts you in control and makes you responsible for performance, maintainability and reliability
- Is finally standardized, but also fast moving and easy to lose track

HTML



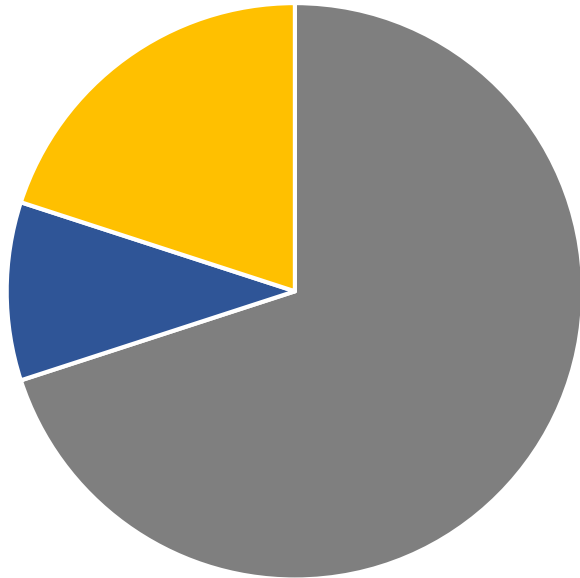
CSS



JavaScript

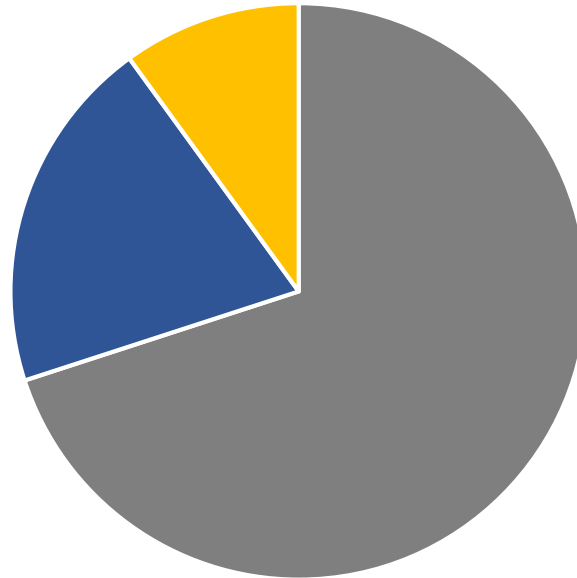


Responsibilities



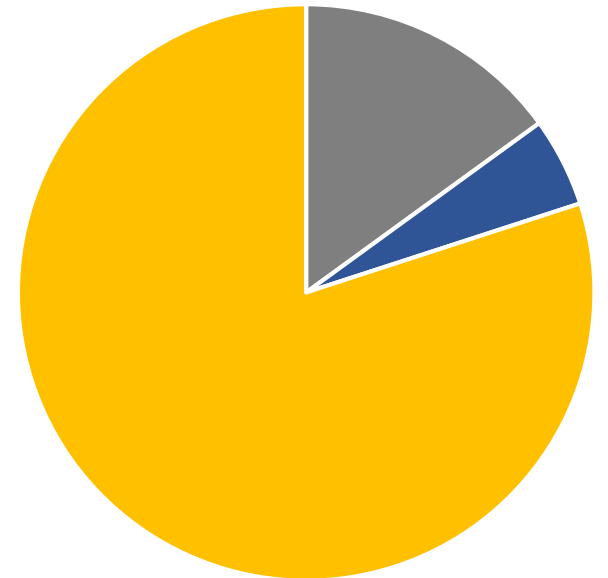
■ Browser ■ Standard ■ You

Responsibilities



■ Browser ■ Standard ■ You

Responsibilities



■ Browser ■ Standard ■ You



We live in great tooling times...

- We have evergreen browsers with in-built development tools
- We have virtualized testing built into browsers (mobile view, simulated connections)
- We have community channels to share our code and solutions and ways to discuss our approaches
- We have excellent, up-to-date documentation



So why don't
we build great
solutions?

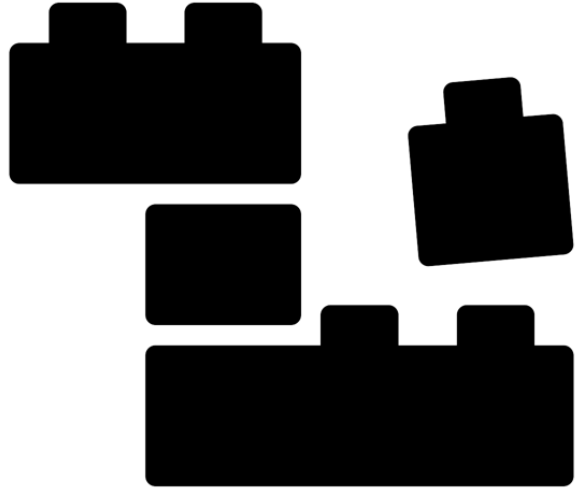
- Resources with upvoting options favour the how over the why
- We're not paid to learn, we're paid to deliver
- Peer pressure makes developers who don't know things afraid to admit it
- There's always a "good enough" way to create a lot in a short amount of time.



Web Developers
need to know a lot...




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



The web is
assembled, not
authored...



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

By @paulcalvano w/ @HTTPArchive data

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





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

% of Third 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...

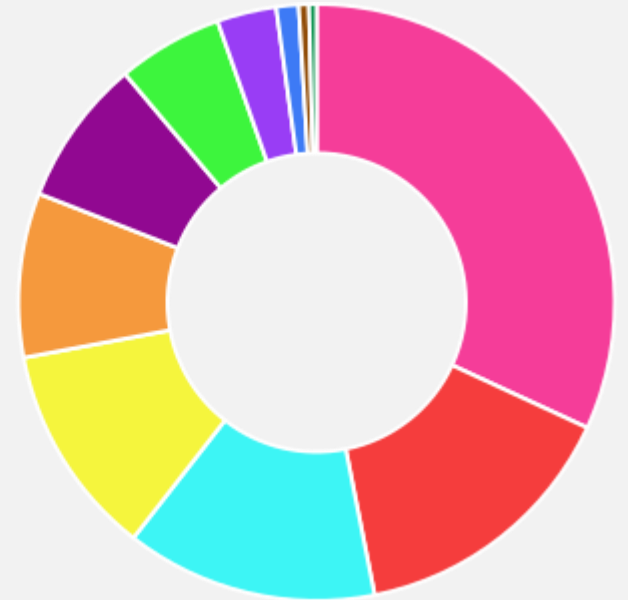
discuss.httparchive.org

  30  61 

<https://discuss.httparchive.org/t/what-is-the-distribution-of-1st-party-vs-3rd-party-resources/100/14>



Third party slowness and cruft...





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.

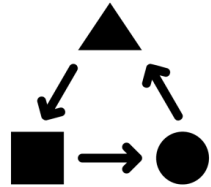
This is 'normal' in modern web development.

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

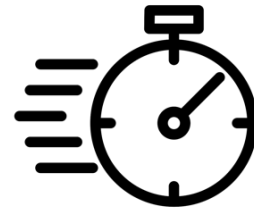
Can you say that your product is OK with...



Accessibility



Interoperability

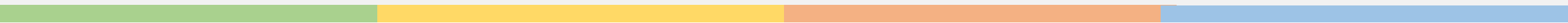


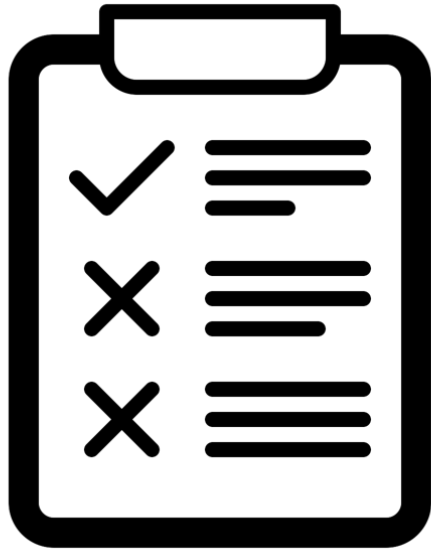
Performance



Security

Let's talk tools...





Browser tool audits...



Audits


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

 Device

- Mobile
- Desktop

Audits

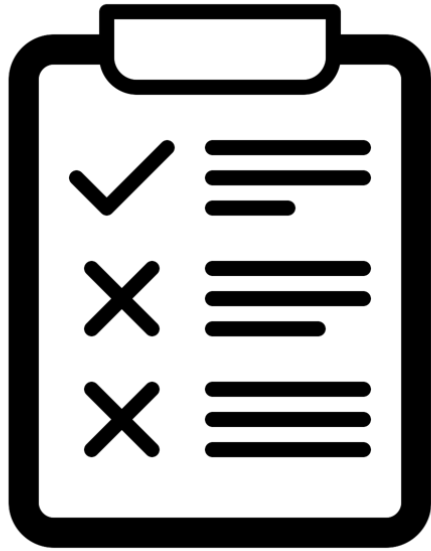
- Performance
- Progressive Web App
- Best practices
- Accessibility
- SEO

 Throttling

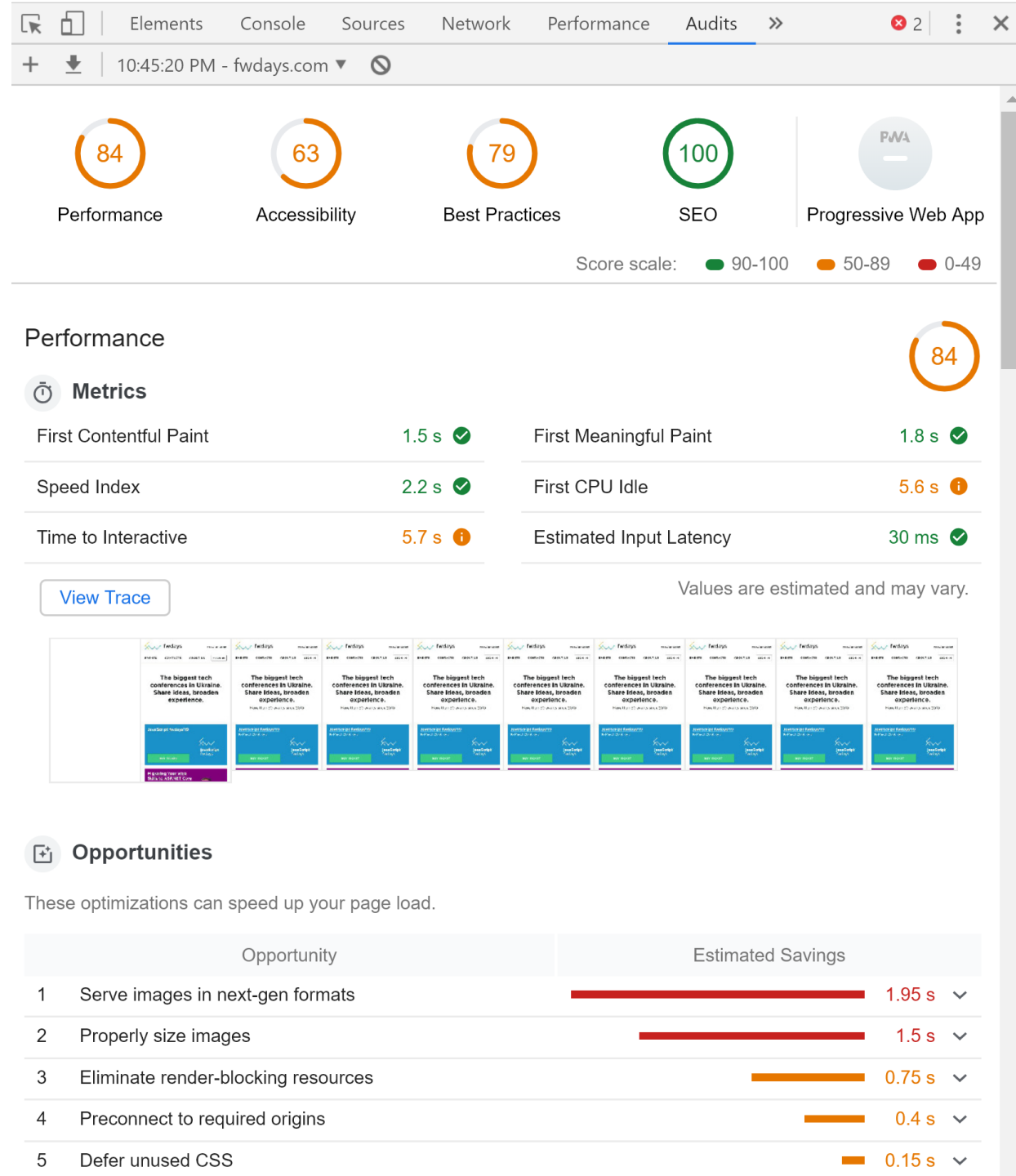
- Simulated Fast 3G, 4x CPU Slowdown
- Applied Fast 3G, 4x CPU Slowdown
- No throttling

Clear storage

Run audits



Auditing report





Accessibility Insights



- Browser extension for Chrome or Windows app – depending on what you want to test
- Open source collaboration of Deque and Microsoft
- Community driven with Github as the communication channel
- Customisable to your bug reporting tool

<https://accessibilityinsights.io/>



Accessibility Insights



Accessibility Insights for Web



[Watch 3-minute video introduction](#)

Launch pad



FastPass

Run two tests to find the most common accessibility issues in less than 5 minutes.



Assessment

Walk through a guided process for assessing accessibility compliance.



Ad hoc tools

Get quick access to visualizations that help you identify accessibility issues.

<https://accessibilityinsights.io/>

- Overview
0% Completed
- 1 Automated checks
- 2 Keyboard
- 3 Focus
- 4 Landmarks
- 5 Headings
- 6 Repetitive content
- 7 Links
- 8 Native widgets
- 9 Custom widgets
- 10 Timed events
- 11 Errors
- 12 Page navigation
- 13 Parsing
- 14 Images
- 15 Language
- 16 Sensory
- 17 Text legibility
- 18 Audio / video

OVERVIEW

This page contains a summary that indicates the progress of your assessment. An assessment is a manual experience in which you navigate through a set of tests that cover all WCAG 2.0 AA success criteria. Each test has one or more requirements that can be:

- Automated
- Assisted
- Manual

Where possible, Accessibility Insights for Web "assists" the test process by generating a list of instances to evaluate and highlighting them on the screen. Accessibility Insights for Web also allows you to manually record failure instances.

Summary



Test details

Automated checks	<input checked="" type="checkbox"/> 48	Page navigation	○○○
Keyboard	○○○○○	Parsing	○
Focus	○○○○○	Images	○○○○
Landmarks	○○○	Language	○○○
Headings	○○○	Sensory	○○○○
Repetitive content	○○○	Text legibility	○○○
Links	○○	Audio / video	○○
Native widgets	○○○○	Multimedia	○○○○○
Custom widgets	○○○○○○	Live multimedia	○

Help

- [Getting started](#)
- [How to complete a test](#)
- [Ask a question](#)

Target page: [Fwdays. IT tech conferences in Ukraine, Kyiv](#)

1 Automated checks

2 Tab stops

Automated checks

Show failures On [↗ Export result](#)

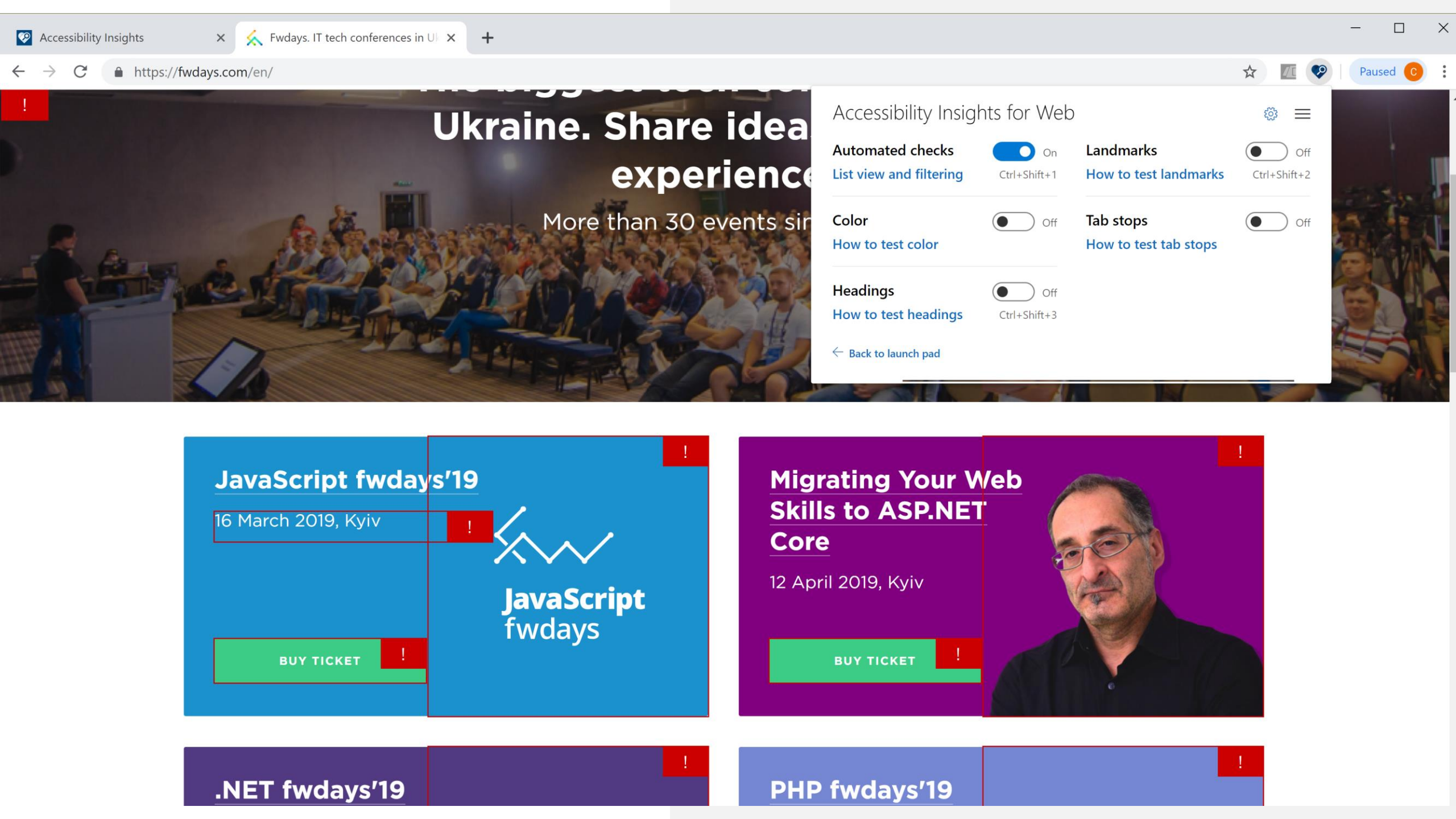
✖ 20 failures were detected. When the Automated Checks toggle is on, failure instances selected in this page are highlighted in the target page.

<input checked="" type="checkbox"/>	>	Path	Snippet	How to fix
<input checked="" type="checkbox"/>	>	color-contrast : Elements must have sufficient color contrast (✖ 9) WCAG 1.4.3		
<input checked="" type="checkbox"/>	>	frame-title : Frames must have title attribute (✖ 1) WCAG 2.4.1		
<input checked="" type="checkbox"/>	>	link-name : Links must have discernible text (✖ 7) WCAG 2.4.4 , WCAG 4.1.2		
<input checked="" type="checkbox"/>	>	meta-viewport : Zooming and scaling must not be disabled (✖ 1) WCAG 1.4.4		
<input checked="" type="checkbox"/>	>	unique-landmark : Landmarks must have a unique role or role/label combination (aria-label OR aria-labelledby) (✖ 2) WCAG 2.4.1		



Failure details

Select a single failure instance from a group in the table above to see more details here.



Ukraine. Share ideas experience

More than 30 events since

Accessibility Insights for Web

Automated checks On
[List view and filtering](#) Ctrl+Shift+1

Landmarks Off
[How to test landmarks](#) Ctrl+Shift+2

Color Off
[How to test color](#)


Tab stops Off
[How to test tab stops](#)

Headings Off
[How to test headings](#) Ctrl+Shift+3

[← Back to launch pad](#)

JavaScript fwdays'19

16 March 2019, Kyiv




**JavaScript
fwdays**

BUY TICKET

**Migrating Your Web
Skills to ASP.NET
Core**

12 April 2019, Kyiv



BUY TICKET

.NET fwdays'19

PHP fwdays'19

Ukraine. Share ideas, broaden experience.

More than 30 events since 2010

Links must have discernible text

Rule name: `link-name`

Path:

`..event-card__photo > a[href$="js-fwdays-2019"]`

Inspect HTML

Copy failure details

File issue

To enable the Inspect HTML button, open the Chrome dev tools (F12).

Fix the following:

Element is in tab order and does not have accessible text

Fix ONE of the following:

- Element does not have text that is visible to screen readers
- aria-label attribute does not exist or is empty
- aria-labelledby attribute does not exist, references elements that do not exist or references elements that are empty
- Element's default semantics were not overridden with `role="presentation"`
- Element's default semantics were not overridden with `role="none"`

< Back

Failure 1 of 1 for this target

Next >

JavaScript fwdays'19

16 March 2019, Kyiv

BUY TICKET

Web

T



NET fwdays'19

DLD fwdays'19

Ukraine. Share ideas, broaden experience.

More than 30 events since 2010

Elements must have sufficient color contrast

✖ Rule name: [color-contrast](#)

Path:
#card-js-fwdays-2019

Inspect HTML Copy failure details File issue

To enable the Inspect HTML button, open the Chrome dev tools (F12).

Fix the following:
Element has insufficient color contrast of 2.06 (foreground color: #ffffff, background color: #3dcc85, font size: 9.0pt, font weight: bold). Expected contrast ratio of 4.5:1

< Back Failure 1 of 1 for this target Next >

JavaScript fwdays'19

16 March 2019, Kyiv

BUY TICKET

Web T




BUY TICKET

.NET fwdays'19

PHP fwdays'19

JavaScript fwdays'19

16 March 2019, Kyiv



JavaScript fwdays

BUY 1 TKT

Annotation 2: points to the JavaScript logo

Migrating Your Web Skills to P.NET Core

12 April 2019, Kyiv




BUY 4 TKT

Annotation 3: points to the title 'P.NET Core'

Annotation 5: points to the speaker's face

.NET fwdays'19

13 April 2019, Kyiv



.NET fwdays


BUY 7 TKT

Annotation 6: points to the title '.NET fwdays'19'

Annotation 8: points to the .NET logo

PHP fwdays'19

1 June 2019, Kyiv



PHP fwdays

BUY 10 TKT

Annotation 9: points to the title 'PHP fwdays'19'

Annotation 10: points to the 'BUY 10 TKT' button

Tech Trends fwdays'19

Highload fwdays'19



The biggest tech conferences in Ukraine. Share ideas, broaden experience.

H1

More than 30 events since 2010

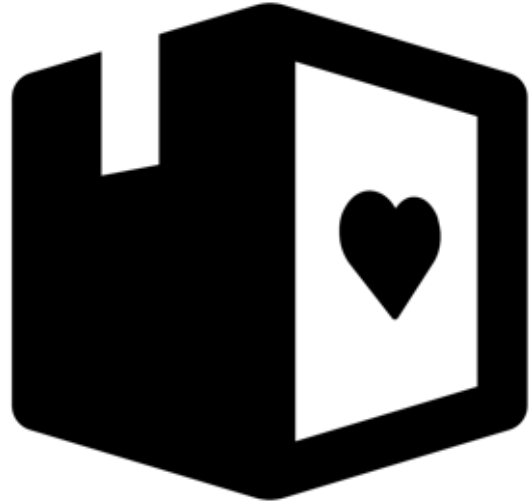
JavaScript fwdays'19 H2

16 March 2019, Kyiv

Migrating Your Web Skills to ASP.NET H2



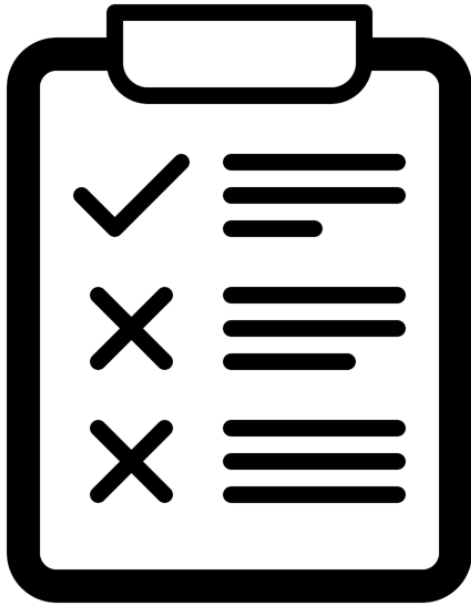
Accessibility Insights for Web Start pressing Tab to start visualizing tab stops. Google Chrome



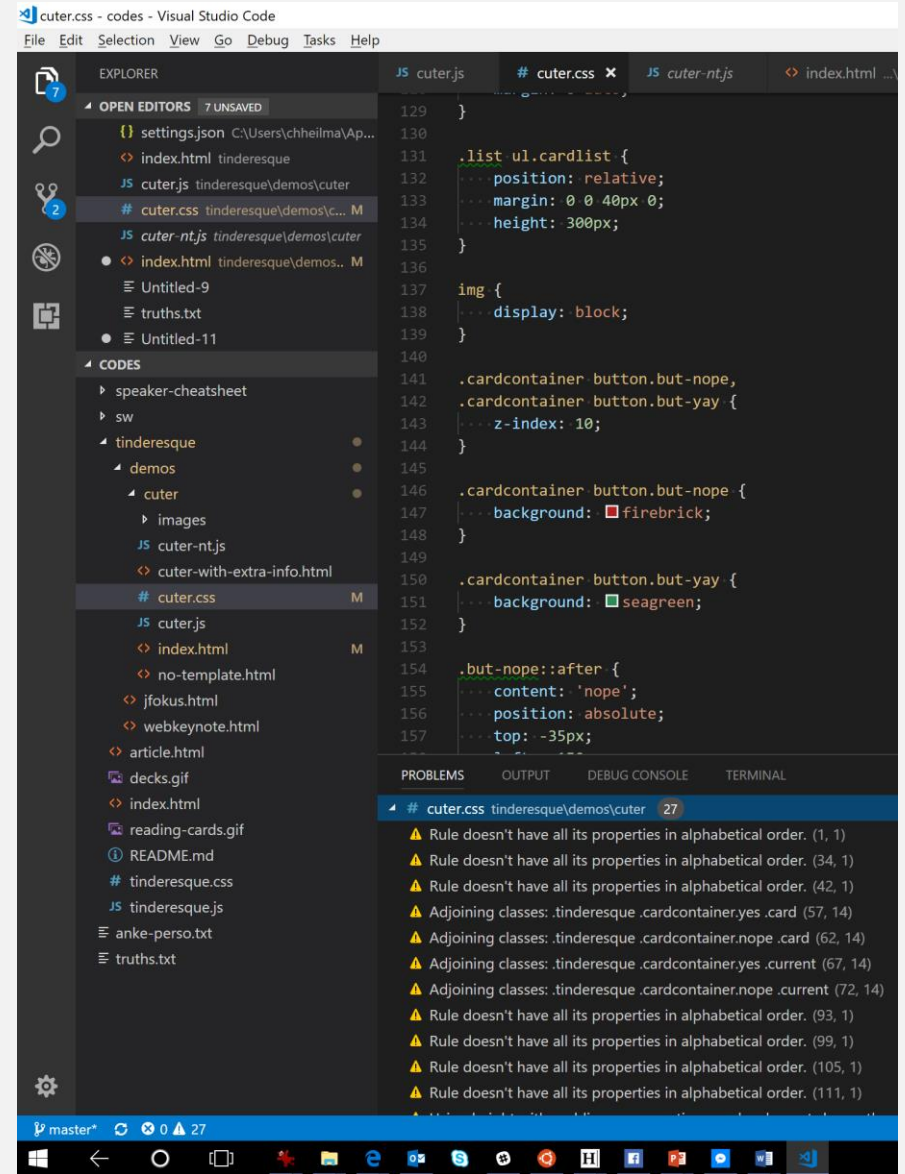
Customisable best practices in context



- 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





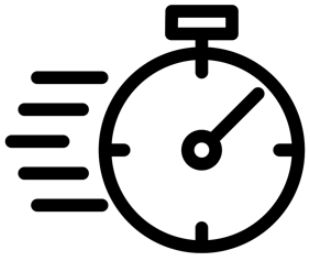
webhint.io

<https://github.com/webhintio>

<https://webhint.io>

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

Things that webhint helps you with...



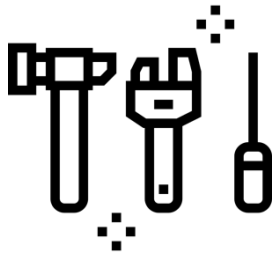
Performance



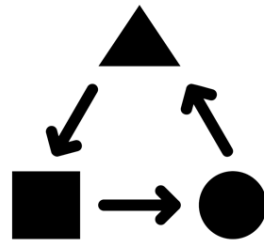
Accessibility



PWA
Readiness



Development
environment



Interoperability



Security

SCANNING 100%

SCAN TIME: 03:00

HINTS

URL: <https://fwdays.com/en/>

DATE: 2019-03-13 22:46

409

YOUR SCAN RESULT LINK: <https://webhint.io/scanner/91c143f5-dbd6-4881-adf8-1511ec186276>

webhint version: 4.4.1 Configuration JSON

Security

expand all

- disown-opener: 13 hints
- no-protocol-relative-urls: 3 hints
- sri: 22 hints
- strict-transport-security: 51 hints
- validate-set-cookie-header: 7 hints
- x-content-type-options: 41 hints
- sslabs: 1 hints

	ACCESSIBILITY	HINTS 1	PASSED 0/1
	COMPATIBILITY	HINTS 53	PASSED 4/7
	PWA	HINTS 1	PASSED 3/4
	PERFORMANCE	HINTS 216	PASSED 1/7
	PITFALLS	HINTS 0	PASSED 0/0
	SECURITY	HINTS 138	PASSED 3/10



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





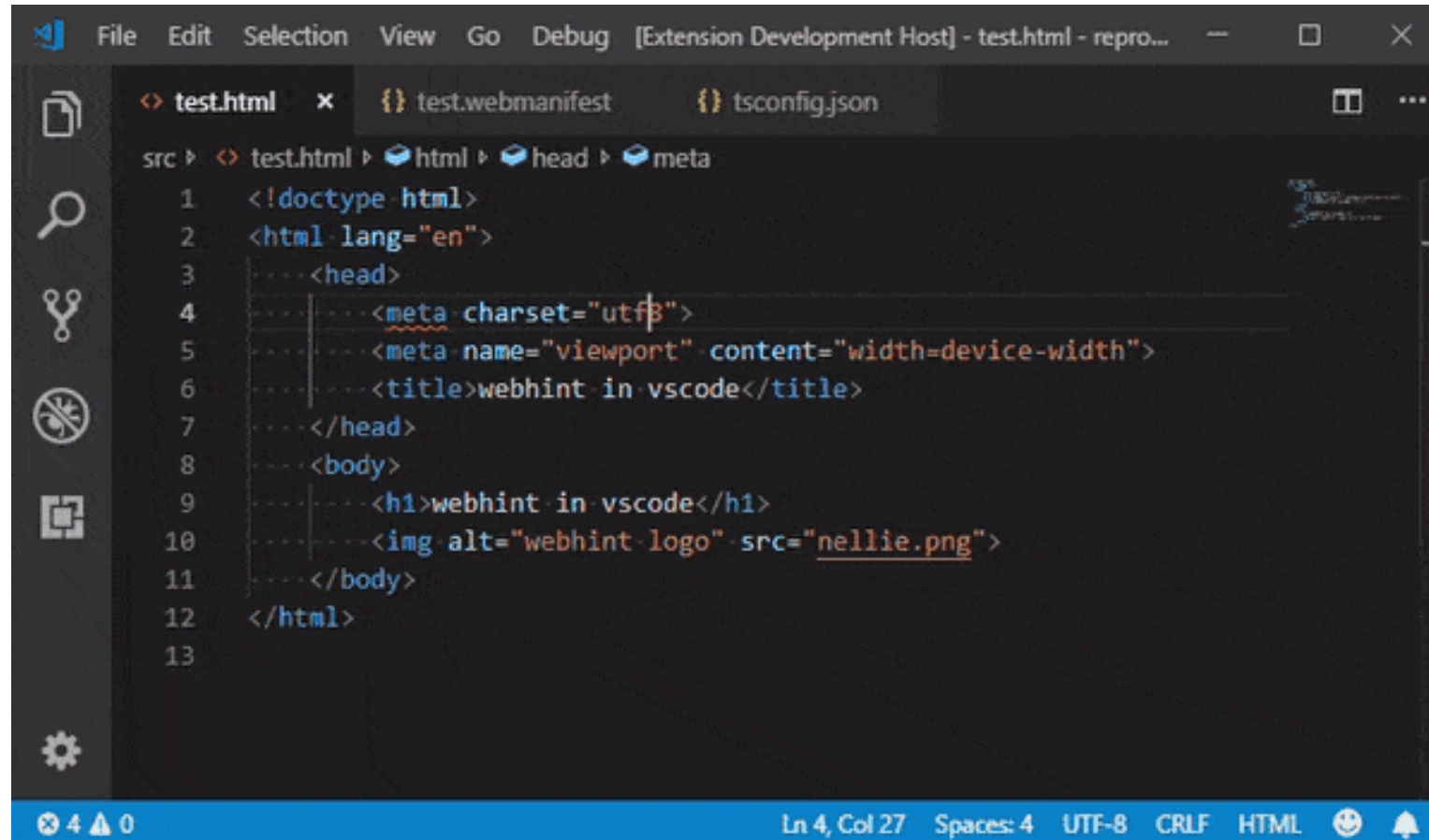
Customise the tests
to your needs...



- 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
```

webhint @code extension



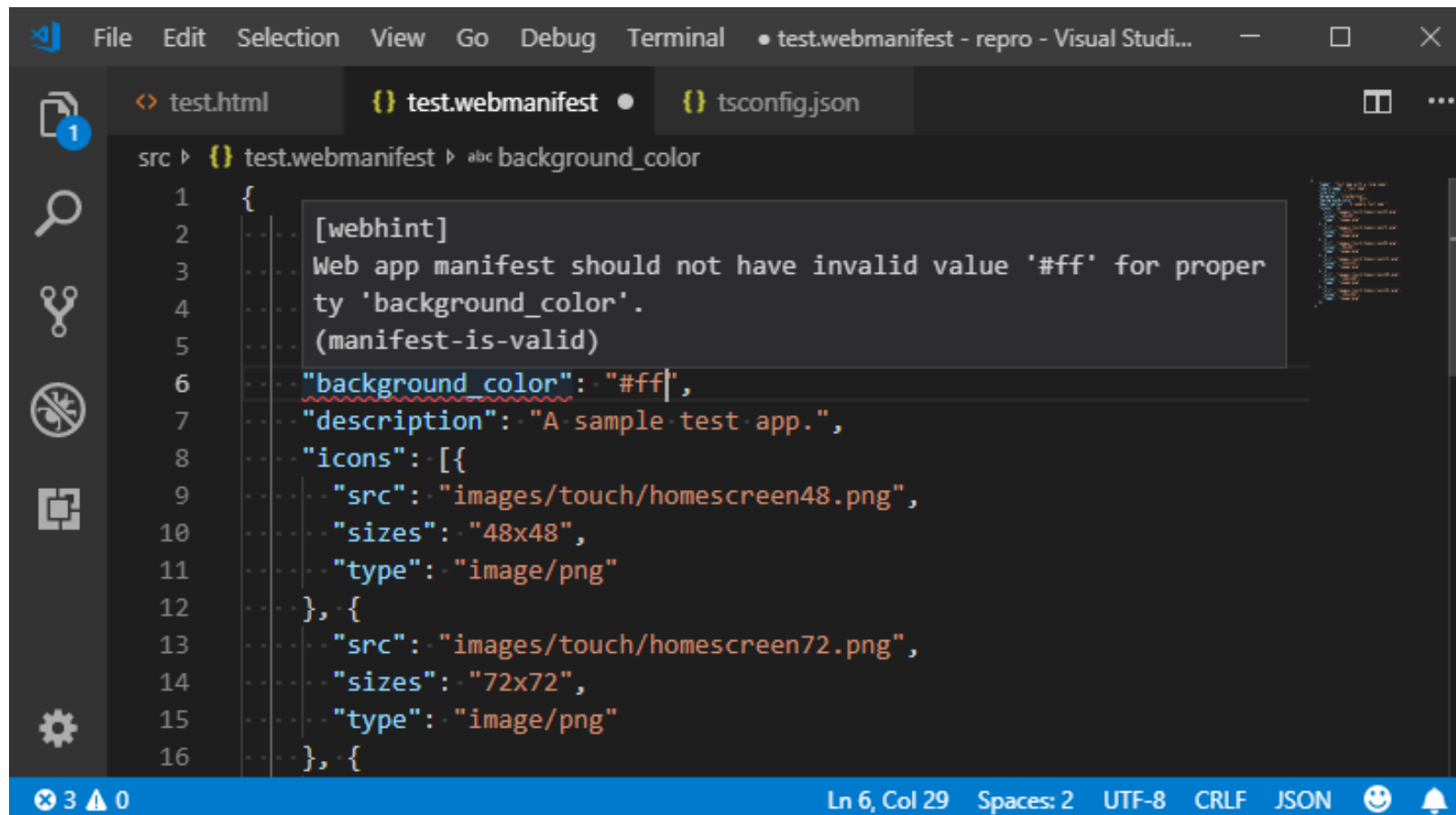
The screenshot shows the Visual Studio Code editor interface. The top menu bar includes File, Edit, Selection, View, Go, and Debug. The title bar indicates the current file is 'test.html' in an 'Extension Development Host'. The editor has three tabs open: 'test.html', 'test.webmanifest', and 'tsconfig.json'. The left sidebar shows the Explorer, Search, Source Control, Run and Debug, and Settings icons. The main editor area displays 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...

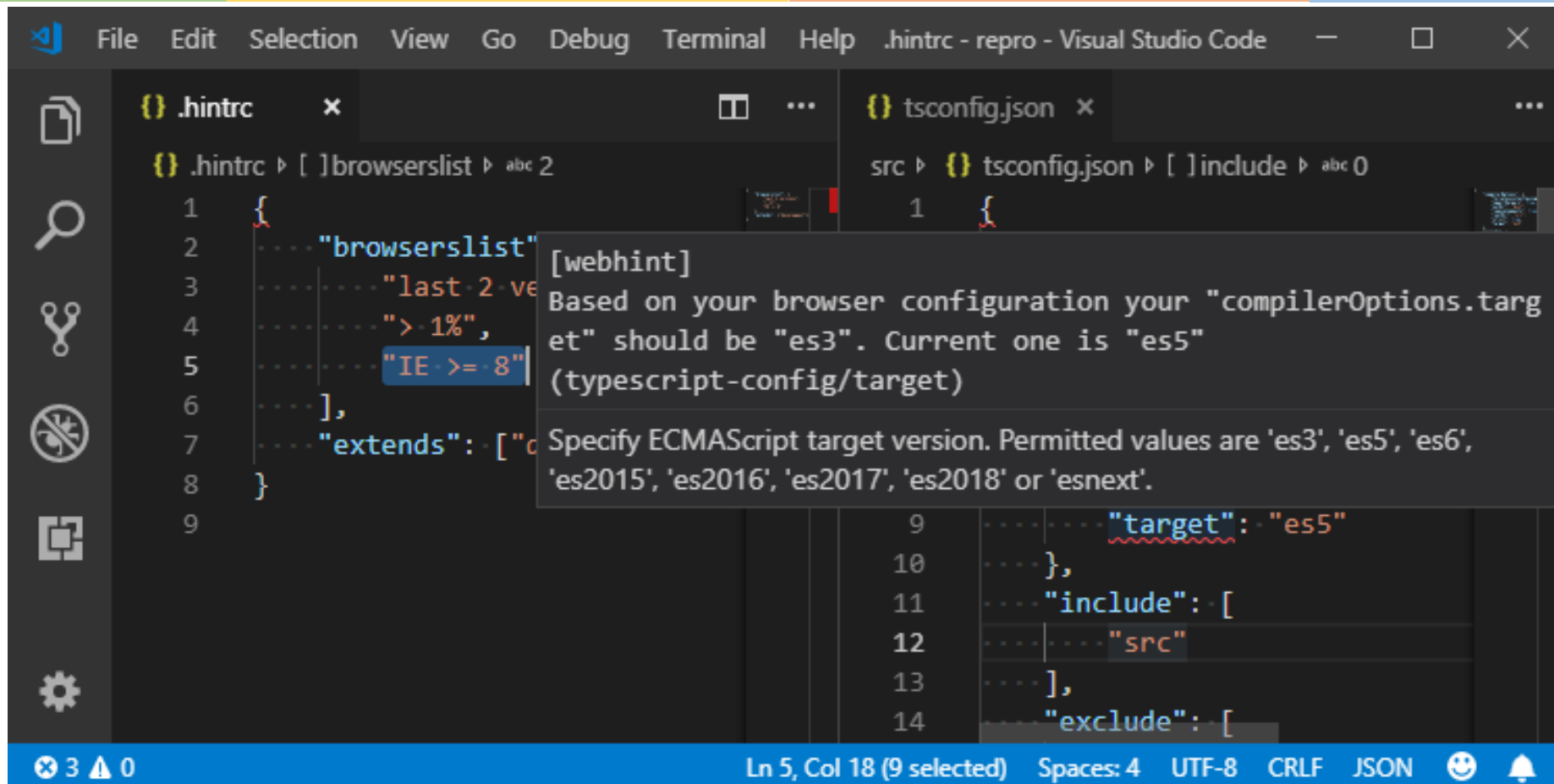


The screenshot shows the Visual Studio Code editor with a TypeScript file named `test.webmanifest` open. The file contains a JSON object representing a web app manifest. A red squiggly line underlines the value `"#ff"` for the `background_color` property. A tooltip error message is displayed over the error, stating: `[webhint] Web app manifest should not have invalid value '#ff' for property 'background_color'. (manifest-is-valid)`. The manifest code is as follows:

```
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  }, {
```

The status bar at the bottom indicates 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 mode is JSON.

Fixing your tooling setup...



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

```
.hintrc x
```

```
1 {
```

```
2   ... "browserslist"
```

```
3     ... "last 2 versions"
```

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

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

```
6   },
```

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

```
8 ]
```

```
9
```

```
tsconfig.json x
```

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

```
1 {
```

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

```
10 },
```

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

```
12   ... "src"
```

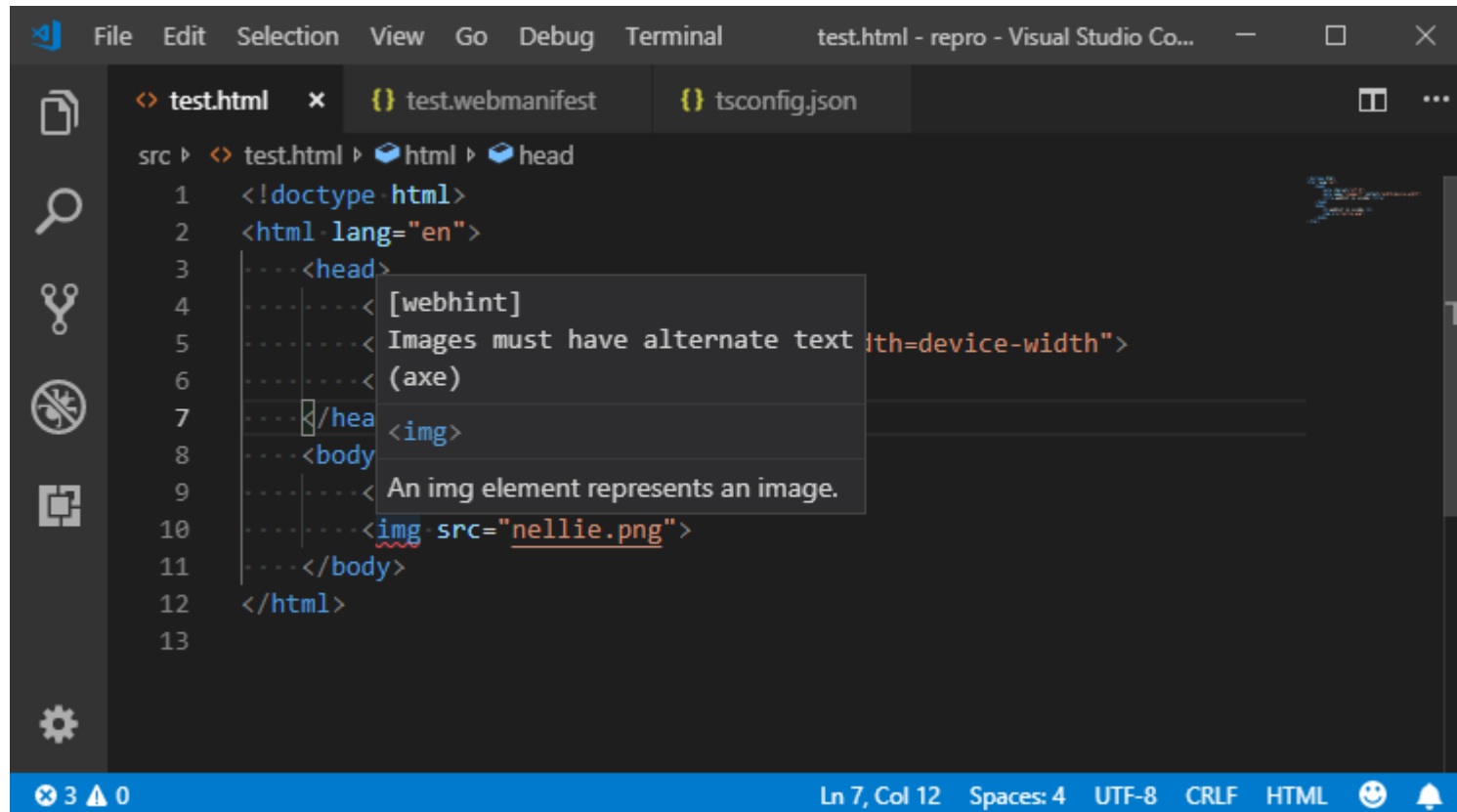
```
13 ],
```

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

[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'.

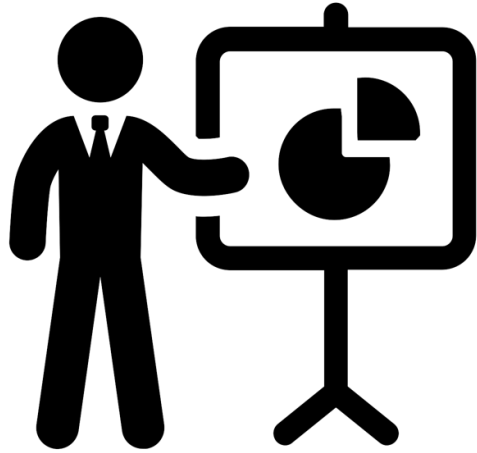
```
Ln 5, Col 18 (9 selected) Spaces: 4 UTF-8 CRLF JSON
```

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...

- The web is a mess
- We are to blame, but often we can't change what is there
- We should not, however, repeat the same mistakes of the past
- We have the tooling to apply accumulated knowledge automatically and be told that we mess up even before we save our code.
- This is a great way to learn and will help the future web.

Thanks!

Chris Heilmann
@codepo8

