Into the web multiverse

$ whoami

- Chris Heilmann @codepo8 – he/him/his
- Principal Program Manager Microsoft Developer Experience (VSCode, Devtools…)
- W3C Member
- 20 years web developer
- eToys, Agilisys, Yahoo, Mozilla…
- Wrote 3 books, contributed to 10(?)…
- Blogged since 2004, loud on Twitter

Yes, the slides are available…

aka.ms/web-multiverse

I've been working on a lot of things…

The web is a roaring success…

- It is hard to imagine a world without the world wide web
- Web technologies are standardised and browser differences are negligible
- Open source has matured from a niche to the technology that runs the web
- Documentation is plentiful, free and interactive

So, why is it all kind of a mess?

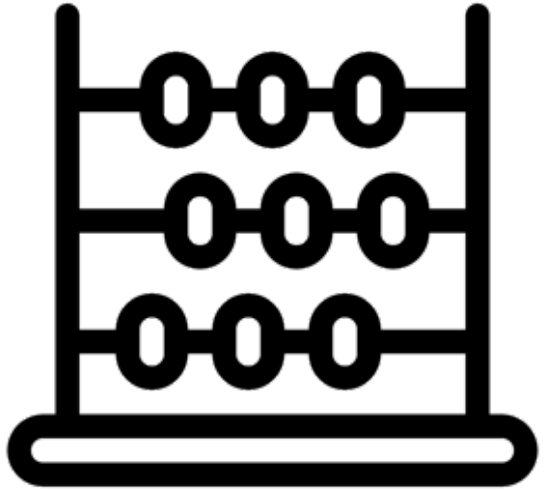
- Blatant lack of accessibility
- Security problems
- Performance issues
- A need for blockers to keep the web safe and a lot less annoying

No shortage of opinions why that is

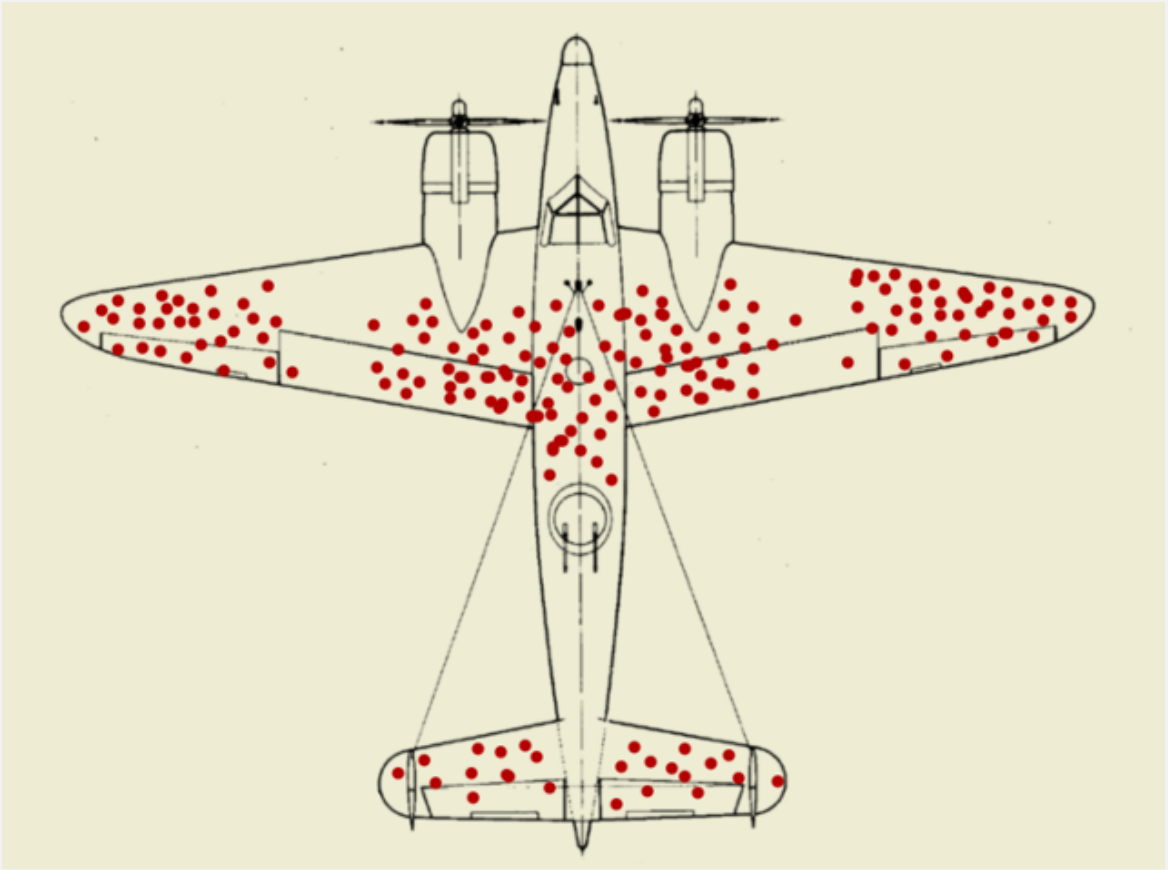
- Users are terrible and have horrible setups
- The web stack is just not good enough
- Greedy corporations plaster the web with ads and see users as sheep to milk (?)
- We don’t have the right tools
- Tech bros build horrible things without consideration
- Aliens

Finding the correct fix

Abraham Wald, statistician



Counting hits and misses…

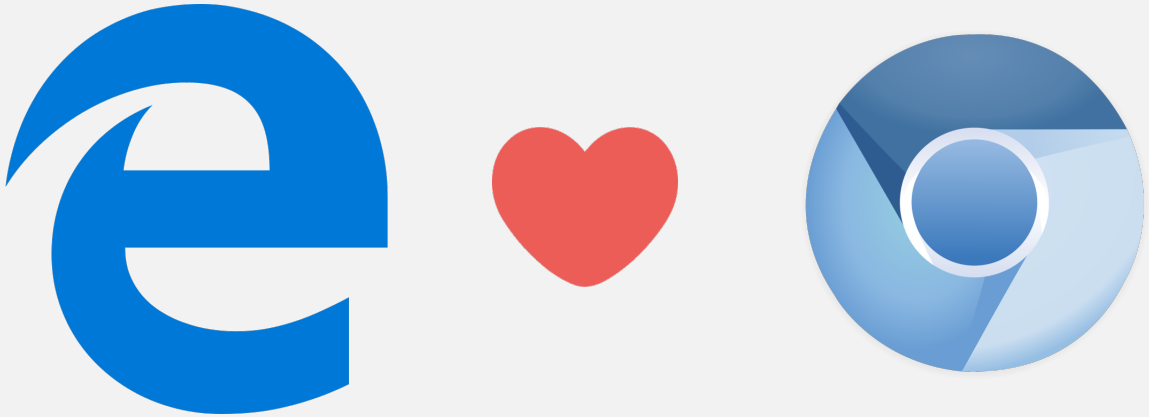


dgsiegel.net/talks/the-bullet-hole-misconception
wikipedia.org/wiki/Survivorship_bias

Sticking to the ratio…

We have 2 ears and 1 mouth – maybe we should spend more time listening and less talking.

I'm in the middle of a launch...



User needs > opinions

aka.ms/edgeinsider

Different users, different needs

The “keeping it current” user

Most of the time of the web I spend on my phone, but I also have a laptop. I user browser $x and I love that I can customize it to my needs. I keep everything I have up-to-date, to stay secure.

The “I am OK with what I have” user

I’ve been using browser $x for a while and I go to only a few places on the web. It annoys me when I need to upgrade my browser or when web sites change all the time. I don’t need new features!

The “I just go online” user

I am on the web for work and on our work computers. I click the icon to get me there and have a few sites I need to go to during the day. I don’t know how this works, our IT department sets these things up for us.

The cautious user

I don’t trust the web, you hear about all sorts of terrible things. So I use ad blockers, and turn off as many things as possible. If a web site needs them and I don’t feel comfortable, I go somewhere else.

The “Stockholm syndrome” user

I’ve used the web for a long time. I don’t trust things that don’t work like those I am used to. Yes, it feels easier, but I want to get a feeling of knowing what I am doing.

Somehow bitter pills to swallow

- To different users, a browser is
  - something they can't control
  - something they cherish and highly customize
  - irrelevant, as all of them work OK enough
  - nothing worth changing their habits for
- Security and privacy aren't things that people control, but trust in the platform to do for them

A fix seems to be somewhere else…

- It might be up to us to fix the web
- This means taking ownership and caring for the needs of different constituencies
- We can't expect users to care enough to fix the web – we need to empower developers, maybe?

Different developers, different needs

“This is a job” developer

I work for a company that defines our tools and computer setup. I know there are probably better things out there, but as I don’t get to use them at work, why bother learning them?

The “switched on” developer

It is frustrating that there are still people out there with old computers and outdated browsers. With a library I can support them as I’d rather learn and try out new things.

The “I am here to deliver” developer

My clients don’t pay extra for fancy new technology. They need to get something that works on their machines. It needs to be legally compliant, you know, accessible so they can’t be sued. That’s why I use a framework.

The “the web is broken” developer

I’ve always worked with higher programming languages and rich UI environments. The web seems like a mess and I don’t want to spend time trying to guess what the outcome of my code is.

The “I’m here to learn” developer

I just started as a web developer and wherever I look, I am confused. There is so much information and often people give contradicting advice. I just want to learn the right thing…

Somehow bitter pills to swallow (2)

- The caring developer is a tiny fraction of the market
- For many, this is a job to make a living
- New developers don't find an empathic, open world, but an avalanche of choice and opinions
- Satisfying developer needs has no direct correlation to creating better user products

Why learn the basics?

- Browsers can't break the web – their forgiveness knows no bounds
- If it works, why bother learning why?
- If it doesn't work, there's ample of resources to learn quick fixes.

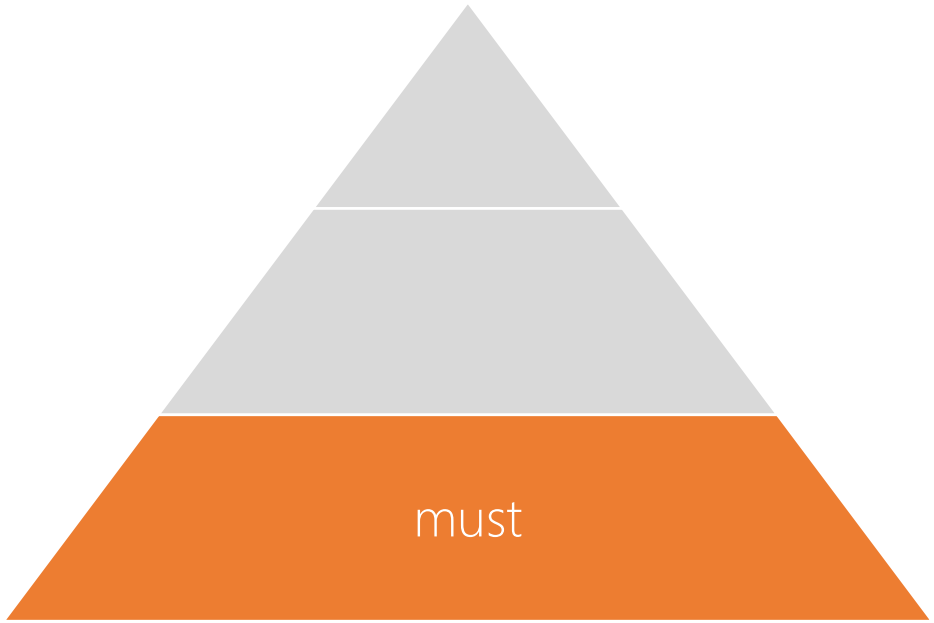
Abstraction attraction

- Learning “the web” isn’t a desirable skill
- Instead, abstractions get you eyeballs on job boards
- Abstractions promise faster turnaround and great products – big companies use them!

Tech innovation isn't the answer

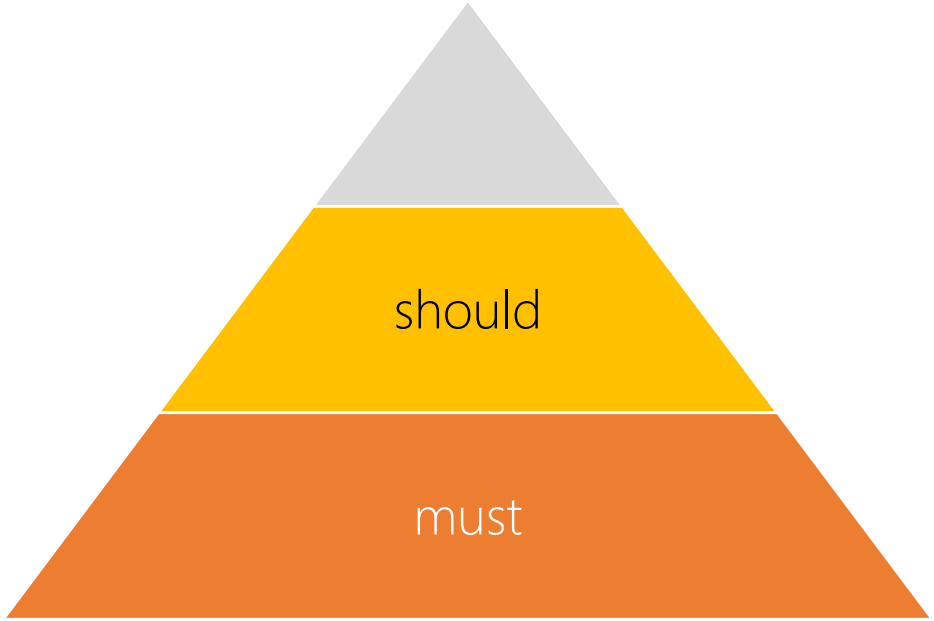
- Constant change in the developer space is a double-edged sword
- We empower a small group of highly motivated people at the cost of overwhelming others
- We innovate to fill a specialist need and call it a success because it is new instead of solving real problems
- We don't find out about real problems as we don't ask people outside our bubble

Focusing on delivery



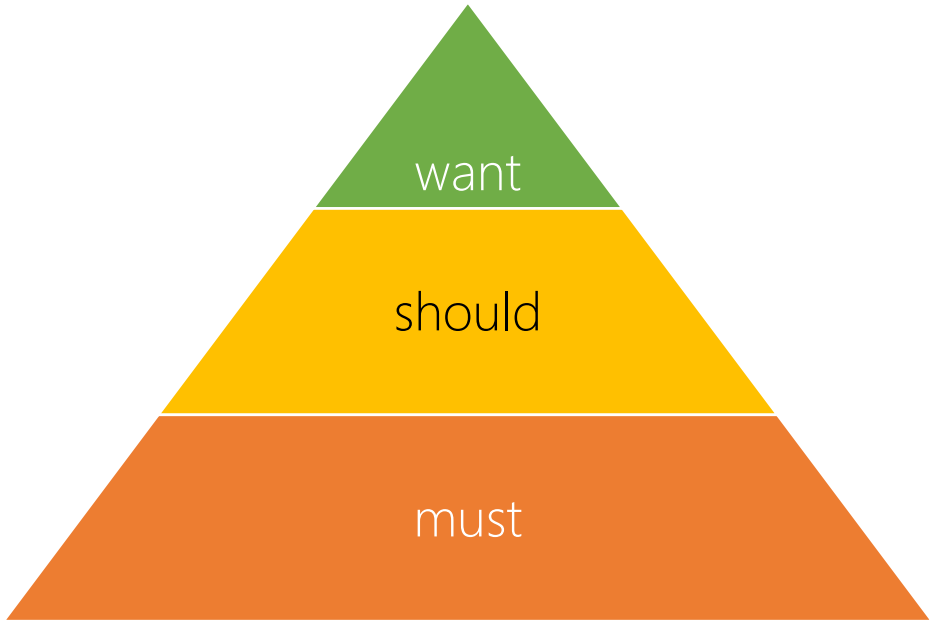
Filling the needs

- Availability
- Accessibility
- Security
- Mutability to user needs



Adding quality

- Enhanced performance
- Extended availability (offline)
- Customisability



Creating delight

- Make full use of the platform
- Predict use cases and prevent simple mistakes
- Offer extended functionality

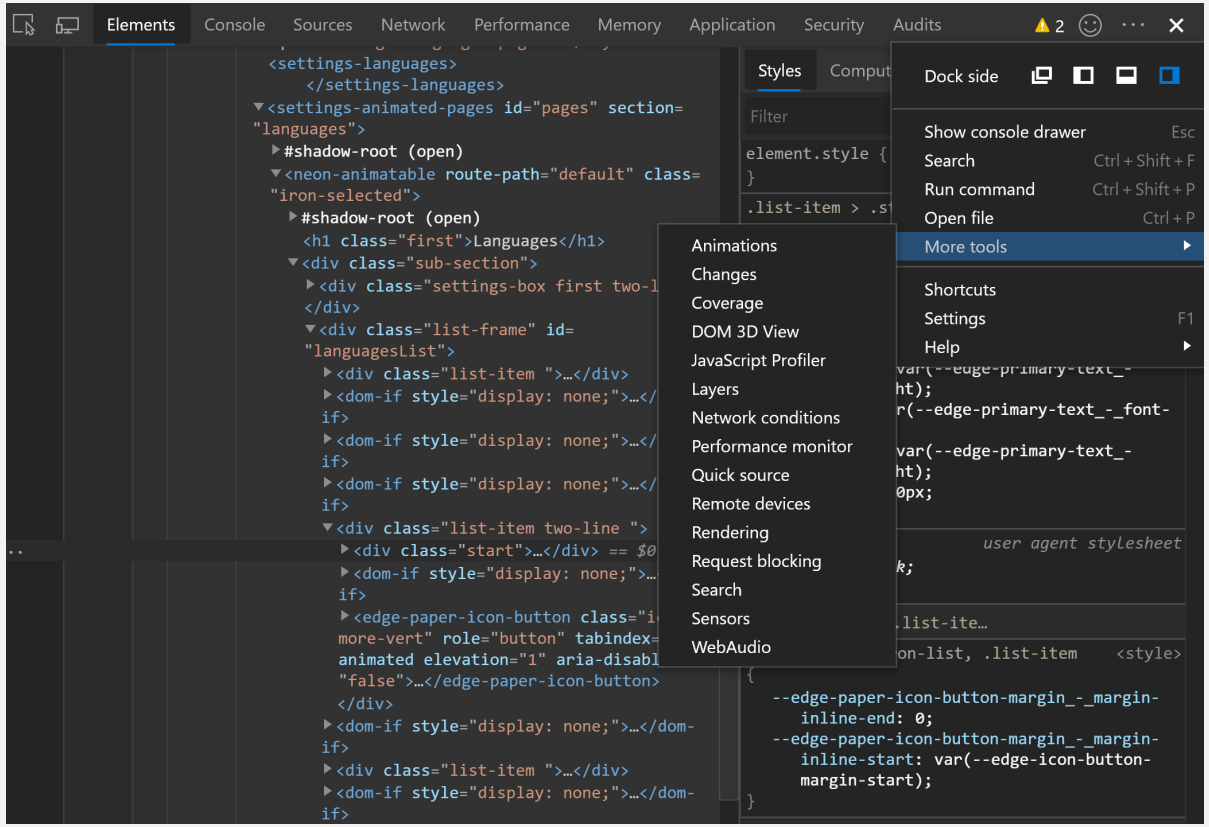
We <3 tools!

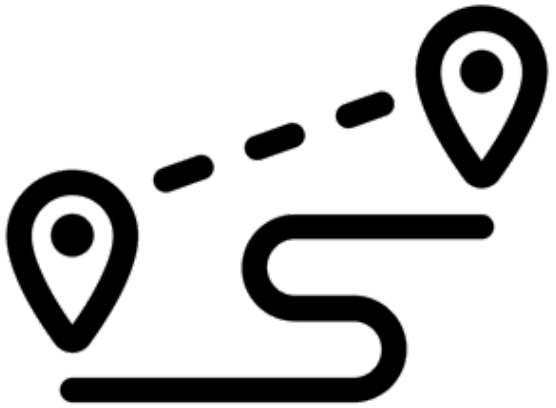
- Browser developer tools
- Editors, extensions and themes
- Terminals and their configuration
- Deployment and optimization toolchains
- Abstractions, Components and libraries

It is all a bit much…

- We overwhelm people who want to start with the web – or just want to do their job
- By demanding people to set up complex dev environments we inhibit diverse contribution
- We don't focus enough on human needs in our tools

Hypothesis: tools are too complex





Tools to help simplify webdev

Node: the glue

- JavaScript on the server
- Enables a highly customisable web infrastructure (servers, build tools…)
- Elevated JavaScript from “how does browser $x break” to a solution for all kind of web tasks

Chromium: the engine

- Open Source
- Multi Platform
- Powers various browsers and App engines (Electron)
- Can be automated and scripted (Puppeteer)

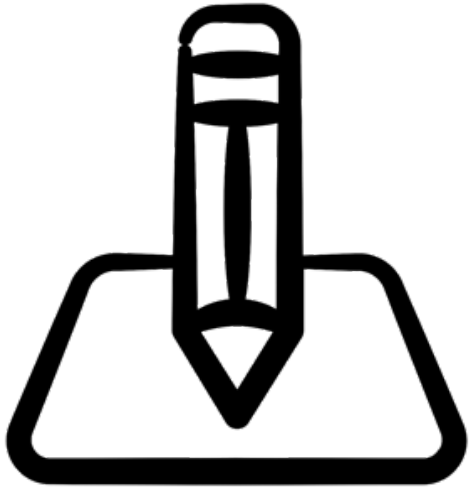
Visual Studio Code: the “hackable” editor

- Open Source
- Multi Platform
- Extensible / Customisable
- Based on Electron/Chromium
- Includes Git version control
- Has built-in terminal
- Lightweight, code focused

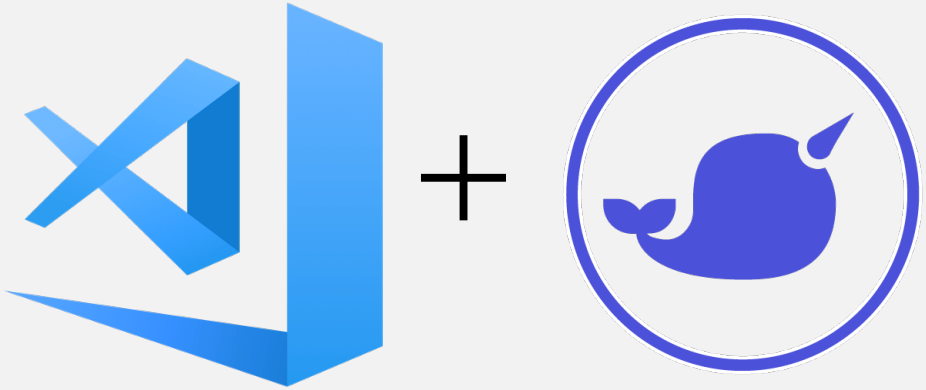
Webhint: testing and best practices

- Testing tool to check web projects for:
  - Accessibility issues
  - Performance problems
  - App readiness
  - Compatibility with standards
  - Common pitfalls to avoid
  - Security issues
- Online service (webhint.io) or node package to integrate in other tools
- Highly customisable

Mixing and matching these tools is powerful



Show problems with your code while you write it…



aka.ms/webhint4code

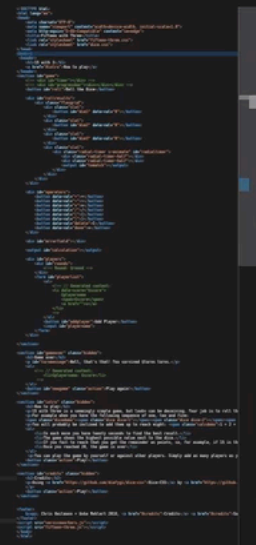
index.html — fifteen-three

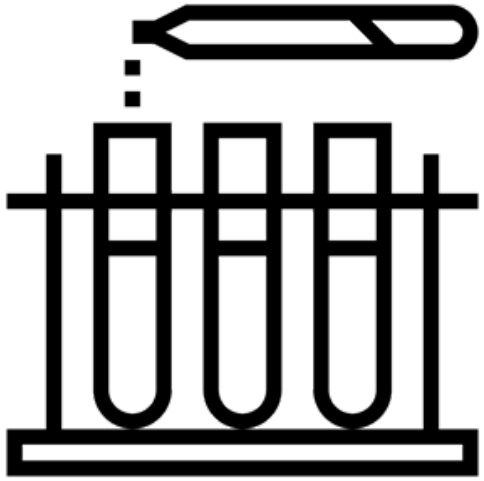
index.html

index.html > html > body

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <meta http-equiv="X-UA-Compatible" content="ie=edge">
    <title>Fifteen with Three</title>
    <link rel="stylesheet" href="fifteen-three.css">
    <link rel="stylesheet" href="dice.css">
</head>
<body>
 <header>
    <h1>15 with 3</h1>
    <a href="#intro">How to play</a>
</header>
<section id="game">
    <!-- <div id="timer"></div> -->
    <!-- <div id="progressbar"><div></div></div> -->
    <button id="roll">Roll the Dice</button>

    <div id="rollresults">
        <div class="flexgrid">
            <div class="slot">
                <button id="die1" data-val="0"></button>
            </div>
            <div class="slot">
                <button id="die2" data-val="0"></button>
```





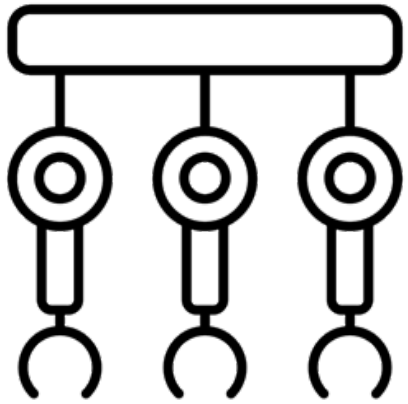
Test your sites in the browser...

+

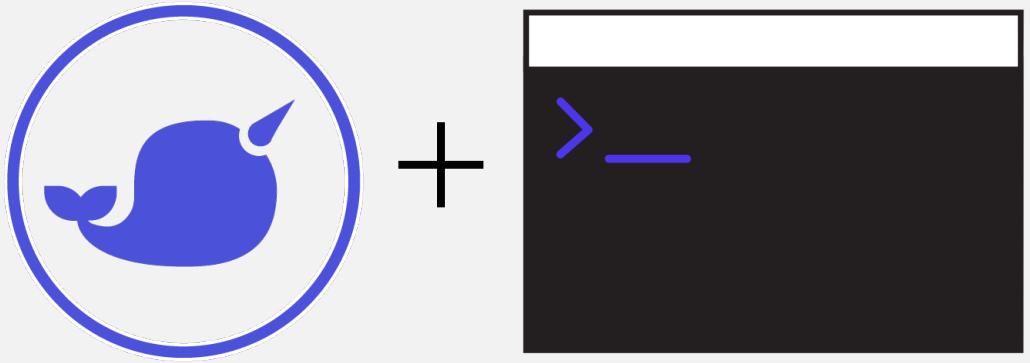
aka.ms/webhint4edge

aka.ms/webhint4firefox

aka.ms/webhint4chrome



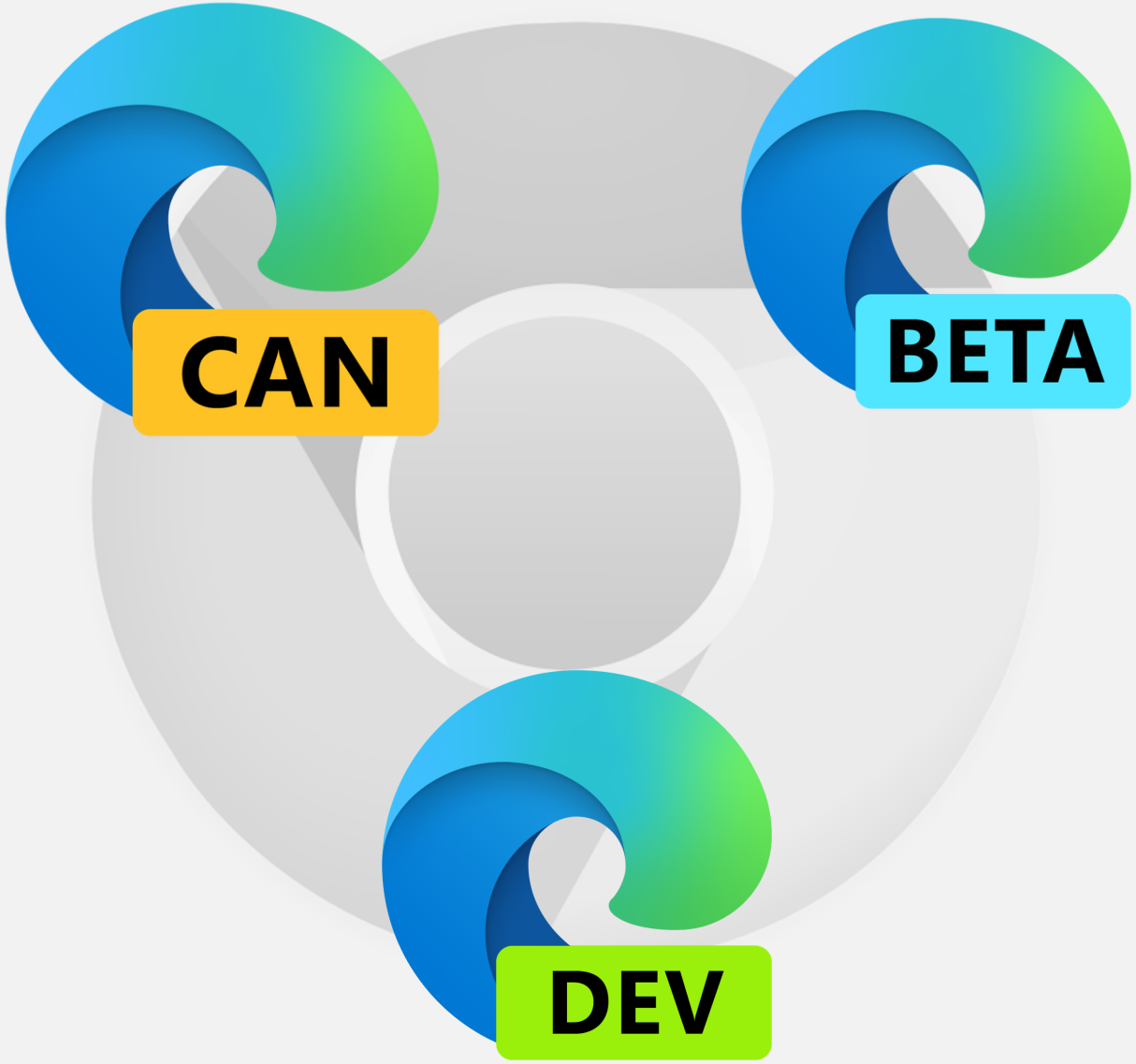
Add automated testing to your build process…



aka.ms/webhint4node

What's next?

Edge on Chromium

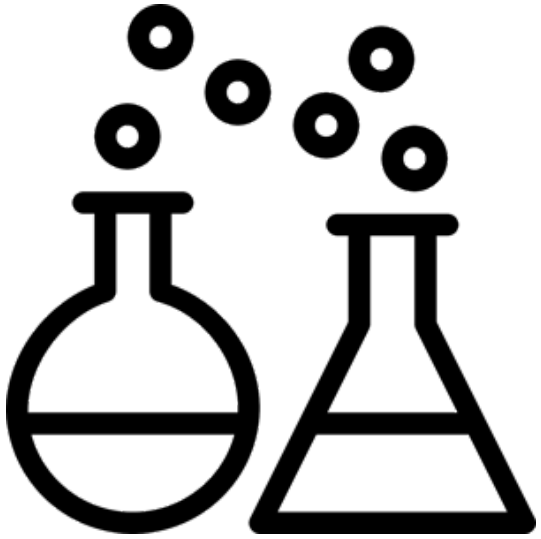


aka.ms/edgeinsider

Edge on Chromium

- Adding a powerful player with lots of users to an important part of the web
- Lots of bug fixes
- Internationalisation of browser and developer tools (30% of VSCode users change the language)
- Accessibility enhancements to browser and devtools

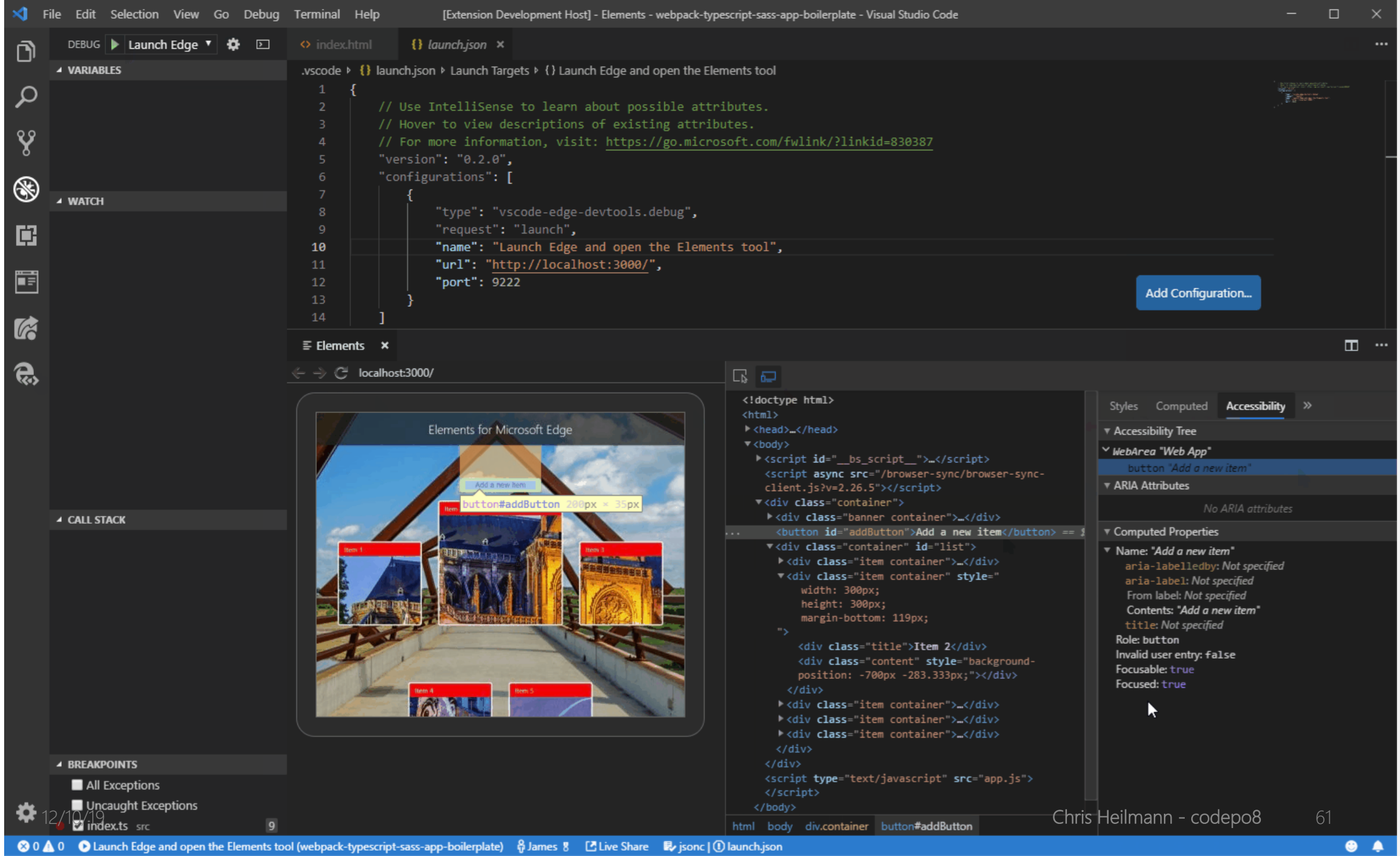
aka.ms/edgeinsider



Elements for Code

- Edge/Chromium developer tools inside VS Code
- No need to switch to the browser to try a few things out
- No need to replicate the same functionality in both places

aka.ms/elements4code


File Edit Selection View Go Debug Terminal Help
[Extension Development Host] - Elements - webpack-typescript-sass-app-boilerplate - Visual Studio Code
DEBUG Launch Edge
VARIABLES
WATCH
CALL STACK
BREAKPOINTS
All Exceptions
Uncaught Exceptions
index.ts src 9
index.html
launch.json
.vscode ▸ launch.json ▸ Launch Targets ▸ {} Launch Edge and open the Elements tool
{
// Use IntelliSense to learn about possible attributes.
// Hover to view descriptions of existing attributes.
// For more information, visit: https://go.microsoft.com/fwlink/?linkid=830387
"version": "0.2.0",
"configurations": [
{
"type": "vscode-edge-devtools.debug",
"request": "launch",
"name": "Launch Edge and open the Elements tool",
"url": "http://localhost:3000/",
"port": 9222
}
]
Add Configuration...
Elements
localhost:3000/
Elements for Microsoft Edge
button#addButton 200px × 35px
<!doctype html>
<html>
<head>…</head>
<body>
<script id="__bs_script__">…</script>
<script async src="/browser-sync/browser-sync-client.js?v=2.26.5"></script>

…
<button id="addButton">Add a new item</button> == $0

…
<div class="item container" style="
width: 300px;
height: 300px;
margin-bottom: 119px;
">
Item 2

…
…
…

<script type="text/javascript" src="app.js">
</script>
</body>
html body div.container button#addButton
Styles Computed Accessibility
Accessibility Tree
WebArea "Web App"
button "Add a new item"
ARIA Attributes
No ARIA attributes
Computed Properties
Name: "Add a new item"
aria-labelledby: Not specified
aria-label: Not specified
From label: Not specified
Contents: "Add a new item"
title: Not specified
Role: button
Invalid user entry: false
Focusable: true
Focused: true
0 0 Launch Edge and open the Elements tool (webpack-typescript-sass-app-boilerplate) James Live Share jsonc | launch.json

What's missing on the web platform?

webwewant.fyi

In summary…

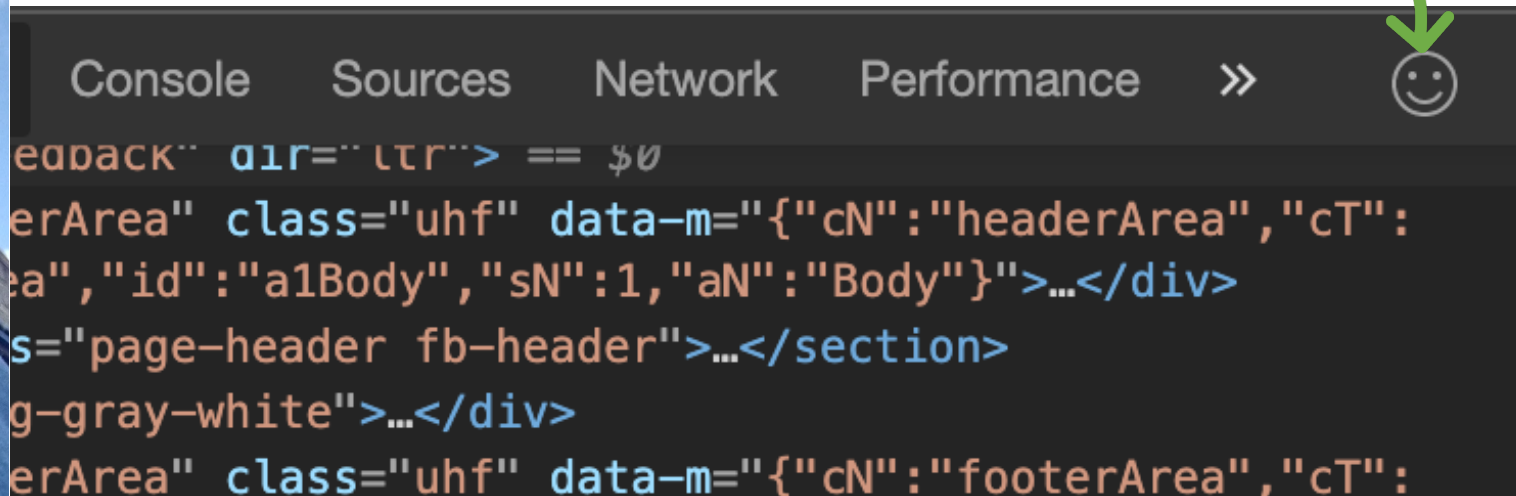
- The web is a diverse place
- We have a chance to make it easier for everyone to create quality products for all end users
- We have open source platforms with open communication channels that power the web
- We accumulated a lot of knowledge, why not add it to our tools to teach the creators of the web in context?

Thanks – your voice matters!

Chris Heilmann

@codepo8

@EdgeDevTools

christianheilmann.com

Click the 😀 in Edge developer tools!