

**This is PDF version
of Web Development
tailored for 2024.**

HTML Version also

available

console.log

**border: 1px solid
red;**

I... love you guys so much.

made with mematic



whoami

===

andrzejfricze

- FullStack for 10 (15?) years. Currently The HEINEKEN Company
- Organizer meet.js Kraków
- Organizer futureconf.tech
- Music producer
- Besides that I read books and go to gym...
- ...and I just got married 🥰

fricze.com/webdev24

Web

Development

tailored for 2024

~~Web Development~~

~~tailored for 2024~~

or: Not your dads

Web Platform

~~Web Development tailored
for 2024~~

~~or: Not your dads
Web Platform~~

or: Not main
Web Platform
tbh, because a
lot has changed
since I started...



Native features

- are generally, faster than libraries and frameworks
- will be available even when your favorite framework changes
- make debugging easier, because that's the language that browsers speak
- sometimes are just magic that cannot be polyfilled in your favorite framework!

**This is a speedrun. I
won't teach you
everything, but you'll
know what to look for
buckle up kiddo 🤠**

Disclaimer

A lot of examples are not main, but are taken from MDN, web.dev, codepen.io, or similar, publicly available sources

A diverse crowd of people is shown at a concert or event. In the center, Taylor Swift is wearing a black dress and has her right arm raised. To her left, a Black man in a suit and tie is smiling. To her right, a woman with long brown hair is also smiling. The background is dark blue with some stage lights visible.

Let's go!



**No more "undefined is not a
function"**

**Safe null checks
in JavaScript**

Nullish coalescing operator ¹

```
let number = 0 || 1;
// number === 1
number = 0 ?? 1;
// number === 0

let string = '' || 'string';
// string === 'string'
string = '' ?? 'string';
// string === ''
```

1. https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Operators/Nullish_coalescing

Optional chaining ¹

```
const ArticleView = (props) => {  
  const value = props.article.value;  
  // Uncaught TypeError: Cannot read  
properties of undefined (reading 'value')  
  
  const maybeValue = props.article?.value;  
  // maybeValue === undefined  
  ...  
};
```

1. https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Operators/Optional_chaining

Native async possibilities 

**Better control
over Promises
and Iterables**

Collect the results of **all given** promises. Even those that threw an error

Promise.allSettled¹

```
// OLD
Promise.all([Promise.resolve(10), Promise.resolve(20)])
  === [10, 20]
// OLD
Promise.all([Promise.resolve(10), Promise.reject(20)])
  === Promise {<rejected>: 20}

// NEW 🔥
Promise.allSettled([Promise.resolve(10), Promise.reject(20)])
  === [
    {status: 'fulfilled', value: 10},
    {status: 'rejected', reason: 20},
  ]
```

1. https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Promise/allSettled

Promise.withResolvers

1

```
const getRequestPromise = () => {
  const {
    promise, resolve, reject
  } = Promise.withResolvers();

  asyncRequest(config, response => {
    const buffer = [];
    response.on('data', data => buffer.push(data));
    response.on('end', () => resolve(buffer));
    response.on('error', reason => reject(reason));
  });

  return promise;
}
```

Array.fromAsync for consuming asynchronous iterables ¹

```
const asyncIterable = (async function* () {
  for (let i = 0; i < 5; i++) {
    await new Promise((resolve) => setTimeout(
      resolve, 10 * i
    ));
    yield i;
  }
})();

await Array.fromAsync(asyncIterable) === [0, 1, 2, 3, 4]
```

1. https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Array/fromAsync

The end of lodash? 🤯

**New methods to
manipulate
Arrays and
Objects**

Easier access to array elements `arr.at()`,

`arr.findLast()`¹

```
const arr = [5, 12, 50, 130, 44];  
arr.at(3) === arr[3] === 130;  
arr.at(-2) === arr[4] === 130;  
  
arr.find((element) => element > 45)  
    === 50  
  
arr.findLast((element) => element > 45)  
    === 130
```

1. https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Array

**Safe state changes in React
with...**

**immutable arrays
methods ¹**

1. https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Array

All those methods return new array instead of mutating

```
Array.prototype.toSorted()
```

```
Array.prototype.toReversed()
```

```
Array.prototype.toSpliced()
```

```
Array.prototype.with()
```

Safe React state update

```
const [posts, setPosts] = useState([]);

setPosts(posts => posts.with(
  3,
  {...posts[3],
  "title": "Updated post title"}
));
```

Group arrays into Objects or Maps ¹

```
Object.groupBy()
```

```
Map.groupBy()
```

1. https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Object/groupBy

Native EventTarget constructor

`.addEventListener()` on your own

objects

```
class Counter extends EventTarget {
  constructor(initialValue = 0) {
    super();
    this.value = initialValue;
  }

  #emitChangeEvent() {
    this.dispatchEvent(new CustomEvent("valuechange", { detail: this.value }));
  }

  increment() {
    this.value++;
    this.#emitChangeEvent();
  }

  decrement() {
    this.value--;
    this.#emitChangeEvent();
  }
}

const counter = new Counter(0);

counter.addEventListener("valuechange", (event) => {
  document.querySelector("#currentValue").innerText = event.detail;
});
```

Can I use those features? What about browsers that don't support them?

@babel/preset-

env 🤖

**Better responsive
design**

Dynamic viewport units ¹

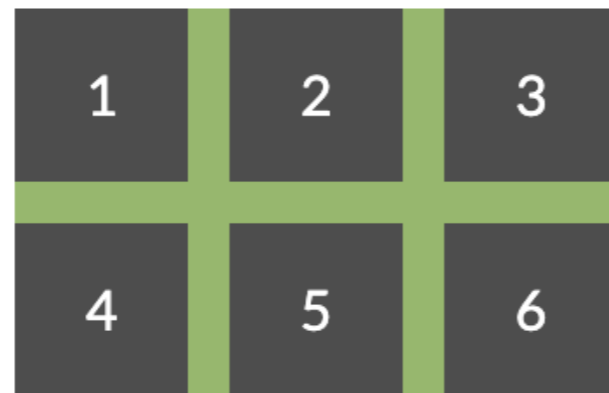
```
height: 45dvh
```

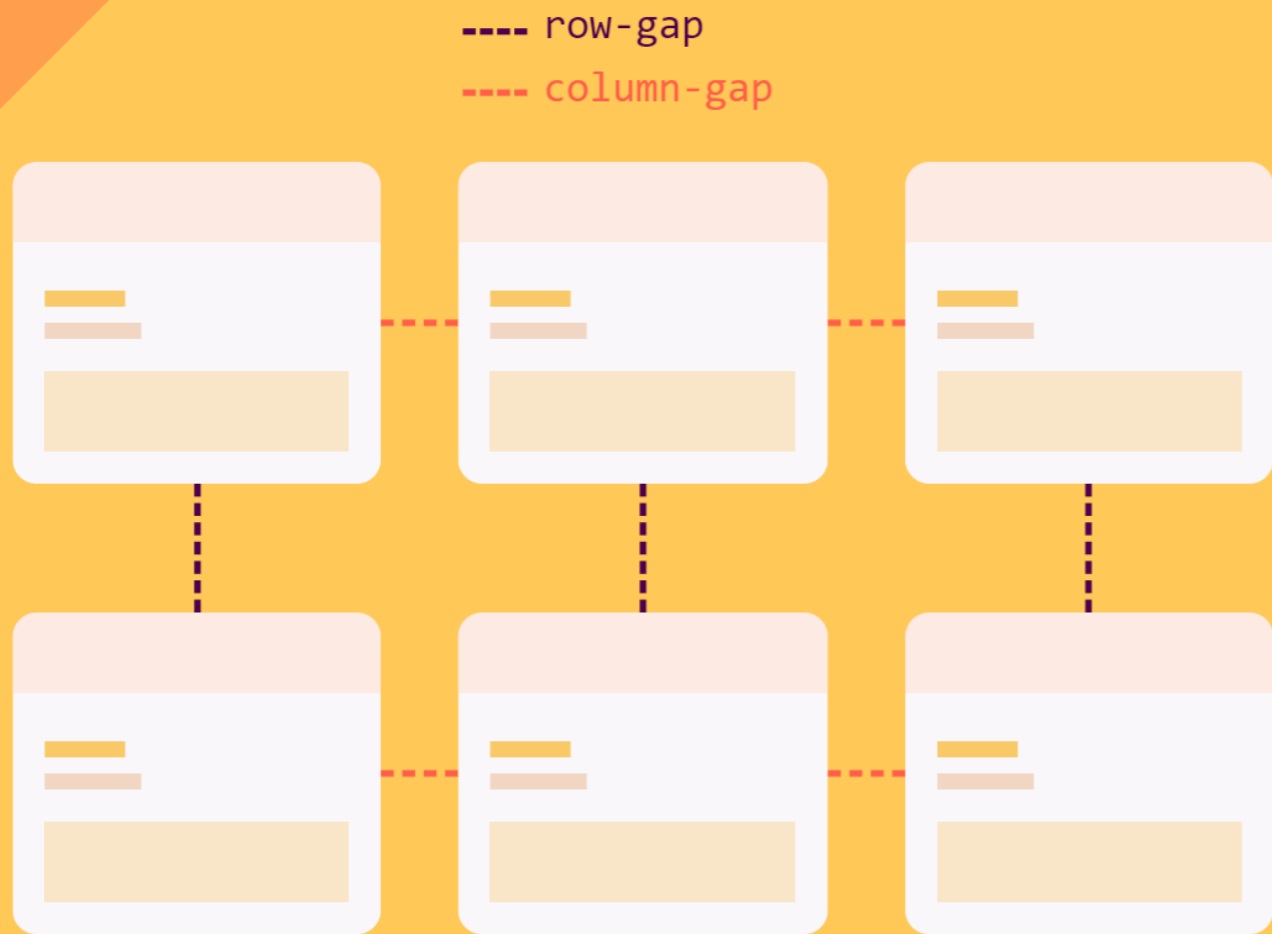
1. <https://web.dev/blog/viewport-units>

No more margin problems with **CSS Gap for Flexbox and Grid**¹

1. <https://developer.mozilla.org/en-US/docs/Web/CSS/gap>

```
.flex-gap {  
  display: inline-flex;  
  flex-wrap: wrap;  
  gap: 12px;  
}
```





```
.grid-gap {  
  display: grid;  
  // gap: 12px 20px;  
  column-gap: 12px;  
  row-gap: 20px;  
}
```

Interop 2023

Container Queries and Container Units

Components **responsive** in relation **to** **container** not whole window

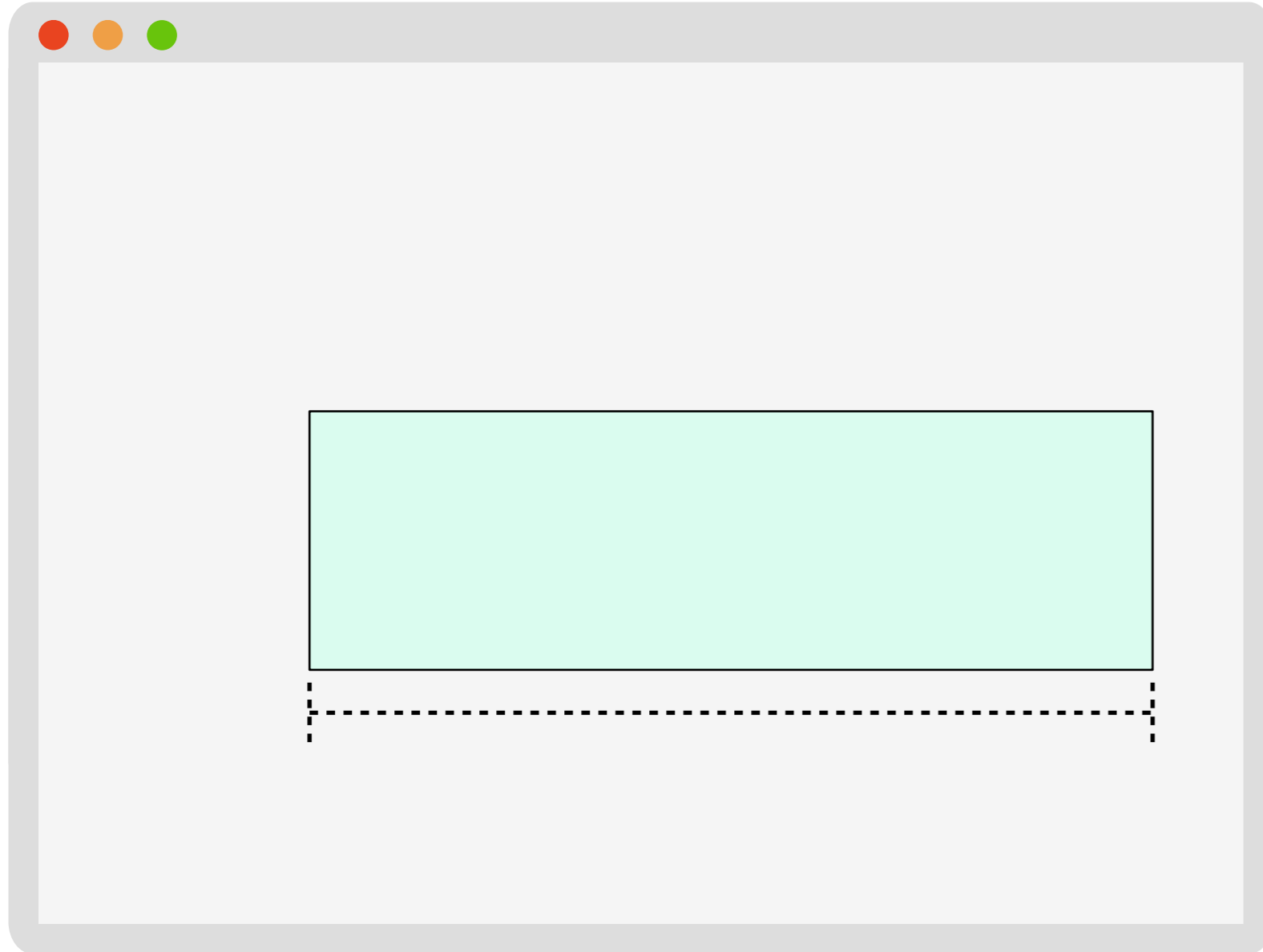
- <https://github.com/GoogleChromeLabs/container-query-polyfill>
- https://developer.mozilla.org/en-US/blog/getting-started-with-css-container-queries/#nested_containers

Media query (viewport size)

```
@media (max-width: 30em){  
  /* stylesheet */  
}
```

Container query (container size)

```
@container (width > 15em){  
  /* stylesheet */  
}
```



**One component adjusting to
given role and position**

<https://codepen.io/wuuuuuuuuut/pen/RwOxVQe>

also...

**Container queries
~~might be~~ are an
alternative to listening
to element resize**

<https://codepen.io/chriscoyier/pen/jOeBzNN>

Interop 2023

Subgrid. Full control over your layout

- <https://codepen.io/wuuuuuuuuuuut/pen/wvZppZE>
- <https://web.dev/articles/css-subgrid>

Want to support older browsers?

Remove container from DOM tree with
`display: contents` **to imitate subgrid**

Supported since 2018

<https://css-tricks.com/get-ready-for-display-contents/>

**New ways to
optimize
performance**

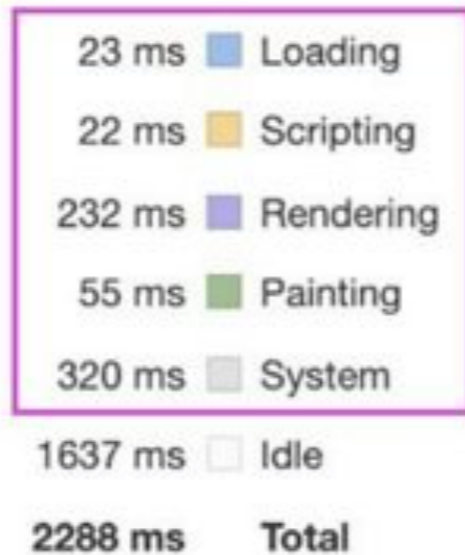
Part of new **containment spec**. Partial implementation in Firefox and Safari

Control rendering with `content-visibility`

- <https://infrequently.org/2020/12/content-visibility-scroll-fix/>
- <https://web.dev/articles/content-visibility>

baseline

chunks with content-visibility: auto



All browsers since 2022

```
contain: strict
```

**for isolating
rendering layers**

Interop 2023

Offscreen Canvas

Performant Three.js on separate thread

- <https://github.com/pmndrs/react-three-offscreen>
- <https://www.fricze.com/demo/three>
- <https://offscreen.pmnd.rs/>
- <https://blog.stackademic.com/optimizing-3d-model-rendering-in-next-js-by-reducing-blocking-time-using-react-fiber-offscreen-11bd39ca7a67>

**New animation
APIs. More fun
designs**

Interop 2023

CSS Motion Path for creating exact animation paths

- <https://codepen.io/prathameshkoshti/pen/oNgVdXV>
- <https://codepen.io/Chokcoco/pen/xxgMPzJ>

CSS Scroll Behavior for smooth scrolling

<https://codepen.io/wuuuuuuuuut/pen/BaEJJYY>

Interop 2023

CSS Clipping and Masking

- <https://codepen.io/michaelgearon/pen/rNmKLO>
- <https://bennettfeely.com/clippy/>

**CSS syntax
improvements**

:is () , :where ()

for shorter CSS selectors¹

1. <https://developer.mozilla.org/en-US/docs/Web/CSS/:is>

Before `:is()`

```
/* Level 0 */
h1 {
  font-size: 30px;
}

/* Level 1 */
section h1,
article h1,
aside h1,
nav h1 {
  font-size: 25px;
}

/* Level 2 */
section section h1,
section article h1,
section aside h1,
section nav h1,
article section h1,
article article h1,
article aside h1,
article nav h1...
```

After `:is()`



```
/* Level 0 */
h1 {
  font-size: 30px;
}
/* Level 1 */
:is(section, article, aside, nav) h1 {
  font-size: 25px;
}
/* Level 2 */
:is(section, article, aside, nav) :is(section, article,
aside, nav) h1 {
  font-size: 20px;
}
```

Interop 2023

`:has()` – real parent
selector!¹

```
label:has(input:invalid) {  
  color: red;  
}
```

1. <https://codepen.io/wuuuuuuuuut/pen/abxEyZr>

**Available since 2022. Polyfill
available**

Control CSS

cascading with

@layer

1 2 3

1. <https://developer.mozilla.org/en-US/docs/Web/CSS/@layer>
2. <https://css-tricks.com/css-cascade-layers/>
3. <https://www.oddbird.net/2022/06/21/cascade-layers-polyfill/>

CSS Demo: @layer

RESET

HTML

CSS

OUTPUT

```
1 @layer module, state;
2
3 @layer state {
4   .alert {
5     background-color: brown;
6   }
7   p {
8     border: medium solid limegreen;
9   }
10 }
11
12 @layer module {
13   .alert {
14     border: medium solid violet;
15     background-color: yellow;
16     color: white;
17   }
18 }
19
```

Beware of the zombies

CSS Demo: @layer

RESET

HTML

CSS

OUTPUT

```
1 @layer state, module;
2
3 @layer state {
4   .alert {
5     background-color: brown;
6   }
7   p {
8     border: medium solid limegreen;
9   }
10 }
11
12 @layer module {
13   .alert {
14     border: medium solid violet;
15     background-color: yellow;
16     color: white;
17   }
18 }
19
```

Beware of the zombies

Works with @import!

```
@import url(headings.css) layer(default);
@import url(links.css) layer(default);
@import url(links-dimmed.css) layer(dimmed);

@layer default {
  audio[controls] {
    display: block;
  }
}
```

Interop 2024

CSS Nesting! ¹

```
// that's native CSS! 🥰  
input {  
  /* styles for input not in a label */  
  border: tomato 2px solid;  
}  
label {  
  /* styles for label */  
  font-family: system-ui;  
  font-size: 1.25rem;  
  & input {  
    /* styles for input in a label */  
    border: blue 2px dashed;  
  }  
}
```

1. https://developer.mozilla.org/en-US/docs/Web/CSS/CSS_nesting/Using_CSS_nesting

Easier declarations, easier animations

Individual transform properties ¹

```
// old transform property
.logo {
  transform: rotate(45deg) scale(2) translate(40px, 0);
}

// new, individual properties
.logo {
  rotate: 45deg;
  scale: 2;
  translate: 40px, 0;
}
```

1. <https://blog.logrocket.com/deep-dive-css-individual-transform-properties/>

**Better web
typography**

Interop 2024

```
text-wrap: balance
```

unbalanced:

<https://codepen.io/wuuuuuuuuut/pen/RwOxVQe>

vs balanced:

<https://codepen.io/wuuuuuuuuut/pen/QWPaMWQ>

No more `:before` and weird borders

`text-decoration` **magic**

(since 2020)

and `1h` unit for vertical rhythm (2023)

<https://codepen.io/wuuuuuuuuut/pen/yLrpoaR>

and `text-emphasis!`

(Because why not?!)

<https://codepen.io/wuuuuuuuuut/pen/MWRrvbz>

CSS Counters

- <https://codepen.io/wuuuuuuuuut/pen/PogoOpL>
- <https://codepen.io/wuuuuuuuuut/pen/QWPaMEr>

CSS

Custom

Highlight API

<https://codepen.io/wuuuuuuuuut/pen/QWPQVxr>

Available since 2021

Build coherent form controls with

accent-color

<https://codepen.io/web-dot-dev/pen/PomBZdy>

**HTML features
you might not be
using**

Lazy loaded images for better LCP and ¹

.webp format for faster loading ²

```

```

1. <https://developer.mozilla.org/en-US/docs/Web/HTML/Element/img#loading>
2. <https://en.wikipedia.org/wiki/WebP>

Widely available since 2019...

<datalist> -

**predefined
autocomplete for
forms ¹**

1. <https://developer.mozilla.org/en-US/docs/Web/HTML/Element/datalist>

HTML Demo: <datalist>

RESET

HTML

CSS

OUTPUT

```
1 <label for="ice-cream-choice">Choose a flavor:
  </label>
2 <input list="ice-cream-flavors" id="ice-cream-
  choice" name="ice-cream-choice" />
3
4 <datalist id="ice-cream-flavors">
5   <option value="Chocolate"></option>
6   <option value="Coconut"></option>
7   <option value="Mint"></option>
8   <option value="Strawberry"></option>
9   <option value="Vanilla"></option>
10 </datalist>
11
```

Choose a flavor:

Chocolate

Coconut

Mint

Strawberry

Vanilla

HTML

Play



```
<label for="tick">Tip amount:</label>
<input type="range" list="tickmarks" min="0" max="100" id="tick" name="tick" />
<datalist id="tickmarks">
  <option value="0"></option>
  <option value="10"></option>
  <option value="20"></option>
  <option value="30"></option>
</datalist>
```

Play

Tip amount:




HTML

Play 



```
<label for="colors">Pick a color (preferably a red tone):</label>
<input type="color" list="redColors" id="colors" />
<datalist id="redColors">
  <option value="#800000"></option>
  <option value="#8B0000"></option>
  <option value="#A52A2A"></option>
  <option value="#DC143C"></option>
</datalist>
```

Play 

Pick a color (preferably a red tone):



Inny...

HTML

Play



```
<input type="time" list="popularHours" />
<datalist id="popularHours">
  <option value="12:00"></option>
  <option value="13:00"></option>
  <option value="14:00"></option>
</datalist>
```

Play

-- : -- ⌚

12:00

13:00

14:00

Inny...

R type

Widely available since 2022

Native modals

with

`<dialog>`¹ and `[inert]`²

1. <https://github.com/GoogleChrome/dialog-polyfill>

2. <https://github.com/WICG/inert/blob/master/README.md#use-cases>

Native modal demo

<https://gui-challenges.web.app/dialog/dist/>

Is it moral to use <dialog> though?

- <https://adrianroselli.com/2019/04/details-summary-are-not-insert-control-here.html>
- <https://adrianroselli.com/2023/08/progressively-enhanced-html-accordion.html>
- <https://samthor.au/2021/in-defence-of-dialog/>

This one is also immoral 😅

Exclusive accordion with HTML `<details>`

- <https://codepen.io/web-dot-dev/pen/xxMBVGd>
- <https://developer.chrome.com/docs/css-ui/exclusive-accordion>
- <https://adrianroselli.com/2023/08/progressively-enhanced-html-accordion.html>

**What's coming to
JS?**

**More functional
features** 

TC39 Stage 2

Pipe operator –

easy function

calls



1

1. <https://github.com/tc39/proposal-pipeline-operator>

From this mess...

```
console.log(  
  chalk.dim(  
    ` $ ${Object.keys(envvars)  
      .map(envar =>  
        `${envar}=${envvars[envar]} `)  
      .join(' ')}  
    ` ,  
    'node',  
    args.join(' '))) ;
```

to this 🎉

```
Object.keys(envvars)
  .map(envar =>
    `${envar}=${envvars[envar]}`)
  .join(' ')
|> ` $ ${%} `
|> chalk.dim(
  %, 'node', args.join(' '))
|> console.log(%);
```

TC39 Stage 2

Records and Tuples – immutable data structures in JavaScript, with cheap equality checks ¹

1. <https://github.com/tc39/proposal-record-tuple>

```
const proposal = #{
  id: 1234,
  title: "Record & Tuple proposal",
  contents: `...`,
  // tuples are primitive types so you can put them in records:
  keywords: #["ecma", "tc39", "proposal", "record", "tuple"],
};

// Accessing keys like you would with objects!
proposal.title === "Record & Tuple proposal"
proposal.keywords[1] === "tc39"

// Spread like objects!
const proposal2 = #{
  ...proposal,
  title: "Stage 2: Record & Tuple",
};
proposal2.title === "Stage 2: Record & Tuple"
proposal2.keywords[1] === "tc39"

// Object functions work on Records:
Object.keys(proposal) === ["contents", "id", "keywords", "title"]
```

**With immutable
data structures
React might,
finally, be fast!**

**Future of CSS?
Animating
everything!**

Interop 2024 – almost there!

Entry / exit animations

<https://developer.chrome.com/blog/entry-exit-animations/>

I dislike the fact that you still need to wait for animations to finish before you remove element from DOM. But hey, it's opportunity to use new `.getAnimations()` API

```
Promise.all(  
  elem  
    .getAnimations({ subtree: true })  
    .map((animation) => animation.finished),  
).then(() => elem.remove());
```

Firefox - behind flag

Safari - no implementation

Polyfill available

Scroll Driven Animations

<https://scroll-driven-animations.style/>

<https://github.com/flackr/scroll-timeline>

Firefox, Safari - no implementation. Can't polyfill 🙄

View Transitions

Animated transitions between separate DOM elements 🌟

- <https://http203-playlist.netlify.app/>
- <https://live-transitions.pages.dev/>
- <https://developer.chrome.com/docs/web-platform/view-transitions/>

Future of DOM

API

DOM Parts API! 1

No more JSX?!

```
<template>
  <section>
    <h1 id="name">{{}}</h1>
    Email: <a id="link" href="{{}}">{{}}</a>
  </section>
</template>
```

1. <https://github.com/WICG/webcomponents/blob/gh-pages/proposals/DOM-Parts.md>

`{{}}` is a reference to a specific place in DOM tree

Quick update without traversing the tree

```
<template>
  <section>
    <h1 id="name">{{}}</h1>
    Email: <a id="link" href="{{}}">{{}}</a>
  </section>
</template>
```

HTML Template Instantiation – Angular templates in browser

<https://github.com/WICG/webcomponents/blob/gh-pages/proposals/Template-Instantiation.md>

Quiz time!

<https://tinyurl.com/wdiquiz>



Prince
Heinrich
Suite

133

That's all for now.

Thanks!

a.fricze@pm.me / fricze.com

Chat with me about:

1. New Web Platform features
2. Presentations, meetups, conferences, workshops
3. Jotai and RTK-Query
4. [Music production \(YT: 23carats beats\)](#).
5. Gym
6. [„Certified Web Platform Inspector” t-shirts \(fricze.com/merch\)](http://fricze.com/merch).