

Latest web features

2019



Rokas Brazdžionis

Front end geek

GitHub github.com/dzonatan

Twitter twitter.com/rdzonatan

rokas.brazdzionis@visma.com

Agenda

- 1~3 standards organizations
- 1~8 latest web features
- Most exciting feature poll
- Q&A

W3C[®] World Wide Web Consortium

- Main standards org. for WWW;
- Its members are companies;
- Has dedicated working groups:
CSSWG, SVGWG, WasmWG, ...
- Uses **snapshots** for specification;
- Follows **W3C Process**;



W3C[®] Standard formation process

Editor's Draft [not an official status]

The absolute freshest version of the specification.



Working Draft (WD)

Officially ready to be reviewed by community.



Candidate Recommendation (CR)

Spec is complete. Feedback from implementers.



Proposed Recommendation (PR)

Accepted by W3C Director.



W3C Recommendation (REC)

Accepted by Advisory Committee.

? WHATWG

- Web Hypertext Application Technology Working Group;
- Formed in 2004 due to slow development of **HTML** in **W3C**;
- Mainly focuses on **HTML/DOM**;
- Numerous other standards: **XMLHttpRequest**, **Fetch**, **Fullscreen API**, **Canvas**, ...
- Uses a **Living Standard**;





Technical Committee 39

- Works under **Ecma International**;
- Maintains **ECMAScript*** language;
- Standard called **ECMA-262**;
- Releases specification annually;
- Spec. edition name: **ESYEAR**;
- Its members are companies;
- Operates by using **TC39 Process**.



*JavaScript

TC39 Process

Stage 0 - Strawman

Free form ideas, reviewed in TC39 meetings.



Stage 1 - Proposal

High level API, examples, selected champion.



Stage 2 - Draft

Formal specification of the syntax and semantics.



Stage 3 - Candidate

Specification complete.



Stage 4 - Finished

Two spec-compliant implementations.
Passes acceptance tests.



ECMAScript

5

6

2016+

next

intl

non-standard

compatibility table

🔗 <https://button.flat>

by [kangax](#) & [webbedspace](#) & [zloiroc](#)

Sort by Engine types ▾

Show obsolete platforms ☐

Show unstable platforms ☐

■ V8
 ■ SpiderMonkey
 ■ JavaScriptCore
 ■ Chrome

 • Minor difference (1 point)

 • Small feature (2 points)

 • Large feature (8 points)

| | | Compilers/polyfills | | | | | | | | | |
|---|---|---------------------|---------|---------------------|---------------------|-----------------|------------------------|----------|-------|---------|---------|
| | | 95% | 5% | 49% | 51% | 39% | 47% | 9% | 1% | 44% | 49% |
| Feature name ▶ | | Current browser | Traceur | Babel 6 + core-js 2 | Babel 7 + core-js 2 | Closure 2019.01 | TypeScript + core-js 2 | es7-shim | IE 11 | Edge 17 | Edge 18 |
| 2016 features | | | | | | | | | | | |
| • exponentiation (**) operator 🗨 | ▶ | 3/3 | 2/3 | 3/3 | 3/3 | 3/3 | 2/3 | 0/3 | 0/3 | 3/3 | 3/3 |
| • Array.prototype.includes 🗨 | ▼ | 3/3 | 0/3 | 3/3 | 3/3 | 2/3 | 3/3 | 2/3 | 0/3 | 3/3 | 3/3 |
| Array.prototype.includes 🗨 | Ⓒ | Yes | No | Yes ^[5] | Yes ^[5] | Yes | Yes ^[6] | Yes | No | Yes | Yes |
| Array.prototype.includes is generic | Ⓒ | Yes | No | Yes ^[5] | Yes ^[5] | Yes | Yes ^[6] | Yes | No | Yes | Yes |
| %TypedArray%.prototype.includes 🗨 | Ⓒ | Yes | No | Yes ^[5] | Yes ^[5] | No | Yes ^[6] | No | No | Yes | Yes |



Accessibility
I18n
Graphics
Media
Security
WebAssembly
Privacy

....

XMLHttpRequest
Fullscreen API
Notifications API
Storage
Streams
Fetch
Console

....

▲ ChrisSD 10 months ago | parent | favorite | on: Microsoft, Google, Mozilla, and Apple Object to W3...

It's sad to see the relationship between WHATWG and W3C has deteriorated to this point. Trying to wrangle a standard from a "living" (i.e. constantly changing) specification was always going to be tough but I'd have hoped both WHATWG and W3C would be able to maintain a working relationship.

▲ chrisseaton 10 months ago [-]

Is there an article with the background on this? Why do we have both the W3C and the WHATWG, and why do the W3C just copy and paste work from WHATWG, if that is indeed what happens?

▲ ChrisSD 10 months ago [-]

I don't know of an article, sorry. A brief history from memory would be that during XHTML days the W3C essentially let the HTML spec languish and people weren't moving to XHTML (at best they were moving to XHTML-like HTML).

So the WHATWG came along (mainly organised by the major browser vendors) and started the HTML spec moving again. This became part of what's known as HTML5.

However WHATWG doesn't exactly make a "standard" it makes a "living standard", which is a constantly shifting document which aims to describe where browsers currently are and what they hope to implement. The W3C decided to keep publishing its own HTML specifications and, as the WHATWG does describe what browsers are trying to do, the W3C's spec has to build at least partly on that work. There are differences though. For example, the W3C requires at least two implementations of a feature for it to be included in their spec.

The WHATWG has always opposed the W3C's spec. They see it as confusing to have two "official" specifications.

▲ lucideer 10 months ago [-]

To put a slightly different spin on the same story as perspective always colours the telling:

W3C decided to deprecate HTML in favour of XHTML. Most of the web quickly moved to XHTML. One individual (an employee at Opera, then Mozilla, finally and currently Google) wrote an oddly influential opinion piece saying the the move to XHTML had been somehow harmful and pushed for the major browser vendors to form a rival non-democratic standards body (WHATWG) to the W3C, which forked and completely redefined HTML.

The W3C, which unlike the WHATWG has many voting members from many backgrounds, not all related to browser making, quite understandably was never fully on board with the new WHATWG HTML spec efforts. However, with the level of adoption and support it received (mainly from being the creation of the powerful browser vendors) W3C were eventually pressured into conceding to advocate for HTML. Which they've done by maintaining a copy, rather than blindly directing people to the work by what for all intents and purposes effectively amounts to a rival organisation, and an extremely undemocratic one at that.

As web developers, we should follow the WHATWG and ignore the W3C, because the W3C have lost the political battle for HTML and we need to get our stuff working on browsers, all of whom follow WHATWG. But that's an unfortunately pragmatic approach that shouldn't amount to acceptance.

▲ Sharlin 10 months ago [-]

> Most of the web quickly moved to XHTML.

This simply is not true. The web moved to an XHTML-like dialect of HTML which was still served as text/html and browsers interpreted it as "HTML soup" because actually serving pages as application/xhtml+xml would have broken the majority of the web because browsers would actually validate them and refuse to display a page *at all* if there was even a single missing close tag.

▲ lucideer 10 months ago [-]

> This simply is not true. The web moved to an XHTML-like dialect of HTML

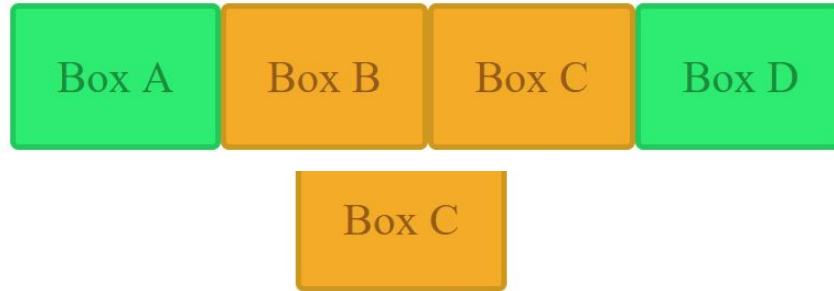
You're thinking of XHTML 1.1 or XHTML 2. That "XHTML-dialect" that everyone switched to was called "XHTML 1.0", which allowed serving as either content type.

display: contents;

```
<div class="container">
  <div class="box">Box A</div>
  <div class="sub-container">
    <div class="box">Box B</div>
    <div class="box">Box C</div>
  </div>
  <div class="box">Box D</div>
</div>
```

```
.container {
  display: flex;
  flex-direction: row;
  align-items: flex-start;
}

.sub-container {
  display: contents;
}
```



Intl.RelativeTimeFormat

```
const rtf = new Intl.RelativeTimeFormat("en", {
  localeMatcher: "best fit", // other values: "lookup"
  numeric: "always", // other values: "auto"
  style: "long", // other values: "short" or "narrow"
});

rtf.format(-1, "day");
// > "1 day ago"

rtf.format(1, "day");
// > "in 1 day"

rtf.format(2, "week");
// > "in 2 weeks"

rtf.format(-1, "month");
// > "1 month ago"
```



```
// locale: en / numeric: auto / style: long
rtf.format(-1, "day");
// > "yesterday"

// locale: en / numeric: always / style: long
rtf.format(1, "hour");
// > "in 1 hour"

// locale: lt / numeric: auto / style: short
rtf.format(-2, "week");
// > "prieš 2 sav."

// locale: ru / numeric: auto / style: narrow
rtf.format(1, "month");
// > "в след. мес."
```


queueMicrotask()

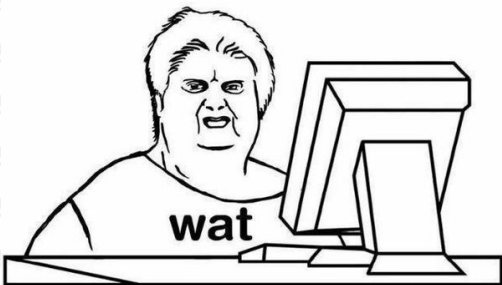


```
setTimeout(() => console.log('A'));  
  
Promise.resolve().then(() => console.log('B'));  
  
console.log('C');  
  
// > "C"  
// > "B"  
// > "A"
```

```

121     private update(): void {
122         if (!this.links || !this.linksWithHrefs || !this.router.navigated) return;
123         Promise.resolve().then(() => {
124             const hasActiveLinks = this.hasActiveLinks();
125             if (this.hasActiveLinks() !== hasActiveLinks) {
126                 this.active = hasActiveLinks;
127                 this.links.forEach((c) => {
128                     this.links.forEach((c) => {
129                         this.addClass(this.element.nativeElement, c);
130                     });
131                     this.renderer.removeClass(this.element.nativeElement, c);

```





```
setTimeout(() => console.log('A'));  
  
queueMicrotask(() => console.log('B'));  
  
console.log('C');  
  
// > "C"  
// > "B"  
// > "A"
```

Resize Observer



```
const ro = new ResizeObserver((changes) => {  
  for (const change of changes) {  
    const { target, contentRect } = change;  
    // ...  
  }  
});  
  
// Observe one or multiple elements  
ro.observe(someElement);
```

Class fields

ES2015


```
class Counter {  
  constructor() {  
    this.count = 0;  
  }  
  
  increment() {  
    this.count++;  
  }  
}
```

Public field


```
class Counter {  
  count = 0;  
  
  increment() {  
    this.count++;  
  }  
}
```

Private field

```
class Counter {  
  #count = 0;  
  
  get count() {  
    return this.#count;  
  }  
  
  increment() {  
    this.#count++;  
  }  
}
```

```
class Counter {  
  #count = 0;  
  
  get count() {  
    return this.#count;  
  }  
  
  increment() {  
    this.#count++;  
  }  
}
```



```
const counter = new Counter();  
counter.#count;  
// > "SyntaxError"
```

RegExp Named Capture Groups



```
const regex = /(\d{4})-(\d{2})-(\d{2})/;  
const result = regex.exec('2015-01-02');  
// result[1] === '2015';  
// result[2] === '01';  
// result[3] === '02';
```



```
const regex = /(?<year>\d{4})-(?<month>\d{2})-(?<day>\d{2})/;  
const result = re.exec('2015-01-02');  
// result.groups.year === '2015';  
// result.groups.month === '01';  
// result.groups.day === '02';
```

prefers-color-scheme



```
/* prefers-color-scheme: no-preference, dark, light */
```

```
@media (prefers-color-scheme: light) {  
  .themed {  
    background: white;  
    color: black;  
  }  
}
```

```
@media (prefers-color-scheme: dark) {  
  .themed {  
    background: black;  
    color: white;  
  }  
}
```

Lorem ipsum

dolor sit amet, consectetur adipiscing elit. Mauris eget odio orci. Pellentesque sed rhoncus augue. Etiam ac luctus elit. Maecenas a pretium augue. Nam sit amet venenatis quam. Vivamus auctor, nunc quis tincidunt pellentesque, nulla nisl pulvinar mauris, at cursus est lacus at leo...

Picture-in-Picture



```
<video id="videoEl" src="https://url.to.a.video"></video>
```



```
// To launch  
await videoEl.requestPictureInPicture();  
  
// To exit  
document.exitPictureInPicture();
```

Join at
[slido.com](https://slido.com/join/5608)
#5608

Google Code Jam

Training Session @Visma

WHAT LANGUAGE DO YOU SPEAK?

Kada?
Kelintā?
Kur?

Kovo 7 dieną, ketvirtadienį

Renkamės 17:30

Giedraičių g. 3, 5 aukštas

```
const _browser_adapter = Adapter} from './generic'

str
name:
element;
in
name
return
: boolean {
name, string}:
Property(element, DomAdapter)); } has
DomAdapter(new Browser
static makeCurrent() { setRoot
ter parse(templateHtml: string)
ter parse(templateHtml: string)
ter parse(templateHtml: string)
extends GenericBrowserDomAda
export class BrowserDomAdapter
```



Thank you

