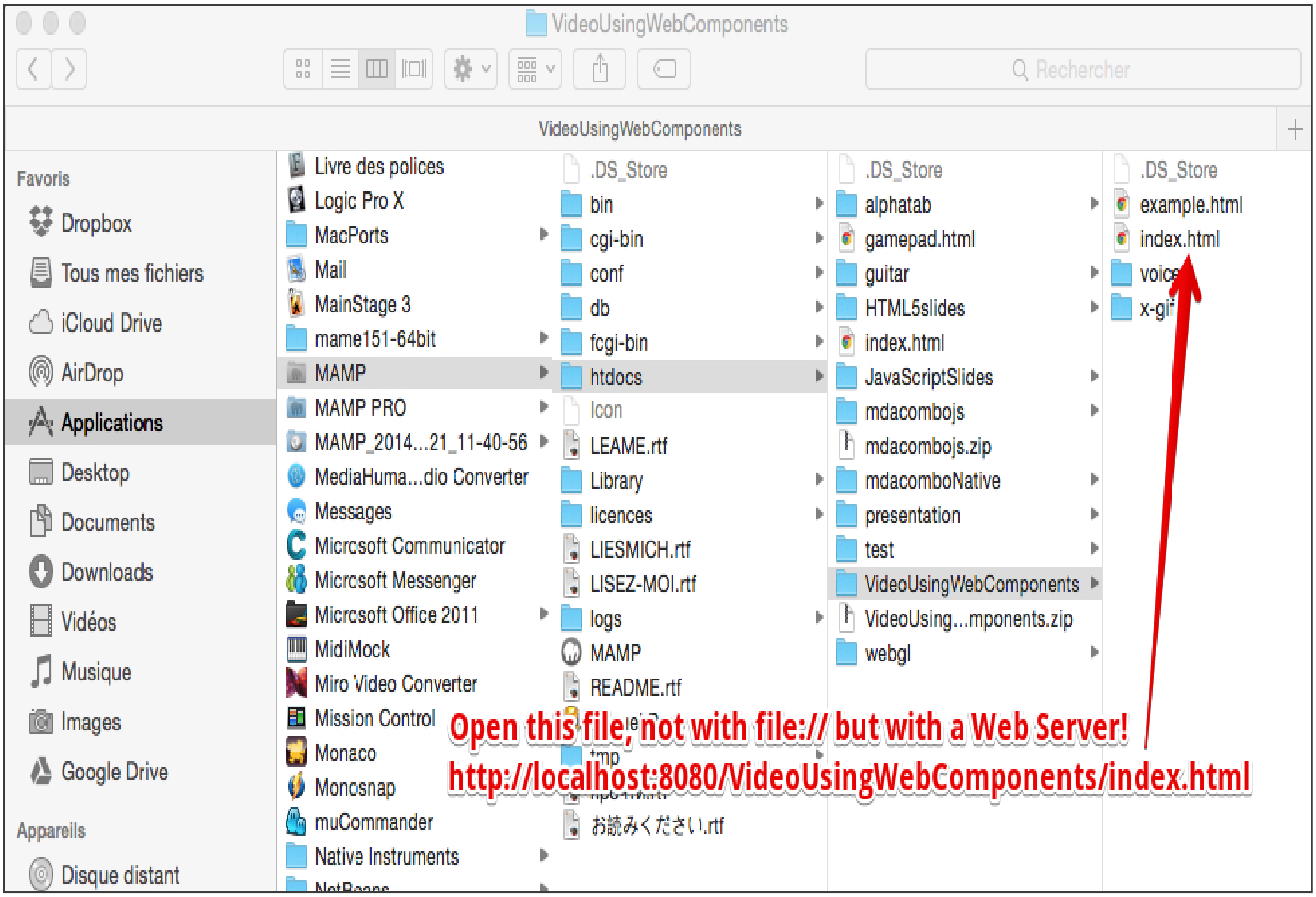
Example from the video

You can download the archive of the example from the video at: VideoUsingWebComponents.zip

You need to unarchive it in your Web server `htdocs` directory of your WAMP/MAMP/LAMP http distribution, for example. Then open the `index.html` file located in the directory.



Open this file, not with file:// but with a Web Server!
http://localhost:8080/VideoUsingWebComponents/index.html

Web components

INTRODUCTION

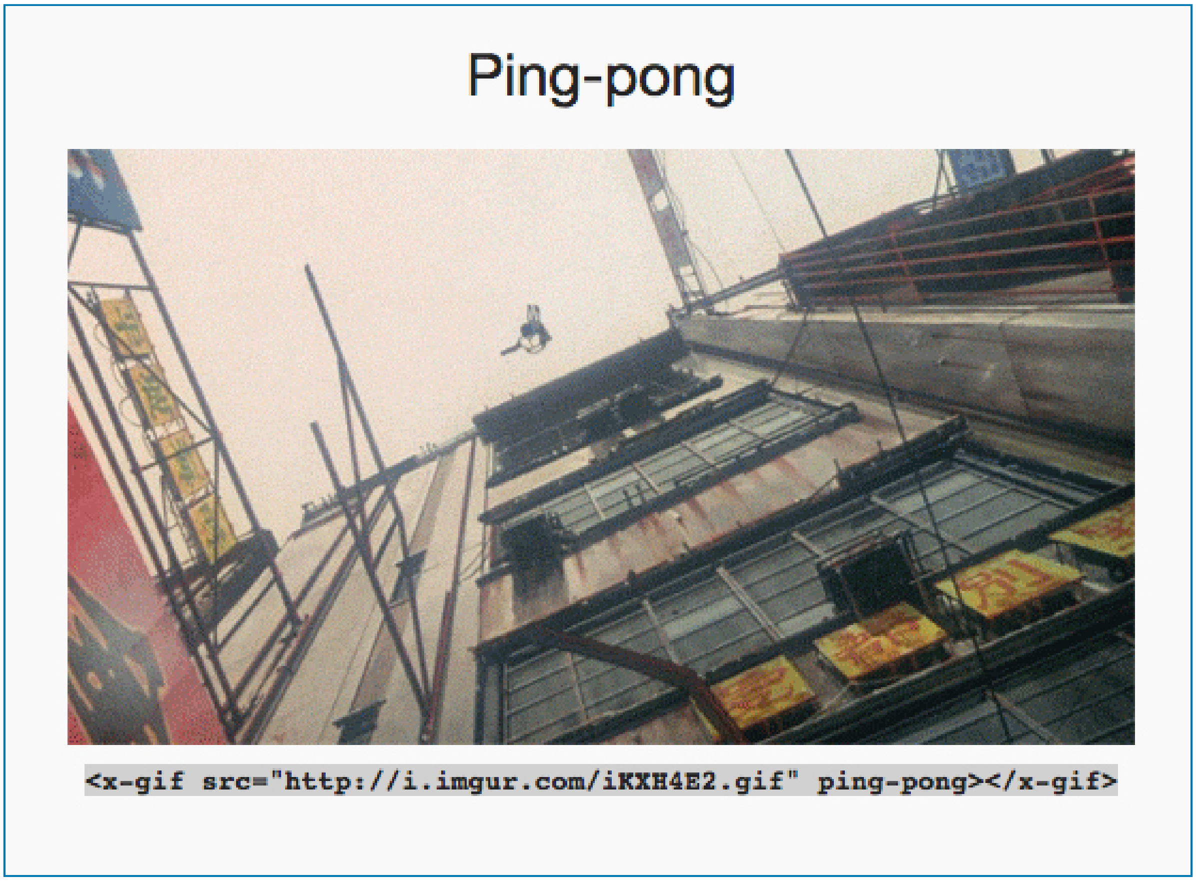
Web components provide a standard way to built your own widgets/components using similar methods to those used by browser developers to build the `<video>`, `<audio>` or `<input type="date">` elements, for example.

Web components enable you *to use custom HTML elements in your HTML documents*, that will render as complex widgets: a better looking calendar, an input text with vocal recognition, a nice chart, etc.

Let's start with an example! This code...:

```
<x-gif src="http://i.imgur.com/iKXH4E2.gif" ping-pong></x-gif>
```

... will render in your document an animated GIF, and it will loop forever in ping-pong mode: the order of the animation will be reversed when the last image is reached and again when the animation goes back to the first image.



Ping-pong

```
<x-gif src="http://i.imgur.com/iKXH4E2.gif" ping-pong></x-gif>
```

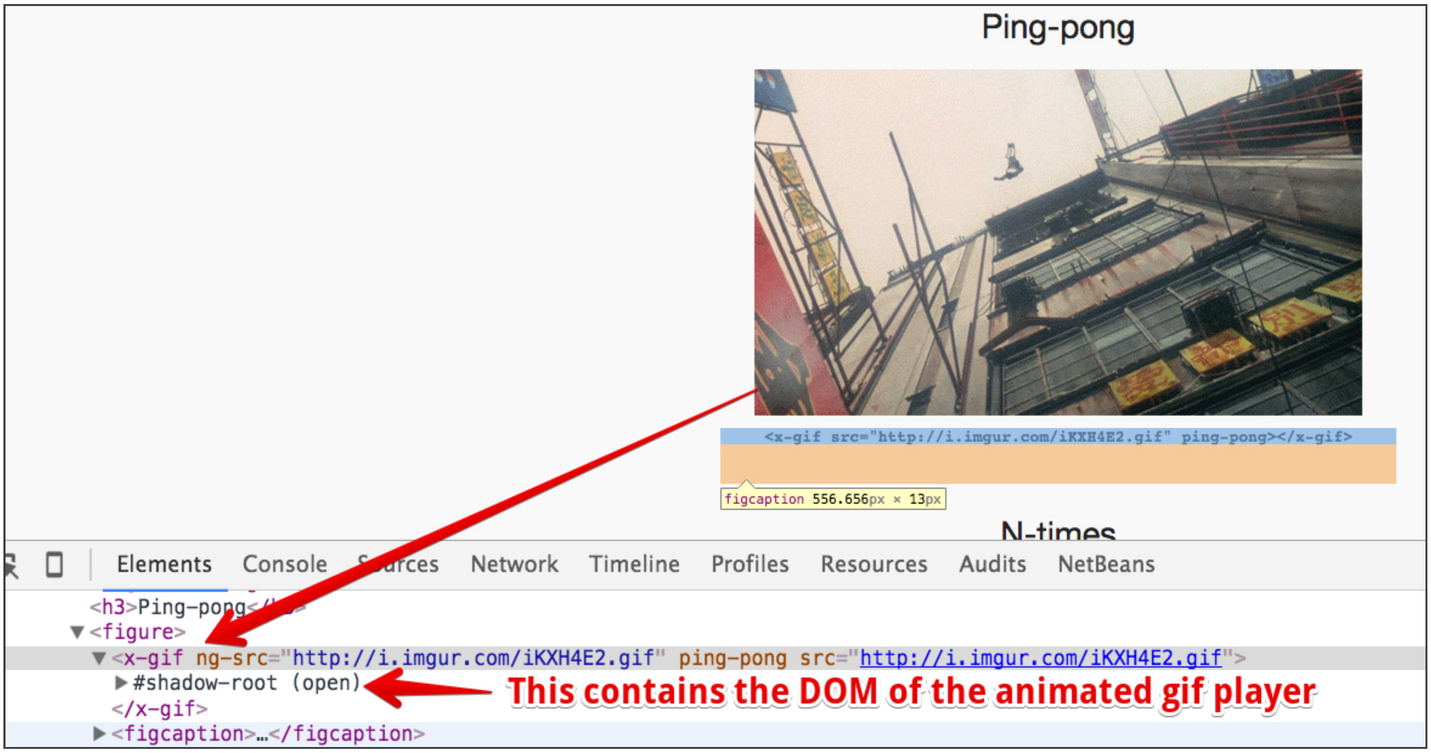
Click the image to see a real demo with the animated GIF playing, or visit this Web site.

If you look at the source of the demo page, you will note the following at the top of the page:

```
<link rel="import" href="dist/x-gif.html">
```

This is new! It's called an "HTML import"! If your browser supports*HTML imports*, you can now import *another HTML document,* that will come with its own HTML, CSS and JavaScript codes, into your HTML page . The code for the animated GIF player, rendered when the browser encounters the *custom HTML element* `<x-gif>`, is located in the imported HTML file (and this HTML file can in turn include or define CSS and JavaScript content).

Even more impressive: if you use the devtools or the right click context menu to view the source of the page, you will not see the DOM of this animated GIF player:


Ping-pong
<x-gif src="http://i.imgur.com/iKXH4E2.gif" ping-pong></x-gif>
figcaption 556.656px × 13px
N-times
Elements
Console
Sources
Network
Timeline
Profiles
Resources
Audits
NetBeans

```
    <h3>Ping-pong</h3>
  ▼<figure>
    ▼<x-gif ng-src="http://i.imgur.com/iKXH4E2.gif" ping-pong src="http://i.imgur.com/iKXH4E2.gif">
      ▶#shadow-root (open)
      </x-gif>
    ▶<figcaption>…</figcaption>
```

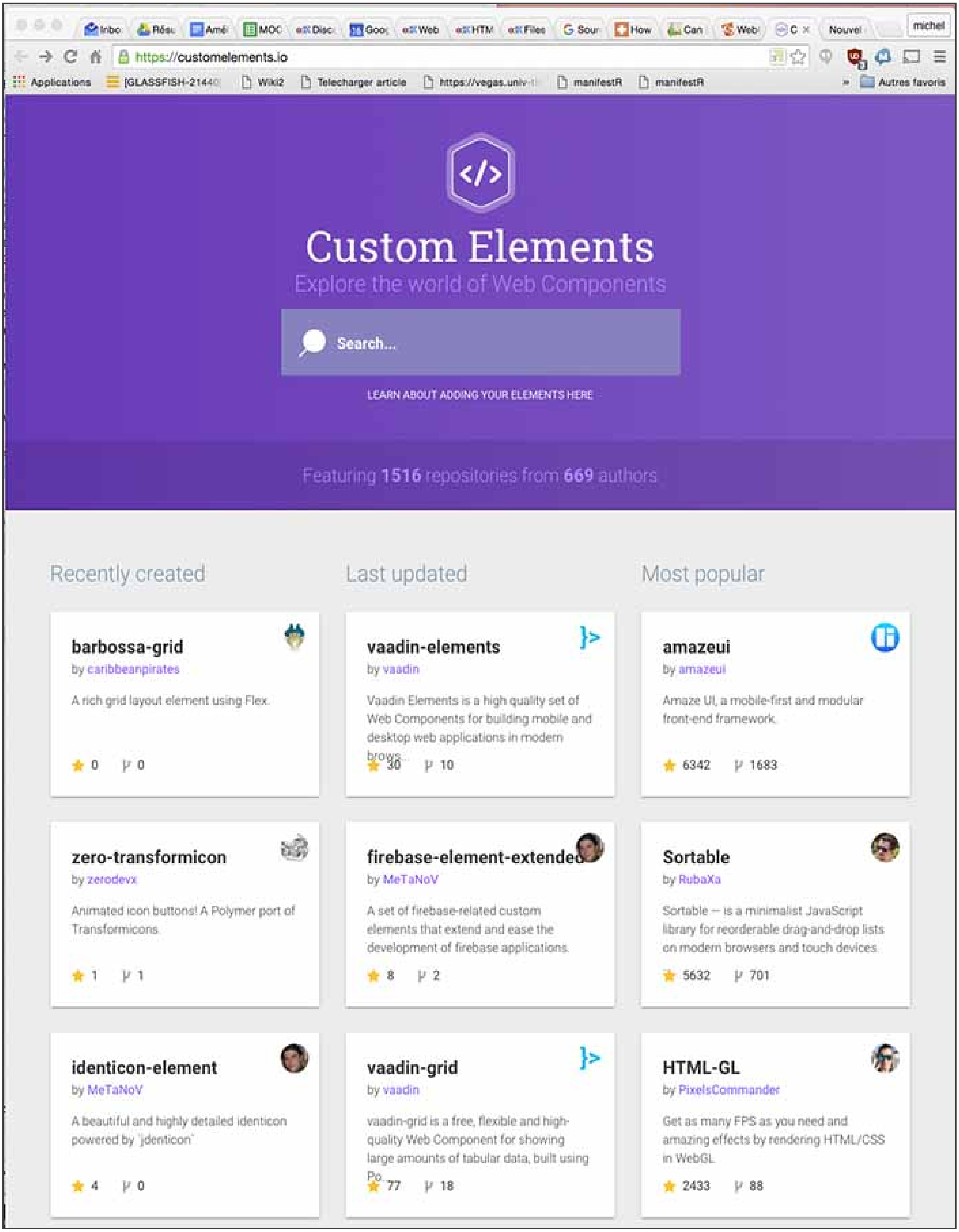
This contains the DOM of the animated gif player

...*and your document will still be valid.* Looking at the source code or at the DOM with the devtool's inspector will not reveal the source code (HTML/JavaScript/CSS) used for creating it.

THERE ARE ALREADY HUNDREDS OF WEB COMPONENTS AVAILABLE

Indeed, you can use many Web components made by others.
The webcomponents.org Web site links to several Web components repositories, such as customelements.io where you will find lots of Web components. Usually, you will need to import the HTML file that defines the components you want to use, and maybe also a polyfill if you want to use them with browsers that do not yet support Web Components.

Example: let's go to the customelements.io home page:



Custom Elements

Explore the world of Web Components

Search...

LEARN ABOUT ADDING YOUR ELEMENTS HERE

Featuring **1516** repositories from **669** authors

Recently created

barbossa-grid
by caribbeanpirates

A rich grid layout element using Flex.

★ 0 ⑂ 0

zero-transformicon
by zerodevx

Animated icon buttons! A Polymer port of Transformicons.

★ 1 ⑂ 1

identicon-element
by MeTaNoV

A beautiful and highly detailed identicon powered by `jdenticon`

★ 4 ⑂ 0

Last updated

vaadin-elements
by vaadin

Vaadin Elements is a high quality set of Web Components for building mobile and desktop web applications in modern brows...

★ 30 ⑂ 10

firebase-element-extended
by MeTaNoV

A set of firebase-related custom elements that extend and ease the development of firebase applications.

★ 8 ⑂ 2

vaadin-grid
by vaadin

vaadin-grid is a free, flexible and high-quality Web Component for showing large amounts of tabular data, built using Po...

★ 77 ⑂ 18

Most popular

amazeui
by amazeui

Amaze UI, a mobile-first and modular front-end framework.

★ 6342 ⑂ 1683

Sortable
by RubaXa

Sortable — is a minimalist JavaScript library for reorderable drag-and-drop lists on modern browsers and touch devices.

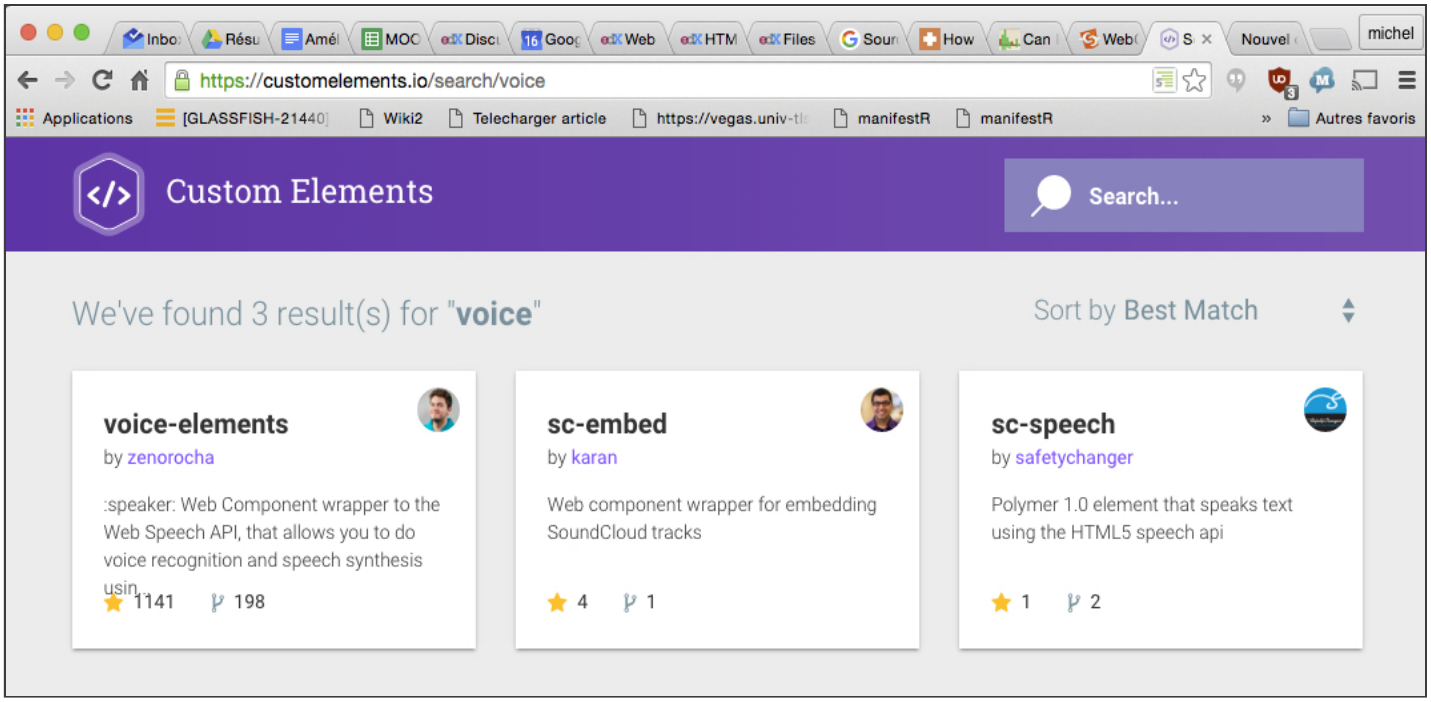
★ 5632 ⑂ 701

HTML-GL
by PixelsCommander

Get as many FPS as you need and amazing effects by rendering HTML/CSS in WebGL

★ 2433 ⑂ 88

We then search for Web components tagged with the "voice" tag and find input fields with voice recognition, and a text area that could vocalize the text:



We've found 3 result(s) for "**voice**"

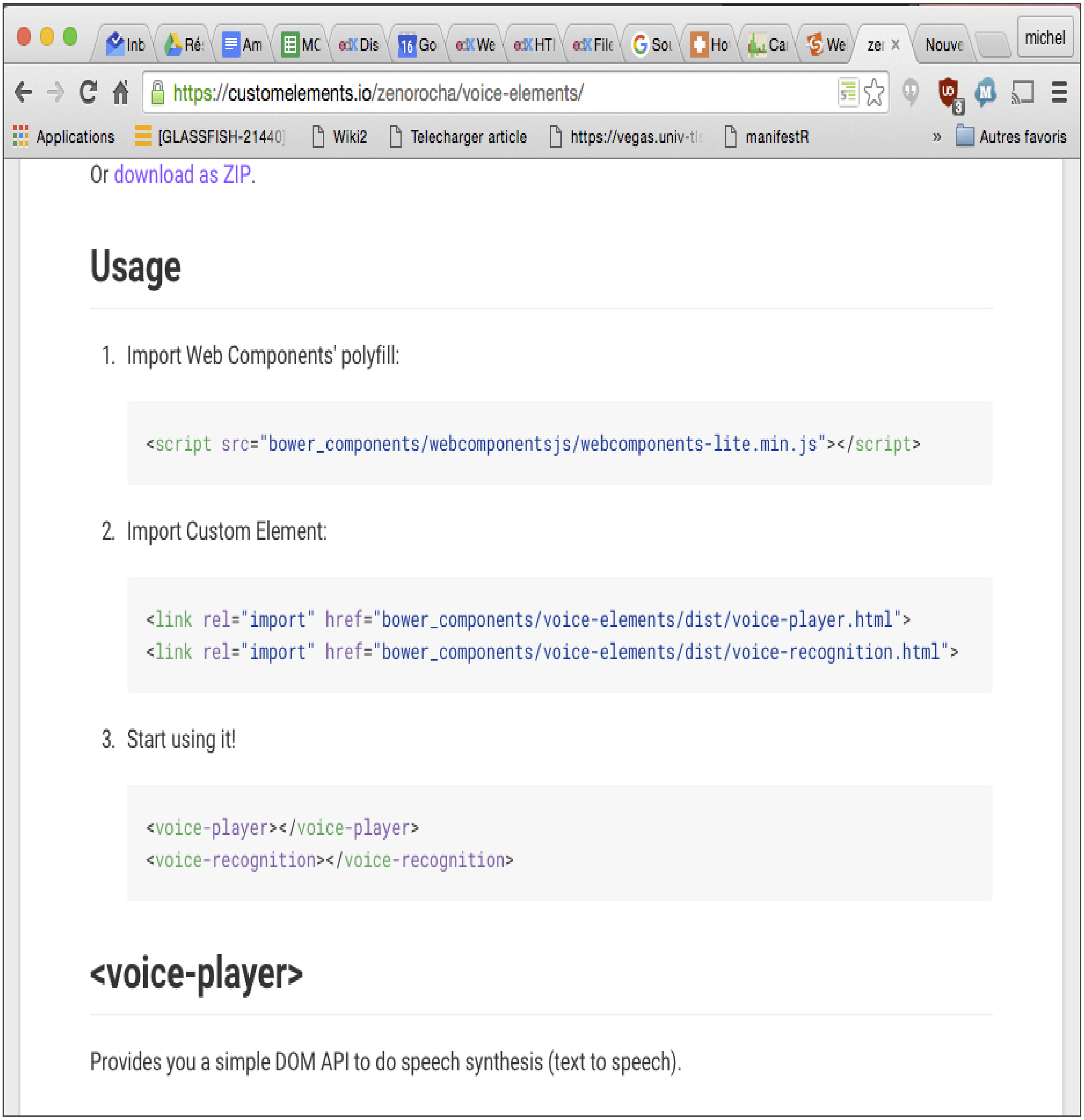
Sort by Best Match

voice-elements
by zenorocha
:speaker: Web Component wrapper to the Web Speech API, that allows you to do voice recognition and speech synthesis usin...
★ 1141 198

sc-embed
by karan
Web component wrapper for embedding SoundCloud tracks
★ 4 1

sc-speech
by safetychanger
Polymer 1.0 element that speaks text using the HTML5 speech api
★ 1 2

Let's click on the first one:



Or download as ZIP.

Usage

1. Import Web Components' polyfill:

```
<script src="bower_components/webcomponentsjs/webcomponents-lite.min.js"></script>
```

2. Import Custom Element:

```
<link rel="import" href="bower_components/voice-elements/dist/voice-player.html">
<link rel="import" href="bower_components/voice-elements/dist/voice-recognition.html">
```

3. Start using it!

```
<voice-player></voice-player>
<voice-recognition></voice-recognition>
```

<voice-player>

Provides you a simple DOM API to do speech synthesis (text to speech).

Now, please try the demonstration of this component!

Also, re-using Web components is easy :-)

Notice that Google with the Polymer project and Mozilla, with the X-Tag project, also offer huge sets of components for creating rich UIs with a common look and feel.

CURRENT SUPPORT

Web components are built on four different APIS

In this lesson, we will talk about "Web components". This is not one single API - rather it's what we call an "umbrella" API, **built on top of four W3C specifications,** that will be detailed in subsequent lessons:

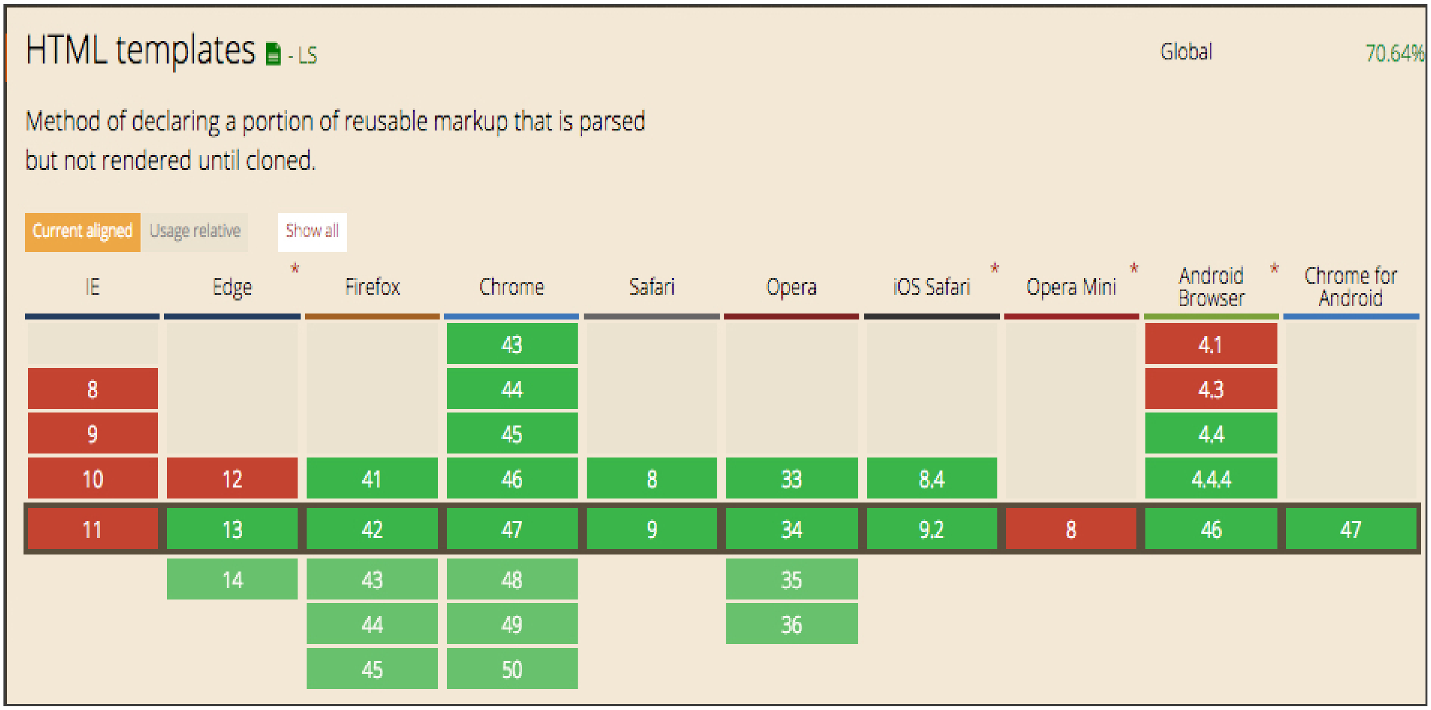
1. The HTML Templates specification (W3C Working Group Note)
2. The Shadow DOM specification (Working Draft)
3. The Custom Elements specification (Working Draft)
4. The HTML Imports specification (Working Draft)

You can check the current support for these APIs here:http://status.modern.ie/ and http://www.caniuse.com.

Currently (as at December 2015), only Google Chrome and Opera natively support these four APIs. Other browsers support only some of them, or have incomplete support. However, polyfills are available, and Web components frameworks, such as Polymer by Google or X-Tags by Mozilla include a polyfill, that adds support for most modern browsers (> 2013).

With a polyfill, Web components can be used in all modern browsers (> 2013)

HTML templates are supported by nearly all modern browsers, including mobile browsers:



HTML templates - LS

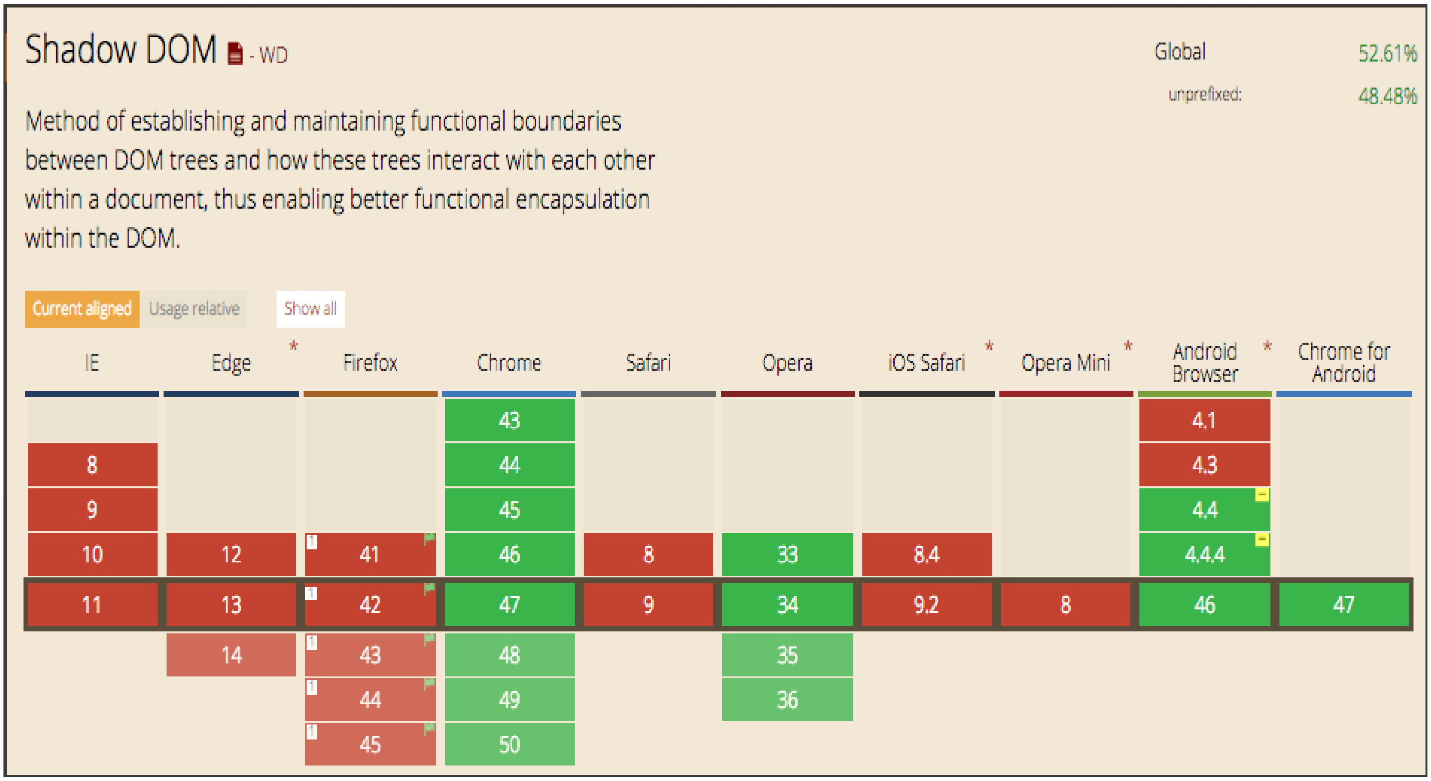
Global 70.64%

Method of declaring a portion of reusable markup that is parsed but not rendered until cloned.

Current aligned | Usage relative | Show all

IE	Edge *	Firefox	Chrome	Safari	Opera	iOS Safari *	Opera Mini *	Android Browser *	Chrome for Android
			43					4.1	
8			44					4.3	
9			45					4.4	
10	12	41	46	8	33	8.4		4.4.4	
11	13	42	47	9	34	9.2	8	46	47
	14	43	48		35				
		44	49		36				
		45	50						

Shadow DOM is supported by Chrome and Opera, and FireFox offers partial support:



Shadow DOM - WD

Global 52.61%

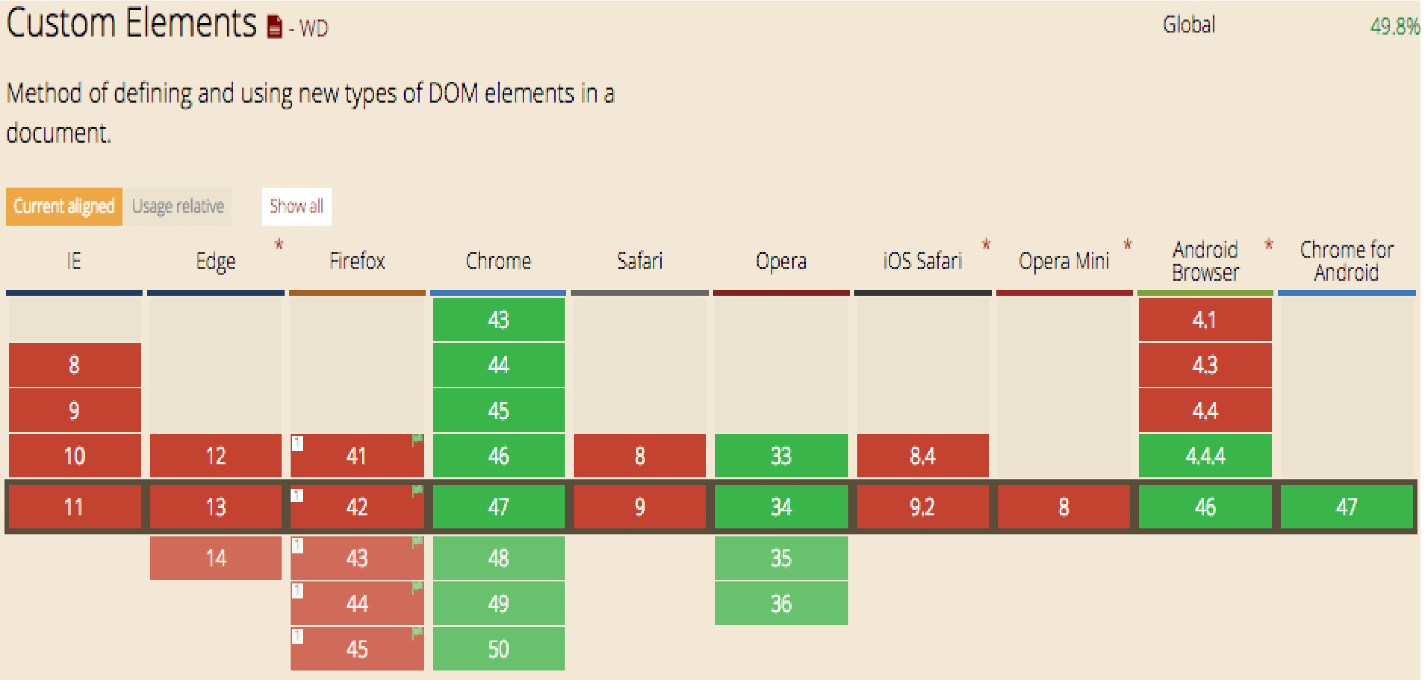
unprefixed: 48.48%

Method of establishing and maintaining functional boundaries between DOM trees and how these trees interact with each other within a document, thus enabling better functional encapsulation within the DOM.

Current aligned | Usage relative | Show all

IE	Edge *	Firefox	Chrome	Safari	Opera	iOS Safari *	Opera Mini *	Android Browser *	Chrome for Android
			43					4.1	
8			44					4.3	
9			45					4.4	
10	12	41	46	8	33	8.4		4.4.4	
11	13	42	47	9	34	9.2	8	46	47
	14	43	48		35				
		44	49		36				
		45	50						

Custom Elements is supported by Chrome and Opera, and FireFox offers partial support:



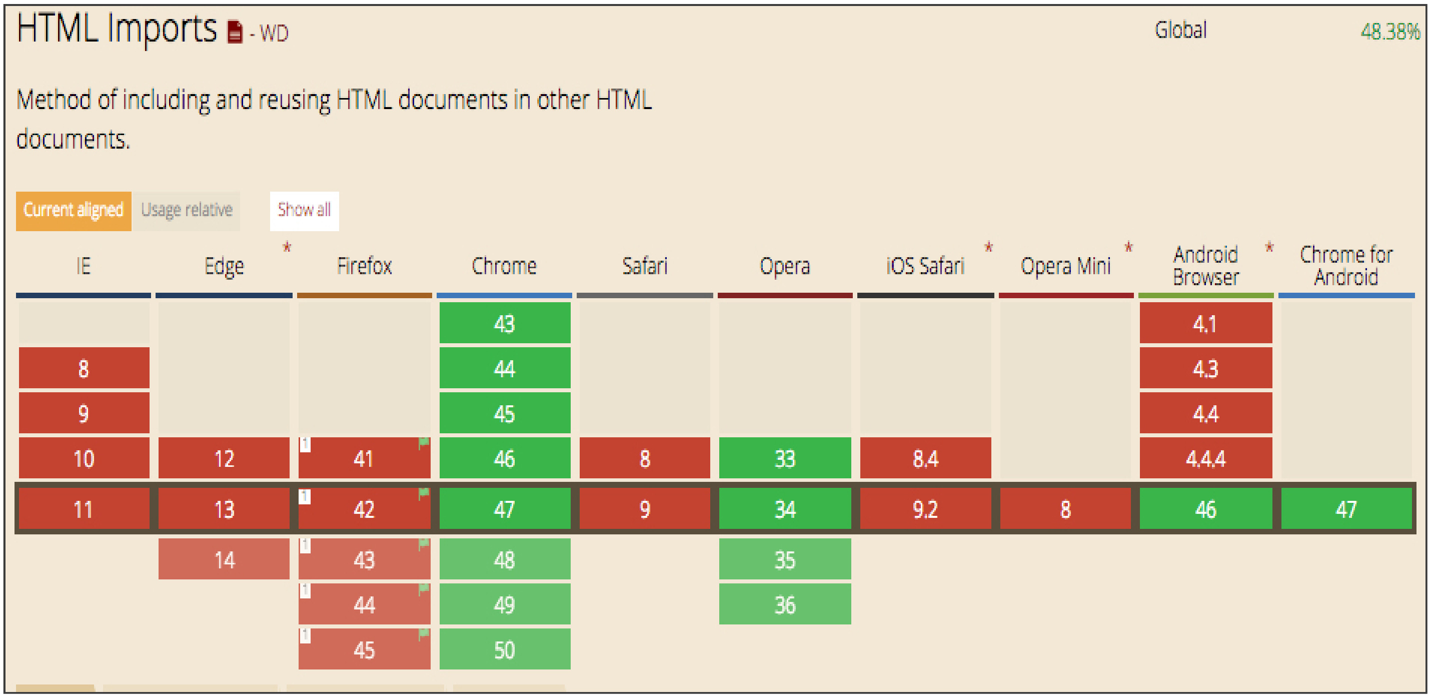
Custom Elements - WD

Global 49.8%

Method of defining and using new types of DOM elements in a document.

IE	Edge *	Firefox	Chrome	Safari	Opera	iOS Safari *	Opera Mini *	Android Browser *	Chrome for Android
			43					4.1	
8			44					4.3	
9			45					4.4	
10	12	41	46	8	33	8.4		4.4.4	
11	13	42	47	9	34	9.2	8	46	47
	14	43	48		35				
		44	49		36				
		45	50						

HTML Imports is supported by Chrome and Opera, and FireFox offers partial support:



HTML Imports - WD

Global 48.38%

Method of including and reusing HTML documents in other HTML documents.

IE	Edge *	Firefox	Chrome	Safari	Opera	iOS Safari *	Opera Mini *	Android Browser *	Chrome for Android
			43					4.1	
8			44					4.3	
9			45					4.4	
10	12	41	46	8	33	8.4		4.4.4	
11	13	42	47	9	34	9.2	8	46	47
	14	43	48		35				
		44	49		36				
		45	50						