

Framework-less frontend development

by Matthias Hryniszak

A long time ago, in a galaxy not far away...



Sch

E





חוקשים
זהר!

احترس

DANGER
MINES!

من

اصل GAM!

Tuesday, January 9th, 2018

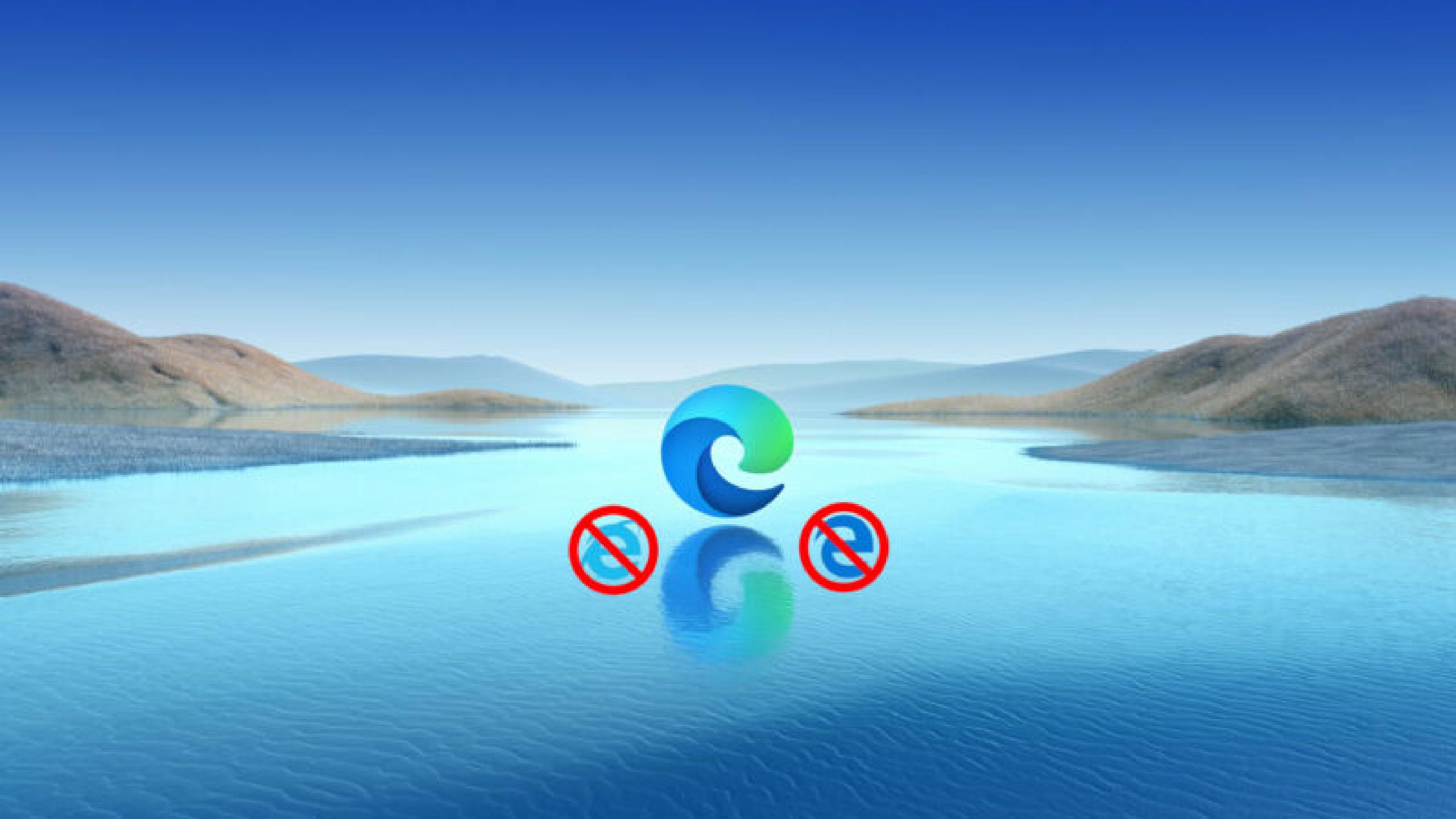
The day the world changed





Tuesday, March 9th, 2021

The day the world changed - again





Let's get a bit more technical

The Basics

HTML/DOM is the representation of concepts

CSS determines how they look like and what is their positioning

JavaScript makes them behave dynamically

In this instalment... JavaScript

JavaScript is no longer the quirky language!

Classes, properties, private fields, immutable references, closures and arrow functions, async functions, modules, object/array destructuring, shorthand object literals, metaprogramming through prototypes and proxies and more!

A fully capable, dynamic, loosely-typed, object-oriented, functional, single-threaded and asynchronous language with prototypal inheritance that rules the world!

...and a huge ecosystem!

1,651,401 packages in public NPM registry

A bit of comparison with alien species (Java, .NET, ...)

411,338 projects in Maven Central

244,792 packages in NuGet

173,485 RubyGems

383,596 Go packages

370,940 PyPI packages

38,436 CPAN (Perl)

1,622,587 combined vs 1,651,401 npm



The only language that runs the same runtime everywhere

...even though Microsoft would like to run .NET apps in the browser too
(Silverlight) just like Sun (and now Oracle) also wanted to make their runtime available in the browser (Java Applets)



JavaScript gave birth to JSON

<xml />

<xml />

<xml />

<xml /><xml />

<xml />

With great power comes great responsibility

With great power comes great responsibility

Learn object-oriented programming and principles and when to use them

With great power comes great responsibility

Learn functional programming and principles and when to use them

With great power comes great responsibility

Learn the APIs (e.g. DOM, `fetch`) instead of reaching for NPM

With great power comes great responsibility

Learn design patterns and SOLID principles

With great power comes great responsibility

Learn refactoring techniques

With great power comes great responsibility

Learn to think like a software programmer - not just web designer

Demo time!

Fun examples

<https://ckeditor.com/blog/implementing-single-file-web-components/>

Final thoughts

When in doubt see <http://caniuse.com>

Prefer polyfills over frameworks

Use frameworks that work with the platform*

Questions?

May the platform be with you!

Blog:

<https://padcom13.blogspot.com>

LinkedIn:

<https://linkedin.com/in/padcom>

This presentation:

<https://bit.ly/2Q5hE7z>