

riganti

 Update
NOW 2020



Nebojte se JavaScriptu (s TypeScriptem)

Tomáš Herceg

CEO @ RIGANTI

Co-founder of Update Conference

Microsoft MVP

tomas.herceg@riganti.cz

[@hercegtomas](https://twitter.com/hercegtomas)

www.tomasherceg.com/blog



Proč se nebát

- JS má špatnou pověst kvůli několika „neintuitivním“ quirkům
- == VS ===
 - <https://dorey.github.io/JavaScript-Equality-Table/>
- Číslování měsíců v Date od nuly
- Automatické doplňování středníků
 - ```
return
 1 + 1;
```
- Je to moderní jazyk se spoustou hezkých funkcí



# Trocha historie

- ECMAScript 1 – 1997
- ECMAScript 2 – 1998
- ECMAScript 3 – 1999
- ... *dlouho nic* ...
- ECMAScript 5 – 2009
- ... *dlouho nic* ...
- ECMAScript 6 – 2015
- Nyní nová verze každý rok

Moment, kde je verze 4? 🙄



# Základní zvířátka

- null
- undefined
- boolean
- number
- string
- object
- symbol



# Undefined

- Rozdíl undefined vs undeclared
- `if (typeof something === "undefined")`
- ~~`undefined = true;`~~
- `void 0`



# Pole

- Vlastnost `length`
  - Poslední prvek, na který se přistoupilo
  - Lze nastavit
- Rozdíl mezi prázdnou buňkou a buňkou, ve které je `undefined`
- `for (item of array)` // nově v ES6



# Stringy

- Pozor na `substr` vs `substring`
- Není rozdíl mezi uvozovkami a apostrofy
- Nově v ES2015
  - ``` tohle umí
  - konce řádků i dosazování: `${expr}``



Co bylo dřív? Slepice nebo vejce?

```
["0", "🐔"].sort()
```





# Čísla

- Dle standardu IEEE 754
  - Stejně jako `double` v .NETu
- `Number.EPSILON`
  - Tolerance při porovnávání: `0.1 + 0.2` nejsou přesně `0.3`
- `Number.MAX_SAFE_INTEGER`
- `num | 0`
  - Zarovnání na 32 bitů
- Rozdíl mezi `isNaN` a `Number.isNaN`



# Objekty

- `obj["test"]` a `obj.test` je to samé

```
var obj = {
 a: 10,
 b,
 fn(b) { },
 ["prop_" + name]: val,
 ...obj2
}
```



# Objekty

- `"a" in obj`
- `for (key in obj)`
- `Object.keys`
- `Object.values`
- `Object.defineProperty`



# Funkce

- `function myFn() { }`
- `var myFn = function() { }`
- `var myFn = () => { };`
  
- arguments
- Spread operátor
  - `function myFn(arg1, arg2, ...otherArgs) { }`



# Jak je to s tím `this`

- Lexical vs Runtime scope

```
var a = 10;
function test() {
 return a;
}
```

Nezáleží na tom, kde je funkce deklarována, ale jak je volána

```
obj.test(); // obj
test(); // global scope
 // nebo undefined
```

```
// co si tam dáte
test.apply(a, ...);
test.call(a, ...);
```



# var, let, const

- var je function-scoped, ne block-scoped
- Hoisting
  - Deklarace (ne přiřazení) se přesune na začátek bloku
- let je block-scoped
- const je block-scoped a nelze přiřadit opakovaně



# Promise

- Kombinace Task a TaskCompletionSource
- Zavoláte resolve nebo reject

```
new Promise(function (resolve, reject) { ... })
 .then(success_func, error_func);
```

- Lze použít async / await



# Moduly

- Zamezují zneužívání globálního scope
- `export function, const, ...`
- `import { věc } from './modul.mjs'`
- `import * as věc from './modul.mjs'`
- `export default ...`
- `import věc from './modul.mjs'`





# Moduly – co s tím v browseru

- `<script type=module>`
- Import začne modul stahovat
- Lze použít vlastní loader
  
- Bundling – webpack, rollup, esbuild
  
- Pozor na stav v modulu – je to singleton



# Třídy

- Oproti C# skoro nejsou potřeba
- Moduly = „singleton služby“
- DTO (Data Transfer Object) = prostý JS objekt
- Co zbývá?
  - Objekty, které mají stav i logiku



# Původní ambice

- Přinést typový systém podobný C#
  - Anders Hejlsberg
- Přidávání vlastních features jazyka
  - Vlastní řešení modulů (tzv. Internal Modules)
    - Podpora CommonJS, AMD, UMD a dalších řešení
  - Namespaces
- Na začátku snaha o dopřednou kompatibilitu s ECMAScript 6
  - Standard se v průběhu času dost měnil a vyvíjel



# Jak začít

- TypeScript je nadmnožina ECMAScriptu
  - JavaScript je zároveň validní TypeScript
  - TypeScript bude možná házet warningy, ale vyprodukuje výstup
- Stačí postupně přidávat typové anotace
  - Type inference je velmi chytrá a domyslí si spoustu věcí sama



# Ekosystém

- Pro běžně používané knihovny existují typové deklarace
  - Soubory s příponou `.d.ts`
  - Lze instalovat pomocí npm
    - `npm install @types/react --save-dev`
  - Obrovská knihovna
    - <https://github.com/DefinitelyTyped/DefinitelyTyped/>
- Velmi dobrá podpora v IDE a editorech
  - Populární např. Visual Studio Code



# TypeScript

- Typové anotace
  - `var age: number = 15;`
- Typová inference

```
interface User {
 id: number;
 name: string;
}
```



# TypeScript

```
type WindowStates = "open" | "closed" | "minimized";
```

```
type Options = "recommended" | { prop: value ... };
```

```
type PostbackError =
 { type: "network", err: NetworkError } |
 { type: "serverError", status: number, ... };
```



# Síla typového systému

- Typový systém podporuje mnoho zajímavých funkcí
  - Generické typy včetně constraintů
  - Union typy – `type1 | type2`
  - Discriminated union
    - `{ "type": "circle", radius: number } | { "type": "rectangle", width: number, height: number }`
  - Intersection typy – `type1 & type2`
  - Literal typy – `"one" | "two"`
  - Enums
  - `keyof`





# DEMO

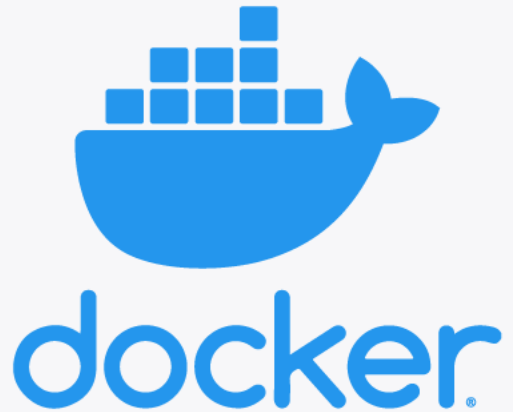
TypeScript v DotVVM

# 4 DNY ŠKOLENÍ

## Docker a Kubernetes v Azure



kubernetes



PRAHA  
BRNO  
PRAHA

od 19. září 2022  
od 20. října 2022  
od 24. listopadu 2022





Update Days  
Frontend 2022

# Neztraťte se ve frontend světě

29. - 30. září 2022

Brick House, Ostrava / Online

[frontend.updatedays.cz](https://frontend.updatedays.cz)





# Update Conference Prague 2022

**.NET - Cloud - Security**

**10. - 11. listopadu 2022**

Praha / Online

[updateconference.net](https://updateconference.net)



riganti

 Update  
NOW 2020



Q&A

**Tomáš Herceg**

CEO @ RIGANTI

Co-founder of Update Conference

Microsoft MVP

[tomas.herceg@riganti.cz](mailto:tomas.herceg@riganti.cz)

[@hercegtomas](https://twitter.com/hercegtomas)

Microsoft takes .NET  
Open Source

November 12<sup>th</sup> 2014