

JavaScript

Cvičení 4

Webová kartografie – úvod

Podzim 2022

Filip Leitner

JAVASCRIPT

- umožňuje tvorbu **dynamických** webových stránek
- **objektově** orientovaný programovací jazyk (OOP)
- **standardizace** - ECMAScript
 - budeme se učit aktuální JavaScript
 - dlouho **ES5**, nyní **ES6**, po novom podľa roku ES2021
 - ... ES.Next
- lze použít jak na **klientovi** (v prohlížeči), tak na serveru
- **loosely typed** - proměnné nemají striktně daný datový typ, je možné jejich typ měnit
- JavaScript *nemá nic společného* s Javou :-)



JavaScript
The Definitive Guide

David Flanagan

O'REILLY



JavaScript:
The Good Parts

Douglas Crockford

O'REILLY | YAHOO! PRESS

PROČ?

- interakce:

- uživatel ↔ *prohlížeč*

- validace formulářů, vizualizace, animace, ...

- *prohlížeč* ↔ server

- dotazování a odesílání dat prostřednictvím internetových protokolů
(nejčastěji HTTP)

- **webové mapy**

PROČ NE?

- **ne na všechno** je potřeba JavaScript!
- řada vizuálních efektů, animací, etc. se dá vytvořit pomocí **CSS** i JavaScriptu
- „Když můžu použít JavaScript nebo CSS - kdy mám použít JavaScript?“ (téměř) **Nikdy.**

[https://hackernoon.com/
in-simple-terms-css-vs-javascript-abc9d709399d](https://hackernoon.com/in-simple-terms-css-vs-javascript-abc9d709399d)

JAK?

- na JavaScript odkazujeme v HTML dokumentu:

```
<script src="js/main.js"></script>  
<script src="https://example.com/js/main.js"></script>  
<script>  
  "use strict"; console.log("Hello  
  world!");  
</script>
```

- v elementu
<head>:

```
<!DOCTYPE html>  
<html>  
<head>  
  <script src="js/main.js"></script> <!-- tady -->  
</head>  
<body>  
...  
</body>  
</html>
```

- načítání skriptů zastavuje načítání stránky
- pokud načítáme objemné skripty, můžeme použít atributy `async` a `defer` viz <https://javascript.info/script-async-defer>

DEKLARACE PROMĚNNÝCH

- **klíčová slova `let`, `const` (dříve `var`)**
 - **`const`** používáme pokud předpokládáme, že se hodnota konstanty nebude měnit, konvence je psát názvy konstant velkými písmeny (`SPEED_OF_LIGHT`, `USER_ID`, `GENDER`, `BLOOD_TYPE`)
 - **`let`** používáme pokud víme, že se hodnota proměnné měnit bude, doporučuji názvy zapisovat v camelCase (`userActivities`, `name`, `age`) nebo spodtržítky (`user_activities`, `name`, `age`)
- **rozdělují se velká a malá písmena**
- **nepoužívá se diakritika**
- **názvy proměnných mohou:**
 - začínat písmenem, `$`, `_`
 - obsahovat písmena, čísla, `$`, `_`
 - `^[a-zA-Z_$][a-zA-Z0-9_$]*$`


```
let name = "John";  
let surname = "Doe";  
let age = 28;  
let hobbies = ["TV shows", "getting murdered"];  
let married = false;  
const GENDER =  
"male";  
let causeOfDeath = undefined;
```

DATOVÉ TYPY

- čísla
 - **strings** - textové řetězce
 - **booleans** - pravdivostní hodnoty
 - **Objekty**
 - null
 - undefined
-
- seznamy (array)
 - funkce
 - Date (Dátum), RegEx, Error...
...Map , Set

primitíva

objekty

```
let speedLimit = 90;
```

```
let array = [1,2,3,4,5,6];
```

```
array[1] + array[2] = 5 // index začíná od 0
```

```
let [lat, lon] = [49.23, 16.4]; // Destructuring assignment
```

```
let message = "Welcome";
```

```
let warning = "The speed limit is " + speedLimit;
```

```
typeof warning === "string"; // returns true
```

```
let warn = currentSpeed > speedLimit; //boolean
```

```
let carModel = null;
```

```
let activity = { // object
```

```
  type: "run",
```

```
  distance: 5632, // meters
```

```
  duration: 1412, // seconds
```

```
  elevation: 273 // meters
```

```
}
```

```
activity.distance //5632
```

```
activity["elevation"] //1412
```

JAVASCRIPT

- <https://www.freecodecamp.org/news/content/images/2019/07/best-js-meme-to-date-2.png>
- <https://stackoverflow.com/questions/7615214/in-javascript-why-is-0-equal-to-false-but-when-tested-by-if-it-is-not-fals>

”USE STRICT”

Zbavuje JavaScript niektorých nástrah - upozorní na chyby

- Disallows global variables. (Catches missing var declarations and typos in variable names)
- Silent failing assignments will throw error in strict mode (assigning `NaN = 5;`)
- Attempts to delete undeletable properties will throw (delete `Object.prototype`)
- Requires all property names in an object literal to be unique (`var x = x1: "1", x1: "2"`)
- Function parameter names must be unique (function `sum (x, x) ...`)
- Forbids octal syntax (`var x = 023;` some devs assume wrongly that a preceding zero does nothing to change the number.)
- Forbids the `with` keyword
- Forbids deleting plain names (`delete x;`)

JAK NA JAVASCRIPT

- 1 V nové složce (např. ukol-01) vytvořte soubory index.html, script.js (index.html můžete zkopírovat z jiného příkladu / svého webu).
- 2 V index.html připojte script.js pomocí **relativní cesty** k souboru:
`<script src="script.js"></script>`
- 3 script.js začněte výrazem **"use strict"**; a zkuste cokoliv vypsát do konzole:

```
"use strict";  
console.log("^\_( _o)_/");
```

- 4 Otevřete index.html v prohlížeči dvoklíkem na soubor ve správci souborů.
- 5 Otevřete vývojářské nástroje - F12 v Chrome a Firefoxu, Option-Command-I v Safari (možná bude potřeba povolit v nastavení).
- 6 Ve vývojářských nástrojích otevřete kartu Console a uvidíte výstup vašich console.log(); příkazů.
- 7 Pro změny ve skriptu:
 - Upravte kód ve script.js.
 - Uložte script.js.
 - Přepněte se do prohlížeče.
 - Klávesou F5 obnovte stránku (není potřeba zavírat záložku a otevírat znovu index.html)

WEBSERVER

- pro webový vývoj se hodí používat webserver
- **VS Code rozšíření Live Server (ritwickdey.liveserver)**
- Případne:
 - NPM knižnica `http-server`

```
http-server -p 8000
```
 - pip knižnica `http-server`

```
python -m http.server 8000
```

PROCVIČOVÁNÍ 1

- Vypište do konzole ASCII art `"_(_o)_/"` alebo
"((...))"
"(O O)"
" \ /" "
(" _ ")

Kvůli některým znakům je potřeba upravit deklaraci

- `This is a 'string'.` `'This is a`
 - `"string".` `"This is a`
 - `\ "string\".`
 - `"We need to escape backslashes - \."`

Použijete: `console.log();`



OPERÁTORY

- standardní aritmetické operace (+, -, *, /, %)
- provádání hodnot (>, >=, <, <=, ===, !==)
 - **používejte** === a !==
porovnává hodnoty stejných datových typů
 - **nepoužívejte** ==, != (pokud nevíte co děláte)
"0" == 0 → vrátí true, často má ale [nepředvídatelné chování](#), proto pro porovnání hodnot používáme první variantu a hodnoty případně převedeme do stejného datového typu pomocí `parseInt()`; `parseFloat()`; `String()`;
- logické operátory
 - && AND
 - || OR
 - ! NOT

OPERÁTOR

```
let speedKmph = distanceM / durationS / 3.6
let isEven = 90 % 2 === 0;
console.log(activity.type !== "bike");
let msg = "Your speed is " + speedKmph;
let message = `Your speed is ${speedKmph}`; //Template literal
```

OPERÁTOR

...ďalšie operátory

- **Addition assignment (+=)**

```
let a = 2;  
let b = 'hello';  
console.log(a += 3); // addition  
// expected output: 5
```

- **Nullish coalescing operator (??)**

```
let foo = null ?? 'default string';  
console.log(foo)  
// expected output: 'default string'
```

- ... <https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Operators>
- **Logical nullish assignment (??=)**
- **Logical AND assignment (&&=)**

PROCVIČOVÁNÍ 2

- Vypište do konzole obsah kruhu o průměru který si zvolíte.
- **Použijete:**
 - `let`
 - `const`
 - aritmetické operace `*`,
<https://javascript.info/operators#exponentiation>
 - `console.log()`;

```
const PI = 3.14; // nebo také Math.PI  
let diameter = ..;
```

PODMÍNKY

```
const BIRTH_YEAR = 1992;
const YEAR = new Date().getFullYear();
if (YEAR - BIRTH_YEAR >= 18) {
  console.log("The user is eligible to drive a car.");
} else {
  alert("You cannot get a driver's license.");
}
// nebo pomocí ternárního operátoru:
const msg = YEAR - BIRTH_YEAR >= 18 ? "User not old enough"
  : "User old enough";
```

PROCVIČOVÁNÍ 3

- **Pokračujte** v kódu z procvičování 2
- Pomocí **podmínky** `if` (`()`) `{}` vypište do konzole zprávu jestli je obsah kruhu větší nebo menší než 30.
- **Použijete:**
 - podmínku `if`, `else`
 - logické operátory `<`, `>`, `<=`, `>=`

```
// vypíše → "Obsah kruhu je větší než 30."  
// nebo → "Obsah kruhu je menší než 30."
```

SWITCH

```
let running = [],
    cycling = [],
    others = [];
switch(activity.type) {
  case "run":
    running.push(activity);
    break;
  case "bike":
    cycling.push(activity);
    break;
  default:
    others.push(activity);
}
```

SMYČKY

```
const LETTERS = ['A', 'B', 'C', 'D'];  
for (let i=0; i < LETTERS.length; i += 1) {  
  console.log(i, ':', LETTERS[i]);  
}
```

```
let i = 0;  
while (i < LETTERS.length) {  
  console.log(i, ':', LETTERS[i]); i +=  
  1; // nebo také i++;  
}
```

```
// nebo druhá varianta for:  
for (const LETTER of LETTERS) {  
  console.log(calculateArea(POLYGON));  
}
```

// dále také for..in a .forEach() → vyzkoušejte :)

SMYČKY

```
const LETTERS = ['A', 'B', 'C', 'D'];

for (const letter in LETTERS){
  console.log(letter) // „0“;„1“ ;„2“; „3“
  console.log(LETTERS[letter]) // „A“ ...
}

for (const letter of LETTERS){
  console.log(letter) // „A“ ...
}

//Aj s INDEXOM
for ([index,letter] of LETTERS.entries()){
  console.log(index,letter) // „0, A“ ...
}
```

FUNKCE

```
function welcome() {  
  alert("Welcome user!");  
}  
welcome();
```

```
const welcome = function () {  
  alert("Welcome user!");  
}
```

```
const welcome = () => {  
  alert("Welcome user!");  
}
```

```
function welcome(name) {  
  alert("Welcome, " + name + "!");  
}
```

```
welcome("Jane");  
welcome("John");
```

```
function isEven(x) {  
  return x % 2 === 0;  
}
```

```
const isResultEven = isEven(4) //true
```

```
console.log(isEven(5));  
console.log(isEven(50));
```

```
function currentYear() {  
    return new Date().getFullYear();  
}  
  
console.log(currentYear());
```

PROCVIČOVÁNÍ 4

- Vytvořte funkci `isLeapYear`, která vrátí (return) jestli je argument `YEAR` přestupný rok nebo ne
- Můžete použít:
 - logické a matematické operátory
 - podmínky `if`, `else`
- Bonus: pokud není zadán argument `YEAR`, použijte aktuální rok

TIPY

- v názvech souborů a cestách k nim **nepoužívejte** diakritiku, mezery a velká písmena
- používejte **angličtinu** v kódu a názvech souborů
- **odsazujte** kód
- používejte **zvýraznění syntaxe** v textovém editoru
- na prvním řádku **každého** skriptu uvádějte **"use strict"**; řada chybných zápisů vyvolá chybovou hlášku místo tichého selhání - **snadnější debugging**
- naučte se používat **vývojářské nástroje** (F12, developer tools, konzole) a debugovat kód

ÚKOL

- vypracujte procvičování 1-4 a následující 3 (až 4) části úkolu
- odevzdávejte buďto samostatný web (index.html, script.js)

nebo jako novou podstránku vašeho webu ← pokud chcete feedback na úpravy webu samotného!

- odevzdat do 18. 10.
- max 5 b., za 4. část bod navíc
- odevzdávat v archivu (web.zip)

ČÁST 1

- Vytvořte funkci `cityPop()`, která vrátí náhodné číslo v rozmezí 10 000–1 000 000
- **Použijete:**
 - `function`
 - aritmetické operace `*`, `+`, `-`
 - `Math.random()`

```
function cityPop() {  
    ...  
}  
  
console.log(cityPop()); // ověření funkčnosti
```

ČÁST 2

- Vytvořte **objekt** s názvy pěti fiktivních měst a počtem jejich obyvatel
- Pro počet obyvatel použijte funkci `cityPop()`
- **Použijete:**
 - `let`
 - objekt `{key: val}`
 - funkci `cityPop()`

```
let cities = {  
  "Domašov": ..., "King's  
  Landing": ...,  
  ...  
}
```

ČÁST 3

- Ze slovníku `cities` odstraňte všechna města pod 500 000 obyvatel
- Zbylá města vypište do konzole ve formátu:
Město: 999999 obyvatel
- **Použijete:**
 - cyklus `for () {}`
 - podmínku `if () {}`
 - `delete`

ČÁST 4 (BONUS)

- Vytvořte si v index.html prázdný seznam nebo tabulku
`<ul id="cities">/ <table id="cities">`
- Do tabulky / seznamu vložte pomocí JavaScriptu název každého zbývajícího města nad 500 000 obyvatel a jeho počet obyvatel
- **Použijete:**
 - cyklus for () {}
 - document.getElementById()
 - document.createElement()
 - element.innerText
 - parentElement.appendChild(element)

Přečtěte si víc a procvičujte

- <https://javascript.info/>
- <https://exercism.io/tracks/javascript>
- e-booky **zdarma**
<https://github.com/getify/You-Dont-Know-JS>
- <https://medium.com/>
- <https://bost.ocks.org/mike/>

Ptejte se

kdykoliv
kdekoliv
jakkoliv
co nejdřív ...

e-mail:
451242@mail.muni.cz