JavaScript

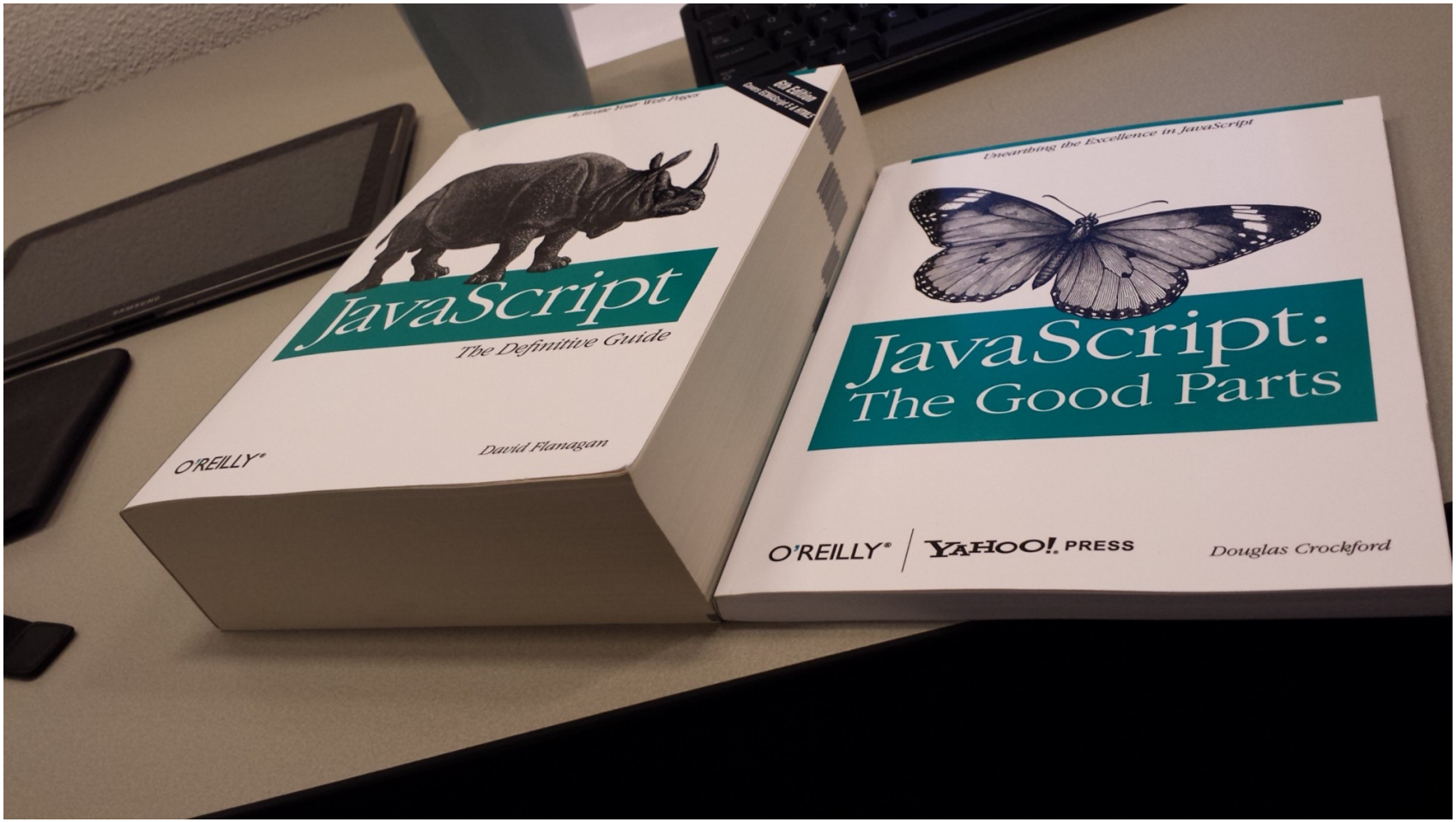
Cvičení 3

Webová kartografie

Podzim 2019

JavaScript

- umožňuje tvorbu **dynamických** webových stránek
- **objektově** orientovaný programovací jazyk (OOP)
- **standardizace** - ECMAScript
  - budeme se učit **ES6**
  - dále **ES5**, ES7, ES8, ES9
  - 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
O'REILLY
David Flanagan
Unearthing the Excellence in JavaScript
JavaScript:
The Good Parts
O'REILLY
YAHOO! PRESS
Douglas Crockford

Proč?

- interakce:
  - uživatel $\leftrightarrow$ *prohlížeč*

    validace formulářů, vizualizace, animace, ...
  - *prohlížeč* $\leftrightarrow$ 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

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

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 vypsat do konzole:

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

4. Otevřete `index.html` v prohlížeči dvouklikem 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 vývojařských 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`)

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 s podtržítky (user_activities, name, age)
- **rozlišují** 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
- **seznamy** (array)
- **objekty**
- null, undefined

```
let speedLimit = 90;
let [lat, lon] = [49.23, 16.4];
let message = "Welcome";
let warning = `The speed limit is ${speedLimit}`;
let warn = currentSpeed > speedLimit;
let carModel = null;
let activity = { // object
  type: "run",
  distance: 5632, // meters
  duration: 1412, // seconds
  elevation: 273 // meters
}

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

Procvičování 1

- Vypište do konzole libovolný ASCII art ze stránky `https://www.asciiart.eu/`, např.:

```
"((...))"
"( O O )"
" \   / "
" (`_`) "
```

- Před některé znaky (" ' \) je nutné vložit zpětné lomítko (\), jinak se nemusí zobrazit.
- **Použijete:** `console.log();`

Operátory

- standardní aritmetické operace (+, -, *, /, %)
- provnávání hodnot (>, >=, <, <=, ===, !==)
  - **používejte** === a !==

    porovnává pouze stejné datové typy
  - **nepoužívejte** ==, !=

    `"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

```
let speedKmph = distanceM / durationS / 3.6
let even = 90 % 2 === 0;
console.log(activity.type !== "bike");
let msg = "Your speed is " + speedKmph;
```

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 radius = …;
```

Podmínky

```
const BIRTH_YEAR = 1992;
let year = 2018;
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 oprátory `<, >, <=, >=`

```
// 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 (polygon in polygons) {
  console.log(calculateArea(polygon));
}
```

Funkce

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

welcome();
```

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

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

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

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

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 selhální - **snadnější debugging**
- naučte se používat **vývojářské nástroje** (**F12**, developer tools, konzole) a debuggovat kód

Úkol

- vypracujte procvičování 1–3 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 14. 11. 24:00
- **bodované** (max 5 b.), za část 4 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
- Tyto města vypište do konzole ve formátu:
  `Město: 999999 obyvatel`
- **Použijete:**
  - cyklus `for () {}`
  - podmínku `if () {}`
  - `delete`
- **Bonus:** místo cyklu a mazání dat použijte metodu `.filter()`

  https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Array/filter

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)`

Jak dál?

Přečtěte si víc a procvičujte

- https://javascript.info/
- 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: leitgeb@mail.muni.cz
discord: https://discord.gg/ppscspX