

JavaScript 1

Matěj Cajthaml

Distanční výuka - 43. - 44. hod
Smíchovská SPŠ | WBA

Abstrakt

Tento dokument vznikl jako podpůrný materiál při **n-té** vlně pandemie CoVID-19 a spojených nemocí. Je určen pouze pro studijní účely. Text byl vytvořen ve školním roce 2020 / 2021 na škole Smíchovská SPŠ. Autorem textu je Matěj Cajthaml jakékoliv připomínky a dotazy buď odesílejte na e-mail matej.cajthaml@ssps.cz nebo na MS Teams.

Obsah

1 Úvod	2
2 JavaScript	2
3 Zápis	2
4 JS na stránce	3
5 Proměnné	3
6 Základní vypisování	4
7 Boolean hodnoty	4
8 Text	4
9 Čísla	5
10 Větvení pomocí podmínek	5
11 Konzultace	6

1 Úvod

Konečně se dostáváme do zajímavých témat webových stránek: již umíme vytvářet HTML, to stylovat a nyní tam chybí již dva poslední dvě části, a to interakce s uživatelem (pomocí skriptovacího jazyku JavaScript) a animace. Nyní se vrhneme na základy JavaScriptu.

2 JavaScript

JavaScript (dále jen JS) je skriptovací jazyk, který je určen pro automatizaci různých akcí na webových stránkách. Již jsem zmiňoval že pomocí JS tvoříme interakci webových stránek s uživatelem. Pomocí JS můžeme např. při kliku provést akci, přidat, odebrat či upravit prvky stránky či odeslat a získat data.

Skriptovací jazyk je jazyk, který je především určen k úpravě již stávajícího systému (pro nás tedy webové stránky) k potřebám zákazníka či uživatele. JS již v dnešní době však není určen jen pro webové stránky, nýbrž jako samostatný programovací jazyk, který nemusí mít s webovými stránkami nic společného. My ho však budeme prozatím používat jen pro původní účely.

JS vzniknul poměrně rychle již před více jak dvaceti lety. Spousta nevýhod se JS “valí” již od začátku. Tento jazyk byl určen na dané věci a postupem času i na další. Kvůli zpětné kompatibilitě - aby fungoval na všech zařízeních, prohlížečích a tak dále, tyto věci v JS i přes nechuť autorů zůstali. V dnešní době se již tyto zastaralé věci nepoužívají a mi se je učit nebudeme.

Spousta lidí si JS plete s dalším velmi známým jazykem a to Javou. Java ale má s JS pramálo společného. JS byl takto pojmenován z marketingových důvodů - aby se lépe uchytil (a vidíte, že se mu to povedlo).

Při programování (nebo skriptování) pomocí JS musí programátor (nebo webař) přemýšlet jinak, než například u programování C#. JS nemá přímé datové typy, tedy, proměnná nemá určený datový typ a ten se může měnit. JS samozřejmě interně datové typy používá, jedná se o: číslo (celé i s plovoucí čárkou), text, pole, funkce, třída a objekt.

Pro začátek si ukážeme jednoduchou práci s JS a potom si ukážeme jako JS funguje s webovými stránkami (tzv. DOMem).

3 Zápis

JavaScript zapisujeme do HTML, přesněji však do tagu script v body, který by měl být vždy na konci stránky.

```
1 <!DOCTYPE html>
2 <html lang="cs">
3 <head>
4   <meta charset="UTF-8">
5   <meta http-equiv="X-UA-Compatible" content="IE=edge">
6   <meta name="viewport" content="width=device-width, initial-scale=1.0">
7
8   <title>JavaScript a Há-nůž</title>
9 </head>
10 <body>
11   <!-- Veškerý obsah -->
12
13   <script>
14
15   </script>
16 </body>
17 </html>
```

Skript JS můžeme importovat do stránky z jiného souboru (či jiné stránky) taktéž pomocí tagu script a atributu src.

```

10 <body>
11   <!-- Veškerý obsah -->
12
13   <script src="js/skript.js"></script>
14 </body>
15 </html>
16

```

Při odevzdávání úkolu však budeme preferovat první možnost (kvůli lepšímu hodnocení). Na každé jiné webové stránce se hodí využívat další soubor (např. v složce js).

V JS můžeme používat “//” k zapsání řádkového komentáře a “/**/” k zapsání víceřádkového komentáře.

4 JS na stránce

Při přidání skriptu na stránku se samotný skript volá tehdy, kdy je načten na stránce - proto se taktéž dává až na konec tagu body, tedy potom, až se načtou všechny ostatní tagy. Po zavolání skriptu se provedou všechny akce (a nebo se čeká např. na uživatele). V DevTools (F12 nebo prozkoumat prvek) máme možnost zobrazit tzv. Konzoli, která nám bude ukazovat chyby, které jsme udělali, nebo např. texty, které si necháme vypsat.

Konzole je velmi silný nástroj a bez něj se při přípravě (a začátcích JS) neobejdete. Do konzole taktéž můžete zapisovat jakýkoliv JS kód. To si ale ukážeme na konzultacích nebo si to laskavý čtenář zkusí sám.

5 Proměnné

V základu můžeme v JS definovat dva typy proměnných a to normální proměnné a konstanty. Z programování víte, že konstanty jsou speciální proměnné, které nelze “měnit”. V JS to však znamená, že nejde měnit jejich datový typ (protože např. u polí se dá u konstant měnit jejich obsah). Konstanty není potřeba využívat, jen je dobré o nich vědět.

Základní proměnnou definujeme pomocí klíčového slova `let`. Po něm následuje jméno proměnné (bez výjimky anglicky, unikátní, přesně a čitelně). Při definování můžeme rovnou nastavit hodnotu proměnné pomocí `rovná se a hodnoty`. Po definování či nastavení hodnoty musí vždy přijít středník.

Když nenastavíme proměnné hodnotu, bude automaticky nastavena na klíčové slovo `undefined`, tj. nenastaveno.

Jak jsme si již řekli, JS nemá datové typy a proto můžeme jakékoliv proměnné přiřadit jakýkoliv typ. Text se obaluje do uvozovek, číslo se píše rovnou (desetinné pomocí tečky) a další věci si ukážeme později.

Datové typy můžeme za běhu měnit. Stačí napsat jméno proměnné, `rovná se a novou hodnotu`.

```

13 <script>
14   let message = "Ahoj!";
15   let unknown;
16   let tax;
17
18   tax = 21.0341;
19   message = "Ahoj <Há-nůž>!";
20 </script>

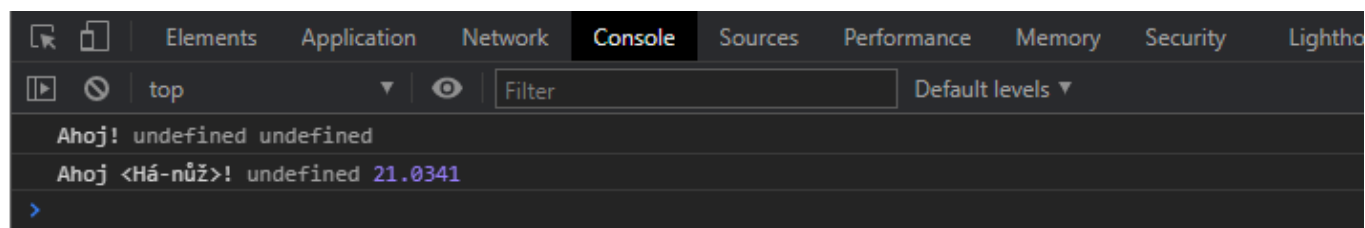
```

6 Základní vypisování

Díky implementované konzoli ve (skoro) všech prohlížečích můžeme vypisovat různý text do stránky. To můžeme udělat pomocí funkce `console.log()`. Tato funkce bere neomezeně parametrů a vypíše jakýkoliv datový typ. Po zápisu musíme znovu uzavřít pomocí středníku.

```
13 <script>
14     let message = "Ahoj!";
15     let unknown;
16     let tax;
17
18     console.log(message, unknown, tax);
19
20     tax = 21.0341;
21     message = "Ahoj <Há-nůž>!";
22
23     console.log(message, unknown, tax);
24 </script>
```

ÚKOL: Tipněte si co vypíše výše uvedený kód.



Stránku si můžete zobrazit ZDE.

7 Boolean hodnoty

V JS, stejně jako např. v C# existují tzv. “booleanovy hodnoty”, tedy pravda a nepravda. Tyto hodnoty jsou v JS reprezentovány klíčovými slovy `true` a `false`.

8 Text

S textem můžeme dělat spoustu věcí. Můžeme například slučovat dva texty dohromady pomocí znaku `+`. Tímto způsobem můžeme do textu přidávat i jiné textové proměnné či čísla.

Z jakékoliv textové proměnné můžeme získat délku pomocí získání vlastnosti `“length”`. Pomocí metody `.charAt(k)` můžeme získat znak na `k`-té pozici.

V jakékoliv textové proměnné můžeme pomocí metody `.includes(str)` zkontrolovat, zda daná hodnota obsahuje jinou textovou hodnotu. Metoda vrátí `bool` - `true` nebo `false`.

Metody `.toUpperCase()` a `.toLowerCase()` vrátí pro daný text hodnotu v kapitálkách (resp. v malých písmenech).

Dvě textové hodnoty taktéž můžeme porovnávat pomocí `“==”`, tento zápis vrátí znovu hodnotu `bool` - `true` nebo `false`.

```

13 <script>
14   let name = "Matěj";
15   let message = "Ahoj!" + " " + "Já se jmenuji " + name + ".";
16
17   console.log(name, message);
18
19   let age = 12;
20   let level = "Tvůj level je " + age + ". (asi)";
21
22   console.log(age, level);
23
24   console.log(level.length, level.charAt(12));
25
26   console.log(level.includes("je"));
27   console.log(level.includes("jest"));
28   console.log(level.includes(age));
29
30   console.log(level.toUpperCase());
31   console.log(level.toLowerCase());
32 </script>

```

Stránku si můžete zobrazit [ZDE](#)..

9 Čísla

Stejně jako např. v C# můžeme v JS čísla sčítat, odečítat, dělit, násobit a v podstatě cokoliv dalšího.

Čísla můžeme porovnávat pomocí "=", "<", ">", "<=" a ">=".

```

13 <script>
14   let age = 17;
15   let year = 2021;
16
17   let lastYear = year - 1;
18
19   let adult = age >= 18;
20   let notCoronaYear = year < lastYear;
21
22   console.log(age, year, lastYear, adult, notCoronaYear);
23
24   let older = age * 2.1;
25   let decades = lastYear / 10;
26
27   console.log(older, decades);
28 </script>

```

Stránku si můžete zobrazit [ZDE](#)..

10 Větvení pomocí podmínek

V JS existuje několik možností větvení, tento týden si ukážeme prostou podmínku (if, else if a else) a příští týdny cykly a switche.

Podmínka se zapisuje pomocí klíčového slova if (p) kde p značí podmínku - hodnot true nebo false. Když bude hodnota true, provedou se příkazy uvnitř podmínky, pokud ne, daný blok se přeskočí. Pomocí else můžeme definovat pokud daná podmínky neplatí a pomocí else if další podmínku, pokud předchozí neplatí.

```

12
13  <script>
14      let year = 2021;
15
16      if (year == 2021) {
17          console.log("Je rok 2021.");
18      }
19
20      if (year != 2021) {
21          console.log("Není rok 2021.");
22      } else {
23          console.log("Je rok 2021.");
24      }
25
26
27      if(year == 2020) {
28          console.log("Je rok 2020.");
29      } else if (year > 2020) {
30          console.log("Je po roce 2020.");
31      } else {
32          console.log("Je před rokem 2020.");
33      }
34  </script>

```

Stránku si můžete zobrazit ZDE.

11 Konzultace

Na konzultacích si ukážeme práci s konzolí, základy JS a ukážeme si chyby, které se Vám mohou stát. Připravte si otázky, protože toto téma je důležité - a je povinné v závěrečné práci.

Tento týden není žádný úkol.