

JAVASCRIPT NA WEBU

Michal Kučera

michal.kucera@czechitas.cz






JAVASCRIPT NA WEBU

Luděk Roleček

ludek.rolecek@czechitas.cz



 @mi_kucera
 @mi_kucera
 in/mikucera
michalkucera.com

Co to je JavaScript?

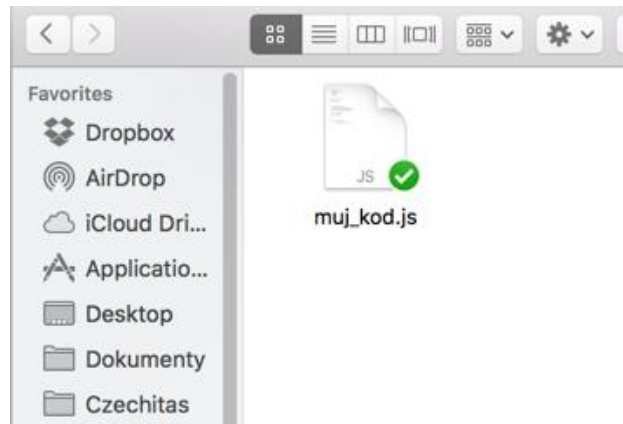
- **Programovací jazyk** - často označován jako skriptovací
- HTML - obsah a struktura stránky
- CSS - vzhled stránky
- JavaScript - chování a interaktivita stránky

Co to je JavaScript?

- Původně navržen pro přidávání drobných efektů na stránku
- Dnes se používá na webu téměř na vše
- Např. velké aplikace typu Facebook, Gmail, apod.

Co to je JavaScript?

- Kód se píše do souborů, které mají koncovku **.js**
- Tyto soubory se připojují do HTML souboru pomocí značky **<script>**
- Značka má atribut **src**, do kterého napíšeme jméno souboru



Co to je JavaScript?

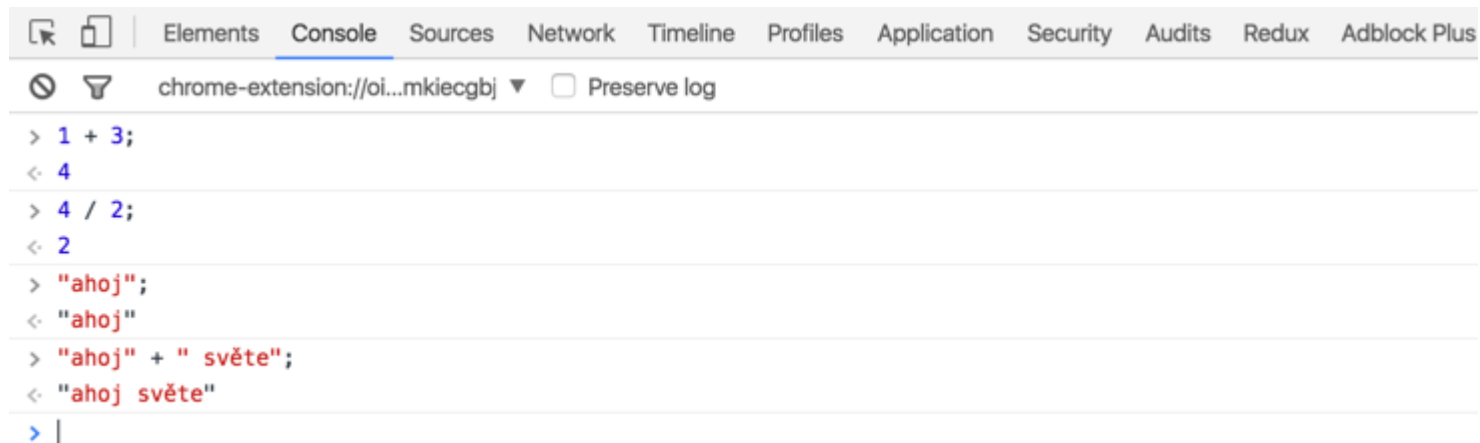
```
<html>
  <head>
    <script src="muj-kod.js"></script>
  </head>

  <body>
    ...
    <!-- nebo těsně před konec body - to preferujeme -->
    <script src="muj-kod.js"></script>
  </body>
</html>
```

Co to je JavaScript?

- Javascript můžeme zkoušet v prohlížeči
- Pomocí F12 otevřeme vývojářské nástroje prohlížeče a přepneme se do záložky Console
- Můžeme začít psát JavaScript, který se okamžitě bude provádět

Vývojářské nástroje v prohlížeči



The screenshot shows the Chrome DevTools interface with the 'Console' tab selected. The address bar indicates the page is a Chrome extension. The console log shows the following sequence of commands and results:

```
> 1 + 3;  
< 4  
> 4 / 2;  
< 2  
> "ahoj";  
< "ahoj"  
> "ahoj" + " světe";  
< "ahoj světe"  
> |
```

Jak si JavaScript zkusit

- Do konzole píšeme jednotlivé příkazy a **enterem** je vykonáme
- Každý příkaz v JavaScriptu by správně měl končit středníkem

`alert("kuk na mě");`



- Výsledek příkazu (pokud je nějaký) se hned vypisuje zpět

Opakování

Funkce

- Sada příkazů, kterou chceme vykonávat opakovaně
- V programování se obecně snažíme vyhnout opakování kódu
- Zjednodušuje a zpřehledňuje kód

Funkce

- Můžeme používat vestavěné funkce JavaScriptu / prohlížeče, jako je např. **alert("ahoj");**
- Funkci vždy voláme jejím jménem, za kterým následují kulaté závorky. V závorkách jsou někdy uvedeny tzv. **parametry funkce** nebo jsou závorky prázdné.

```
alert("ahoj");
```

```
posli();
```

Vlastní funkce

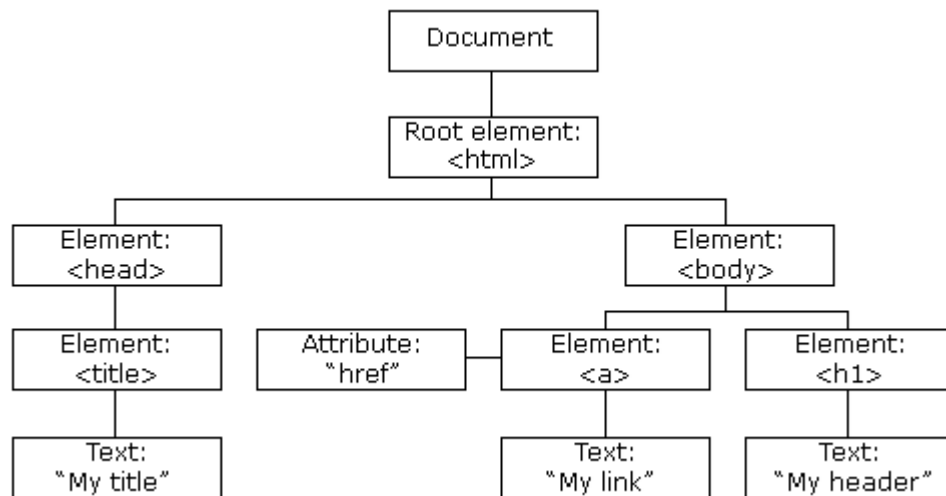
- Můžeme si napsat vlastní funkce.

```
function pozdrav(jmeno) {  
    alert("Ahoj " + jmeno);  
}
```

Úkoly

DOM - objektový model dokumentu

- Každý HTML prvek na stránce je reprezentován objektem, ke kterému můžeme pomocí JS přistupovat




Výběr prvku na stránce

- Abychom mohli upravovat JavaScriptem prvek na stránce, musíme nejprve získat objekt tohoto prvku v DOM
- Musíme prvek na stránce **“najít”**

Výběr prvku na stránce

```
let nadpis;
```

```
nadpis = document.querySelector("h1");
```



Nalezený prvek si uložíme do proměnné **nadpis**

Vyhledáváme v celém dokumentu

Hledáme jeden prvek

Popis prvku, který hledáme.
Zapíše se stejně, jako kdybychom psali selektor pro prvek v CSS

Selektory

- Aneb to, co píšeme do uvozovek dovnitř
document.querySelector("...");

h1 HTML prvek

.foto Prvek s CSS třídou class="foto"

#cedule Prvek s id="cedule"

.menu a Odkaz uvnitř prvku s třídou menu

Změna vlastností prvku

- Když jsem získal objekt prvku na stránce pomocí **document.querySelector("...");** mohu v JavaScriptu měnit jeho vlastnosti
- Přímou nastavovat CSS styly
- Přidávat / odebírat CSS třídy
- Měnit atributy prvky
- Měnit textový obsah prvku

Přímá změna CSS stylu prvku

```
let nadpis;
```

```
nadpis = document.querySelector("h1");
```

```
nadpis.style.color = "red";
```

Přímá změna CSS stylu prvku

- CSS vlastnosti se v JavaScriptu zapisují pomocí tzv. camel case. Vlastnosti se zapisují dohromady bez mezer a pomlček, první písmeno je malé, další slova v názvu začínají velkým písmenem.

background-color	→	backgroundColor
font-size	→	fontSize

Změna textového obsahu prvku

```
let nadpis;
```

```
nadpis = document.querySelector("h1");
```

```
nadpis.textContent = "Ahoj";
```

Měníme pouze textový obsah prvku, nemůžeme vložit jiné značky nebo formátování

Změna HTML obsahu prvku

```
let odstavec;
```

```
odstavec = document.querySelector("p");
```

```
odstavec.innerHTML =
```

```
    "Umíme <strong>JavaScript</strong>";
```

Pomocí innerHTML můžeme do obsahu prvku přidávat i jiné HTML značky (zde např. tučný text)

Přidání/odebrání CSS třídy pomocí classList

Jako vždy si nejprve najdeme v dokumentu příslušný prvek:

```
let nadpis;  
nadpis = document.querySelector("h1");
```

```
nadpis.classList.add("csstrida");
```

↑
Našemu vybraný
nadpis

↑
Seznam CSS tříd
aplikovaných na
objektu

↑
Do seznamu
přidáme

↑
CSS třídu s tímto názvem

Výsledek je stejný, jako bychom v HTML napsali `<h1 class="csstrida">Můj nadpis</h1>`

Přidání/odebrání CSS třídy pomocí classList

Stejným způsobem můžeme CSS třídy z objektu odebírat:

```
nadpis.classList.remove("csstrida");
```

Můžeme také třídu na objektu přepínat:

- pokud třída na objektu neexistuje, tak se přidá
- pokud třída na objektu existuje, tak se naopak odebere

```
nadpis.classList.toggle("csstrida");
```

Úkoly

Události

- Událost nastává na prvku na stránce ve chvíli, kdy se s prvkem něco děje
- Může se jednat např. o kliknutí na prvek, najetí kurzoru myši, stisk klávesy, změna hodnoty formulářového pole, apod.
- Událostem nasloucháme a ve chvíli, kdy k nim dojde, na ně můžeme reagovat

Nejpoužívanější události

click	kliknutí na prvek
mouseover	kurzor myši najel nad prvek
mouseout	kurzor myši odjel z prvku pryč
keydown	stisk klávesy
change	změna hodnoty pole ve formuláři
load	kompletní načtení prvku (typicky celé stránky)

A spousta dalších, viz např. <https://developer.mozilla.org/en-US/docs/Web/Events>

Nasloucháme událostem

Jako vždy si nejprve najdeme v dokumentu příslušný prvek:

```
let tlacitko;  
tlacitko = document.querySelector("a");
```

```
tlacitko.addEventListener("click", priKliknuti);
```

↑
Na našem
tlačítku

↑
Nasloucháme
události

↑
Kliknutí myši

↑
Když k události dojde,
zavoláme tuto funkci

Reagujeme na kliknutí

```
let tlacitko;  
tlacitko = document.querySelector("a");  
  
tlacitko.addEventListener("click", priKliknuti);  
  
function priKliknuti() {  
    // zde je náš kód, který se provede  
    // při kliknutí na tlačítko  
}
```