

JavaScript

Cvičení 5 – 2.část

Webová kartografie – úvod

Podzim 2024

Filip Leitner

DOM

- Document Object Model
- rozhraní pro práci s HTML dokumenty
- prostřednictvím DOM můžeme **číst konkrétní části dokumentu, měnit jeho strukturu, styl a obsah**
- implementace DOM v prohlížečích je založena na standardech, ale mezi prohlížeči se **může měnit**

Přečtěte si: https://developer.mozilla.org/en-US/docs/Web/API/Document_Object_Model/Introduction

```
document.getElementById('id').style.backgroundColor = 'red'
```

Ukážky manipulácie dom elementov

- <https://phuoc.ng/collection/html-dom/>

//2. najděte jeden element pomocí jeho ID a změňte barvu jeho pozadí

```
const el = document.getElementById('redBackground');  
el.style.backgroundColor = 'red'; //https://developer.mozilla.org/en-
```

//3. vyberte si část své webové stránky a odstraňte ji z DOM

/*****/

```
const element = document.getElementById('toBeRemoved');  
element.remove() //Odstrání vybraný element
```

```
/*  
//4. vytvořte nový DOM element a připojte jej na konec <body>  
*/  
const p = document.createElement('p'); // Vytvorí element 'p'. Bez  
/*  
* Je možné element ľubovolne upravovat  
*/  
p.innerText = 'Custom text'; // Vloženie obsahu  
p.style.color = 'red'; //zmena farby písma  
p.setAttribute('id', 'dynamicallyAddedId') //Nastavenie atribútu id  
document.body.appendChild(p); //Pripojí vytvorený element (uložený  
// Alternatíva s getElementByTagName  
// document.getElementsByTagName('body')[0].appendChild(p);  
////////////////////////////////////
```

Alternativa

```
querySelector(selectors)
```

The [Document](#) method `querySelector()` returns the first [Element](#) within the document that matches the specified selector, or group of selectors. If no matches are found, `null` is returned.

`selectors`

A string containing one or more selectors to match. This string must be a valid CSS selector string; if it isn't, a `SyntaxError` exception is thrown. See [Locating DOM elements using selectors](#) for more about selectors and how to manage them.

```
querySelectorAll(selectors)
```

The [Document](#) method `querySelectorAll()` returns a static (not live) [NodeList](#) representing a list of the document's elements that match the specified group of selectors.

```
const el = document.querySelector("div.user-panel.main input[name='login']");
```

<https://www.smashingmagazine.com/2013/11/an-introduction-to-dom-events/>

- **oznámení** že se ve stránce něco stalo
- např. kliknutí na tlačítko, pohyb myši, posuv obsahu stránky, odeslání formuláře, ...
- **eventům** můžeme v kódu **naslouchat** a na jejich základě provádět další akce (kontrolovat obsah formuláře, zobrazit další obsah, odstraňovat obsah, ...)

```
element.addEventListener(<event-name>, <callback>,  );
```

- **event-name** (string) This is the name or type of event that you would like to listen to. It could be any of the standard DOM events (**click**, **mousedown**, **touchstart**, **transitionEnd**, etc.) or even your own custom event name (we'll touch on custom events later).
- **callback** (function) This function gets called when the event happens. The **event** object, containing data about the event, is passed as the first argument.

```
const FORM = document.getElementById("login");  
FORM.addEventListener("submit", function(event) {  
    alert("The form was submitted!");  
});
```



```
element.addEventListener(<event-name>, <callback>, <use-capture>);
```

```
const FORM = document.getElementById("login");  
FORM.addEventListener("submit", function(event) {  
    alert("The form was submitted!");  
});
```

ÚKOL 3

- 1 stáhněte si vzor formuláře z <https://gist.github.com/a32a22e3c6dfa4d8b4da5fae6e193a2c>
- 2 pomocí funkce `console.log()` vypište hodnotu zadanou do formuláře **<form>** po jeho odeslání (submit)
- 3 upravte kód tak, aby se stránka po odeslání formuláře neobnovila (googlete ...)
- 4 využijte kód z domácího úkolu (převeďte výpočet obsahu kruhu na funkci) a výsledek vypište do nového elementu **<p>**

Bude se hodit:

- https://www.w3schools.com/js/js_events.asp
- `event.target`
- `addEventListener("submit", function(event) { .. });`
- `element.innerText parent.appendChild(element)`

ÚKOL 3

- doplňte také výpočet obvodu
- (bonus) vytvořte nový element `<div>`, který bude mít tvar kruhu
 - 1 při každém odeslání formuláře změňte jeho rozměry tak aby odpovídaly zadané hodnotě v pixelech
 - 2 změňte jeho barvu pozadí na náhodnou barvu

ÚKOL 3

- <https://gist.github.com/FilipLeitner/2d6b3523a457b93ac2ffd2db910bcc2d>

ÚKOL 4

- 1 stáhněte si vzor šachovnice z <https://gist.github.com/766e2741e4c9c501c522fad47acae6e1>
- 2 pomocí CSS zvýrazněte element při najetí myší
- 3 po kliknutí myší změňte vzhled elementu
 - použijte funkci pro přepnutí (přidání nebo odebrání) class selected
- 4 přidejte do stránky element `<p>`, ve kterém budete aktualizovat počet vybraných prvků

ÚKOL 4

- 1 <https://gist.github.com/FilipLeitner/ca6e8add5df62973e367edc993835392>

- vytvořte formulář **HTML formulář** <form>, který bude sloužit k převodu:
 - **zeměpisných souřadnic** z jednoho formátu do jiného
např. stupně, minuty → stupně, minuty, vteřiny || stupně, minuty, vteřiny -> decimal
 - **nebo vzdáleností / plochy** podle daného měřítka mapy (možnost změny jednotky)
 - **nebo výpočtu vzdálenosti** po ortodromě mezi dvěma body danými souřadnicemi
 - **nebo jiný užitečný formulář** - napište mi s návrhem :)
- pro formulář vytvořte **validaci vstupu**, uživateli se zobrazí text ve stránce jako varování
 - mělo by být možné zadat jen **kladné hodnoty**
 - v případě souřadnic zkontrolovat možné rozmezí hodnot
 - **zamyslete sa nad ďalšími obmedzeniami ktoré logicky vyplývajú z kontextu vášho formulára**
 - PS: nech to vyzerá aspoň trošku k svetu
- uživateli se výsledek objeví jako text ve stránce (např. element <p>) odevzdat do 19. 11. **bodované** (max 20 b.)

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/>
<https://html5dom.dev/loop-over-a-nodelist/>

Ptejte se

kdykoliv

kdekoliv

jakkoliv

co nejdřív ...