

# Web programiranje – Vežbe 7

## JavaScript i DOM

---

### Primer – registracija vozila

HTML stranica **registracija.html** sadrži formu za produžavanje registracije. Klikom na dugme “Submit”, podaci iz forme se šalju na stranicu **registrovani\_korisnici.html** koja sadrži listu trenutno registrovanih korisnika.

JavaScript kod se može definisati unutar HTML-a u script tagu, ali je dobra praksa pisati JavaScript u posebnim fajlovima sa ekstenzijom **.js**. Potrebno je kreirati novi fajl koji pod nazivom **registracija.js**, koji za sada sadrži samo funkciju:

```
// Ako funkcija vrati true, validacija je uspesna i forma šalje podatke
// Ako funkcija vrati false, validacija nije uspešna i browser ostaje na
// trenutnoj stranici
function validacija(forma){
    return true;
}
```

Zatim je potrebno povezati ga sa HTML fajlom **registracija.html**, tako što se u head tagu doda script tag sa apsolutnom ili relativnom putanjom do JavaScript fajla:

```
<script src="registracija.js"></script>
```

Za putanje je preporučeno koristiti relativne putanje.

Moguće je povezati JavaScript funkcije sa HTML elementima kroz određene attribute. Na primer, moguće je povezati funkciju koja će se pozvati kada se klikne na dugme “Submit” u formi.

Potrebno je promeniti formu u fajlu **registracija.html**, tako što se doda atribut *onsubmit* iz koda ispod. Ono što ovim postizemo je da kada se klikne na submit dugme forme, poziva se funkcija *validacija*.

```
<form action="registrovani_korisnici.html" method="get" onsubmit="return
validacija(this)">
```

Funkcije se mogu pozivati za različite događaje, npr. na klik ili na unos vrednosti sa tastature. Više informacija videti [ovde](#). (pristupano u maju 2018.)

JavaScript omogućava manipulisanje DOM (Document Object Model) elementima, koji su reprezentacija HTML elemenata u memoriji browsera. Moguće je dodavanje, brisanje i izmena postojećih DOM objekata za vreme izvršavanja programa. Pristupanje DOM elementima je moguće kroz ugrađenu promenljivu **document**, na više različitih načina:

- **document.getElementById** – nalazi element sa prosleđenim id-jem
- **document.getElementsByTagName** – nalazi elemente sa prosleđenim imenom
- **document.getElementsByClassName** – nalazi elemente sa prosleđenom klasom

- **document.getElementsByTagName** – nalazi elemente sa prosleđenim tagom
- **document.querySelector / document.querySelectorAll** – nalazi prvi element / sve elemente koji odgovaraju prosleđenom CSS upitu

Promena atributa DOM elemenata se vrši tako što mu se promeni odgovarajuće polje na željenu vrednost. Polje **attributes** sadrži attribute odgovarajućeg HTML elementa (npr. name i href), dok polje **style** sadrži stilove (npr. boja). Promeniti funkciju validacija tako da proverava da li je polje za ime prazno:

```
function validacija(forma) {
    let imeEl = document.getElementsByName('ime')[0];
    // Vrednost unešena u polje za ime
    let ime = imeEl.value;
    // False ako je ime null, undefined ili prazan string
    if (!ime) {
        // Postavi boju inputa na crvenu
        imeEl.style.background = 'red';
        return false;
    }
    return true;
}
```

Dinamičko kreiranje elemenata se vrši pozivom funkcije **document.createElement(tagName)**. Novokreiranim elementom se dalje manipuliše kao i sa svakim drugim DOM elementom. Sadržaj DOM elementa se može promeniti na dva načina:

1. Upotrebom polja **innerHTML**. Za novu vrednost se može proslediti bilo koji string.  
**NAPOMENA:** potencijalno sigurnosni rizik jer se može uneti i maliciozni kod.
2. Pozivanjem **element.appendChild(childElement)** metode DOM elementa. Ova metoda prima drugi DOM element i stavlja ga iza poslednjeg deteta. Ovaj pristup je bezbedan jer prihvata samo elemente kreirane pozivom funkcije **document.createElement**, ali zahteva više napora.

Na stranici **registrovani\_korisnici.html** je potrebno dinamički prikazati listu registrovanih korisnika. Potrebno je proći kroz listu korisnika i za svaki kreirati `<tr>` element i dodati ga u tabelu. Kod na sledećoj strani radi baš to. Dodati ga u fajl **registrovani\_korisnici.js**.

**NAPOMENA:** **window.onload** se za sada može smatrati “main metodom” stranice. O događajima i obrađivačima događaja više na sledećim vežbama.

**SAVET:** Koristiti konzolu browsera za testiranje (**F12** ili **ctrl+shift+j** za Chrome, **ctrl+shift+k** za Firefox). Tu je moguće kucati JavaScript kod i direktno ga izvršavati u otvorenoj stranici (Console tab). Takođe je moguće postavljati *breakpoint*-ove i debugovati kod (Sources tab).

```
// Funkcija koja se poziva kada se učitava stranica
window.onload = function(event) {
    // Lista: [objekat1, objekat2 ...]
    // Objekat: {'polje': vrednost}
    let users = [
        {'ime': 'Petar', 'prezime': 'Petrović', 'jmbg': '123321'},
        {'ime': 'Marko', 'prezime': 'Marković', 'jmbg': '111111'},
        {'ime': 'Stevan', 'prezime': 'Stevanović', 'jmbg': '123456'},
        {'ime': 'Stevan', 'prezime': 'Stevanović', 'jmbg': '123456'}
    ];

    let tabela = document.getElementsByTagName('table')[0];

    for (user of users) {
        // Kreiraj novi red (tr)
        let userTr = document.createElement('tr');
        // Kreiraj polja reda (td)
        let imeTd = document.createElement('td');
        let prezimeTd = document.createElement('td');
        let jmbgTd = document.createElement('td');

        // Dodaj tekst u svako polje
        imeTd.appendChild(document.createTextNode(user.ime));
        prezimeTd.appendChild(document.createTextNode(user.prezime));
        jmbgTd.appendChild(document.createTextNode(user.jmbg));

        // Dodaj polja u red
        userTr.appendChild(imeTd);
        userTr.appendChild(prezimeTd);
        userTr.appendChild(jmbgTd);

        // Dodaj red u tabelu
        tabela.appendChild(userTr);
    }
};
```

## Zadaci

1. Validirati da forma nema prazna polja.
2. Preuzeti parametre forme iz **registracija.html** i dodati ih u listu korisnika u tabeli **registrovani\_korisnici.html**. Trenutna putanja se nalazi u promenljivoj **window.location.href** i treba je parsirati kao URL kod HTTP zahteva.
3. Validirati da ime i prezime počinju velikim slovom, da JMBG se sastoji od 13 cifara.
4. Ako se izabere motorno vozilo dodati polje kubikaža u formu.  
Dodati paragraf koji sadrži tekst: "EKO KLASA"  
Menjati boju u zavisnosti od vrednosti polja kubikaža (HINT: provera se vrši na događaj unosa sa tastature).  
Ako je izabrano električno ili hibridno boja teksta treba da bude u skladu sa kubikažom:
  - <600 - zelena
  - 600-1200 - žuta
  - >1200 – crvena

## Korisni linkovi

- Atributi vezani za događaje: [https://www.w3schools.com/tags/ref\\_eventattributes.asp](https://www.w3schools.com/tags/ref_eventattributes.asp)
- Konvencije kodiranja u JavaScriptu: <https://github.com/airbnb/javascript>
- Dokumentacija za DOM elemente: [https://www.w3schools.com/jsref/dom\\_obj\\_all.asp](https://www.w3schools.com/jsref/dom_obj_all.asp)

- Dokumentacija za ugrađenu promenljivu **document**:  
[https://www.w3schools.com/jsref/dom\\_obj\\_document.asp](https://www.w3schools.com/jsref/dom_obj_document.asp)
- Mozilla dokumentacija za JavaScript:  
<https://developer.mozilla.org/bm/docs/Web/JavaScript/Reference>
- WC3 dokumentacija za JavaScript: <https://www.w3schools.com/jsref/default.asp>