Univerzitet u Sarajevu
Elektrotehnički fakultet Sarajevo
Odsjek za računarstvo i informatiku
Predmet: Web tehnologije

Tutorijal br. 3: JavaScript

Literatura:

- Tutorijal JavaScript.
- Mozilla MDN https://developer.mozilla.org/en-US/docs/Web/JavaScript
- http://jsfiddle.net/ - stranica na kojoj možete isprobati kôd.

Uvod u tutorijal

Sljedeći HTML kôd snimite kao fajl *digitron.html* te ga otvorite u web browseru.

```
<!DOCTYPE HTML>
<HTML>
<HEAD>
  <META http-equiv="Content-Type" content="text/html; charset=utf-8">
  <TITLE>Tutorijal 3, Uvod</TITLE>
</HEAD>
<BODY>
  <H1>Digitron</H1>
  <INPUT type="text" id="sabirak1" size="5"> +
  <INPUT type="text" id="sabirak2" size="5"> =
  <INPUT type="text" id="zbir" size="5">
  <INPUT type="button" id="dugme" value=" Izračunaj"
onClick="sabiranje();">
  <SCRIPT type="text/javascript">
  function sabiranje() {
    var a = document.getElementById("sabirak1").value;
    var b = document.getElementById("sabirak2").value;
    var c=a+b;
    document.getElementById("zbir").value = c;
  }
  </SCRIPT>
  <P>Unesite sabirke i kliknite na dugme Izračunaj.</P>
</BODY>
</HTML>
```

Diskusija (hintovi na sljedećoj stranici):

- Gdje se nalazi kôd koji vrši sabiranje i u kojem programskom jeziku je napisan? Kako određujemo da će se taj kod izvršiti klikom na dugme Izračunaj?
- Zašto rezultat sabiranja nije tačan? Kako to možemo riješiti?
- Miješanje HTML i JavaScript kôda nije preporučljivo jer je otežano čitanje i održavanje.

Modifikujte ovu stranicu tako da je sav JS kôd u zasebnoj datoteci *sabiranje.js*.

Hintovi:

- JavaScript kôd funkcije sabiranje() se nalazi unutar <SCRIPT>...</SCRIPT> taga. Poziv ove funkcije specificiran je atributom onClick na dugmetu <INPUT type="button"> . U knjizi možete naći i mnoge druge korisne događaje (events) kojima možete pridružiti JavaScript kôd, npr. ondblclick (dvostruki klik), onmouseover (prelazak mišem), onchange (promjena vrijednosti, npr. za <INPUT type="text">) ...
- Poljima forme pristupili smo preko Document objekta, o čemu će više riječi biti na sljedećem predavanju i tutorijalu.
- Atribut value je tipa string. Operator + primijenjen na vrijednosti numeričkih tipova vrši sabiranje, a na vrijednosti string tipa vrši spajanje stringova (concatenate). U JavaScriptu promjenljivima ne moramo davati specifičan tip, čak se tip može i promijeniti za vrijeme postojanja varijable! Npr. ako napišete ovakav kôd:

```
var x = 1+2; // Numericki tip
x = "broj "+x;
alert(x);
```

  Na ekranu će se ispisati tekst: "broj 3". Promjenljiva x je prvobitno bila numeričkog tipa, ali kada smo primijenili operator + na string i broj rezultat je string (broj je najprije konvertovan u string), pa je taj rezultat pridružen varijabli x te je i ona postala znakovnog tipa.
- Za konvertovanje stringa u int možemo koristiti funkciju *parseInt*.
- Za izdvajanje JavaScript kôda u zasebnu datoteku koristimo ovakav HTML tag:

```
<SCRIPT src="sabiranje.js"></SCRIPT>
```

  Ovaj tag treba postaviti u HTML dokument tamo gdje želite da se JS kod izvrši (ako je sav kod u funkcijama onda je svejedno). No ostaje činjenica da se u onClick atributu nalazi JavaScript kôd. Preporučljivo je da pridruživanje eventa dugmetu vršite na ovaj način:

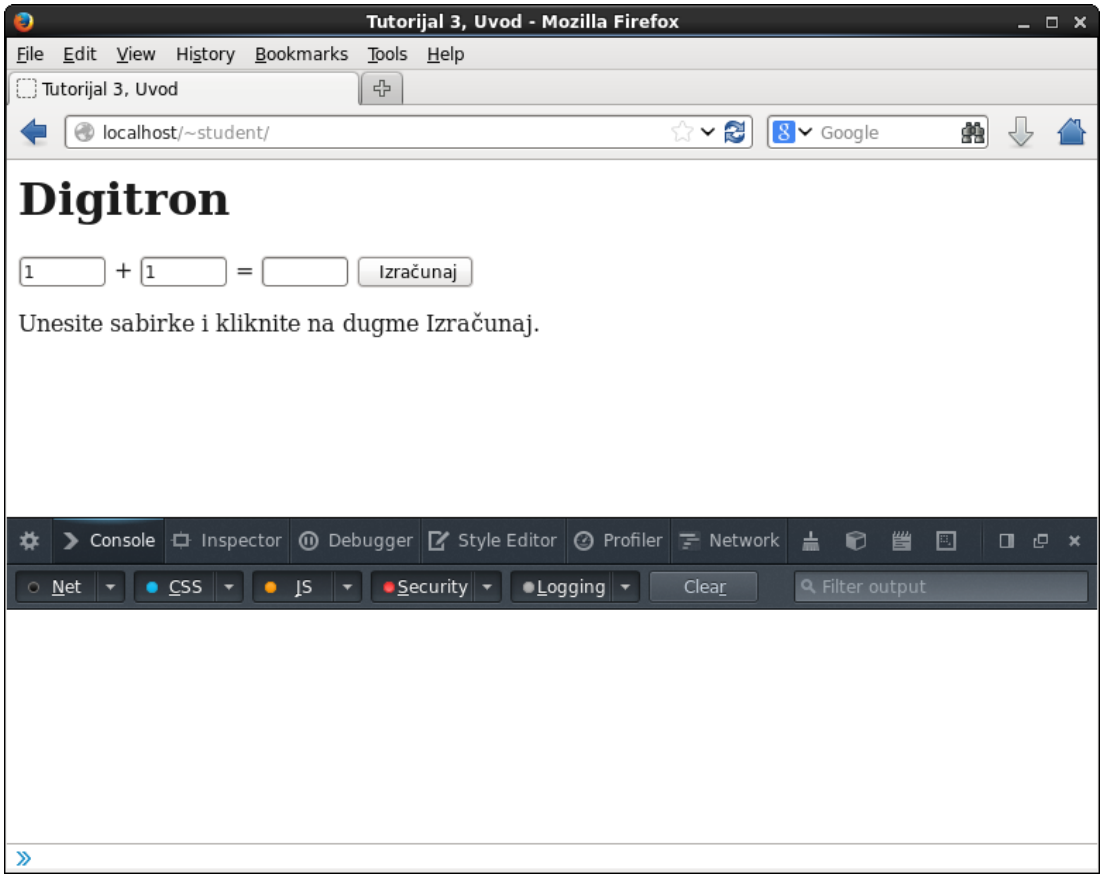
```
// Dugme sa IDom "dugme"
var dugme = document.getElementById("dugme");
// Funkcija koja se izvršava klikom na dugme
dugme.addEventListener( "click", function( ev ) {
    alert( "Zdravo" );
}, false);
```

- Ovaj kod treba dodati u JS fajl i to van funkcije. No addEventListener nije podržan u IE7!
- Na primjeru iznad možemo vidjeti *anonimnu funkciju* ili *funkcijski izraz*. Ova funkcija je deklarisana unutar samog poziva dugme.addEventListener, nije joj dodijeljeno ime (tako da se ne može pozivati osim klikom na dugme), prima jedan parametar (ev) koji nije iskorišten, a tijelo te funkcije je jedna naredba: alert("Zdravo"). Anonimne funkcije su vrlo često korištene u JavaScript programima. Također možemo i pridružiti funkciju varijabli:

```
var nesto = function( ev ) { alert( "Zdravo" ); }
nesto();
```

Web developer tools

Pritiskom na kombinaciju tipki Ctrl+Shift+I (Firefox i Chrome) dobije se alat pod imenom Web Developer Tools.



- Kartica *Console* nudi osnovnu JavaScriptom komandnu ljusku. U ovom prozoru možete otkucati neku komandu (npr. `alert('Zdravo')`), te vidjeti greške i upozorenja koje je proizveo JavaScript interpreter. Inače, u slučaju greške interpreter jednostavno prestane raditi.
    - Kada radite debugging vaše skripte često je potrebno da ispišete vrijednosti nekih varijabli. Možete ih upisati u prozor konzole funkcijom `console.log('Proba')`.
- Kartica *Inspector* vam omogućuje da kliknete na bilo koji element web stranice, saznate o kojem tagu/tagovima se radi, gdje se nalaze u stranici, te da direktno editujete CSS svojstva.
- *Debugger* pruža uobičajene opcije za debugging JavaScript kôda kao što su breakpoints, step/trace, watches itd. Kod ovoga treba biti pažljiv jer se zna desiti da se kompletan preglednik zaglavi.
- *Style Editor* je zgodnija varijanta za editovanje CSS listova.

- *Profiler* vam omogućuje da pratite vrijeme izvršenja pojedinih dijelova (funkcija) vašeg JS kôda kako biste optimizovali program.
- *Network* vam daje informaciju o mrežnoj komunikaciji (upućeni zahtjevi prema serveru ili serverima te koliko je koji trajao).

U slučaju Chrome-a debugging opcije se nalaze pod karticom *Sources*, a profiling je drugačije riješen i nalazi se pod *Profiles* i *Audits*.

Zadatak 1.

Pokrenite sljedeći JavaScript kôd. Da li razumijete šta se ovdje dešava i zašto?

```
var odgovor = prompt("Kako se zoves?", "Imenom i prezimenom(default)");
if (odgovor!=null && odgovor!="")
{
     var r=confirm("Pritisnite OK da prikazete ime u alertboxu a Cancel za
prikaz direktno na stranici");
     if (r==true)  // ili if(r)
           alert(odgovor);
     else
           document.write(odgovor);
}
```

Doradite uneseni kôd na sljedeći način:

Postaviti korisniku pitanje da unese neki tekst korištenjem prompt dijaloga. Nakon unosa, tekst ispisati u alert boxu obrnutim redoslijedom tj ako smo unijeli "ovo je test" ispisati "tset ej ovo".

Zadatak 2.

Korištenjem document.write i petlje napraviti JavaScript funkciju koja u trenutni HTML dokument dodaje tablicu množenja 10x10 koja izgleda kao na slici (dat je samo gornji lijevi ugao):

X	1	2	3	4	...
1	1	2	3	4	5
2	2	4	6	8	10
3	3	6	9	12	15
4	4	8	12	16	20

Za stiliziranje tabele koristiti CSS koji se, kao i JavaScript, treba nalaziti u zasebnoj datoteci. Korištene boje su: #CCCCCC (siva) i #FFF2CC (žućkasta). Font je sans-serif. Zaglavlja redova i

kolona su boldirana. Rubovi tabele su širine 1 pixel.

Savjet
Za ovaj zadatak nemojte koristiti dugme koje iscrtava tabelu na click! **document.write** metoda se može koristiti samo za vrijeme učitavanja HTML dokumenta jer u suprotnom zamjenjuje tekući dokument novim (čime ujedno gubimo CSS koji je tražen zadatkom). Ispravan način izmjene aktuelnog dokumenta je putem DOM manipulacije odnosno atributa **innerHTML** što je tema sljedećeg tutorijala.

Zadatak 3.

Dat je sljedeći HTML kôd:

```
<TEXTAREA ID="tekst" ROWS=10 COLS=30></TEXTAREA>
<BR>
<INPUT TYPE="button" ID="dugme" VALUE="Zamijeni">
```

Napisati JavaScript funkciju koja će klikom na dugme transformisati tekst unesen u tekstualno polje tako da svi brojevi (ne cifre!) budu pretvoreni u tekstualnu formu. Tako npr. tekst "ovo je broj 2586 i broj 123" treba biti zamijenjen stringom "ovo je broj dvijehiljadepetstotinaosamdesetišest i broj stodvadesettri" (bez navodnika).

Savjet
Napravite niz stotica (čiji su elementi "stotinu", "dvijestotine"...), desetica i jedinica. Posebni slučajevi su brojevi 11, 12, 13... 19. Za brojeve veće od 999 radite sljedeće: prvu cifru posmatrate kao zaseban broj (npr. "pet") nakon čega ide string "hiljada" i zatim preostale tri cifre. Slično i za veće brojeve, slična logika i za milone. Brojeve veće od milijardu ne morate obrađivati.

Zadatak 4.

Dat je sljedeći HTML kôd:

```
f(x)=<INPUT TYPE="text" ID="funkcija"><BR>
<INPUT TYPE="button" ID="dugme" VALUE="Crtaj"><BR>
<CANVAS ID="slika" WIDTH="200" HEIGHT="200">Browser ne podržava
canvas</CANVAS>
```

Napisati JavaScript funkciju koja će klikom na dugme u predviđenom CANVAS prostoru iscrtati grafik funkcije napisane u tekstualnom polju "funkcija". Za ovaj tutorijal možete podržati samo polinome odnosno funkcije oblika: 2x^2-5x+6. Za vježbu kod kuće možete dodati podršku i za korjenovanje, trigonometrijske funkcije i sl.

Kako iscrtati grafik funkcije?
Najprije obradite primljeni string kako biste prepoznali funkciju koja se iscrtava. Recimo možete u stringu tražiti podstringove x^2, x^3, x..., izdvojiti brojeve

ispred ovih podstringova (uključujući i predznak) i funkcijom parseInt dobiti vrijednost koeficijenta. Ove vrijednosti možete stavljati u neki niz. Napravite JavaScript funkciju koja izračunava vrijednost funkcije f(x) za neko primljeno x koristeći taj niz.

Pretpostavimo da se koordinatni početak nalazi na koordinatama (100,100) platna (canvas-a). Potrebno je da odredite skalu odnosno razmjer vašeg crteža. Recimo možete definisati da se na koordinatama (0,100) nalazi vrijednost -5, a na (200,100) +5, dakle iscrtava se grafik funkcije na intervalu [-5,5].

Najprije izračunajte vrijednost funkcije za x=-5. Npr. f(-5)=2. Pošto se nula nalazi na 100 a 5 na 200, 2 se nalazi na koordinati 140. Pomjerite kursor naredbom context.moveTo na koordinate (0,140) pri čemu 0 odgovara vrijednosti x=-5 a 140 vrijednosti y=2.

```
context.moveTo(0,140)
```

Sada petljom izračunavajte vrijednosti f(x) za svako x između -5 i 5 pri čemu koristite takav korak da bude izračunata vrijednost za svaki pixel od 0 do 200. Dakle korak je 5/100=0,05. Za svaku takvu vrijednost izvršite naredbu `context.lineTo(x,y)` na koordinate koje ćete izračunati po sličnom principu. Na kraju dodajte naredbu `context.stroke()`.

Kod kuće možete dodatno poboljšati program tako što će se sivom bojom iscrtati koordinatne ose, a razmjer i lokacija koordinatnog početka se može mijenjati po želji (zoom-in, zoom-out).