# Menjanje HTML-a pomoću JavaScripta

Jedna od najčešće korišćenih metoda u HTML Document Object Model-u je metoda getElementById() koja kao rezultat vraća element koji ima ID atribut sa navedenom vrednošću, i koristi se skoro svaki put kada želite da manipulišete ili dobijete informacije o delementu u vašem dokumentu.

Vraća null ako nema elemenata sa navedenim ID-om.

ID mora biti jedinstven unutar stranice. Međutim, ako postoji više od jednog elementa sa navedenim ID-om, metoda getElementById() vraća prvi element u izvornom kodu.

document.getElementById("demo").innerHTML = "Hello World";

innerHTML metoda je takođe deo Document Object Model-a i omogućava Javascript kodu da manipuliše prikazom elemenata na websajtu. Odnosno omogućava čitanje i izmenu elenenata unutar HTML taga.

Korišćenjem innerHTML, možete promeniti sadržaj stranice bez osvežavanja stranice. Ovo može učiniti web stranicu bržom i responzivnijom na korisnički unos. Unutar HTML svojstva se koristiti zajedno sa getElementById().

Primer:

<p id="demo">JavaScript moze promeniti HTML.</p>

<script>

document.getElementById('demo').innerHTML = 'JavaScript je promenio HTML';

</script>

U prvom redu smo paragrafu “JavaScript moze promeniti HTML.” Dodali atribut ID sa vrednošću “demo“.

Nakon toga smo pomoću JavaScript metode document.getElementById('demo').innerHTML pozvali element sa id-om “demo“ i promenili njegov sadržaj na 'JavaScript je promenio HTML'.

* Kada se koristi innerHTML bitan je redosled elemenata, innerHTMLmetoda se uvek mora nalaziti nakon HTML elementa nad kojim se primenjuje.

Korišćenje metode kao u gore navedenom primeru nema puno smisla jer se sav kod izvrši prilikom učitavanja stranice, ali možemo napraviti da se izvršava na neku akciju, kao u primeru ispod.

<p id="demo">JavaScript moze promeniti HTML.</p>

<button type="button" onclick="document.getElementById('demo').innerHTML = 'JavaScript je promenio HTML'">promeni</button>

Sada opet imamo paragraph koji sadrži atribut ID sa vrednošću “demo“, ali JavaScript metodu smo ubacili unutar onclick atributa HTML elementa button, odnosno dugme.

Prilikom učitavanja stranice prvobitni sadržaj paragrafa “JavaScript moze promeniti HTML.”  
Tek pritiskom na dugme pokrećemo JavaScript metodu koja će promeniti sadržaj paragrafa.

Prilikom osvežavanja stranice sadržaj paragrafa se opet vraća na prvobitnu vrednost.Tako da možemo iskorisiti metodu window.location.reload() koju ubacimo unutar onclick atributa drugog dugmeta, da dobijemo mogućnost vraćanja na prvobitnu vrednost pomoću dugmeta.

<button type="button" onClick="window.location.reload()">reset</button>

Zadatak 2.

Izmeniti prethodni primer za izracunjavanje povrsine i obima kruga tako da se sad rezultat vraca pomocu getElementById() . innerHTML metoda.

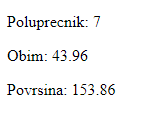
* Poluprečnik se unosi pomoću prompta i čuva u promenjivoj.
* Na osnovu unetog poluprečnika se izračunavaju obim i površina.

Formule:

P=r2π

O=2rπ

* Rezultat se ispisuje u formatu:



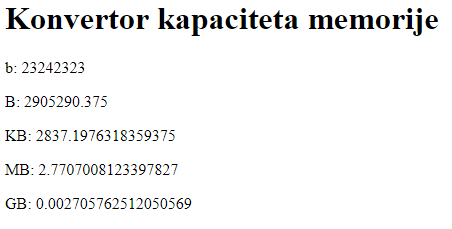
Korisceni elementi su <p> i <span>

Zadatak 3.

Potrebno je napisati JavaScript program koji unetu vrednost kolicine memoruje u bitovima konvertuje u bajtove, KB, MB, GB.

* Kolicina memoruje u bitovima se unosi pomoću prompta i čuva u promenjivoj.
* Nakon toga se izracunavaju ostale jedinice.

Rezultat se ispisuje u formatu:



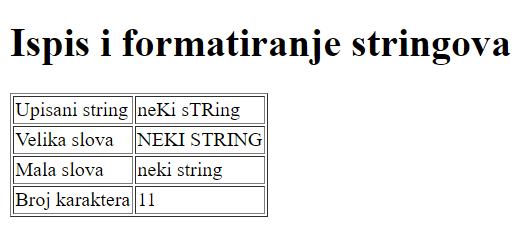
Korisceni elementi su <p> <span> i <h1> za naslov.

Zadatak 4.

Potrebno je napisati JavaScript program koji uneti string formatira i ispisuje duzinu.

* String se unosi pomoću prompta i čuva u promenjivoj.
* Nakon toga se pomocu ugradjenih metoda string pretvara u sva velika slova, u sva mala slova i ispisuje se duzina stinga.
* Ispis rezultata se vrsi unutar tabele.

Tagovi za tabelu <table> <tr> <td>

****