

JavaScript

Što je to?

JavaScript je *skriptni* programski jezik primarno namijenjen upotrebi na web stranicama kako bi im dali određenu dozu dinamičnosti. Službeno ime JavaScripta je ECMAScript čiji standard je donijela organizacija ECMA International. Standard ECMA-262 donešen je 1997., a ISO ga je prihvatio 1998. Autor jezika je Brendan Eich koji ga je razvio za Netscapeov Navigator 2.0 (1996.).

JavaScript kod izvršava web browser koji pregledava stranicu na kojoj se taj kod nalazi. Stoga se ovakav oblik skriptiranja naziva skriptiranje na klijentskoj strani (eng. client side scripting).

Važno: JavaScript i Java su dva **različita** jezika, čija je jedina poveznica slična sintaksa koju dijele s C-om i C++-om.

JavaScript

Što mogu s njim?

JavaScript može

- ubaciti dinamički sadržaj na web stranicu

JavaScript

Što mogu s njim?

JavaScript može

- ubaciti dinamički sadržaj na web stranicu
- reagirati na događaje (npr. prelazak miša preko nekog objekta, klik miša, završetak učitavanja stranice, ...)

JavaScript

Što mogu s njim?

JavaScript može

- ubaciti dinamički sadržaj na web stranicu
- reagirati na događaje (npr. prelazak miša preko nekog objekta, klik miša, završetak učitavanja stranice, ...)
- čitati i mijenjati sadržaj HTML dokumenta

JavaScript

Što mogu s njim?

JavaScript može

- ubaciti dinamički sadržaj na web stranicu
- reagirati na događaje (npr. prelazak miša preko nekog objekta, klik miša, završetak učitavanja stranice, . . .)
- čitati i mijenjati sadržaj HTML dokumenta
- izvršiti validaciju podatka (ako korisnik šalje podatke serveru korisno je validirati podatke na klijentovoj strani prije nego ih pošaljemo serveru, jer na taj način rasterećujemo server)

JavaScript

Što mogu s njim?

JavaScript može

- ubaciti dinamički sadržaj na web stranicu
- reagirati na događaje (npr. prelazak miša preko nekog objekta, klik miša, završetak učitavanja stranice, . . .)
- čitati i mijenjati sadržaj HTML dokumenta
- izvršiti validaciju podatka (ako korisnik šalje podatke serveru korisno je validirati podatke na klijentovoj strani prije nego ih pošaljemo serveru, jer na taj način rasterećujemo server)
- detektirati koji browser korisnik koristi (na taj način možemo korisniku prezentirati stranicu prilagođenu njegovom browseru)

JavaScript

Što mogu s njim?

JavaScript može

- ubaciti dinamički sadržaj na web stranicu
- reagirati na događaje (npr. prelazak miša preko nekog objekta, klik miša, završetak učitavanja stranice, . . .)
- čitati i mijenjati sadržaj HTML dokumenta
- izvršiti validaciju podatka (ako korisnik šalje podatke serveru korisno je validirati podatke na klijentovoj strani prije nego ih pošaljemo serveru, jer na taj način rasterećujemo server)
- detektirati koji browser korisnik koristi (na taj način možemo korisniku prezentirati stranicu prilagođenu njegovom browseru)
- kreirati “kolačiće” (eng. cookies) (mehanizam zapisivanja i čitanja podataka na korisnikovom računalu)

JavaScript

Gdje dolazi kod?

Kako JavaScript kod izvršava browser, treba ga ubaciti unutar HTML dokumenta. To možemo izvesti na dva načina. Prvi način je da kod ubacimo unutar

```
<script type="text/javascript">i</script>
```

tagova, što je pogodno za kratke kodove. Veće kodove možemo snimiti unutar posebne .js datoteke, i onda ga unutar HTML dokumenta uključiti pomoću

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

Ako se JavaScript kod nalazi unutar `body` elementa, onda će se izvršiti odmah prilikom učitavanja stranice, ukoliko to ne želimo, nego se taj kod treba izvršavati nakon nekog specijalnog događaja, onda ga treba staviti unutar `head` elementa i pozvati ga kada želimo da se izvrši. (Kako se JavaScript kod poziva nakon nekog događaja, vidjet ćemo kasnije.)

JavaScript

osnove

- JavaScript je (kao i C) *case sensitive* tj. razlikuje velika i mala slova.

JavaScript

osnove

- JavaScript je (kao i C) *case sensitive* tj. razlikuje velika i mala slova.
- Naredbe (eng. statements) se razdvajaju prelaskom u novi red ili točka-zarezom (;). Stavljati točku-zarez na kraju linije nije obavezno, ali je uobičajeno.

JavaScript

osnove

- JavaScript je (kao i C) *case sensitive* tj. razlikuje velika i mala slova.
- Naredbe (eng. statements) se razdvajaju prelaskom u novi red ili točka-zarezom (;). Stavljanje točka-zarez na kraju linije nije obavezno, ali je uobičajeno.
- Blokovi se (kao i u C-u) odvajaju vitičastim zagradama ({ i }).

JavaScript

osnove

- JavaScript je (kao i C) *case sensitive* tj. razlikuje velika i mala slova.
- Naredbe (eng. statements) se razdvajaju prelaskom u novi red ili točka-zarezom (;). Stavljati točku-zarez na kraju linije nije obavezno, ali je uobičajeno.
- Blokovi se (kao i u C-u) odvajaju vitičastim zagradama ({ i }).
- Komentari (kao u C-u) pišu unutar /* i */ ili (kao u C++-u) nakon //.

JavaScript

varijable

- Pravila za imenovanje varijabli su identična kao i u C-u.

JavaScript

varijable

- Pravila za imenovanje varijabli su identična kao i u C-u.
- Varijable nemaju tipove tj. u svaku varijablu možemo spremiti vrijednosti bilo kojeg tipa. (Vrijednosti imaju tipove!)

JavaScript

varijable

- Pravila za imenovanje varijabli su identična kao i u C-u.
- Varijable nemaju tipove tj. u svaku varijablu možemo spremiti vrijednosti bilo kojeg tipa. (Vrijednosti imaju tipove!)
- Deklaracija varijabli izgleda ovako:

```
var a;  
var x = 5;  
var carname="Volvo";
```

- Varijable možete deklarirati na bilo kojem mjestu u kodu.

JavaScript

varijable

- Pravila za imenovanje varijabli su identična kao i u C-u.
- Varijable nemaju tipove tj. u svaku varijablu možemo spremiti vrijednosti bilo kojeg tipa. (Vrijednosti imaju tipove!)
- Deklaracija varijabli izgleda ovako:

```
var a;  
var x = 5;  
var carname="Volvo";
```

- Varijable možete deklarirati na bilo kojem mjestu u kodu.
- Varijable nije obavezno deklarirati. U slučaju da varijabla nije deklarirana, njeno prvo korištenje smatra se deklaracijom.

JavaScript

operatori (1)

- aritmetički operatori (+ - * / % ++ --) i operatori pridruživanja (= += -= *= /= %=) su istovjetni C-ovskim operatorima

JavaScript

operatori (1)

- aritmetički operatori (+ - * / % ++ --) i operatori pridruživanja (= += -= *= /= %=) su istovjetni C-ovskim operatorima
- operator + primjenjen na stringovima označava konkatenciju. **Oprez**, 5 je broj, a "5" je string. Ako "zbrajamo" string i broj, onda se broj pretvara u string, dakle 5+5 je 10, ali 5+"5" je "55"

JavaScript

operatori (1)

- aritmetički operatori (+ - * / % ++ --) i operatori pridruživanja (= += -= *= /= %=) su istovjetni C-ovskim operatorima
- operator + primjenjen na stringovima označava konkatenciju. **Oprez**, 5 je broj, a "5" je string. Ako "zbrajamo" string i broj, onda se broj pretvara u string, dakle 5+5 je 10, ali 5+"5" je "55"
- uz C-ovske operatore uspoređivanja (== != > < >= <=) postoji i operator ===
- === znači jednako i po vrijednosti i po tipu

```
5 == 5      // true
5 === 5     // true
5 == "5"    // true
5 === "5"   // false
```

JavaScript

operatori (2)

- logički operatori (`&&` `||` `!`) istovjetni su C-ovskim
- postoji i ternarni uvjetni operator (`? :`) koji je također istovjetan C-ovskom

JavaScript

grananje

- `if..else` kao u C-u
- `switch` kao u C-u

JavaScript

petlje

- `for` kao u C-u
- `while` kao u C-u
- `do..while` kao u C-u
- Za razliku od C-a, postoji i `for..in` petlja (vidi primjer) koja omogućava elegantno iteriranje kroz članove niza.

JavaScript

funkcije

- Funkcije se definiraju slično kao u C-u, no kako varijable nemaju tipove, ne navode se tipovi argumenata niti tip povratne vrijenosti

```
// funkcija s 3 argumenta:  
function naziv_funkcije(var1, var2, var3){  
    tijelo funkcije  
}
```

```
// funkcija bez argumenata:  
function naziv_funkcije2(){  
    tijelo funkcije  
}
```

- Funkcija može (ne mora) imati povratnu vrijednost. Vrijednost se vraća pomoću naredbe `return` (kao u C-u).
- Kako bi bili sigurni da se vaše funkcije neće izvršavati dok ih ne pozovete, držite njihove definicije unutar `head` elementa. (primjer malo kasnije)

JavaScript

popup boxes

- Alert Box – služi za predočavanje neke informacije korisniku. Kada se pojavi alert box, korisnik mora kliknuti na “OK” prije nego što nastavi s radom.

```
alert("neki tekst");
```


JavaScript

popup boxes

- Alert Box – služi za predočavanje neke informacije korisniku. Kada se pojavi alert box, korisnik mora kliknuti na “OK” prije nego što nastavi s radom.

```
alert("neki tekst");
```

- Confirm Box – koristi se kada korisnik treba nešto potvrditi. Kada se pojavi alert box, korisnik mož kliknuti na “OK” ili “Cancel” prije nego što nastavi s radom. Klik na “OK” vraća `true`, a klik na “Cancel” vraća `false`.

```
confirm("neki tekst");
```

JavaScript

popup boxes

- Alert Box – služi za predočavanje neke informacije korisniku. Kada se pojavi alert box, korisnik mora kliknuti na “OK” prije nego što nastavi s radom.

```
alert("neki tekst");
```

- Confirm Box – koristi se kada korisnik treba nešto potvrditi. Kada se pojavi alert box, korisnik mož kliknuti na “OK” ili “Cancel” prije nego što nastavi s radom. Klik na “OK” vraća `true`, a klik na “Cancel” vraća `false`.

```
confirm("neki tekst");
```

- Prompt Box – koristi se kada korisnik treba unijeti neke podatke. Kada se pojavi alert box, korisniku se prezentira polje u koje može unijeti tražene podatke, te mož kliknuti na “OK” ili “Cancel” prije nego što nastavi s radom. Klik na “OK” vraća unesene podatke, a klik na “Cancel” vraća `null`.

```
prompt("opis", "defaultna vrijednost");
```

JavaScript

dogadjaji (eng. events)

Neki češće korišteni događaji:

- onload, onUnload – aktiviraju se kada korisnik dođe na stranicu (učita je), odnosno ode s nje
- onmousemove – aktivira se kad se pomakne miš
- onmousedown, onmouseup – aktiviraju se kad se pritisne, odnosno otpusti neka tipka miša (dok je miš nad elementom)
- onmouseover, onmouseout – aktiviraju se kad se kursor miša nađe nad elementom, odnosno skloni s njega.

Kompletnu listu događaja možete vidjeti na

http://www.w3schools.com/jsref/jsref_events.asp

primjer

(vidi primjer)

JavaScript ima veliku količinu predefiniраних типова objekata (klasa). Objekti u JavaScriptu imaju

- svojstva (eng. *properties*) – to su vrijednosti pridružene objektu

```
var txt = "Hello, world!"  
document.write(txt.length) // ispisuje: 13
```

- metode (eng. *methods*) – to su akcije (funkcije) koje može izvršavati objekt

```
var str="Hello world!";  
document.write(str.toUpperCase()); // ispisuje:  
// HELLO, WORLD!
```

JavaScript

klasa String

- Varijable u kojima imamo spremljen neki string su u stvari objekti tipa String. Pomoću metoda i svojstava tih objekata možemo manipulirati pohranjenim tekстом.
- Pogledajte `primjer`
- Poptpun popis svih svojstava i metoda klase String možete pronaći na http://www.w3schools.com/jsref/jsref_obj_string.asp

JavaScript

klasa String

- Varijable u kojima imamo spremljen neki string su u stvari objekti tipa String. Pomoću metoda i svojstava tih objekata možemo manipulirati pohranjenim tekstom.
- Pogledajte `primjer`
- Poptpun popis svih svojstava i metoda klase String možete pronaći na http://www.w3schools.com/jsref/jsref_obj_string.asp

JavaScript

klasa Array

- Klasa Array služi za spremanje više vrijednosti u jednu varijebalu, tj. Array objekti su nizovi.
- Pogledajte primjer
- Poptpun popis svih svojstava i metoda klase String možete pronaći na http://www.w3schools.com/jsref/jsref_obj_array.asp

JavaScript

još neke klase

- `Math` – klasa sa matematičkim funkcijama i konstantama
- `Date` – klasa za rad s datumima
- `Window` – reprezentira prozor browsera (odgovara body elementu)
- `Navigator` – informacije o browseru
- `Screen` – informacije o ekranu
- `History` – sadrži posjećene URL-ove
- `Location` – informacije o trenutnom URL-u

HTML Document Object Model (DOM)

uvod

HTML DOM je W3C standard koji definira kako se web stranica reprezentira pomoću skupa objekata. HTML DOM nije dio JavaScripta, nego se radi o posebnom standardu koji koriste razni client side scripting jezici, pa tako i JavaScript. Svakom elementu na stranici pridružen je jedan objekt koji ima svoja svojstva i metode pomoću kojih možemo baratati njime.

Listu HTML DOM objekata možete naći na

http://www.w3schools.com/js/js_obj_htmlDOM.asp.

Korištenje HTML DOM-a, možete vidjeti i na ovom primjeru.