

INTERNET TEHNOLOGIJE I VEB DIZAJN

Uvod u JavaScript programski jezik

predavač: dr Miloš Antonijević

Serverski i klijentski programski jezici

- Programski jezici za razvoj Veb aplikacija se dele na **serverske** i **klijentske**.
- Serverski jezici se izvršavaju na serveru i kao izlaz daju HTML kod koji se šalje klijentu (npr. Web browser-u).
- Klijentski jezici se izvršavaju na strani klijenta, odnosno u Web browser-ima. Veb stranica koju browser dobije od servera može u sebi da sadrži kod pisan u nekom klijentskom jeziku, koji browser interpretira i rezultat prikazuje u okviru Veb stranice.
- Kod pisan u serverskim programskim jezicima je skriven za klijenta, vidljiv je samo izlaz izvršavanja koda.
- U slučaju klijentskih jezika, kod ne može da se sakrije od klijenta, jer je integrisan u sadržaj HTML koda ili se nalazi u odvojenom fajlu (kao CSS) koji browser dobije od servera.

JavaScript

- Osnovna primena JS jezika jeste da obezbedi interaktivnost između korisnika i Veb stranice.
- Integrisan je u sve poznatije Web browser-e, može da se piše u bilo kom tekstualnom editoru, pa ne zahteva instalaciju nikakvog dodatnog okruženja.
- JavaScript jeste osetljiv na mala i velika slova (Case sensitive, za razliku od HTML) i objektno je orijentisan.
- Ako se piše u okviru samog HTML koda, nalazi se u okviru **script** elementa (`<script></script>`).
- JavaScript **nije** Java.
- JavaScript je pisan za Veb, dok Java predstavlja samostalan programski jezik u kome se mogu razvijati sve vrste aplikacija.

Razlike između JavaScripta i Jave

JAVA	JavaScript
Samostalni programski jezik	Skriptni jezik (nema samostalno okruženje)
Stroga definicija promenljivih	Slobodna definicija promenljivih
Podržava višenitno (multi-threading) programiranje	Ne podržava višenitno programiranje
Java kod se kompajlira	JavaScript se interpretira (od strane browser-a)
Različita sintaksa	

Integracija JS u HTML kod

- JS kod može da se koristi u okviru same HTML stranice ili u obliku eksternog fajla.
- Ako se koristi u okviru HTML stranice, JS kod se navodi u okviru **<script></script>** tagova. U HTML 5, <script> tag ne zahteva nikakav atribut da bi se preciziralo da je u pitanju JS kod.
- Pošto se HTML kod izvršava redosledom kojim je pisan, kada naiđe na script, on će ga izvršiti. Ukoliko postoje neka obaveštanja ili čekanje na odgovor korisnika u okviru JS koda, dalji kod HTML stranice se neće učitavati dok korisnik ne odreaguje.
- JS se može nalaziti unutar head ili body elementa.
- Ako se JS kod nalazi u eksternom fajlu (snimljenom sa .js ekstenzijom), može se pozvati iz HTML koda korišćenjem src atributa u okviru script taga.

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

Definisanje promenljivih

- Promenjive se definišu navođenjem imena promenjive nakon rezervisane reči **var**.
- Imena promenljivih i funkcija (metoda) su case sensitive.
- Ime promenjive ne može da sadrži space, a može da sadrži slova, brojeve, donju crtu (_) i znak dolara (\$).
- Prvi karakter u imenu promenjive ne može da bude broj.
- Vrednost promenjive se dodelju znakom jednakom (=).
- U JavaScriptu dvostruki (") i jednostruki (') navodnici imaju isto značenje.
- Brojevi se dodeljuju kao vrednost promenjive bez navodnika, a nizovi karaktera (stringovi) u okviru navodnika.
- Blokovi naredbi se završavaju sa tačka-zarez (;)
- Komentari u JS-u mogu biti jednoredni (//) ili višeredni (između /* i */)

Aritmetičke i logičke operacije

- Korišćenje aritmetičkih operacija je nasleđeno iz C programskog jezika.
- Osnovni aritmetički operatori su:

+ **-** ***** ******(stepenovanje) **/** **%**(ostatak pri deljenju)
++(inkrementiranje) **--**(dekrementiranje)

- Kada se operator nalazi ispred promenjive onda se prvo obavi operacija, pa se dodeli vrednost, a kada se nalazi iza promenjive, onda se dodeljivanje vrednosti obavi pre izvršenja operacija

```
var a = 1;
```

```
b = a++;      (b=1, a=2)
```

```
b = ++a;      (b=2, a=2)
```

- Logički operatori su **&&**(and) **||**(or) **!**(not)

Operatori dodele vrednosti

- Vrednost promenjive se dodeljuje korišćenjem operatora jednako ili kombinacijom aritmetičkog operatora i znaka jednako

= dodeljuje vrednost promenjive sa desne strane promenjivoj sa leve

+= dodaje vrednost promenjive sa desne strane promenjivoj sa leve i dodeljuje vrednost

-= oduzima vrednost promenjive sa desne strane od promenjive sa leve i dodeljuje vrednost

...

- Odnosno:

`a += b` je isto što i `a = a + b;`

`a /= b` je isto što i `a = a / b;`

...

Prikaz i unos podataka

- Prikaz podataka u okviru Veb stranice se najjednostavnije obavlja korišćenjem naredbe **document.write** ili u formi prozora naredbom **alert**. Komanda **alert** blokira renderovanje stranice dok korisnik ne izvrši interakciju sa otvorenim prozorom.
- Unos podataka u okviru Veb stranice se obavlja preko HTML forme, a unos putem posebnog (iskaćućeg) prozora se obavlja metodom **prompt** (npr. `prompt("Unesite tekst i pritisnite OK:", "Tekst")`).
- Uneti tekst se može dodeliti nekoj promenljivoj (npr. `var unetitekst = prompt("Unesite tekst i pritisnite OK:", "Tekst")`).
- Metoda **confirm** prikazuje iskaćući prozor koji sadrži pitanje i dve opcije – OK i Cancel (npr. `confirm("želite li da prikazete alert prozor?")`).
- Metoda **confirm** vraća vrednost tipa boolean, pa se najčešće koristi u kombinaciji sa uslovnom naredbom (**if**).

Rad sa datumom i vremenom

- Za razliku od serverskih programskih jezika koji podatke o vremenu uzimaju sa servera, klijentski jezici podatke dobijaju od sistemskog vremena, odnosno operativnog sistema na klijentskom računaru.
- Korišćenjem klase Date moguće je pribavljati i manipulirati podacima o vremenu.

```
var date = new Date();  
alert(date.getHours());
```

- Slično **get** metodama postoje i **set** metode koje menjaju odgovarajući podatak datuma.

getFullYear()	pribavlja podatak o godini
getMonth()	pribavlja podatak o mesecu
getDate()	pribavlja podatak o danu u mesecu
getDay()	pribavlja podatak o rednom broju dana u nedelji
getHours()	pribavlja podatak o satima
getMinutes()	pribavlja podatak o minutima
getSeconds()	pribavlja podatak o sekundama
getMilliseconds()	pribavlja podatak o milisekundama

Rad sa datumom i vremenom (2)

- Objekat klase **Date** sadrži informaciju o broju milisekundi koji je prošao od početka UNIX epohe, odnosno od 01.01.1970. (<https://www.epochconverter.com/>)
- Konstruktor klase Date može sadržati nula ili više parametara koji ukazuju na datum koji se želi kreirati.

```
var date = Date('December 17, 1995 13:24:00');
```

- dateString parametar je string u formatu koji prepoznaje Date.parse() metoda
(<https://datatracker.ietf.org/doc/html/rfc2822#page-14>)

```
alert(Date.parse('12 Dec 2021 14:12:34 GMT'))
```

new Date()
new Date(value)
new Date(dateString)
new Date(year, monthIndex)
new Date(year, monthIndex, day)
new Date(year, monthIndex, day, hours)
new Date(year, monthIndex, day, hours, minutes)

Naredbe za kontrolu toka izvršavanja

if

```
if (uslov) {  
    }  
else if {  
    }  
else {  
    }
```

switch

```
switch(promenjiva) {  
    case a:  
        break;  
    case b:  
        break;  
    default:  
    }
```

Petlje

for

```
for (pocetni izraz; uslov; promena izraza) {  
    }  
}
```

while

```
while(uslov) {  
    }  
}
```

do

```
do {  
    } while (uslov);
```

forEach (za nizove)

```
objekat.forEach(function(item, index) {  
    alert(index + “.” + item);  
})
```

Korišćenje funkcija

- Deklaracija funkcija u JS-u počinje rezervisanom rečju **function**, nakon koje sledi ime funkcije.
- Parametri funkcije se navode u zagradama () koje prate ime funkcije.
- Telo (kod) funkcije se definiše između uglastih zagrada { }.
- Ukoliko funkcija vraća neku vrednost, ona se navodi nakon ključne reči **return**.

```
function ime_funkcije(param1, param2,...) {  
    // telo funkcije  
  
    return povratna_vrednost;  
}
```

Opseg važenja promenljivih

- Po opsegu važenja promenjive se dele na **globalne** i **lokalne** (u skoro svim programskim jezicima, a u nekima postoje i super globalne).
- Lokalne promenjive su one koje imaju svoju vrednost i definisanost u okviru programskog bloka u kome su kreirane (npr. u okviru neke funkcije).
- Globalne promenjive čuvaju svoje vrednost unutar celog JS koda, odnosno Veb stranice, tj. važe izvan funkcija.
- Super globalne (ne postoje u JS) su promenjive koje svoje važenje protežu između više odvojenih Veb stranica.

```
var glob_a = 2;
```

```
var glob_b = 3;
```

```
function zbir_brojeva() {  
    var zbir = glob_a + glob_b;  
}
```

PITANJA?