


 COURSE DEVELOPMENT PROGRAM +
 financed by
 Austrian
 Development Cooperation

Programiranje za Internet



predavanja
 v.as.mr. Samir Lemeš
 slemes@mf.unze.ba

22. JavaScript

- Varijable i automatsko pretvaranje tipova
- Pisanje vrijednosti i komentara
- Svojstva objekata i nizovi
- Kontrola toka programa
- Operatori i njihovi prioriteti

Varijable i automatsko pretvaranje tipova

- Postoji ograničenje za varijable u JavaScriptu: ne smiju počinjati brojevima.
- Prvi znak smije biti samo slovo ili oznaka podvlake (_).
- Ne postoji ograničenje dužine imena varijable.
- Za deklarisanje varijabli u JavaScriptu koristi se naredba **var**

Varijable i automatsko pretvaranje tipova

- JavaScript ne zahtijeva da varijable budu deklarirane na početku programa
- U JavaScriptu ne postoje tipovi podataka.
- JavaScript obavlja dinamičko pretvaranje podataka tijekom izvršavanja skripte
- JavaScript će sam odrediti kojeg tipa treba biti pojedina varijabla onog trenutka kada je uvedete u program.

Varijable i automatsko pretvaranje tipova

- Primjer, ako koristite dvije varijable koje su prvi puta spomenute:
`Varijabla1="15"`
`Varijabla2=5`
- Tada će **varijabla1** biti string (zbog toga što je njen sadržaj obuhvaćen navodnicima), dok će **varijabla2** biti cjelobrojnog tipa.

Varijable i automatsko pretvaranje tipova

- `X = varijabla1 + varijabla2`
`Y = varijabla2 + varijabla1`
- Pretvaranje se obavlja s lijeva na desno. Operand s desne strane operatora (desno od plusa) bit će pretvoren u tip operanda s lijeve strane operatora.
 - X: varijabla2 će prvo biti pretvorena u string, a zatim će taj string biti "zbrojen" sa stringom pohranjenom u varijabli varijabla1. Nova vrijednost varijable x (koja je u međuvremenu također postala tip string) bit će "155".
 - Y: varijabla1 se iz stringa mora pretvoriti u cijeli broj i zatim pomnožiti s vrijednošću pohranjenom u varijabli varijabla2. Varijabla y bit će cjelobrojnog tipa, a sadržavat će rezultat množenja 5 sa 15, odnosno broj 75.

Variable i automatsko pretvaranje tipova

- Ne može se sve konvertovati.
- Naprimjer, string se može pretvoriti u cijeli broj samo ako ne sadrži niti jedno slovo.
- Drugi problem je u nedefiniranim operacijama:
`x = varijabla1 - varijabla2`
- Varijabla `varijabla2` očito se mora pretvoriti u string, ali operacija oduzimanja stringova nije definirana. U svim će takvim slučajevima doći do ispisivanja pogreške u prozoru browsera.

Pisanje vrijednosti i komentara

- Stringovi se navode pod navodnicima, koji mogu biti i jednostruki (tzv. literal) i dvostruki.
- Primjeri stringova su "Znak" i 'Znak'.
- Posebne znakove potrebno je unutar stringova navoditi kao prekidne nizove (*escape sequence*).

Pisanje vrijednosti i komentara

Escape sekvenca	Značenje	Opis
\	novi red	nastavak u novom redu
\n	LF	prelazak u novi red (line feed)
\t	HT	tabulator
\b	BS	backspace
\r	CR	prelazak u novi red (carriage return)
\f	FF	prelazak na novu stranu (form feed)
\\	\	kosa crta
\'	'	apostrof
\"	"	navodnik

Pisanje vrijednosti i komentara

- Cijeli brojevi mogu se pisati u decimalnom, oktalanom ili heksadecimalnom obliku.
- Heksadecimalni brojevi se pišu uz prefiks 0x (0x2FC4).
- Brojevi s pomičnim zarezom mogu se pisati i u eksponencijalnom obliku, npr. 2E-5.
- Logičke varijable mogu poprimiti vrijednost TRUE ili FALSE.

Pisanje vrijednosti i komentara

- Operatori za upoređivanje:
 - jednakost (`==`),
 - nejednakost (`!=`),
 - veće od (`>`),
 - manje od (`<`),
 - manje ili jednako (`<=`)
 - veće ili jednako (`>=`).
- Komentari se pišu između para oznaka `/*` i `*/`, pri čemu se mogu protezati i kroz više redova, ili se navode iza oznake `//`, pa se u tom slučaju komentarom smatra sve što se nalazi do kraja reda.

Svojstva objekata i nizovi

- Objektima se pristupa tako da se prvo navede ime objekta, a zatim osobina koju treba promijeniti ili kojoj treba pristupiti.
- Naprimjer:
`Knjiga.stranica=1050`
će osobini `stranica` od objekta `knjiga` pridružiti navedenu vrijednost. Ako osobina `stranica` ne postoji u objektu `knjiga`, ona će mu biti dodata.

Svojstva objekata i nizovi

- S nizovima se u JavaScriptu radi na sličan način kao što se pristupalo objektima.
- Primjer:
`Knjiga ["stranica"]=1050`
- Nizovima se općenito pristupa tako da se unutar uglatih zagrada navede indeks člana koji se želi promijeniti ili čija se vrijednost želi dobiti.

Kontrola toka programa

- Naredba `if ... else` za postavljanje logičkih uvjeta ima istu sintaksu kao u Javi ili C++.
- Umjesto naredbe `if ... else` može se opet koristiti i skraćeno pisanje pomoću upitnika i dvotačke.

```
if (logički izraz) {  
  ...  
}  
else {  
  ...  
}
```

Kontrola toka programa

- Naredbe za stvaranje petlji, `for` i `while` također se pišu jednako kao u Javi ili C++.

```
for (izraz; logički izraz; izraz) {  
  ...  
}
```

- Naredbe `while` i `continue` mogu se, kao i u Javi i C++, koristiti za iskakanje iz petlje i nastavljjanje izvršavanja od neke tačke programa.

Operatori i njihovi prioriteti

Prioritet	Operator	Opis
1	()	zgrade se koriste za promjenu prioriteta
2	+ +	inkrementiranje (unarni operator)
	- -	dekrementiranje
	!	logički komplement (unarni operator)
	~	unarni komplement (unarna negacija)
	+	unarni pozitivni predznak
	-	unarna negacija
3	*	množenje
	/	dijeljenje
	%	ostatak cjelobrojnog dijeljenja (modulo)
4	+	sabiranje
	-	oduzimanje

Operatori i njihovi prioriteti

Prioritet	Operator	Opis
5	<<	shift bitova ulijevo
	>>	shift bitova udesno
	>>>	shift bitova udesno uz popunjavanje nulama
6	<	manje od
	<=	manje ili jednako
	>	veće od
	>=	veće ili jednako
7	= =	jednako
	! =	različito od (nejednako)
8	&	AND
9		OR
10	: =	dodjeljivanje vrijednosti