

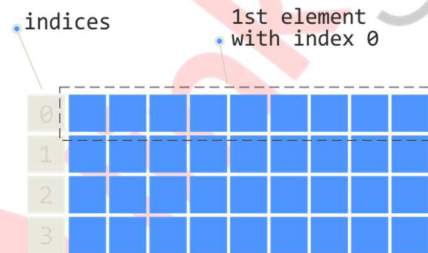
Višedimenzionalni nizovi

U prethodnoj lekciji bilo je reči o osnovama korišćenja nizova u JavaScript jeziku. Tako su ilustrovane osnovne osobine nizova i tehnike za manipulaciju njihovim članovima. Ipak, u prethodnoj lekciji isključivo je bilo reči o jednoj, osnovnoj vrsti nizova. Takvi nizovi se drugačije nazivaju jednodimenzionalni nizovi

Kada se za neki niz kaže da je jednodimenzionalan, misli se da su njegovi članovi prosti tipovi podataka (`number`, `string`, `boolean`). Ipak, članovi nizova mogu biti i složeni, odnosno kompozitni tipovi podataka, pa je potpuno legitimno da članovi nizova budu neki drugi nizovi. U takvoj situaciji se govori o nizovima sa više dimenzija, odnosno o višedimenzionalnim nizovima, koji će biti predmet lekcije koja je pred vama.

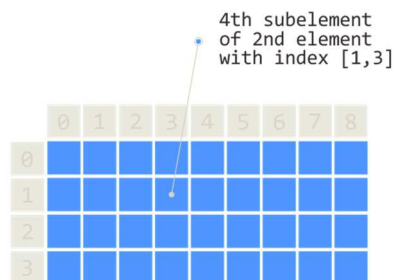
Pojam višedimenzionalnih nizova

Za nizove u JavaScript jeziku se može reći da su vrlo fleksibilni, zato što dozvoljavaju lako dodavanje i uklanjanje elemenata i smeštanje vrednosti različitih tipova. Tako nizovi nisu ograničeni samo na proste tipove, već je unutar njih moguće smestiti i kompozitne, odnosno složene vrednosti. To sve na kraju znači da je unutar jednog niza moguće smestiti više drugih nizova ili objekata. U takvoj situaciji se dobijaju nizovi sa više dimenzija (slika 16.1).



Slika 16.1. Višedimenzionalni niz

Na slici 16.1. je prikazana struktura jednog višedimenzionalnog niza sa četiri elementa. Pri tome je svaki od elemenata zaseban niz (svi oni su predstavljeni plavom bojom). Tako se na primer može reći da je prvi element niza, na poziciji 0, zapravo još jedan niz. S obzirom na to da su i sami elementi nizovi, i oni poseduju svoje indekse (slika 16.2).



Slika 16.2. Višedimenzionalni niz (2)

Sada su, pored indeksa glavnog niza, prikazani i indeksi pojedinačnih elemenata. Na taj način četvrti element drugog elementa glavnog niza jednoznačno je određen indeksima [1, 3].

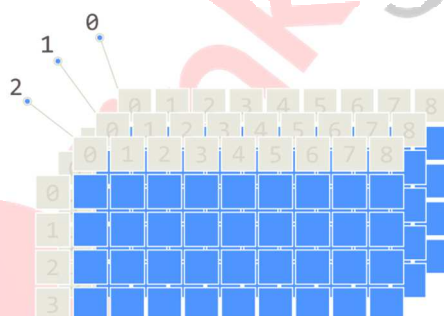
Takođe, pojedinačni nizovi koji se nalaze unutar glavnog niza ne moraju imati isti broj elemenata (slika 16.3).

	0	1	2	3	4	5	6	7	8
0	[0,0]	[0,1]	[0,2]	[0,3]	[0,4]	[0,5]			
1	[1,0]	[1,1]	[1,2]	[1,3]	[1,4]	[1,5]	[1,6]	[1,7]	[1,8]
2	[2,0]	[2,1]	[2,2]	[2,3]					
3	[3,0]	[3,1]	[3,2]	[3,3]	[3,4]				

Slika 16.3. Višedimenzionalni niz (2)

Slika 16.3. ilustruje jedan višedimenzionalni niz, čiji elementi imaju različit broj podelemenata.

Svi do sada prikazani višedimenzionalni nizovi imali su dve dimenzije, pa se zbog toga oni drugačije nazivaju dvodimenzionalni nizovi. Ipak, višedimenzionalni nizovi mogu imati proizvoljan broj dimenzija (dve, tri, četiri...).



Slika 16.4. Trodimenzionalni niz

Na slici 16.4. prikazan je jedan višedimenzionalni niz sa tri dimenzije. On ima tri elementa (0, 1 i 2), od kojih je svaki po jedan dvodimenzionalni niz.

Zbog čega su nam potrebni višedimenzionalni nizovi?

U nastavku ove lekcije mi ćemo se fokusirati na nizove sa dve dimenzije. Nizovi sa više od dve dimenzije se veoma retko koriste, pogotovu kada je u pitanju frontend programiranje i razvoj web sajtova. Ipak, u ovom trenutku se može postaviti pitanje opravdanosti korišćenja nizova i sa dve dimenzije.

Nizovi sa dve dimenzije veoma su korisni za predstavljanje podataka u tabelarnoj formi. Tako se nizovi sa slike 16.1, 16.2. i 16.3. sada mogu predstaviti na sledeći način (slika 16.5).

0	1	john	04as89...	john@mail.com
1	2	emma	7dh9hh...	emma@mail.com
2	3	ben	9fgj44...	ben@mail.com
3	4	ava	dfh57h...	ava@mail.com

Slika 16.5. Tabelarni prikaz podataka

Uvidom u sliku 16.5. stvari postaju mnogo jasnije. Na slici 16.5. zapravo je prikazana jedna tabela sa četiri reda i četiri kolone. U tabeli su navedeni pristupni podaci korisnika nekog sajta. U prvoj koloni su prikazani identifikacioni brojevi korisnika (1, 2, 3...). Druga kolona koristi se za prikaz korisničkih imena. Treća kolona namenjena je prikazu lozinki, a četvrta za prikaz email adresa.

Ovakva vrsta tabelarnih podataka veoma lako se može predstaviti korišćenjem dvodimenzionalnih nizova. Tako se prikazana tabela može predstaviti korišćenjem niza sa dve dimenzije i četiri člana. Svaki od pojedinačnih nizova koji predstavljaju podatke jednog korisnika bili bi članovi takvog niza. Stoga su na slici 16.5. njihovi indeksi obeleženi sasvim levo, sivom bojom (0, 1, 2, 3).

Kreiranje dvodimenzionalnih nizova

Kreiranje dvodimenzionalnih nizova veoma je lako i ne zahteva korišćenje bilo kakve specijalne sintakse. U nastavku će biti prikazano nekoliko načina na koje je tabelu sa slike 16.5. moguće pretvoriti u dvodimenzionalni niz.

Najlakše je započeti kreiranje jednog praznog niza:

```
var users = [];
```

Promenljiva `users` predstavlja jedan prazan niz koji je dobijen navođenjem otvorene i zatvorene uglaste zagrade. Elemente je sada ovakvom nizu moguće dodavati na sledeći način:

```
users[0] = [1, "john", "04as89v4v", "john@gmail.com"];
users[1] = [2, "emma", "7dh9hh4t4", "emma@gmail.com"];
users[2] = [3, "ben", "9fghj44858", "ben@gmail.com"];
users[3] = [4, "ava", "dfh57h24gv", "ava@gmail.com"];
```

Korišćenjem već viđene sintakse sada su kreirana četiri (4) elementa unutar niza `users`.

Svaki od elemenata je pojedinačni niz sa podacima jednog korisnika (engl. *user*).

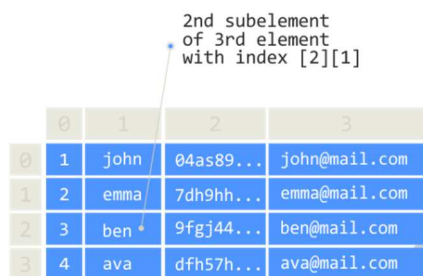
Sve ovo je moglo biti urađeno i na nešto drugačiji način:

```
var users = [[1, "john", "04as89v4v", "john@gmail.com"],
             [2, "emma", "7dh9hh4t4", "emma@gmail.com"],
             [3, "ben", "9fghj44858", "ben@gmail.com"],
             [4, "ava", "dfh57h24gv", "ava@gmail.com"]];
```

Sada je kompletan proces deklaracije i inicijalizacije dvodimenzionalnog niza obavljen unutar jedne naredbe. Kôd je nešto kompaktniji, ali manje pregledan i teži za održavanje.

Čitanje podataka dvodimenzionalnih nizova

Čitanje podataka dvodimenzionalnih nizova obavlja se korišćenjem dva para uglastih zagrada, unutar kojih se navode dva indeksa. Prvi indeks predstavlja poziciju jednodimenzionalnog niza unutar dvodimenzionalnog niza. Drugi indeks označava element unutar jednodimenzionalnog niza (slika 16.6).



0	1	2	3
0	1	john	04as89... john@mail.com
1	2	emma	7dh9hh... emma@mail.com
2	3	ben	9fgj44... ben@mail.com
3	4	ava	dfh57h... ava@mail.com

Slika 16.6. Tabelarni prikaz podataka

Uzimajući u obzir sve što je rečeno, ukoliko je potrebno doći do vrednosti `ben`, koja se nalazi unutar prikazanog dvodimenzionalnog niza, dovoljno je napisati:

```
console.log(users[2][1]);
```

Nakon izvršavanja prikazane naredbe, unutar konzole se dobija:

```
ben
```

Pitanje

Dat je sledeći niz:

```
var arr = [[1, " John", " Johnson", "1111111111111"],  
[2, " Robert", " Anderson", "2222222222222"],  
[3, " Thomas", " Miller", "3333333333333"]];
```

Na koji način se može doći do vrednosti `Anderson`?

- `arr[1][1]`
- **`arr[1][2]`**
- `arr[2][3]`
- `arr[3][2]`

Objašnjenje:

Pristup vrednostima višedimenzionalnih nizova nije ništa drugačiji od pristupa vrednosti običnim nizovima u JavaScriptu. Elementima se pristupa korišćenjem uglastih zagrada – `[]`. Prilikom rada sa dvodimenzionalnim nizovima neophodno je koristiti dva para takvih zagrada i unutar njih navesti dva indeksa. Prvi indeks se odnosi na glavni niz, a drugi na podniz.

Prolazak kroz dvodimenzionalne nizove

U prethodnoj lekciji ilustrovani su načini za prolazak kroz nizove, i to korišćenjem nekoliko različitih petlji koje postoje u JavaScriptu. Sve prikazane petlje primenljive su i nad višedimenzionalnim nizovima, uz malu razliku u logici prolaska. Stoga će u nastavku biti prikazani različiti načini za prolazak kroz dvodimenzionalne nizove.

Prethodni primer u ovoj lekciji ilustrovao je ispisivanje vrednosti `ben` unutar konzole. Reč je o korisničkom imenu jednog od korisnika koji su *spakovani* unutar dvodimenzionalnog niza. Ali šta ukoliko je potrebno ispisati korisnička imena svih korisnika? Svakako, moglo bi se napisati nešto ovakvo:

```
console.log(users[0][1]);  
console.log(users[1][1]);  
console.log(users[2][1]);  
console.log(users[3][1]);
```

Na ovaj način, unutar konzole se dobija:

```
john  
emma  
ben  
ava
```

Ipak, zamislite da niz `users` poseduje nekoliko hiljada elemenata. Jasno je da u takvoj situaciji formulisanje pojedinačnih naredbi ne dolazi u obzir, već je neophodno iskoristiti neku od petlji. Korišćenje `for` petlje za ispis svih korisničkih imena može da izgleda ovako:

```
for (let i = 0; i < users.length; i++) {  
    console.log(users[i][1]);  
}
```

Ovakva petlja obaviće ispis svih korisničkih imena, bez obzira na to da li unutar niza postoji jedan korisnik ili hiljadu korisnika. Korišćenjem svojstva `length` obavlja se čitanje dužine glavnog niza, a zatim se takva vrednost koristi kao uslov za završetak petlje. Unutar tela petlje, u svakoj iteraciji se obavlja pristup korisničkom imenu korisnika, odnosno elementu sa indeksom 1. Indeks glavnog niza (unutar prvog para uglastih zagrada) dinamički se postavlja na osnovu vrednosti brojača `i`.

Nešto komplikovaniji scenario predstavlja čitanje svih krajnjih elemenata dvodimenzionalnih nizova. Za obavljanje takvog posla neophodno je kreirati dve petlje i to jednu unutar druge:

```
for (let i = 0; i < users.length; i++) {  
    for (let j = 0; j < users[i].length; j++) {  
        console.log(users[i][j]);  
    }  
}
```

Na ovaj način se unutar konzole dobija:

```
1
john
04as89v4v
john@gmail.com
2
emma
7dh9hh4t4
emma@gmail.com
3
ben
9fghj44858
ben@gmail.com
4
ava
dfh57h24gv
ava@gmail.com
```

Sada se može videti da je unutar konzole obavljen ispis svih pojedinačnih elemenata niza. To je postignuto korišćenjem dve petlje, jedne unutar druge. Spoljašnjom petljom prolazi se kroz glavni niz, a unutrašnjom kroz elemente svakog pojedinačnog niza. Bitno je primetiti da su prilikom formiranja petlji iskorišćeni brojači različitih naziva (*i* i *j*).

Primer – Prolazak kroz dvodimenzionalni niz

U narednim redovima će biti prikazan još jedan primer prolaska kroz dvodimenzionalni niz.

Niz kroz koji je potrebno *proći* izgleda ovako:

```
let arr = [
  ['apple', 'orange', 'pear'],
  ['carrots', 'beans', 'peas'],
  ['cookies', 'cake', 'muffins', 'pie']
];
```

Prvi način za ispisivanje svih elemenata ovakvog niza jeste korišćenje dve *for* petlje:

```
for (let i = 0; i < arr.length; i++) {
  for (let j = 0; j < arr[i].length; j++) {
    document.write(arr[i][j] + "<br/>");
  }
  document.write("<hr>");
}
```

Spoljašnjom *for* petljom se prolazi kroz tri člana dvodimenzionalnog niza, a unutrašnjom kroz elemente jednodimenzionalnih nizova. Vrednosti se štampaju direktno unutar HTML dokumenta korišćenjem *document.write()* metode. Takođe, nakon štampe svih vrednosti svakog pojedinačnog niza, obavlja se ispis i jednog *hr* HTML elementa, koji će odvojiti članove pojedinačnih nizova.

Nešto kompaktniji način za obavljanje istog posla jeste upotreba `for...of` petlji:

```
for (let outer of arr) {  
  for (let value of outer) {  
    document.write(value + "<br/>");  
  }  
  document.write("<hr>");  
}
```

Sada su `for` petlje zamenjene `for...of` petljama, kojima se na nešto jednostavniji način dolazi do krajnjih vrednosti. Nije potrebno deklarirati brojače i voditi računa o njihovom uvećavanju. U obe situacije unutar HTML dokumenta se dobija identičan efekat.

Rezime

- Članovi nizova mogu biti i složeni, odnosno kompozitni tipovi podataka, pa je potpuno legitimno da članovi nizova budu neki drugi nizovi, čime se dobijaju višedimenzionalni nizovi.
- Nizovi sa dve dimenzije se nazivaju dvodimenzionalni nizovi.
- Nizovi sa dve dimenzije veoma su korisni za predstavljanje podataka u tabelarnoj formi.
- Pristup elementima dvodimenzionalnog niza podrazumeva korišćenje dva indeksa i dva para uglastih zagrada.
- Svi pristupi korišćeni nad nizovima sa jednom dimenzijom primenljivi su i nad dvodimenzionalnim nizovima.

