

Sigurnosni aspekti programiranja

Definisanje uzorka (1)

- JavaScript funkcije se često upotrebljavaju za provjeru unijetih podataka od strane klijenta.
JavaScript ima razvijenu podršku za razne vrste provjera i one se obavljaju na klijentskoj strani.
- Uzorak se još naziva i regularni izraz (regular expression) i može se definisati na dva načina:
 - `var ime_uzorka = new RegExp("primjer")`
 - `var ime_uzorka = /primjer/`
- Na oba načina se formira objekat uzorka koji se naziva `ime_uzorka` i kome odgovara svaki string koji u sebi sadrži podstring primjer.

Definisanje uzorka (2)

```
- var uzorak = new RegExp("HTML")
```

```
- var uzorak = /HTML/
```

- Prvim se poziva RegExp konstruktor, a u drugom se sadržaj uzorka piše između početnog i krajnjeg znaka / (slash)

```
- var uzorak = new RegExp("s$")
```

```
- var uzorak = /s$/
```

- Simbol \$ označava kraj stringa. Sada promjenljiva uzorak odgovara bilo kom stringu koji se završava sa s.

Karaktereri koji se koriste u uzorku

Karakter	Predstavlja
alfanumerički znak	sebe
\d	Bilo koja cifra od 0 do 9
\D	Bilo koji karakter koji nije cifra
\w	Bilo koji karakter (slova a-z, A-Z, 0-9 i _)
\W	Neki specijalni karakteri (na primjer: @)
\s	Neki bijeli karakter (tab, nova linija, ...)
\S	Neki karakter koji nije bijeli
.	Bilo koji karakter (osim nove linije)
[...]	Bilo koji karakter naveden između []
[^...]	Bilo koji karakter koji nije naveden između []
[\b]	Brisanje unazad

Znakovi za ponavljanje

Karakter	Predstavlja
$\{n,m\}$	Prethodni element se ponavlja najmanje n puta i ne više od m puta
$\{n,\}$	Prethodni element se ponavlja n ili više puta
$\{n\}$	Prethodni element se ponavlja TAČNO n puta
$?$	Prethodni element se ne pojavljuje ili se pojavljuje samo jednom. Ekvivalentno izrazu $\{0,1\}$
$+$	Prethodni element se ponavlja jednom ili više puta. Ekvivalentno izrazu $\{1,\}$
$*$	Prethodni element se ne pojavljuje ili se ponavlja više puta. Ekvivalentno izrazu $\{0,\}$

Primjeri (1)

- `/ [abc] /`
 - predstavlja jedno pojavljivanje slova a ili slova b ili slova c. String "c" ispunjava uslove definisane uzorkom, ali string "s" ne ispunjava definisane uslove.
- `/ [^abc] /`
 - predstavlja karakter koji nije slovo a ili slovo b ili slovo c

Primjeri (2)

- Primjer za petocifreni poštanski broj
 - `/\d\d\d\d\d/`
 - `/\d{5}/`
- `/\d{2,4}/`
 - uzorak koji označava 2, 3 ili 4 pojavljivanja cifara
- `/\w{3}\d?/`
 - uzorak koji označava tačno 3 pojavljivanja slova i opciono jedne cifre
 - primjer: web8, ana, iva

Primjeri (3)

- `/\s+Internet\s+/
– uzorak koji označava string "Internet" sa jednim ili više prostora prije ili poslije stringa.`
- `/[a-z]+\d+/
– uzorak koji označava jedno ili više malih slova praćenih jednom ili više cifara.`

Znakovi za alternativu, grupisanje i sidrenje

Karakter	Predstavlja
	Alternative. Pojavljuje se ili samo desni ili samo lijevi dio uzorka u stringu.
(. . .)	Grupisanje simbola u jedan objekat nad kojim se mogu koristiti *, +, ?,
^	Pretragu uzorka na početku znakovnog niza
\$	Pretragu uzorka na kraju znakovnog niza

Primjeri (4)

- `/ab|cd|ef/`
 - uzorak koji označava pojavljivanje ab ili cd ili ef
- `/\d{3}|[A-Z]{4}/`
 - uzorak koji označava pojavljivanje 3 cifre ili 4 velika slova
- `/java(script)/`
 - uzorak koji označava pojavljivanje stringa "java" ili stringa "javascript"

Primjeri (5)

- $/ (ab | cd) + | ef /$ ili
- $/ ab+ | cd+ | ef /$
 - uzorak koji označava pojavljivanje stringa "ef"
ili pojavljivanje jednom ili više puta stringa "ab"
ili pojavljivanje jednom ili više puta stringa "cd"

Atributi

Atribut	Značenje
i	case-insensitive ispitivanje
g	globalno izvršavanje (pronalaženje svih pojavljivanja definisanog uzorka)
M	rad sa više linija

Ispitivanje uzoraka pomoću metoda

- `search()` - traži određeni uzorak u tekstu
- `replace()` - traži određeni uzorak u tekstu i zamjenjuje ga nekim stringom
- `match()` - formira niz koji sadrži samo traženi uzorak
- `split()` - dijeli string određenim uzorkom (uzorak je kao separator)

search() metod

- Ispituje da li u stringu postoji definisani uzorak
- Rezultat je pozicija prvog pojavljivanja uzorka ili -1, ako ne pronade uzorak
- Primjer1:

```
x = /Script/i  
y = "JavaScript".search(x);
```
- Kao rezultat izvršavanja ovog primjera promjenljiva y će dobiti vrijednost 4
- Ovaj metod ne podržava globalnu pretragu, tj. ignoriše upotrebu atributa g u okviru definicije uzorka

replace() metod

- Ispituje da li u stringu postoji uzorak i ako postoji zamijeni uzorak unutar stringa nekim drugim stringom
- Metod ima dva argumenta, prvi je uzorak, a drugi je string koji treba da zamijeni uzorak
- Primjer2:

```
"html: HTML se uci na IP".replace(/HTML/, "JAVA")
```
- Ovaj metod podržava globalnu primjenu, pa ako se u okviru uzorka navede i g atribut, ovaj metod će izvršiti zamjenu svakog uzorka koji pronade u okviru stringa

match() metod

- Vrlo sličan search() metodu, samo umjesto pozicije vraća niz elemenata sa svim pojavljivanjima definisanog uzorka, ako je definisan atribut g.
- Primjer:
 - "1 plus 2 jednako je 3".match(/\d+/g)
 - Rezultat:
["1", "2", "3"], jer je uzorak definisan kao pojavljivanje cifre, jednom ili više puta, u cijelom stringu

split() metod

- Ima jedan argument - uzorak!
- Rezultat je niz koji se dobija kada se string podijeli argumentom (uzorkom) kao separatorom
- Primjer:
 - `"123, 456, 2009 , 3141".split(/\s*,\s*/)`
 - Rezultat je `["123", "456", "2009", "3141"]`,
jer je uzorak definisan sa određenim brojem blanko znakova prije i poslije zareza, uključujući zarez

Metodi objekta RegExp

- `exec()`
- `test()`

exec()

- Ovaj metod je sličan string metodu match(). Razlika je u tome što kod ovog metoda argument je string, a primenjuje se na uzorku, dok je kod match() obrnuto.
- Rezultat izvršavanja exec() je niz koji sadrži rezultate ispitivanja, definisane na isti način kao i metod match().
- Za razliku od match() metoda exec() vraća isti rezultat ako postoji atribut g i ako ne postoji.

lastIndex

- Ako se metodi `exec()` proslijedi regularni izraz sa indikatorom `g`, u svojstvo `lastIndex` objekta klase `Regex` upisuje se **pozicija prvog znaka poslije odgovarajućeg podniza**.
- Kada se metoda `exec()` ponovo pozove za isti regularni izraz, počinje pretraživanje od pozicije zadate vrijednošću svojstva `lastIndex`.
- Ovo ponašanje omogućava da ponovljene pozive metode `exec()` izvršavamo kroz petlju, kako bi se pristupilo svim podnizovima u znakovnom nizu podudarnim sa regularnim izrazom.

Primjer

```
var pattern = /Java/g;
var text = "JavaScript je mnogo zabavniji nego
    Java!";
var result;
while((result = pattern.exec(text)) != null)
{
    alert("Pronadjen '" + result[0] + "'" +
        " na poziciji " + result.index + ";
    sljedeca pretraga pocinje od " +
        pattern.lastIndex);
}
```

This page says

Pronadjen 'Java' na poziciji 0; sljedeca pretraga pocinje od 4

OK

This page says

Pronadjen 'Java' na poziciji 35; sljedeca pretraga pocinje od 39

OK

test()

- Ova metoda se ponaša kao `exec()` tj. vraća vrijednost `true`, ako njen rezultat nije `null`.
- Počinje da pretražuje znakovni niz počevši od pozicije zadate svojstvom `lastIndex` (isto kao `exec()`!!!) i ako nađe odgovarajući podniz, zadaje tom svojstvu vrijednost pozicije prvog znaka neposredno poslije nađenog podniza.
- Svojstvo `lastIndex` postoji samo ako regularni izraz ima indikator `g`, u suprotnom metode `exec()` i `test()` zanemaruju svojstvo `lastIndex` bez indikatora `g`.