

# INTERNET TEHNOLOGIJE I VEB DIZAJN

Regularni izrazi i rad sa kolačićima

predavač: dr Miloš Antonijević

# Pojam i osnovni koncepti

- Regularni izrazi (eng. *regular expressions*) predstavljaju obrasce koji definišu pravila za pronalaženje kombinacija karaktera u stringu.
- Najčešće se koriste za filtriranje (ili proveru) unetih podataka u skladu sa očekivanom vrstom ulaznih podataka.
- U JavaScript-u se koristi u sklopu **match()**, **matchAll()**, **replace()**, **replaceAll()**, **search()** i **split()** metoda **String**-a i u okviru **exec()** i **test()** metoda objekta **RegExp**.
- Obzirom da korisnici ulazne podatke unose preko elemenata forme, podatke treba proveriti pre slanja serveru kako bi se server rasteretio nepotrebnih poziva.
- Iako je provera ulaznih podataka na strani klijenta potrebna, ona nije dovoljna da bi se podacima "verovalo" na serverskoj strani.
- Iako je input text polje primarni način kroz koji korisnik može da unese proizvoljni (ili zlonamerni) tekst, **neophodno** je vršiti proveru **svih** ulaznih podataka na serverskoj strani (čak i onih dobijenih od npr. elementa input range).

# Modifikatori i kvantifikatori

- Modifikatori utiču na način obavljanja pretrage obrasca. Najčešće se koriste dva modifikatora:
  - **i** – modifikator koji omogućava pretragu neosetljivu na mala i velika slova
  - **g** – izvršava globalnu pretragu celog stringa, odnosno, ne zadržava se na prvom preklapanju
- Kvantifikatori određuju broj karaktera ili izraza koji se koriste u pretrazi. Najčešće se koriste:
  - **\*** - nula ili više pojavljivanja
  - **+** - jedno ili više pojavljivanja
  - **?** – nula ili jedno pojavljivanje
  - **{n}** – n pojavljivanja
  - **{n,m}** – n do m pojavljivanja
  - **{n, }** – najmanje n pojavljivanja
  - **{abc\$}** – pojavljivanje string abc na kraju stringa koji se pretražuje
  - **^abc** - pojavljivanje string abc na početku stringa koji se pretražuje

# Definisanje opsega karaktera

- Opseg karaktera koji je pretražuju se može definisati navođenjem unutar uglastih zagrada []:
  - **[xy]** – traženje karaktera navedenih unutar zagrada (ili opsega između navedenih karaktera)
  - **[^xy]** – traženje karaktera koji ne pripadaju navedenim karakterima (ili opsegu između navedenih karaktera)
- Primer:
  - **[0-9]** – traženje brojeva
  - **[a-z]** – traženje malih slova
  - **[A-Z]** – traženje svih slova
  - **[A-Za-z0-9]** – traženje brojeva i slova
  - **[^0-9]** – traženje svega što nisu brojevi

# Karakteristi sa posebnim začenjem

- \w - traženje samo slova
- \W - traženje ne slovnihi karaktera
- \d - traženje cifara
- \D - traženje ne-cifarskih karaktera
- \s - traženje razmaka (space)
- \S - traženje grupe karaktera bez razmaka
- \n - traženje karaktera za novi red
- \r - traženje karaktera za povratak (return)
- \t - traženje tab karaktera

# RegEx metode

- Metoda **test()** objekta RegExp proverava da li se regularni izraz korišćen prilikom kreiranja objekta RegExp poklapa sa stringom koji je naveden kao argument metode.

```
var str = 'singidunum je zakon';  
var regex = new RegExp(/[A-Za-z0-9]{10}/, 'g');  
console.log(regex.test(str));  
console.log(regex.lastIndex);
```

- Metoda **exec()** objekta RegExp vraća niz koji sadrži pronađeni rezultat u stringu navedenom u argumentu, u odnosu na regularni izraz naveden u konstruktoru. U slučaju da ne pronađe preklapanje, metoda vraća NULL.

```
var array;  
while ((array = regex.exec(str)) !== null) {  
    console.log("Found "+array[0]+" . Next starts at "+regex.lastIndex);  
}
```

## RegEx metode (2)

- Metoda **search()** objekta String pretražuje preklapanje između regulatornog izraza navedenog u argumentu metode i stringa. Metoda vraća indeks prvog karaktera pronađenog preklapanja ili -1 ako obrazac nije pronađen.

```
const regex = /^[^\w]/;  
console.log(str.search(regex));
```

- Metoda **match()** objekta String vraća rezultat preklapanja između regulatornog izraza navedenog u argumentu i stringa. Vraćeni rezultat je niz pronađenih preklapanja.

```
var str = 'Singidunum je Zakon';  
var regex = /[A-Z]/g;  
var found = str.match(regex);  
console.log(found[0]);
```

## RegEx metode (3)

- Metoda **split()** objekta String razdvaja string u odnosu na regulatorni izraz naveden u argumentu metode. Metoda vraća niz delova stringa.

```
var reci = str.split(' '); // str.split(/\s/)
console.log(reci[1]);
```

- Metoda **replace()** objekta String vraća novi string u kome je preklapanje između regulatornog izraza navedenog u prvom argumentu metode i stringa, zamenjeno vrednošću navedenom u drugom argumentu metode. Ako je u regularnom izrazu naveden modifikator **g** (global), sva preklapanja će biti zamenjena, u suprotnom, samo prvo.

```
var regex = /[\\s]/g;
console.log(str.replace(regex, '_'));
```



# Primeri regularnih izraza

- Brojevi indeksa:

`/^(\d){1,4}\/(\d){2}/`

- Nazivi slika:

`/^\S+\.(gif|jpg|png)$/`

- Lozinke:

`/[^A-Za-z\d]{5,12}$/`

- Decimalni broj:

`/^\d*(\.\d*)?$/`

- Heksadecimalni kod boje:

`/^#?([a-f]|[A-F]|[0-9]){6}$/`

# Kolačići (cookies)

- Kolačići su znakovni nizovi koji se najčešće koriste za čuvanje stanja sesije prilikom HTTP komunikacije.
  - Kada server pošalje odgovor klijentu putem HTTP protokola konekcija se zatvara i server ne pamti nikakve podatke o klijentu (HTTP je stateless protokol).
  - Veb pregledači snimaju kolačiće na disk i vezuju ga za određeni server (sajt, domen).
  - Da bi server pamtio stanje klijenta, Veb browser šalje kolačiće u sklopu zaglavlja HTTP zahteva koji šalje serveru.
  - Kolačići mogu da imaju definisano ime, datum isteka, putanju i bezbednost
  - Sintaksa za kreiranje kolačića je  
name=value;expires=date;path=path;domain=domain;secure
- `document.cookie="username=John Doe; expires=Thu, 18 Dec 2022 12:00:00 UTC";`
- **expires** polje je u formatu koji vraća **toGMTString()** metoda objekta Date. Ako nije definisano, kolačić nestaje prilikom zatvaranja browser-a.

# Polja kolačića

- Polje **path** se odnosi na putanju sajta na kome se primenjuje kolačić. Samo stranice koje se nalaze na navedenoj putanji mogu postavljati i preuzimati vrednost kolačića. U slučaju da je polje prazno, stranica koja je postavila kolačić može da i koristiti.
- Polje **domain** se odnosi na domen koji može čitati vrednost kolačića. Ukoliko vrednost nije postavljena samo domen koji je postavio kolačić može upravljati njime. Ukoliko je setovan domen i poddomeni mogu koristiti kolačić.
- **Secure** opcija određuje da će se za prenos podataka ka serveru koristiti SSL (HTTPS).
- Primena **HttpOnly** opcije u kolačiću sprečava JavaScript da pristupi kolačiću preko `document.cookie` polja.
- Da bi se kolačić izbrisao dovoljno je postaviti vrednost polja `expires` na neki prošli datum.

**PITANJA?**