

# Web dizajn i programiranje

Prof. dr.sc. Dragutin Kermek  
*Sveučilište u Zagrebu*  
*Fakultet organizacije i informatike*  
*Pavlinska 2, Varaždin 42000*  
**dkermek@foi.hr**

11. dio

## Programiranje na strani korisnika

**Uvod u Javascript jezik i koncept programiranja.**  
**Objektni model dokumenta (Document Object Model - DOM).**  
**Programiranje događaja.**  
**Razvoj dinamičkih osobina dokumenata.**  
**Dozvoljeni izrazi (Regular Expressions - RegExp).**  
**Web 2.0., Ajax**  
**jQuery**



## RegExp

To su predlošci koji se koriste za uspoređivanje kombinacije znakova u stringovima.

Postoje u mnogi skriptnim jezicima kao **Perl** i **PHP**, jezicima ljuske kao **awk** i **sed**, programskim jezicima kao **Java** i **sl**.

Kreiraju se na dva načina:

- korištenje inicijalizatora objekta

```
RE = /predložak/[zastavice]
```

```
re = /ab+c/g
```

- pozivom konstruktora objekta **RegExp**

```
re = new RegExp("predložak"[, zastavice])
```

```
re = new RegExp("ab+c", "gi")
```



## RegExp – nastavak 1.

### Atributi:

**global**

- da li se testira dozvoljeni izraz prema svim mogućim pokrivanjima u stringu ili samo prema prvom.

**ignoreCase**

- da li se ignorira veličina slova kod pokušaja pokrivanja u stringu.

**input**

- string prema kojem se pokriva dozvoljeni izraz.

**lastIndex**

- index kod kojeg počinje sljedeće pokrivanje.

**lastMatch**

- posljednji pokriveni karakter.

**lastParen**

- posljednji podstring, ako postoji.

**leftContext**

- podstring koji je posljednji pokriven.

**multiline**

- da li se pretražuje u stringu koji ima više linija.

**prototype**

- dopušta dodavanje svojstava svim objektima.

**rightContext**

- podstring koji slijedi posljednji pokriveni.

**source**

- tekst predloška.



## RegExp – nastavak 2.

### Funkcije:

- exec** - izvršava traženje za pokrivanje u njegovom string parametru
- test** - ispituje pokrivanje u njegovom string parametru (true, false).



## RegExp – nastavak 3.

### Korištenje posebnih znakova:

\

- za znakove koji se obično doslovno tretiraju određuje da sljedeći znak je poseban i da se doslovno ne interpretira.
- npr, **/b/** pokriva znak 'b'. Stavljanjem znaka backslash ispred b, **/b/**, karakter postaje poseban i znači da pokriva granicu riječi.
- ili-
- za karaktere koji se obično posebno tretiraju određuju da sljedeći znak nije poseban i treba ga doslovno interpretirati
- npr, **\*** je poseban znak koji znači 0 ili više pojava prethodnog znaka
- npr, **/a\*/** pokriva 0 ili više a. Za pokrivanje **\*** doslovno, potrebno je da prethodi backslash; **/a\\\*/** pokriva 'a\*'.



## RegExp – nastavak 4.

- ^** - pokriva početak.
  - npr, **/^A/** ne pokriva 'A' u "u A," ali pokriva "Au A."
- \$** - pokriva kraj.
  - npr, **/t\$/** ne pokriva 't' u "tor", ali pokriva u "bat"
- \*** - pokriva prethodni znak 0 ili više puta.
  - npr, **/bo\*/** pokriva 'boooo' u "ide booooo" i 'b' u "bilo jednom", ali ne i "na zapadu".
- +** - pokriva prethodni znak 1 ili više puta. Odgovara **{1,}**.
  - npr, **/a+/** pokriva 'a' u "kaktus" i sve a u "kaaaaaaaj."
- ?** - pokriva prethodni znak 0 ili 1 puta.
  - npr, **/e?le?/** pokriva 'ele' in "element" i 'le' "leglo"
- .** - (decimalna točka) pokriva bilo koji karakter osim nove linije.
  - npr, **/n/** pokriva 'an' i 'on' u "taj ban svira svaki ton", ali ne 'noj'.



## RegExp – nastavak 5.

- (x)** - pokriva 'x' i pamti pokrivanje.
  - npr, **/boo/** pokriva i pamti 'boo' u "boo baa." Pokrivani podstring može se dobiti u polju elemenata **[1]**, ..., **[n]**
- x|y** - pokriva 'x' ili 'y'.
  - npr, **/plavi|crveni/** pokriva 'plavi' u "plavi auto" i 'crveni' u "crveni mak"
- {n}** - gdje je **n** pozitivni broj. Pokriva točno **n** pojava prethodnog karaktera.
  - npr, **/a{2}/** ne pokriva 'a' u "bar" ali pokriva sva a u "kaanu," i prva dva a u "baaanana."
- {n,}** - gdje je **n** pozitivni broj. Pokriva najmanje **n** pojava prethodnog znaka.
  - npr, **/a{2,}/** ne pokriva 'a' u "bar", ali pokriva sva a u "kaanu".
- {n,m}** - gdje su **n** i **m** pozitivni brojevi. Pokriva najmanje **n** i najviše **m** pojava prethodnog znaka.
  - npr, **/a{1,3}/** ne pokriva "kuk", ali pokriva 'a' u "bar," prva dva a u "kaanu" i prva tri a u "kaaaaaaanu". Važno je da kada pokriva "kaaaaaaanu", pokriva "aaa", premda string ima više a od njega.



## RegExp – nastavak 6.

- [xyz]** – skup znakova. Pokriva bilo koji znak u zagrada. Može se odrediti niz znakova korištenjem crte.
- npr, **[abcd]** isto je kao **[a-d]**. Oni pokrivaju 'b' u "bik" i 'c' u "cik".
- [^xyz]** – negacija ili komplement skupa znakova. To je, on pokriva bilo koji znak koji nije u zagrada. Može se odrediti niz znakova korištenjem crte.
- npr, **[^abc]** isto je kao **[^a-c]**. Oni odgovaraju 'r' u "aro" i 'h' u "hac".
- [\b]** - pokriva backspace. (Ne zamjeniti s \b.)
- \b** - pokriva granicu riječi, kao što je praznina. (Ne zamjeniti s [\b].)
- npr, **\bhlw/** pokriva 'no' u "noga", **\bka\b/** pokriva 'ka' u "ruka"
- \B** - pokriva ono što nije granica riječi.
- npr, **\Bhlw/** 'on' u "ponder", i **\Bhlw/** pokriva 'di' u "web dizajn"



## RegExp – nastavak 7.

- \cX** - gdje je X kontrolni karakter. Odgovara kontrolnom karakteru u stringu.
- npr, **\cM/** pokriva control-M u stringu.
- \d** - pokriva broječni karakter. Ekvivalentno **[0-9]**.
- npr, **\d/** ili **/[0-9]/** pokriva '2' u "B2B"
- \D** - pokriva ne broječni karakter. Ekvivalentno **[^0-9]**.
- npr, **\D/** ili **/[^0-9]/** pokriva 'B' u "B2B"
- \f** - pokriva form-feed.
- \n** - pokriva linefeed.
- \r** - pokriva carriage return.
- \s** - pokriva jednu bijelu prazninu, uključujući space, tab, form feed, line feed. Ekvivalentno **[\f\n\r\t]**.
- npr, **\slw\*/** pokriva ' bar' u "night bar"
- \S** - pokriva pojedinačni karakter koji nije bijela praznina. Ekvivalentno **[^\f\n\r\t]**.
- npr, **\S\lw\*** pokriva 'night' u "night bar."
- \t** - pokriva tab
- \v** - pokriva vertikalni tab.



## RegExp – nastavak 8.

- \w** - pokriva bilo koji alfanumerički karakter uključujući underscore.  
Ekvivalentno **[A-Za-z0-9\_]**.  
•npr, **/\w/** pokriva 'a' u "alka", '5' u "\$5.28", i '3' u "3D."
- \W** - suprotno od **\w**. Ekvivalentno **[^A-Za-z0-9\_]**.  
•npr, **/\W/** or **/[^A-Za-z0-9\_]/** pokriva '%' u "50%."
- \n** - gdje je *n* pozitivan broj.
- \o~~ctal~~**
- \x~~hex~~** - gdje je **\o~~ctal~~**/oktalna vrijednost ili **\x~~hex~~** je heksadecimalna vrijednost value. Dopušta uključivanje ASCII kodova u dopuštene izraze.



## RegExp – primjer 1

```
<body>
  <script type="text/javascript">
    function provjeri() {
      var re = new RegExp(document.obrazac.predlozak.value,
        document.obrazac.zastavice.value);
      var ok = re.test(document.obrazac.tekst.value);
      if (!ok)
      {
        alert("Tekst ne poklapa predložak!");
        return false;
      }
      else {
        alert("OK!");
        return true;
      }
    }
  </script>
```



## RegExp - primjer 1 / 1.

```
<form name="obrazac">
  Ulazni tekst <input name="tekst" value="Programiranje za Web"><br>
  Predložak <input name="predlozak"><br>
  Zastavice <input name="zastavice"><br>
  <input type="button" value=" Provjeri " onclick="provjeri();" >
</form>

</body>
```

RE/Primjer01 – RegExp



## RegExp - primjer 1 / 2.

Ulazni tekst	Predložak	Zastavice	Test
Programiranje za Web	p		F
Programiranje za Web	p	i	T
Programiranje za Web	P		T
Programiranje za Web	Prom		F
Programiranje za Web	Prom?		T
Programiranje za Web	^P		T
Programiranje za Web	Web\$		T
Programiranje za Web	\bza\b		T
Programiranje za Web	^\\w*ra\\.ra\\w*\\b		T
Programiranje za Web	(Web)   (Internet)\$		T
Programiranje za Web	[abc]		T
Programiranje za Web	[abc] [abc]		F
06.12.2002.	\\d\\d\\.\\d\\d\\.\\d{4}\\.		T
6.1.2003.	\\d\\d\\.\\d\\d\\.\\d{4}\\.		F
6.1.2003.	\\d?\\d\\.\\d?\\d\\.\\d{4}\\.		T



## RegExp - primjer 1 / 3.

Ulazni tekst	Predložak	Test
dkermek@foi.hr	^\w{2,}@(\w{2,}\.){1,2}\w{2,}\$	T
dkermek@barok.foi.hr	^\w{2,}@(\w{2,}\.){1,2}\w{2,}\$	T
dkermek@prof.barok.foi.hr	^\w{2,}@(\w{2,}\.){1,2}\w{2,}\$	F
dkermek@foi.hr	^\w{2,}@(\w{2,}\.){1,2}\w{2,}\$	F
dkermek@foi.hr.	^\w{2,}@(\w{2,}\.){1,2}\w{2,}\$	F
dragutin.kermek@foi.hr	^\w{2,}@(\w{2,}\.){1,2}\w{2,}\$	F
Programiranje za Web	^\w{2,}@(\w{2,}\.){1,2}\w{2,}\$	F



## RegExp - primjer 2

```
<body>

<script type="text/javascript">
    function provjeri() {
        var re = new RegExp(document.obrazac.predlozak.value,
                             document.obrazac.zastavice.value);
        var polje = re.exec(document.obrazac.tekst.value);
        var a = "";

        if (polje === null)
        {
            alert("Nema pokrivanja!");
            return false;
        }

        for (i = 0; i < polje.length; i++)
            a += polje[i] + " - ";

        alert(a);
        return true;
    }

</script>
```





## RegExp - primjer 2 / 1.

```
<form name="obrazac">
  Ulazni tekst <input name="tekst" value="Programiranje za Web"><br>
  Predložak <input name="predlozak"><br>
  Zastavice <input name="zastavice"><br>
  <input type="button" value=" Provjeri " onclick="provjeri();" >
</form>

</body>
```

[RE/Primjer01\\_1 – RegExp](#)



## RegExp - primjer 2 / 2.

Ulazni tekst	Predložak	Z.	Izlazni tekst
Programiranje za Web	Pro		Pro -
Programiranje za Web	Pro\w+		Programiranje -
Programiranje za Web	\w+		Programiranje -
Programiranje za Web	\w+\s\w+\s\w+		Programiranje za Web -
Programiranje za Web	(\w+)\s(\w+)\s(\w+)		Programiranje za Web - Programiranje - za - Web -



## Korištena i dodatna literatura

- ⇒ <http://www.angelfire.com/tx4/cus/notes/javascript.html>
- ⇒ [http://dir.yahoo.com/Computers\\_and\\_Internet/Programming\\_and\\_Development/Languages/JavaScript/](http://dir.yahoo.com/Computers_and_Internet/Programming_and_Development/Languages/JavaScript/)
- ⇒ <http://webreference.com/programming/javascript/>
- ⇒ <http://www.angelfire.com/tx4/cus/notes/javascript.html>
- ⇒ <http://javascript.internet.com/>
- ⇒ <http://www.jsworkshop.com/js15examples.html>
- ⇒ <http://www.devguru.com/Technologies/ecmascript/quickref/regexp.html>
- ⇒ <http://www.sislands.com/javascript/week5/regexp.htm>
- ⇒ <http://www.webmasterbase.com/article/286>
- ⇒ <http://www.webreference.com/js/column5/>

