



Interneto technologijos

JavaScript
Reguliarios išraiškos



Reguliarios išraiškos

- Skirtos šablono atitikimo paieškai tekste
 - Pvz., ar yra žodis "kuku" tekste: "abc**kuk**ude"
 - šablonas būtų: `kuku`
 - ar yra "a" raidžių eilutė tekste "bc**aaa**de"
 - Šablonas būtų: `a+`
- Sukurti reguliarią išraišką (šabloną) galima dviem būdais:
 - `var re = /ab+c/;`
 - `var re = new RegExp("ab+c");`
 - Šis šablonas nusako tekstą pavidalu: `abb...bc`



Specialūs simboliai

- * žymi prieš tai einantį simbolį 0, 1, 2, ... , daug kartų.
Pvz.: /Go*/ atitiks fragmentą 'Goo' eilutėje "Google", fragmentą 'G' eilutėje "Gandras", bet nieko neatitiks eilutėje "Boo".
- + žymi prieš tai einantį simbolį 1, 2, ... daug kartų.
Pvz.: /b1+/ atitinka fragmentą 'b11' eilutėje "ab112c", bet nieko neatitinka eilutėje "abc".
- ? žymi prieš tai einantį simbolį 0 arba 1 kartą.
Pvz.: /aky?las/ atitinka fragmentą 'aklas' eilutėje "aklas" bei fragmentą 'akylas' eilutėje "akylas"
- {n} (n – skaičius) žymi prieš tai einantį simbolį lygiai n kartų.
Pvz.: /b{2}c{3}/ atitinka fragmentą 'bbccc' eilutėje "abbcccd".
- {n,} žymi prieš tai einantį simbolį n arba daugiau kartų.
- {n,m} žymi prieš tai einantį simbolį nuo n iki m kartų.



Specialūs simboliai

^ žymi eilutės pradžią.

Pvz.: `/^a/` atitiks pirmą raidę 'a' eilutėje "arbūzas", bet ne antrą 'a'.

\$ žymi eilutės pabaigą.

Pvz.: `/s$/` atitiks paskutinę raidę 's' eilutėje "sausainis", bet ne pirmą ir ne antrą.

. (taškas) žymi vieną bet kokį simbolį, išskyrus eilutės pabaigos simbolį.

Pvz.: `/.a.a.a./` atitinka fragmentą 'bananas' eilutėje "žalias bananas".

x|y žymi x arba y.

Pvz.: `/juodas|baltas|geltonas/` atitinka fragmentą 'baltas' eilutėje "baltas sniegas".

(x) žymi x ir įsimena x atminties įrenginyje. Įsimintas reikšmes galima pasiekti indeksais [1], [2], ir t.t.

Pvz.: `/prof (.+)/` atitinka fragmentą 'prof Jonaitis' eilutėje "prof Jonaitis" ir įsimena 'Jonaitis'.



Specialūs simboliai

[xyz] žymi vieną simbolį iš simbolių x, y, ir z aibės.

Pvz.: `/[abls]{5}/` atitinka fragmentą 'labas' eilutėje "labas rytas".

[x-y] žymi vieną simbolį iš simbolių intervalo nuo x iki y. `[a-d]` reiškia tą patį kaip ir `[abcd]`.

[^xyz] žymi vieną simbolį, nelygų simboliams x, y ir z. T.y., aibės papildinys.



Specialūs simboliai

\d = [0-9] t.y. skaičius.

\D = [^0-9] t.y. ne skaičius.

\s = [\f\n\r\t\v\u00A0\u2028\u2029] t.y. "white space" simbolis. Visų rūšių simboliai, kurie atrodo kaip tarpai.

\S = [^ \f\n\r\t\v\u00A0\u2028\u2029] t.y. ne tarpo simbolis.

\w = [A-Za-z0-9_] t.y. simboliai, iš kurių sudaromi žodžiai.

\W = [^A-Za-z0-9_] t.y. ne žodžio simboliai.



Reguliarių išraiškų vykdymas

- `\` – escape simbolis. Jei reikia ieškoti "+", rašome `"\+"`
- `RegExp` turi du metodus:
 - `exec()` – metodas, kuris vykdo paiešką tekste ir grąžina informacijos masyvą (aprašyta kitoje skaidrėje)
 - `test()` – atlieka paiešką ir grąžina `true` arba `false`.
Pvz.:

```
if ( ! /^[a-z]{4}\d{4}$/ .test( "abab1234" ) ) {  
    alert( "Blogas vartotojo vardas" );  
}
```

RegExp.exec() pavyzdys

```
myRe=/d(b+)d/g;
```

```
myArray = myRe.exec("cdbbdbsbz");
```

| | | | |
|---------|-----------|--|----------------------|
| myArray | | The matched string and all remembered substrings. | ["dbbd" , "bb"] |
| | index | The 0-based index of the match in the input string. | 1 |
| | input | The original string. | "cd bb dbsbz" |
| | [0] | The last matched characters. | "dbbd" |
| | [1]...[n] | The parenthesized substring matches, if any. The number of possible parenthesized substrings is unlimited. | [1] = bb |
| myRe | lastIndex | The index at which to start the next match. (This property is set only if the regular expression uses the <code>g</code> option, described in Executing a Global Search, Ignoring Case, and Considering Multiline Input.) | 5 |
| | source | The text of the pattern.Updated at the time that the regular expression is created, not executed. | "d(b+)d" |



String objekto metodai

| | | |
|----------------|---|--|
| search | A <code>String</code> method that tests for a match in a string. It returns the index of the match, or -1 if the search fails. | |
| replace | A <code>String</code> method that executes a search for a match in a string, and replaces the matched substring with a replacement substring. | |
| | <code>\$\$</code> | Inserts a '\$'. |
| | <code>\$&</code> | Inserts the matched substring.. |
| | <code>\$n</code> or <code>\$nn</code> | Where <i>n</i> or <i>nn</i> are decimal digits, inserts the <i>n</i> th parenthesized submatch string. |



String.replace pavyzdys

- Tarkime norime tekste visus žodžius „knyga“ (visais linksniais) pakeisti į „brošiūra“ (atitinkamais linksniais):
 - knyga – brošiūra, knygos – brošiūros, ...

```
var regExp =  
    /knyg(a|os|ai|ą|oje|ų|oms|as|omis|ose)/g;  
var text = "gera knyga, dvi knygos, mano knygai...";  
var result = text.replace(regExp, "brošiūr$1");
```



String objekto metodai

split

A String method that uses a regular expression or a fixed string to break a string into an array of substrings.

```
names="Harry Trump ;Fred Barney; Helen Rigby";  
re = /\s*;\s*/; // skirtukas  
nameList = names.split (re);
```



Reguliarių išraiškų raktai

```
re = /šablonas/raktai;  
re = new RegExp("šablonas", ["raktai"]);
```

- g – globali paieška (ieškoma ne tik iki pirmo sutapimo, bet ir toliau)
- i – case insensitive – mažosios ir didžiosios raidės sutampa
- m – multiline - \n simbolis reiškia naujos eilutės pradžią, kurią atitiks ^ simbolis
- Pvz.:

```
re = /\w+\s/g; // kelios raidės ir vienas tarpas  
str = "fee fi fo fum";  
myArray = str.match(re);  
// myArray bus ["fee ", "fi ", "fo "]
```