

Regular Expressions in JavaScript

Search and replace in strings using patterns

Javascript Fundamentals

Telerik Software Academy
<https://telerikacademy.com>



Regex

Follow us



Table of Contents

- Regex overview
- Regex in JavaScript
 - Creating a Regex
 - Regex methods and properties
 - Use
 - Flags
- Regex special characters



Regular Expressions Overview

What is a RegEx and what is it useful for?

Follow us



Telerik Academy Regular Expressions Overview

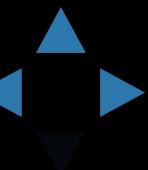
- A regular expression is a set of characters that defined a pattern used to match character combinations in strings
 - Very powerful for find/replace type of operations
- Some examples when regular expressions are powerful:
 - Find and extract data from a document
 - Extract image source from HTML, extract exceptions/errors from logs
 - Validate data provided as text:
 - Passwords, emails, mobile numbers, url

Follow us



Regex Syntax

- Regular expressions are an extremely powerful tool implemented in most languages
- Yet, regular expressions have their own syntax and use of special characters
 - Difficult to remember unless used frequently
 - [MDN Regex reference](#)
- Regular expressions can be tested at:
 - <http://www.regexr.com/>
 - <https://regex101.com/>



Regex in JavaScript

Using Regular Expressions in Javascript

Follow us



Creating Regex in JavaScript

- Regular expressions are built-in in JavaScript
 - Can be created using a regex literal or a function constructor
 - Regex literals are useful for static expressions
 - The function constructor comes into play when the expression depends on other parameters

Follow us

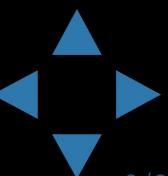


- The following will match 'Telerik Academy', 'Academy', 'kitty', 'yummy'

```
// literal syntax  
const literalRegex = /y$/g;
```

- The following will match 'Telerik Academy', 'Todor'

```
// function constructor syntax  
const constructorRegex = new RegExp('^T', 'g');
```



Telerik Academy Regex methods and properties

- Full list of properties and methods on [MDN](#)
- `RegExp.test` – searches for a match in a given string. Returns `true` or `false`
- `RegExp.exec` - searches for the next match in a given string
 - Returns an array of all captured groups for the found match or `null`.
- `String.match` – searches for a match in a string
 - Returns an array of information or `null` on a mismatch

Follow us



Telerik Academy Regex methods and properties

- `String.replace` – replaces the matched substring with a replacement substring
 - Returns the new string
- `String.split` – breaks a string into an array of substrings, using a regex or a string search for matches
 - Returns an array
- `String.search` – tests for a match in a string
 - It returns the index of the match, or -1 if the search fails

Follow us



Regular Expression Flags

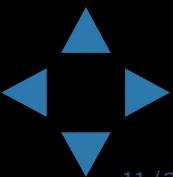
- Regular expression have optional flags that allow for global and case insensitive searching
 - These flags can be used separately or together in any order

Regular expressions have four optional flags that allow for global and case insensitive searching. These flags can be used separately or together in any order, and are included as part of the regular expression.

Table 4.4 Regular expression flags.

Flag	Description
g	Global search.
i	Case-insensitive search.
m	Multi-line search.
y	Perform a "sticky" search that matches starting at the current position in the target string.

Follow us



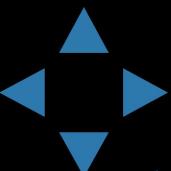
- Match all mentions of Telerik Academy initiatives
 - `RegExp.test`, `String.match`, `RegExp.exec`

```
let academyRegex = /telerik\s(software\s|algo\s|kids\s)?academy/gi;

let text = 'George is studying JavaScript at Telerik Academy ',
          'while Jimmy goes to Telerik Kids Academy.';
// true
let isMatch = text.test(academyRegex);

// will contain array of matches substrings or null
let matches = text.match(academyRegex);

// get matches and matched groups one by one
let currentMatch;
while(currentMatch = academyRegex.exec(text)) {
    console.log(currentMatch);
}
```



- Replace all whitespaces, tabs and newlines with a single space
 - `String.replace`

```
let text = 'text    with    lots of      spaces\n' +
           '        and lots of tabulations    ';
console.log(text.replace(/\s\s+/g, ' '));
```

- Split a JavaScript expression to get it's operands
 - `String.split`

```
let expression = '4+5*count-initialCount+1';
let operands = expression.split(/\+|\*|/-/);
console.log(operands);
```



- Search for the first occurrence of a pattern match
 - `String.search`

```
let text = 'JavaScript is awesome!';
let index = text.search(/is/);
console.log(index);
```



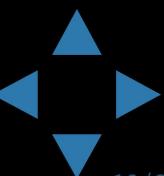
Regex Special Characters

Groups, ranges, word boundaries...

Follow us



- Regular expressions have a set of special characters, that have a different behavior
 - Characters for matching multiple characters
 - Characters for matching whitespace
 - Characters for matching digits
 - Characters for matching letters
 - Etc...
- Complete list of special characters can be found [here](#)



Telerik Academy Special Characters in Regex

- * – The preceding character/group is matched 0 or more times
- + – Almost the same behaviour as * - the preceding character/group is matched 1 or more times
- ? – The preceding character/group is matched 0 or 1 times
- .(dot) – matches any single character except the newline character

Telerik Academy Special Characters in Regex

- | - Matches one pattern or the other
- [xyz] - Character set - Matches any of the characters
- [x-z] - Character set - Matches any one between the characters range
- [^xyz] - Inverted characters set - Matches all other characters

Follow us



Telerik Academy Special Characters in Regex

- $\{N\}$ – matches exactly N occurrences of the preceding character/group
- $\{N, M\}$ – matches at least N and at most M occurrences of the preceding character/group
- $^$ - matches the start of the string
- $$$ matches the end of the string

Follow us

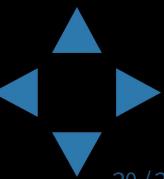




Telerik Academy Special Characters in Regex

- \s – matches a single white space character, including space, tab, form feed, line feed
- \S – matches a single character other than white space
- \d – matches a digit character
 - Equivalent to [0-9]

Follow us



Telerik Academy Special Characters in Regex

- \D – matches any non-digit character
 - Equivalent to [^0-9]
- \w – matches any alphanumeric character including the underscore
- \W – matches any non-alphanumeric or underscore character

Follow us



Using Regular Expressions in Practice

Solving practical problems

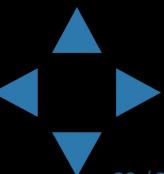
Follow us



Username validation

- The username:
 - can contain only lower or capital latin letters, digits and underscore _
 - it's length must be between 4 and 15, inclusive
 - must start with a capital letter
- Test your regular expression with the following:

```
['Chris11', '', 'Joe', 'Peter_356', '123george',
 '__proto__', 'ImATooLongUsername15', 'J0hn_', 'scripter']
```



Extract all image sources

- Extract all image source from given HTML markup
 - No matter relative or absolute
- Test your regex on the HTML of telerikacademy.com

Follow us

