

Regular Expressions in JS

Fast ways to search and replace string

JavaScript Fundamentals

Telerik Software Academy

<http://academy.telerik.com>

Table of Contents

- ◆ Regular Expression Overview
- ◆ Special Characters for beginning and end
- ◆ Regex in JavaScript
 - ◆ Regex literal and RegExp function constructor
 - ◆ Special Characters in Regex

Regular Expressions

Overview

Regular Expressions Overview

- ◆ A regular expression is a set of patterns used to match character combinations in strings
 - ◆ Find and extract data from a document
 - ◆ Validate content supplied in a form before it is submitted like:
 - ◆ Telephone numbers
 - ◆ SSN/EGN
 - ◆ Email addresses
 - ◆ Anything that follows a pattern

- ◆ Regular expressions are an extremely powerful tool implemented in most languages
- ◆ Yet, regular expressions have their own syntax and usage of special characters
 - ◆ Difficult to remember if you use them infrequently
- ◆ Regular expressions can be tested at:
 - ◆ <http://www.regexr.com/>

Regex Special Characters for Beginning and End

^ and \$

Regex Special Characters for Beginning and End

◆ Special Characters:

- ◆ ^ - matches the beginning of input

◆ `^T`

◆ **Matches:** 'Telerik Academy', 'Telerik', 'Theta'

◆ **Does not match:** 'Academy', 'Good Telerik'

- ◆ \$ - matches the end of input

◆ `y$`

◆ **Matches:** 'Telerik Academy', 'Academy', 'yummy'

◆ **Does not match:** 'Telerik', 'Good Telerik'

Regex Special Characters for Beginning and End

Live Demo

Creating and Using Regex in JavaScript

How to use it on the JS

Creating Regex in JavaScript

- ◆ Regular expressions are built-in in JavaScript
 - ◆ Can be created using a regex literal or using the RegExp function constructor:

- ◆ Regex literal:

```
var regex = \^T\;
```

Useful for static
expressions

- ◆ RegExp function constructor:

```
var regex = new RegExp('^\T\');
```

Useful for when the
expressions depends
on something

Using Regex in JavaScript

- ◆ In JavaScript, the regex is used over strings
 - ◆ There are five methods for regexes:
 - ◆ **RegExp#test** – tests for a match in a string
 - ◆ Returns true or false
 - ◆ **String#match** – searches for a match in a string
 - ◆ Returns an array of information or null on a mismatch
 - ◆ **String#replace** – replaces the matched substring with a replacement substring
 - ◆ Returns the new string

Using Regex in JavaScript

- ◆ In JavaScript, the regex is used over strings
 - ◆ There are five methods for regexes (cont):
 - ◆ **String#split** – **breaks** a string into an array of substrings, using an expression or a fixed string
 - ◆ 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

Creating and Using Regex in JavaScript

Live Demo

Special Characters in Regular Expressions

For matching all kinds of stuff

Special Characters in Regex

- ◆ The 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...
- ◆ Full list of special characters can be found at:
 - ◆ https://developer.mozilla.org/en/docs/Web/JavaScript/Guide/Regular_Expressions#Using_special_characters

Special Characters in Regex: *

◆ Special Characters in Regex:

- ◆ * – The preceding character is matched 0 or more times
 - ◆ **a***
 - ◆ Matches: 'alaaaaaaaa bala'
 - ◆ Does not match: 'Good Telerik', 'Doncho Minkov'
 - ◆ Remark: Da*oncho
 - ◆ Matches: 'Doncho Minkov'
 - ◆ 'a' is matched 0 times

Special Characters: *

Live Demo

Special Characters in Regex: +

- ◆ Special Characters in Regex:
 - ◆ + – The preceding character is matched 1 or more times
 - ◆ **a+**
 - ◆ Matches: 'alaaaaaaaaa bala'
 - ◆ Does not match: 'Doncho Minkov', 'Good Telerik'
 - ◆ Remark: **Da+oncho**
 - ◆ Does not match: 'Doncho Minkov'

Special Characters: +



Live Demo

Special Characters in Regex: ?

- ◆ Special Characters in Regex:
 - ◆ ? – The preceding character is matched 0 or 1 times
 - ◆ **T?**
 - ◆ **Matches:** 'Telerik is Telerik'
 - ◆ **Does not match:** 'Academy'

Special Characters: ?

Live Demo

- ◆ Special Characters in Regex:
 - ◆ **.(dot)** – matches any single character except the newline character
 - ◆ 
 - ◆ **Matches:** 'Telerik is Telerik'
 - ◆ **Remark:** 
 - ◆ **Matches any whole string**

Special Characters: . (dot)

Live Demo

Special Characters in Regex: |

- ◆ Special Characters in Regex:
 - ◆ | – Matches one pattern or the other
 - ◆ **T|A**
 - ◆ **Matches: 'Telerik Academy'**

Special Characters: |

Live Demo

Special Characters in Regex: []

- ◆ Special Characters in Regex:
 - ◆ [xyz] – Character set
 - ◆ Matches any one of the enclosed characters
 - ◆ **[TAy]**
 - ◆ Matches: 'Telerik Academy'

Special Characters: []

Live Demo

Special Characters in Regex: []

◆ Special Characters in Regex:

- ◆ [x-z] – Character set
 - ◆ Matches any one between the characters range
 - ◆ [A-Z]
 - ◆ Matches: 'Telerik Academy'
 - ◆ [A-z]
 - ◆ Matches: 'Telerik Academy'
 - ◆ [a-q]
 - ◆ Matches: 'Telerik Academy'
 - ◆ [0-9]
 - ◆ Matches: 'John in 19-years-old'

Special Characters: [] Range

Live Demo

Special Characters in Regex: [^xyx]

- ◆ Special Characters in Regex:

- ◆ [^xyx] – A negated or complemented character set
 - Matches anything that is not enclosed in the brackets
 - `[^ea]+`
 - Matches: 'Telerik Academy'
 - Does not match: 'aaaaaeeeeaaa', 'aaaa', 'eeee'

Special Characters: [^xyz]

Live Demo

Special Characters in Regex: {}

- ◆ Special Characters in Regex:

- ◆ {N} – matches exactly N occurrences
 - ◆ Where N is a positive number
 - ◆ `[A-z]{5}`
 - ◆ Matches: 'Telerik Academy', 'Doncho Minkov'
 - ◆ Does not match: 'JS is the best'
- ◆ {N, M} – matches at least N and at most M occurrences of the preceding character
 - ◆ Where N and M are positive integers
 - ◆ `[A-z]{4, 5}`
 - ◆ Matches: 'Telerik Academy', 'JS is best'
 - ◆ Does not match: 'Ivo is the MAN'

Special Characters: {}

Live Demo

Other Special Characters in Regex

◆ 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]
- ◆ /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

Special Characters: []

Live Demo

Regular Expression Flags

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

Table 4.4 Regular expression flags.

Flag	Description
g	Global search.
i	Case-insensitive search.
m	Multi-line search.

Regular Expression Flags

Live Demo

Regular Expressions in JS

Questions?

Free Trainings @ Telerik Academy

- ◆ "Web Design with HTML 5, CSS 3 and JavaScript" course @ Telerik Academy



- ◆ html5course.telerik.com

- ◆ Telerik Software Academy

- ◆ academy.telerik.com



- ◆ Telerik Academy @ Facebook

- ◆ facebook.com/TelerikAcademy



- ◆ Telerik Software Academy Forums

- ◆ forums.academy.telerik.com

