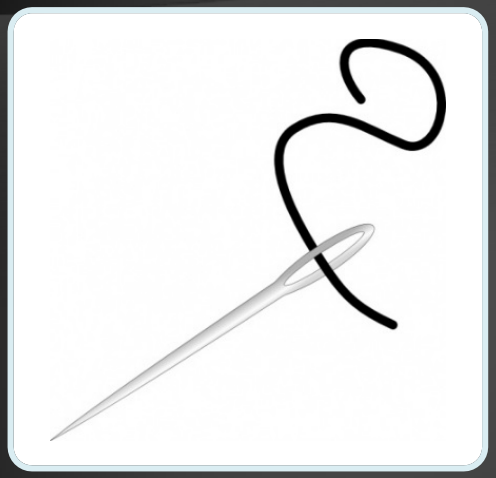
Telerik Academy



Strings

Working with strings in JavaScript

Doncho Minkov

Senior Technical Trainer

http://minkov.it

Telerik Software Academy

http://academy.telerik.com

Telerik Academy

Table of Contents

- Strings in JavaScript
- String Wrapper
- String Methods
  - `trim`, `concat`, `charAt`, `substr`, `indexOf`
- String Concatenation
- Escaping
- Useful Extensions
  - Trimming
  - Padding

Telerik Academy

Strings in JavaScript

Telerik Academy

String in JavaScript

- A string is a sequence of characters
  - Text enclosed in single (' ') or double quotes (" ")

```
var str1 = "Some text saved in a string variable";
var str2 = 'text enclosed in single quotes';
```

- String is a primitive type
  - It is copied / passed by value
- String is also immutable
  - Every time a string is changed, a new string is created

Telerik Academy

String Wrapper

String Wrapper

- As string is a primitive type, it has a object wrapper type
- Primitive types keep only their value
  - When a property is called, the JS engine converts the primitive into its corresponding object type and calls the property

```
var str = "sample";
str.length;
```

same as

```
var str = "sample";
var tempStr = new String(str);
tempStr.length;
```

- Since primitive type wrappers are of type object, properties can be attached to them

Telerik Academy

String Wrapper

Live Demo

Telerik Academy

From Object to Primitive Type

- JavaScript have a simple parsing
  - From string to number
- Conversion from primitive to object type is introduced
  - `new String('…')` creates a string object
  - `String(strObject)` creates a primitive string

```
var base = "string";
var strObj = new String(base);
var str = String(strObj);
```

Telerik Academy

String Methods

Working with Strings

String Methods

- `string.length`
  - Returns the number of characters in the string
- Indexer (`string[index]`)
  - Gets a single-character string at location `index`
  - If index is outside the range of string characters, the indexer returns `undefined`
    - `string[-1]` or `string[string.length]`
- `charAt(index)`
  - Gets a single-character string at location `index`
  - Much like the indexer

String Methods (2)

- string.concat(string2)
  - Returns a new string – the concatenation of the two strings
- string.contains(substring)
  - Determines if a substring in present in the string
  - Returns True/False
- str.startsWith(substr) and str.endsWith(substr)
  - Determines if str starts/ends with substring
  - Returns True/False

String Methods (3)

- string.indexOf(substring [,position])
  - Returns the left-most occurrence of substring in string, that is after position
    - Position is optional and has default value of `0`
  - If string doesn't contain substring, returns `-1`
- string.lastIndexOf(substring [,position])
  - Returns the right-most occurrence of substring in string, that is before position
    - Position is optional, default value is string.length
  - If string doesn't contain substring, returns `-1`

String Methods (4)

- string.split(separator)
  - Splits the string by separator and returns an array of strings, containing the separated parts
  - Separator can be a regular expression
- string.trim()
  - Removes whitespace from the beginning and end of the string
- str.trimLeft(), str.trimRight()
  - Remove whitespace from the left/right side of the string

String Methods (5)

- string.substr(start, length)
  - Returns a substring, starting from start and counting length characters
  - length is optional
- string.substring(start, end)
  - Returns a substring, starting from start and ending at end
- string.valueOf()
  - Returns the primitive value of the object string

Telerik Academy

String Methods

Live Demo

Telerik Academy

String Concatenation

How to concatenate strings

Telerik Academy

String Concatenation

- String is an immutable type
  - A value cannot be changed
  - Instead a new string is created
- There are a few ways to concatenate strings

```
var strConcat1 = str1 + str2;
var strConcat2 = str.concat(str2);
```

- Concatenating strings is slow operation
  - Each concatenation allocates new memory

Telerik Academy

String Concatenation (2)

- String concatenation is one of the most used operations with strings
  - Yet it is hard to optimize it
  - Each browser makes optimizations of its own
  - Old browsers concatenate very slow with +
- If you support older browsers, use array.join("") for concatenation

```
[].push(srt1,str2,str3,…).join("");
```

- Works like string builder
  - Slower in modern browsers

Telerik Academy

String Concatenation

Live Demo

Telerik Academy

Escaping

String Escape

- What is escaping?
  - Replacing reserved characters with their escape sequence
  - Prevents JavaScript injection
- When using JavaScript client-side reserved characters are '<', '>', '&', "'" and """

```
document.body.append(
"<script>"+
  "document.location = "http://bad_place.com"+
"</script>");
```

String Escaping (2)

- Escaping is done by just replacing the reserved characters with their escape sequence
  - Can be attached to the string prototype

```
String.prototype.htmlEscape = function (){
  var escapedStr = String(this).replace(/&/g, '&amp;');
  escapedStr = escapedStr.replace(/</g, '&lt;');
  escapedStr = escapedStr.replace(/>/g, '&gt;');
  escapedStr = escapedStr.replace(/"/g, '&quot;');
  escapedStr = escapedStr.replace(/'/g, "&#39");
  return escapedStr;
}
```

Escaping

Live Demo

Telerik Academy

Useful Extensions and Shims

Telerik Academy

String Extensions - Trim

- `string.trim()`, `string.trimLeft()`, `string.trimRight()`
  - Supported in all modern browsers
  - For older browsers use shim
- `string.trimChars(chars)`, `string.trimLeftChars(chars)`, `string.trimRightChars(chars)`
  - Trim no-whitespace characters
  - No native implementation

Trimming

Live Demo

Padding

- `str.padLeft(count [,char])`, `str.padRight(count [,char])`
  - Pads a string to the left/right
  - Fills the padding with whitespace or character
  - No native implementation

```
String.prototype.paddLeft = function (count, char) {
  char = char || " ";
  if(char.length > 1) return String(this);
  if(str.length >= count) return String(this);
  var s = String(this);
  for (var i = 0, len = this.len; i < count - len; i++) {
    s = char + s;
  }
  return s;
}
```

Padding

Live Demo

Telerik Academy

Strings

http://academy.Telerik.com

Exercises

1. Write a JavaScript function reverses string and returns it
   Example: "sample" → "elpmas".
2. Write a JavaScript function to check if in a given expression the brackets are put correctly.

   Example of correct expression: `((a+b)/5-d)`.

   Example of incorrect expression: `)(a+b))`.

Exercises (2)

3. Write a JavaScript function that finds how many times a substring is contained in a given text (perform case insensitive search).

   Example: The target substring is "`in`". The text is as follows:

```
We are living in an yellow submarine. We don't
have anything else. Inside the submarine is very
tight. So we are drinking all the day. We will
move out of it in 5 days.
```

The result is: 9.

Exercises (3)

4. You are given a text. Write a function that changes the text in all regions:
   - <upcase>text</upcase> to uppercase.
   - <lowcase>text</lowcase> to lowercase
   - <mixcase>text</mixcase> to mix casing(random)

```
We are <mixcase>living</mixcase> in a <upcase>yellow
submarine</upcase>. We <mixcase>don't</mixcase> have
<lowcase>anything</lowcase> else.
```

The expected result:

```
We are LiVinG in a YELLOW SUBMARINE. We dOn'T have
anything else.
```

Regions can be nested

Exercises (4)

5. Write a function that replaces non breaking white-spaces in a text with
6. Write a function that extracts the content of a html page given as text. The function should return anything that is in a tag, without the tags:

```
<html><head><title>Sample
site</title></head><body><div>text<div>more
text</div>and more...</div>in body</body></html>
```

result:

```
Sample sitetextmore textand more...in body
```

Exercises (5)

7. Write a script that parses an URL address given in the format:

```
[protocol]://[server]/[resource]
```

and extracts from it the `[protocol]`, `[server]` and `[resource]` elements. Return the elements in a JSON object.
For example from the URL `http://www.devbg.org/forum/index.php` the following information should be extracted:

```
{protocol: "http",
 server: "www.devbg.org",
 resource: "/forum/index.php"}
```

Exercises (6)

8. Write a JavaScript function that replaces in a HTML document given as string all the tags … with corresponding tags [URL=…]…/URL]. Sample HTML fragment:

```
<p>Please visit <a href="http://academy.telerik.
com">our site</a> to choose a training course. Also
visit <a href="www.devbg.org">our forum</a> to
discuss the courses.</p>
```

```
<p>Please visit [URL=http://academy.telerik.
com]our site[/URL] to choose a training course.
Also visit [URL=www.devbg.org]our forum[/URL] to
discuss the courses.</p>
```

Exercises (7)

9. Write a function for extracting all email addresses from given text. All substrings that match the format `<identifier>@<host>…<domain>` should be recognized as emails. Return the emails as array of strings.
10. Write a program that extracts from a given text all palindromes, e.g. "ABBA", "lamal", "exe".

```
var str = stringFormat("Hello {0}!","Peter");
//str = "Hello Peter!";
```

Exercises (8)

11. Write a function that formats a string using placeholders:

```
var str = stringFormat("Hello {0}!","Peter");
//str = "Hello Peter!";
```

The function should work with up to 30 placeholders and all types

```
var format = "{0}, {1}, {0} text {2}";
var str = stringFormat(format,1,"Pesho","Gosho");
//str = "1, Pesho, 1 text Gosho"
```

Exercises (9)

12. Write a function that creates a HTML UL using a template for every HTML LI. The source of the list should an array of elements. Replace all placeholders marked with –{…}– with the value of the corresponding property of the object. Example:

```
<div data-type="template" id="list-item">
 <strong>-{name}-</strong> <span>-{age}-</span>
/div>
```

```
var people = [{name: "Peter", age: 14},…];
var tmpl = document.getElementById("list-item").innerHtml;
var peopleList = generateList(people,template);
//peopleList = "<ul><li><strong>Peter</strong>
<span>14</span></li><li>…</li>…</ul>"
```

Telerik Academy

Free Trainings @ Telerik Academy

- "C# Programming @ Telerik Academy
  - csharpfundamentals.telerik.com

- Telerik Software Academy
  - academy.telerik.com

- Telerik Academy @ Facebook
  - facebook.com/TelerikAcademy

- Telerik Software Academy Forums
  - forums.academy.telerik.com