

JavaScript part 3

A scripting language based on
objects

1/30/2011

1

JavaScript - Pattern matching

Telephone numbers in Toronto?

416-123-1234

1234-234-12345

My name is :

Mathilda

Ma@#\$er

The postal code for Canada is:

Letter digit letter digit letter digit M2C 5B8

123X34

1/30/2011

2

JavaScript - Pattern matching

JavaScript has powerful pattern matching capabilities based on regular expressions. There are two approaches to pattern matching in JavaScript: one that is based on [RegExp](#) object and one that is based on methods of the [String](#) object. The regular expressions used by these two approaches are the same.

We use a pattern to look for specified characters in a string.

Example:

```
var str1 = "this is an example";  
var position = str1.search(/is/);
```

The simplest pattern matching method is [search\(\)](#), which takes a pattern as parameter. The search method returns the position in the String object where the pattern matched. If there is no match, search returns -1.

1/30/2011

3

JavaScript - Pattern matching

Two categories of characters in patterns:

- a. normal characters (match themselves)
- b. metacharacters (can have special meanings in patterns--do not match themselves)

`\ | () [] { } ^ $ * + ? .`

- a period is a special metacharacter - it matches any character except newline
the pattern `/liga.e/` matches **ligase**, **ligate**

- metacharacters can themselves be matched by being immediately preceded by a backslash.
the pattern `/2\.3/` matches **2.3**
the pattern `/2.3/` matches **2.3**, **243**, **2q3**, **223**

1/30/2011

4

JavaScript - Pattern matching

- Put a sequence of characters in brackets, and it defines a set of characters, any one of which matches

`[abcd]` matches a, b, c or d in the string

- Dashes can be used to specify spans of characters in a class

`[a-d]` matches a, b, c or d in the string

- A caret at the left end of a class definition means the opposite

`[^0-9]` matches any character distinct from decimal digits

1/30/2011

5

JavaScript - Pattern matching

Because they are often used some character classes are predefined and can be specified by their names.

Abbr.	Equiv. Pattern	Matches
<code>\d</code>	<code>[0-9]</code>	a digit
<code>\D</code>	<code>[^0-9]</code>	not a digit
<code>\w</code>	<code>[A-Za-z_0-9]</code>	a word character
<code>\W</code>	<code>[^A-Za-z_0-9]</code>	not a word character
<code>\s</code>	<code>[\r\t\n]</code>	a whitespace character
<code>\S</code>	<code>[^ \r\t\n]</code>	not a whitespace character

1/30/2011

6

JavaScript - Pattern matching

Quantifiers in braces

Quantifier	Meaning
<code>{n}</code>	exactly n repetitions
<code>{m,}</code>	at least m repetitions
<code>{m, n}</code>	at least m but not more than n repetitions

```
var str1 = "candy";
var str2="caandy";
var position1 = str1.search(/a{2}/);
var position2 = str2.search(/a{2}/);
```

1/30/2011

7

JavaScript - Pattern matching

*	means zero or more repetitions
<code>/\d*/</code>	means zero or more digits
<code>/5*/</code>	means zero or more 5
+	means one or more repetitions
<code>/\d+/</code>	means one or more digits
<code>/x+/</code>	means one or more x
?	means zero or one
<code>/\d?/</code>	means zero or one digit
<code>/4?/</code>	means zero or one 4

```
var str1 = "abba";
var str2="12345";
var position1 = str1.search(/1*c/);
var position2 = str2.search(/a+23/);
var position3 = str2.search(/\d+3/);
```

1/30/2011

8

JavaScript - Pattern matching

The pattern can be forced to match only at the left end with **^**; at the right end with **\$**

`/^Lee/` matches "Lee Ann" but not "Mary Lee Ann"

`/Lee Ann$/` matches "Mary Lee Ann", but not "Mary Lee Ann is nice"

The anchor operators (**^** and **\$**) do not match characters in the string--they match positions, at the beginning or end

The **i** modifier tells the matcher to ignore the case of letters

`/oak/i` matches "OAK" and "Oak" and "oAk" and ..

1/30/2011

9

JavaScript - Pattern matching

```
/\d.\d\d/  
/\D\d\D/  
/\w\w\w/  
/aw{3}z/  
/xy{5}bm/  
/x*y+z?/  
/\d+.\d*/  
/[A-Za-z]\w*/  
/\bis\b/  
/^pearl/  
/gold?/  
/gol?d/  
/pe^ar!/  
/\d+\s[A-Z][a-z]+/
```

1/30/2011

10

JavaScript - Pattern matching

The `replace()` method is used to replace substrings of the String object that match the given pattern.

`replace(pattern, string)`

The **g** modifier can be attached to the pattern if the replacement is to be global in the string, in which case the replacement is done for every match in the string. The matched substrings of the string are made available through the predefined variables `$1`, `$2`, and so on.

```
var str = "Some rabbits are rabid";  
str.replace(/rab/g, "tim");
```

str is now "Some timbits are timid"
variables `$1` and `$2` are equal to "rab"

1/30/2011

11

JavaScript - Pattern matching

The `match()` method takes a single parameter, a pattern. If the pattern has the **g** modifier, the returned array has all of the substrings of the string that matched. Without the **g** modifier, first element of the returned array has the matched substring, and the remainder of the array has the matches of parenthesized parts of the pattern, if there are any.

`match(pattern)`

```
var str = "My 3 kings beat your 2 aces";  
var matches = str.match(/[ab]/g);
```

matches is the array ["b", "a", "a"]

```
var str1 = "I have 234 dollars, but I need 555";  
var matches1 = str1.match(/(\d+)([^\d]+)(\d+)/);
```

matches1 is the array

```
["234 dollars, but I need 555", "234", "dollars, but I  
need ", "555" ]
```

1/30/2011

12

JavaScript - Pattern matching

The `split()` method of String splits its object string into substrings, based on a given string or pattern. The substrings are returned in an array.

`split(pattern)`

```
var str1 = "grapes:apples:oranges";  
var fruit = str1.split(":");
```

fruit is the array [grapes, apples, oranges]

JavaScript - Errors finding

Internet Explorer

- Select Internet Options from the Tools menu
- Choose the Advanced tab
- Uncheck the Disable script debugging box
- Check the Display a notification about every script error box
- Now, a script error causes a small window to be opened with an explanation of the error

FireFox

- Select Tools and Error Console
- A small window appears to display script errors
- Remember to Clear the console after using an error message – avoids confusion