Regex

Maher Khan

Recitation 4

Date: 21st September, 2018

slides and codes: https://github.com/maher460/Pitt-CS1520-Recitations

Office hours today from 6:00 PM to 8:00 PM

Use

Regular expressions are used to perform pattern matching
and "search-and-replace"
functions on text.

JS syntax (overview)

```
var re = /pattern/;

// boolean
re.test('string');

// null or Array [“whole match”, $1, $2, …]
re.exec(‘string');

//new string
re.replace(re, “string”)
```

Basic

/pattern/flags

- two forward slashes to show the beginning and end of the pattern

- flags: (optional) indicate how to search (modifier).
  Ex: /th/g (the g flag means "global" match - i.e. find all)

Example: Capital?

Write a JavaScript program to test the first character of a string is uppercase or not.

Test:

- Maher
- Pitt
- New York

Solution

regex = \^[A-Z]\

Syntax: Anchors

^	Matches the beginning of input	^a : “alice” not “banana” ^: “a line\nand break” => 2 match
$	Matches the end of input.	a$: “obama” not “name”
.	Matches any single character except a newline	a.b : “a0b”, “acb” not “ab”
\b	Word boundary	\bdog\b : “dog”, “dog ” not “doggy”
\B	Not word boundary	\Bdog\B : “doggy”, “adogg” not “dog”

Syntax: Anchors

\s	Whitespace (space, tab, newline,...)	\sc: matches “a cat” not “xcat”
\S	Not white space	\Sc: matches “cat” not “a cat”
\d	Digit	\d: matches “1” in “1abc”
\D	Not digit	\D: matches “b” in “123b”
\w	Word (alphanumeric character including _)	\w: matches “a” in “a%”
\W	Not word	\W: matches “%” in “a%”

Syntax: Quantifiers

*	Matches the preceding character 0 or more times	zo* : “z” and “zoo”
+	Matches the preceding character 1 or more times	zo+ : “zoo” not “z”
?	Matches the preceding character 0 or 1 time	ca?r : only “car” and “cr”
{n}	Matches exactly n times.	o{2} : “fooood” not “bob”
{n,}	Matches at least n times	o{2,} : “fooood” not “bob”
{n,m}	Matches the preceding character n to m times	o{2,3} : “food”, “foood” not “fod”

Syntax: Group & Range

|	or	(z|w)o : “zo” and “wo”
(...)	Matches subexpression and remembers the match	(ab): “abc” -> 1st group match “ab”
(?:...)	Matches subexpression and do not remembers the match (the match doesn’t in array)	(ab): “abc” -> No group
(?=...)	Matches only if there is a following pattern	win(?=7): “win” in “win7” not “win95”
(?!...)	Matches only if there is not a following pattern	win(?!=7): “win8” not “win7”
[...]	Range	c[ma-c]: “cm”, “cb” not “cd” c[A-E0-5]: “cB”, “c1” not c6
[^...]	Not in range	c[^ab]: “cc” not “ca”

Syntax: Modifier

g	global match (find all)	/is/g : “this is” => two match
i	Case-i-nse-nsitive	/my/i : “My” and “my”
m	Multiple line	/is/m : “this is”

Syntax: Replacement

$&	insert the whole regex match	"1a2b".replace(/\d+/g, "($&)") => "(1)a(2)b"
$1,… $9	Insert the text matched by one of the first 9 capturing groups.	"abc".replace(/(a)(b)(c)/g, "$3$2$1") => "cba"
$`	Insert the part of the subject string to the left of the regex match	"abc".replace(/b/, "$`") => “aac”
$'	Insert the part of the subject string to the right of the regex match	"abc".replace(/b/, "$'") => “acc”

Example: Regexing Emails

Write a pattern that matches e-mail addresses.
The personal information part contains the following ASCII characters:

- Uppercase (A-Z) and lowercase (a-z) English letters.
- Digits (0-9).
- Characters ! # $ % & ' * + - / = ? ^ _ ` { | } ~
- Character .(period, dot or fullstop) provided that it is not the first or last character and it will not come one after the other.

Solution

regexp = /^\w+([\.-]?\w+)*@\w+([\.-]?\w+)*(\.\w{2,3})+$/

Questions?