**Quiz 05:**

**1. The ‘$’ present in the RegExp object is called a**

[ ] a) character

[ ] b) matcher

[ ] c) metacharacter

[ ] d) metadata

**2. Consider the following statement containing regular expressions**

**var text = "testing: 1, 2, 3";**

**var pattern = /\d+/g;**

**In order to check if the pattern matches, the statement is**

[ ] a) text==pattern

[ ] b) text.equals(pattern)

[ ] c) text.test(pattern)

[ ] d) pattern.test(text)

**3. The regular expression to match any one character not between the brackets is**

[ ] a) […]

[ ] b) [^]

[ ] c) [^…]

[ ] d) [\D]

**4. What does /[^(]\* regular expression indicate ?**

[ ] a) Match one or more characters that are not open paranthesis

[ ] b) Match zero or more characters that are open paranthesis

[ ] c) Match zero or more characters that are not open paranthesis

[ ] d) Match one or more characters that are open paranthesis

**5. What will be the result when non greedy repetition is used on the pattern /a+?b/ ?**

[ ] a) Matches the letter b preceded by the fewest number of a’s possible

[ ] b) Matches the letter b preceded by any number of a

[ ] c) Matches letter a preceded by letter b, in the stack order

[ ] d) None of the mentioned

**6. What does the subexpression /java(script)?/ result in ?**

[ ] a) It matches “java” followed by the optional “script”

[ ] b) It matches “java” followed by any number of “script”

[ ] c) It matches “java” followed by a minimum of one “script”

[ ] d) None of the mentioned

**7. What is the most essential purpose of parantheses in regular expressions ?**

[ ] a) Define pattern matching techniques

[ ] b) Define subpatterns within the complete pattern

[ ] c) Define portion of strings in the regular expression

[ ] d) All of the mentioned

**8.** The method that performs the search-and-replace operation to strings for pattern matching is

[ ] a) searchandreplace()

[ ] b) add()

[ ] c) edit()

[ ] d) replace()

**9. What would be the result of the following statement in JavaScript using regular expression methods ?**

[ ] a) Returns [“123″”456″”789”]

[ ] b) Returns [“123″,”456″,”789”]

[ ] c) Returns [1,2,3,4,5,6,7,8,9]

[ ] d) Throws an exception

**10. Consider the following code snippet**

**var pattern = /Java/g;**

**var text = "JavaScript is more fun than Java!";**

**var result;**

**while ((result = pattern.exec(text)) != null)**

**{**

**alert("Matched '" + result[0] + "'" +" at position " + result.index +"; next search begins at " + pattern.lastIndex);**

**}**

**What purpose does exec() solve in the above code ?**

[ ] a) Returns the same kind of array whether or not the regular expression has the global g flag.

[ ] b) Returns different arrays in the different turns of iterations

[ ] c) Both a and b

[ ] d) None of the mentioned

Answers:

1. c

Explanation : The ‘S’ is a special metacharacter that matches the end of a string.

2. d

Explanation : The given pattern is applied on the text given in the paranthesis.

3. c

Explanation : The [^…] character class is used to match or draw any one character not between the brackets.

4. c

Explanation : We should always be careful while using \* and ? as repetition characters as they may match zero instances of whatever precedes them, they are allowed to match nothing.

5. a

Explanation : Using non greedy repetition may not always produce the results you expect. /a+?b/ matches the letter b preceded by the fewest number of a’s possible.

6. a

Explanation : The subexpression /java(script)?/ matches “java” followed by the optional “script”.

7. b

Explanation : When a regular expression is successfullyy matched against a target string, it is possible to extract the portions of the target string that matched any particular paranthesized subpattern. The essential purpose of parantheses in regular expressions is to define subpatterns within the complete pattern.

8. d

Explanation : The replace() method performs a search-and-replace operation. It takes a regular expression as its first argument and a replacement string as its second argument

9. b

Explanation : The split() method can take regular expressions as its arguments. The split() method generally breaks the string on which it is called into an array of substrings, using the argument as a separator.

10. a

Explanation : exec() returns the same kind of array whether or not the regular expression has the global g flag. Recall that match() returns an array of matches when passed a global regular expression. exec(), by contrast, always returns a single match and provides complete information about that match. When exec() is called on a regular expression that has the g flag, it sets the lastIndex property of the regular expression object to the character position immediately following the matched substring. When exec() is invoked a second time for the same regular expression, it begins its search at the character position indicated by the lastIndex property. If exec() does not find a match, it resets lastIndex to 0.