**正则问题check**

**1、给定正则表达式/^(SE)?[0-9]{12}$/，满足此匹配条件的字符串是：（）**

A、 "123456789123"

B、 "SI12345678"

C、 "1234567890"

D、 "ESX1234567Y"

**2、给定正则表达式 /^([1-9]|[1-9][0-9]|[1-9][0-9][0-9])$/，满足此匹配条件的字符串是：（）**

A、 "010"

B、 "0010"

C、 "127"

D、 "10000"

**3、给定正则表达式 /^[0-5]?[0-9]$/，满足此匹配条件的字符串是：（）**

A、 "99"

B、 "009"

C、 "0009"

D、 "10"

**4、给定正则表达式 /^\d+$/，满足此匹配条件的字符串是：（）**

A、 "12.345"

B、 "1,234.45"

C、 "+12.34"

D、 "1234567890"

**5、浮点数的正则表达式 是/[-+]?\b[0-9]\*\.?[0-9]+\b/，满足此匹配条件的恰好匹配的字符串是：（）**

A、 "1234.45"

B、 "1,234.45"

C、 "1+1=2"

D、 ".34"

**6.科学计数法浮点数正则表达式 /[-+]?\b[0-9]\*\.?[0-9]+(?:[eE][-+]?[0-9]+)?\b/，满足此匹配条件的恰好匹配的字符串是：（）**

A、 "1,234.45"

B、 ".34"

C、 "12."

D、 "-12.34"

**7、匹配一个字符串中两个相邻单词（它们之间可以有一个或者多个空白，如空格、制表符或者任何其他UNICODE空白符）的正则表达式是（ ）。** A、 /\b(\b+)\s+\1\b/

B、 /\b(\w+)\s+\b/

C、 /\b(\w\*)\s+\1\b/

D、 /\b(\w+)\s+\1\b/

**8、已知代码如下，则结果ns变量的值是（ ）。var s='The AAA is a good AAAA.'; var r=/A/; var ns=s.replace(r,'a');**

A、 The aAA is a good AAAA.'

B、 The aaa is a good aaaa.'

C、 The AAA is a good aAAA.'

D、 The aAA is a good aAAA.'

**9、已知代码如后面，则结果result[0]的值是（ ）。var url=/(\w+):\/\/([\w.]+)\/(\S\*)/; var text="Please visit my blog at http://zhuxianzhong.blog.51cto.com/"; var result=text.match(url);**

A、 http://zhuxianzhong.blog.51cto.com/'

B、 http'

C、 zhuxianzhong.blog.51cto.com'

D、 '

**10、给定字符串"<p>第二，3G资费起反作用。</p>"，求一正则式，能够匹配这当中的内容（ ）。**

A、 /<p>(.\*?)</p>/

B、 /<p>\*</p>/

C、 /<p>(\*)</p>/

D、 /<p>(\*?)</p>/

**11、软件开发中常用的匹配一个html标记的正则表达式是/</?[a-z][a-z0-9]\*[^<>]\*>/，则符合此格式要求的是（ ）。**

A、 <a<>

B、 <\>

C、 abc

D、 </body>

**12、已知"JavaScript".search(/a(.)a/);，则这个函数返回值是：（）**

A、 2

B、 3

C、 4

D、 1

**13、有关String的search函数，说法错误的是：（）**

A、 "JavaScript".search(/script/i);将返回子串起始位置4

B、 如果search的参数不是正则表达式，则首先会通过RegExp构造函数将它转换成正则表达式。

C、 search支持全局检索

D、 忽略参数regexp的lastIndex属性，总是从string的开始位置搜索

**14、String函数replace的语法格式是stringobject.replace(regexp/substr,replacement)。说法正确的是：（）**

A、 参数replacement不是必需参数。

B、 replacement可以是字符串，也可以是函数。

~~C、 replacement中的$字符没有什么特定的含义。~~

D、 ECMAScript第5版定义replace()的replacement参数可以用一个函数来代替字符串。

**15、String类match函数的语法格式是stringObject.match(searchvalue) 或者 stringObject.match(regexp)，下列说法错误的是：（）**

A、 如果regexp参数不是RegExp对象，则需要首先把它传递给RegExp构造函数，将其转换为RegExp对象。

B、 函数的返回值存放匹配结果的数组。该数组的内容依赖于regexp是否具有全局标志g。

C、 全局匹配返回的数组的内容与非全局匹配返回的数组内容是一样的。

D、 代码'1 plus 2 equals 3'.match(/\d+/g); 将找出一个字符串中的所有数字。

**16、"123, 456, 789".split(/\s\*,\s\*/);的返回结果是：（）**

A、 ["123","456","789"]

B、 ["123", " 456", "    789"]

C、 ["123", " 456 ", "    789"]

D、 ["123 ", " 456 ", "    789 "]

**17、有关RegExp类exec函数错误的结论是：（）**

A、 exec总是包含它返回的数组中的每一个匹配项的全部细节，无论regexp是否为一个全局模式。

B、 exec是所有RegExp和String模式匹配方法中最强大的一个。

C、 exec在string中搜索匹配regexp的文本。如果它找到一个匹配项，它将返回一个由匹配结果组成的数组；否则，它将返回null。返回数组的元素0是匹配的文本。

D、 不可以重复调用exec()，以便循环遍历一个字符串中所有的匹配项。

**18、有关RegExp类的test方法正确的结论是：（）**

A、 test用于测试字符串string是否包含匹配参数regexp的文本，如果包含则返回1；否则，返回0。

B、 /java/i.test("ECMAScript");将返回true

C、 r.test(s)等价于(r.exec(s)!=null)

D、 /java/i.test("JavaScript");返回false

**19、匹配一个英文句子（假设句子最后没有标点符号）最后一个单词的正则表达式有（ ）。**

A、 \b(\w+)\s\*$

B、 \b(\w+)\s+$

C、 \s(\w+)\s\*$

D、 \b(\w+)\b\*$

**20、已知Visa卡号可能有13位或者16位，且首位总是为4。则用于匹配Visa卡号的正则表达式有（ ）。**

A、 /^5[1-5][0-9]{14}$/

B、 /^5[1-5]\d{14}$/

C、 /5[^1-5][0-9]{14}$/

D、 /^(4\d{12}(?:\d{3})?)$/

**21、给定正则表达式 /\b0[xX][0-9a-fA-F]+\b/，满足此匹配条件的字符串有（ ）。**

A、 "0XDEADBEEF"

B、 "12345678"

C、 "deadbeef"

D、 "12"

**22、在HTML文件中经常遇到注释行，对应这种注释行的正则表达式是/<!--.\*?-->/，满足此匹配条件的字符串有（ ）。**

A、 "<html>"

B、 "<p>First paragraph</p>"

C、 "<a href="http://www.regexbuddy.com/">Link</a>"

D、 "<!--More boring stuff omitted-->"

**23、GUID数据用于标志对象的唯一性，其对应的正则表达式是/\b[A-F0-9]{8}(?:-[A-F0-9]{4}){3}-[A-F0-9]{12}\b/，则下面不符合此格式要求的假GUID数据有：（ ）**

A、 "12345678-90AB-CDEF-1234-567890ABCDEF"

B、 "X2345678-90AB-CDEF-1234-567890ABCDEX"

C、 "2345678-90AB-CDEF-1234-567890ABCDE"

D、 "1234567890ABCDEF1234567890ABCDEF"

**24、符合/\b([a-z0-9]+(-[a-z0-9]+)\*\.)+[a-z]{2,}\b/g的域名(或者其中的子串与之相匹配的)有（ ）。**

A、 www.regexbuddy.com

B、 www.regular-expressions.info

C、 http://www.cocoachina.com/

D、 <https://nodejs.org/>

**25、已知MasterCard信用卡必须包含16位数字。在这16个数字中，前两个数字必须是51-55之间的数字。则如下的正则表达式中合乎匹配MasterCard信用卡的有（ ）。**

A、 /^5[1-5][0-9]{14}$/

B、 /^5[1-5]\d{14}$/

~~C、 /5[^1-5][0-9]{14}$/~~

~~D、 /^5[1-5][0-9]{14,14}$/~~

**26、以下说法错误的有：（ ）**

A、 JS中如果正则表达式中出现“(?=p)”部分，它有可能是一种分组。

B、 JS中如果正则表达式中出现“(?!p)”部分，它有可能是一种分组。

C、 JS正则表达式理论中?（包括与其他字符相结合的情况）至少有四种含义。

D、 \B用于匹配单词边界的字符，与\b含义正好相反。

**27、IPv4的精确匹配的正则表达式为 /\b(?:(?:25[0-5]|2[0-4][0-9]|1[0-9][0-9]|[1-9]?[0-9])\.){3}(?:25[0-5]|2[0-4][0-9]|1[0-9][0-9]|[1-9]?[0-9])\b/，则与之相匹配的符合此表达式（或者其中的子串）有（ ）。**

A、 0.0.0.0

B、 255.255.255.255

~~C、 256.256.256.256~~

D、 127.0.0.1

**28、匹配一个html标记的正则表达式是/</?[a-z][a-z0-9]\*[^<>]\*>/，则符合此格式要求的有（ ）。**

A. <html>

B. </body>

C. </?a>

D. <br>