



链滴

表示数值的字符串

作者: [yudake](#)

原文链接: <https://ld246.com/article/1519049858104>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

题目描述

请实现一个函数用来判断字符串是否表示数值（包括整数和小数）。例如，字符串 "+100", "5e2", "-23", "3.1416" 和 "-1E-16" 都表示数值。但是 "12e", "1a3.14", "1.2.3", "+-5" 和 "12e+4.3" 都不是。

解题思路

正则表达式的实现。

用一个 hasNumber 标记是否已经存在有效数字，此时存在以下几种情况：

- 第一个字符如果是 "+/-"，直接跳过，进入下一字符；
- 本字符是 0，有以下几种情况：
 - 如果 hasNumber 存在有效数字，则进入下一字符；
 - 如果 hasNumber 无有效数字，则下一字符必须是 "."，且 "." 后有有效数字，进入子函数 isNumericHasDot
 - 如果下一位不是 "."，返回 false。
- 如果本字符是 "."，则之后必须存在有效数字；
- 如果本字符是 "e/E"，则有两种情况：
 - 如果本字符是最后一位，返回 false；
 - 如果不是最后一位，进入子函数 isNumericHasE。
- 其他情况，返回 false。

代码

```
public class Solution {
    public boolean isNumeric(char[] str) {
        if (str == null || str.length == 0)
            return false;
        boolean hasNumber = false;
        for (int i = 0; i < str.length; i++) {
            if (i == 0 && (str[i] == '+' || str[i] == '-'))
                continue;
            if (!hasNumber && str[i] == 0)
                if (i == str.length-1 || i == str.length - 2) {
                    return false;
                } else if (str[i+1] == '.') {
                    return isNumericHasDot(str, i+2);
                } else {
                    return false;
                }
            if (!hasNumber && str[i] > '0' && str[i] <= '9')
                hasNumber = true;
            if (hasNumber && (str[i] == 'e' || str[i] == 'E'))
                return (i+1 < str.length) && isNumericHasE(str, i+1);
            if (str[i] == '.')

```

```

        return (i+1 < str.length) && isNumericHasDot(str, i+1);
    if (str[i] < '0' || str[i] > '9')
        return false;
    }
    return hasNumber;
}

private boolean isNumericHasDot(char[] str, int index) {
    boolean hasNumber = false;
    for (int i = index; i < str.length; i++) {
        if (!hasNumber && str[i] >= '0' && str[i] <= '9')
            hasNumber = true;
        if (hasNumber && (str[i] == 'e' || str[i] == 'E'))
            return (i+1 < str.length) && isNumericHasE(str, i+1);
        if (str[i] < '0' || str[i] > '9')
            return false;
    }
    return hasNumber;
}

private boolean isNumericHasE(char[] str, int index) {
    boolean hasNumber = false;
    for (int i = index; i < str.length; i++) {
        if (i == index && (str[i] == '+' || str[i] == '-'))
            continue;
        if (!hasNumber && str[i] > '0' && str[i] <= '9')
            hasNumber = true;
        if (!hasNumber && str[i] == '0')
            return false;
        if (str[i] < '0' || str[i] > '9')
            return false;
    }
    return hasNumber;
}
}
}

```