

Regular Expressions Practice

Matching

Usernames

Some institutions issue usernames that follow the pattern "One or more lower-case alphabetic characters, followed by one or more digits." Which of the following regular expressions would match (only) valid usernames under that policy? You can assume that usernames are case insensitive, and that your regular expression only needs to match their lowercase form.

- | | |
|------------------------------|------------------------------|
| a. <code>\w\d</code> | d. <code>\w+\d+</code> |
| b. <code>[a-z]+\d+</code> | e. <code>[a-z]*\d+</code> |
| c. <code>^[a-z]+\d+\$</code> | f. <code>^[a-z]*\d+\$</code> |

If an institution didn't want to allow arbitrarily long usernames but, instead, only wanted to accept usernames with five or six alphabetic characters followed by one or two digits, how would your regular expression need to change?

Ook!

Ook! is an esoteric language with a limited set of legal instructions. Those commands can (with some simplification) be described by the regular expression: `r'^0ok\?0ok[.!]$|^0ok[.!]0ok[.!?]$'`

Based on the above definition, which of the following are valid Ook! instructions?

- | | |
|------------------------------|----------------------------------|
| a. <code>0ok.</code> | f. <code>0ok.0ok.</code> |
| b. <code>0ok.0ok!</code> | g. <code>0ok!0ok.</code> |
| c. <code>0ok?0ok!</code> | h. <code>0ok?0ok?</code> |
| d. <code>0ok0ok</code> | i. <code>0ok0ok0ok</code> |
| e. <code>0ok.0ok.0ok.</code> | j. <code>0ok?0ok!0ok.0ok!</code> |

Regular Expressions Practice

Writing regular expressions

In this part, you will write a regular expression to capture (a slightly simplified definition of) valid email addresses.

Usernames

Username

Names in email addresses can only contain the following characters:

```
ABCDEFGHIJKLMNOPQRSTUVWXYZ  
abcdefghijklmnopqrstuvwxyz  
0123456789  
._-
```

In addition, the following rules hold:

- A period `.` cannot be the first or last character
- Two consecutive periods are not allowed
- Must be at least one character long

Write a regular expression to capture valid email usernames:

Domain names

```
ABCDEFGHIJKLMNOPQRSTUVWXYZ  
abcdefghijklmnopqrstuvwxyz  
0123456789-.
```

- The first and last characters cannot be a dash `-` or period `.`
- Two consecutive periods are not allowed.
- There must be at least one period.
- The top-level (right-most) domain component must be a 2- or 3-character lower-case alphabetic string.

Write a regular expression to capture valid email domain names:

Regular Expressions Practice

Username and Domain

Write a combined regular expression to represent valid email addresses under our definition:

Invent your own

Write your own regular expression:

Describe your regular expression in English:

Fold your paper over so that only your English description is visible, then pass it to your neighbor. Have them try to write a regular expression capturing your description: