

BREAKING THE ICE

WITH
REGULAR EXPRESSIONS

Contents

01 The String Story

→ 02 Crew Emails

03 Confirmative

04 Multi-line Strings

05 Capture Groups

Analyze Our New Subject String

We now have our list of names for our voyage.



Let's work on
verifying their
emails!

Subject

sara@example.com

Starts with a word

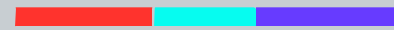
Followed by an @ character

Followed by another word

Starting the Regular Expression

Regular Expression

`/\w@\w/`



Remember: `\w` only searches for 1 word-like character.

Subject

`sara@example.com`



Pattern Goals

1 word character repeated 1 or more times

✓ @ symbol

1 word character repeated 1 or more times

Matching Full Words

Regular Expression

`\w+@\w+/\`

Subject

`sara@example.com`

Because the "." is not a word, it is not matched

Pattern Goals

- ✓ 1 word character repeated 1 or more times
- ✓ @ symbol
- ✓ 1 word character repeated 1 or more times
- dot literal
- com

A Successful Match

Regular Expression

`/\w+@\w+.\w+/'`

Subject

`sara@example.com`



we successfully matched!

Pattern Goals

- ✓ 1 word character repeated 1 or more times
- ✓ @ symbol
- ✓ 1 word character repeated 1 or more times
- ✓ dot literal
- ✓ com

Matching the .com

Regular Expression

`/\w+@\w+.\w+/
`

Subject

`sara@example.com`


`sara@example!com`




Pattern Goals

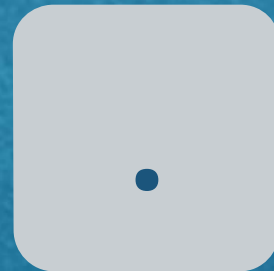
- ✓ 1 word character repeated 1 or more times
- ✓ @ symbol
- ✓ 1 word character repeated 1 or more times
- ✗ dot literal
- ✓ com



Hold up – an exclamation point shouldn't be matching here

The . Wildcard

The "." is a wildcard metacharacter



Escape the "." with a backslash to match a literal period



The "." wildcard matches any character except newline



Escaping a Character in a Pattern

using a backslash, we escape special characters

Regular Expression

`/\w+@\w+\.\w+/'`

Subject

`sara@example.com` ✓

`sara@example!com`

Pattern Goals

- ✓ 1 word character repeated 1 or more times
- ✓ @ symbol
- ✓ 1 word character repeated 1 or more times
- ✓ dot literal
- ✓ com

Great! The incorrect email is not matched.

Escaping Characters

+ → Matches a character 1 or more times

\+ → Matches literal “+” character

. → Matches any character except newline

\. → Matches literal “.” character

Being More Specific About the Top-level Domain (TLD)

Regular Expression

`/\w+@\w+\.[a-z]{2,4}/i`

4 common TLDs

Subject

`sara@example.net`



OR / alternation operators not evaluating properly

Pattern Goals

- ✗ 1 word character repeated 1 or more times
- ✗ @ symbol
- ✗ 1 word character repeated 1 or more times
- ✗ dot literal
- ✓ com or net or org or edu

Grouping Together TLDs

Regular Expression

`\w+@\w+\.(com|net|org|edu)/i`

The parentheses group our TLDs together

Subject

`sara@example.net`



Pattern Goals

- ✓ 1 word character repeated 1 or more times
- ✓ @ symbol
- ✓ 1 word character repeated 1 or more times
- ✓ dot literal
- ✓ com or net or org or edu

Problem: Need to Ignore Characters Before and After a Subject

Regular Expression

```
\w+@\w+\.(com|net|org|edu)/i
```

Subject

sara@example.comlksh



We get a partial match but shouldn't!
These characters should cause the match to fail.

Pattern Goals

- ✓ 1 word character repeated 1 or more times
- ✓ @ symbol
- ✓ 1 word character repeated 1 or more times
- ✓ dot literal
- ✓ matches com or net or org or edu
- ✓ no text before or after email

Introducing Anchors



`/^learnbydoing$/'`

There cannot be anything before here

There cannot be anything after here

`learnbydoing`

Adding Anchors to Our Pattern

Regular Expression

`/^\\w+@\\w+\\. (com|net|org|edu)$ /i`

start matching at the beginning of the line

stop matching at the end of the line

Pattern Goals

- ✓ 1 word character repeated 1 or more times
- ✓ @ symbol
- ✓ 1 word character repeated 1 or more times
- ✓ dot literal
- ✓ matches com or net or org or edu
- ✓ no text before or after email

Subject

`sara@example.com` ✓

`sara@example.comlksh`

Great - our invalid email is no longer matched!