ELIXIR & REGULAR EXPRESSIONS

Aamer F Rakla

67-495 – Advanced Topics in Information Systems/Innovation in Information Systems

WHAT ARE REGULAR EXPRESSIONS?

"In computing, a regular expression is a string that is used to describe or match a set of strings, according to certain syntax rules."

BASIC REGEX SYNTAX

- [] Range Specification
- \w word character
- \W non-word character
- \s whitespace
- \S non-whitespace
- \d decimal digit
- \D non-decimal digit
- \b word boundary
- \B non-word boundary
- * 0 or more of the preceding
- + 1 or more of the preceding

- {m} m of the preceding
- {m,} m or more of the preceding
- {m,n}m through n of the preceding
- ? 0 or 1 of the preceding
- . non-newline char
- ^ start of line
- $ end of line
- (a|z)//[az] a or z
- [^az] not a or z
- [a-z] a through z
- (foo) grouping

DEMONSTRATION

EXERCISE

Provide a regex pattern that will successfully match against the pass list but also not match those in the fail list.

Add the pattern that will pass all pass cases and not pass all fail cases listed below:

```
phonePattern = ~r/                                           /

_______________________________________________________________

phonePass = ["4123456789", "412-456-7890", "412.456.7890",
"(412) 345-6789", "412 345-6789"]

phoneFail = ["14123456789", "412-EAT-FOOD", "412.4567.890",
"345-6789"]
```

EXERCISE

Provide a regex pattern that will successfully match against the pass list but also not match those in the fail list.

Add the pattern that will pass all pass cases and not pass all fail cases listed below:

```
phonePattern = ~r/^\(?\d{3}\)?[ .-]?\d{3}[.-]?\d{4}$        /

_______________________________________________________________

phonePass = ["4123456789", "412-456-7890", "412.456.7890",
"(412) 345-6789", "412 345-6789"]

phoneFail = ["14123456789", "412-EAT-FOOD", "412.4567.890",
"345-6789"]
```

SOURCES

- 272 Slides: http://67272.cmuis.net/files/67272/lecture/slides/16/Ruby_and_Regex.pdf
- Elixir 1.2.5 Documentation: http://elixir-lang.org/docs/stable/elixir/Regex.html
- Elixir Quick Reference & RegEx Tester: http://www.elixre.uk/
- A Brief Introduction to Regular Expressions: http://chimera.labs.oreilly.com/books/1234000001642/apb.html
- Email RegEx: http://emailregex.com/