Regular Expressions

// # define regular expression in ruby

Name.match(/^[|da-z]+\d$/);

# RegExp Syntax

/a/ literal Character

\? Escape character literally

/.ob/ Wildcard Character (.)

/[RrBb]ob/ # Specific matching of the characters

/[aeiou]/ Character Class

/[a-z]ob/ Character Class (Range)

/[A-Fa-f0-9]/ # will match any of the characters within the brackets

/[^A-Fa-f0-9]/ #will match any character that isn’t those

/[0-9]/ == /\d/ #numbers

/a\*/ Operator/ Will match zero or more instanced of a

/.+/ matches 1 or more characters

/Jon \*Doe/

/a?/ It operates on the things previous to it, and match 0 or 1 on the previous

It will match not

/a{3}/ {} = one clause/statement Will on match the string “aaa”

/a{3,5}/ match a range

/^ruby/ Lines. Beginning of the expression

/ruby$/ Line: End of the line -> last character must be “y”

/(R|B)ob/ Or

/(R|B)+/ Will match RRRRR or BBBBB etc

/(?:R|B)+/ Not to capture what’s in the brackets

/(?: = Don’t capture the stuff

<words> can be used to name

/(?<day>\d{1,2})\/(?<month>\d{1,2})\/(?<year>d\{4})/ #named captures

## Capture Email. [.+@.+\.(com|net)](mailto:.+@.+\.(com|net))

### Phone Verification

(020\d{2})\s?(\d{6})

### Email Verification

(\w+@[a-zA-z\_]+?\.[a-zA-Z\_]{2,5})

### G.Sub()

>> "hello".sub('l', '\*')

=> "he\*lo"

>> "hello".gsub('l', '\*')

=> "he\*\*o"

Regex and Ruby