Regular Expressions

What are they and what can they do for you?

```
\b[A-Z0-9._%+-]+@[A-Z0-9.-]+\.[A-Z]{2-4}\b
```

Source: http://www.regular-expressions.info/email.html

So...what is a regular expression?

A Regular Expression is a pattern for a string.

What can you do with Regular Expressions?

Test a string

Extract a string

Change a string

Creating Regular Expressions

/RegularExpression/

Scenario #1
Testing Strings

“dog and cat”

"dog and cat"

/cat/

"dog and cat"

/cat/

/cat/ =~ "dog and cat"

=~

/cat/ =~ “dog and cat”

```
Last login: Mon Oct 29 13:02:14 on ttys003
20:52:57: ~$ irb
1.9.3-p194 :001 > /cat/ =~ "dog and cat"
 => 8
1.9.3-p194 :002 >
```

```
1.9.3-p194 :002 > /gerbil/ =~ "dog and cat"
 => nil
1.9.3-p194 :003 >
```

match

/cat/.match("dog and cat")

Valid Emails

thing@thing.thing

Rubular

Alternation
/Nell | Renee/

Metacharacters

- . any single character
- * character can appear any number of times

/.*/

matches ANYTHING

Range

[a-d] [1-4]

Shorthand!

/w stands for any word character.

It's the same as:
[a-zA-Z0-9_]

Metacharacters

+ character occurs one or more times

```
1.9.3-p194 :003 > def valid_email(email)
1.9.3-p194 :004?>   if /\w+@\w+\.\w+/ =~ email
1.9.3-p194 :005?>     puts "Valid Email"
1.9.3-p194 :006?>     else
1.9.3-p194 :007 >       puts "Invalid Email"
1.9.3-p194 :008?>     end
1.9.3-p194 :009?>   end
```

Scenario #2
Extracting Strings

Shorthand!

/d stands for any digit

/D stands for any non-digit

Repetition!

\d{3}

Looks for exactly 3 digits

Make It Optional!

-?

Means the hyphen is now optional. It can occur 0 or 1 time.

Scenario #3
Changing Strings

"Renee is the teacher of the class right now. Renee is teaching about regular expressions."

"Renee is the teacher of the class right now. Renee is teaching about regular expressions."

sub(/pattern/, “text”)

gsub(/pattern/, “text”)

(replace ALL the matches)

sub and gsub create new strings.

sub! and gsub! change the original strings

Welcome to the wonderful world of Regular Expressions!