WEEK 2: Regular Expressions

-- Code Blocks --
All functions in ruby take a hidden block argument T / F

Block Examples	How to call block argument
Block with no args: { puts “hello” }	yield
Block with 1 arg: { |x| puts x }	yield(“hello”)
Block with 2 args: { |x,y| puts (x+y) }	yield(1,7)

EX:

```
def f                          f { |x| puts (x ** 2) }       =>      25
      yield 5
end
```

What is the output each of the following methods that use code blocks?

```
a = [1,2,3,4]
h = {“a” => 1, “b” => 7}
s = 0
```

```
(1)   a.each { |x| s = s + x }
      puts s

(2)   h.each { |k,v| puts “#{k} is #{v}” }

(3)   a.map { |x| x * 2 }
```

-- Regex --
Define and test the structure of strings.

Literals:	/^hello$/	Only matches the string “hello”
	/hello/	Only matches strings with “hello” as a substring
Character classes:	/[A-Za-z]/	Matches any single alphabetic character
	/[0-9]/	Matches any single digit
	/\s/	Matches any single space character

Operators:	*	Can have zero or more of the preceding character, class, or literal
	+	Must have one or more of the preceding ----------
	?	Can have zero or one of the preceding ----------
	{x}	Matches exactly x occurrences of the preceding ----------

Special characters that represent something in regex must be escaped if you want to match a string that contains them. Special characters include ^, $, \, ?, *, +, etc. For instance, if you want to match a string that contains a dollar sign "$" you would write

/^.*\$.*$/

This reads: I can have any character (represented by the period) any number of times (including zero) followed by a dollar sign (escaped by adding a backslash in front of it) followed by any character any number of times. Strings matched by this expression include "$", "danny$", "$1.17", "GOOD$$BYE"

To extract information from a regex, surround the part of interest in parentheses or use scan:

```
str = "I am 21 years old and 6 feet tall"
if str =~ /I am (\d+) years old and (\d+) feet tall/ then
      puts "Age: #{$1} years old"   # the numbers correspond to the
      puts "Height: #{$2} feet"     # order of parentheses in the regex
end
str.scan(/I am \d+ years old and \d+ feet tall/)   =>   ["21", "6"]
```

Try to come up with a regular expression to match the following (reference rubular for help):

(a) A standard phone number with 10 digits "(123) - 456 - 7890"
(b) An email address ending with either gmail or yahoo
(c) Any sentence phrased as a question "How are you doing?"
(d) A bit string with a length that is a multiple of 4
(e) A string with any letter appearing more than twice "carrier" (**hard**)

-- Challenge --
Can you think of any form or pattern of string that cannot be matched by a regular expression?