Ruby Trivia 3
Question 1:

What is the value of the global variable $$_$?
Answer 1:
The String last read by gets.
Bonus Question:

How can you list all global variables?
Bonus Question:

How can you list all global variables?

Answer: Use the Kernel#global_variables method.
Bonus Question:

How many global variables does Ruby define?

global_variables.count
Bonus Question:

How many global variables does Ruby define?

global_variables.count

Answer: 54.
Question 2:

What does Ruby’s `-n` switch do?
Causes Ruby to assume the following loop around your script, which makes it iterate over file name arguments like `sed -n` or `awk`.

```ruby
while gets
  ...
end
```
Bonus Question:

What does Ruby's `-p` switch do?
Acts like the -n switch, but prints the value of variable \$_ at the each end of the loop. For example:

ruby -p -e \$_.tr! "a-z", "A-Z" < file
Question 3:

What thread-local variable can only store four possible values?
Question 3:

What thread-local variable can only store four possible values?

Hint #1: Those values are 0, 1, 2, and 3.
Question 3:

What thread-local variable can only store four possible values?

Hint #1: Those values are 0, 1, 2, and 3.
Hint #2: The value is 0 by default and can only increase.
Answer 3:

Trick question because it looks like a global variable, even though it behaves like a thread-local variable.
Answer 3:

There used to be $SAFE = 4 but it was removed in Ruby 2.1. Supposedly, it was only ever used by one company in Japan.
Bonus Question:

How can you check if an object is trusted?
Bonus Question:

How can you check if an object is trusted?

Answer: Use the Kernel#tainted? method.
module Kernel
  def tainted?
    return false
  end
end

Question 4:

What happens if you do this?
The tainted? method will always return false, but Ruby will still track tainted state via an internal FL_TAINT flag.
Question 5:

How can you mark a tainted object as safe?
Answer 5:

Kernel#untaint

a.k.a. Kernel#trust

$SAFE = 1
foo = gets.trust
eval(foo)
Thanks for playing!

Follow @sferik on Twitter for more Ruby trivia and practice.