Ruby master - Bug #14935

Inconsitent behaviour with puts and enumerator with different block format.

07/24/2018 12:29 PM - puneet.sutar@gmail.com (Puneet Sutar)

Status:	Rejected		
Priority:	Normal		
Assignee:			
Target version:			
ruby -v:	ruby 2.3.1p112 (2016-04-26 revision 54768) [x86_64-darwin16]	**Backport:**	2.3: UNKNOWN, 2.4: UNKNOWN, 2.5: UNKNOWN

Description

Case: 1

```
  ~ ruby
puts [1,2,3].map do |i|
  puts i
end
#<Enumerator:0x007ffe930b76d0>
  ~
# block does get evaluated.
```

Case 2:

```
  ~ ruby
puts [1, 2, 3].map { |i| puts i }
1
2
3



  ~
# Block gets evaluated.
```

My question is Shouldn't both cases give consistent output.

History

#1 - 07/24/2018 03:54 PM - shan (Shannon Skipper)

#puts returns nil, so when you're mapping you end up with [nil, nil, nil], which prints with #puts as three blank lines. If you use p instead of puts inside the block, the block will return i instead of nil and you'll see the same output.

#2 - 07/24/2018 05:13 PM - lucasbuchala (Lucas Buchala)

I think this is because do ... end blocks has lower precedence than {...} blocks. As documented in [1]. So, probably, not a bug.

```
method1 method2 { ... }     # method2's block
method1 method2 do ... end  # method1's block
```

[1] http://ruby-doc.org/core-2.5.1/doc/syntax/calling_methods_rdoc.html#label-Block+Argument

#3 - 07/25/2018 12:33 AM - nobu (Nobuyoshi Nakada)

- Status changed from Open to Rejected