Ruby master - Feature #15463

oneshot coverage does not allow counting code lines without coverage

12/25/2018 12:28 PM - grosser (Michael Grosser)

Status:	Open	
Priority:	Normal	
Assignee:		
Target version:		

Description

Problem:
I'm using https://github.com/grosser/single_cov to alert when uncovered lines where added to the codebase after each test run.

I'd love to use oneshot coverage for that to make it faster, but noticed that this does not work since there is no way to find out which lines of a given files are uncovered.

My naive implementation failed https://github.com/grosser/single_cov/pull/17 since to know what uncovered lines are I need to know which lines ruby considers code,
otherwise I'd be alerting on comments etc that are uncovered.

Example: loading this file would result in oneshot coverage of [2], loading an executing would result on [2,3]
but since I don't have a way of knowing that line 3 is code and supposed to be covered, I cannot show it as uncovered code.

```
# Foo <- not code and not covered = OK
def a <- code and covered = OK
1 <- code and not covered = FAIL
end <- not code and not covered = OK
```

Solutions:

- oneshot could be implemented as true/false/nil array for covered, uncovered, no-code (maybe oneshot_lines: :full)
- oneshot could restturn the same hash as regular coverage, but with only 1/0 (maybe oneshot_lines: :hash)
- allow lines + oneshot_lines enabled at the same time and make lines stop counting at 1

History

#1 - 12/25/2018 12:30 PM - grosser (Michael Grosser)

- Description updated

#2 - 12/25/2018 12:52 PM - mame (Yusuke Endoh)

Thank you for playing with oneshot coverage. You may want to use Coverage.line_stub.

```
$ cat t.rb
# Foo
def a
  1
end

$ ruby -rcoverage -e 'p Coverage.line_stub("t.rb")'
[nil, 0, 0, nil]
```

nil means that the corresponding line is uncoverable. 0 means that the corresponding line is coverable (lines ruby considers code).