

## Ruby 初級者向けレッスン第5回（解答）

かずひこ@株式会社 ネットワーク応用通信研究所

2006 年 1 月 28 日

### 演習問題

- テストコード (test\_stack.rb)

```
require 'mystack'
require 'test/unit'

class TestMyStack < Test::Unit::TestCase
  def setup
    @stack = MyStack.new
  end

  def test_empty?
    assert(@stack.empty?, 'a new stack is empty.')
  end

  def test_push_and_peek
    @stack.push(3)
    assert_equal(3, @stack.peak, 'peek returns the last value.')
  end

  def test_push_and_length
    @stack.push(3)
    assert_equal(1, @stack.length, 'push increments the length.')
    @stack.push(5)
    assert_equal(2, @stack.length, 'push increments the length.')
  end

  def test_push_and_empty?
    @stack.push(3)
    assert_equal(false, @stack.empty?, 'a stack with data is not empty.')
  end

  def test_push_push_pop_and_length
```

```
    @stack.push(3)
    @stack.push(5)
    @stack.pop
    assert_equal(1, @stack.length, 'pop decrements the length.')
  end

  def test_push_push_pop_and_peek
    @stack.push(3)
    @stack.push(5)
    assert_equal(5, @stack.peak, 'peek returns the last value.')
    @stack.pop
    assert_equal(3, @stack.peak, 'peek returns the last value.')
  end
end
```

- スタッククラスのコード (mystack.rb)

# Forwardable モジュールを用いたスタックの実装

```
require 'forwardable'
```

```
class MyStack
  extend Forwardable

  def initialize
    @s = [] # 委譲するオブジェクトの準備
  end

  # 同名メソッドへの委譲
  def_delegators(:@s, :push, :pop, :length, :empty?)

  # 別名メソッドへの委譲
  def_delegator(:@s, :last, :peek)
end
```