

*yancya* には  
*Refinements* が  
わからぬ

A investigative report of  
refinements

*yancya*

*RubyHiroba 2014*  
*2014/09/21*

# 自己紹介



● @yancya

● 何事も始めは「失敗」から始まる

# オープンクラス



## 特徴的な Ruby の機能

```
class Fixnum
  # 既存のクラスにメソッドを足したり
  def plusplus
    self + 1
  end
  # メソッドを上書きしたり出来る
  def succ
    self + 2
  end
end
```

# オープンクラス



```
(1..10).map{|n| n.succ}  
#=> [3, 4, 5, 6, 7, 8, 9, 10, 11, 12]
```

```
(1..10).map(&:succ)  
#=> [3, 4, 5, 6, 7, 8, 9, 10, 11, 12]
```

```
(1..10).map{|n| n.plusplus}  
#=> [2, 3, 4, 5, 6, 7, 8, 9, 10, 11]
```

```
(1..10).map(&:plusplus)  
#=> [2, 3, 4, 5, 6, 7, 8, 9, 10, 11]
```

# オープンクラス



こんなモンキーパッチばかり  
していると、プログラマがク  
ラスを信じられなくなってい  
まうぞ

# オープンクラス



- こんなモンキーパッチばかりしていると、プログラマがクラスを信じられなくなってしまうぞ
- yancya は「気をつける」のが苦手だろ

# Refinements



スコープを限定してモンキーパッチする仕組みがあるんだぞ

```
# module 内にモンキーパッチを書いておく
module PlusPlus
  refine Fixnum do
    def plusplus
      self + 1
    end

    def succ
      self + 2
    end
  end
end
```

# Refinements



using 以降の、同一ファイル内のコードにのみ、影響するんだ

```
# さっきのモジュールを using の引数に渡してやる  
using PlusPlus
```

```
(1..10).map{|n| n.succ}  
#=> [3, 4, 5, 6, 7, 8, 9, 10, 11, 12]
```

```
(1..10).map{|n| n.plusplus}  
#=> [2, 3, 4, 5, 6, 7, 8, 9, 10, 11]
```



# 色々試してみよう



```
using PlusPlus
```

```
(1..10).map(&:succ)
```

```
#=> [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
```

```
# 上書きされてない
```

```
(1..10).map(&:plusplus)
```

```
#=> in `each': super: no superclass method `plusplus'
```

```
#=> for 1:Fixnum (NoMethodError)
```

```
# そんなメソッドは無い
```

# そ、そんなはずは



```
using PlusPlus
```

```
# 動くかな
```

```
1.plusplus #=> 2
```

```
# よし、動くぞ
```

```
# plusplus ってあるよねー
```

```
1.respond_to? :plusplus
```

```
#=> false 「無いですけど?」「えっ」
```

# 何が悪いのか


---



- `&:symbol` っ  
て、`:symbol.to_proc.call`  
`(arg)` と同じだったような

# to\_proc, send



 やはり...

```
using PlusPlus
```

```
1.send(:succ) #=> 2
```

```
1.send(:plusplus) #=> NoMethodError
```

```
:succ.to_proc[1]    #=> 3
```

```
:plusplus.to_proc[1] #=> NoMethodError
```

# 気をつける



- メソッドが、どこのスコープで実行されるのか、きちんと気をつけないとダメ
- send などのメタプログラミングメソッドを使うと、スコープがファイル外へすっ飛んでいってしまうことがあるので

# 結論



- yancya に Refinements は、まだ早すぎる
- 気をつけなくてもキチンと書けるようになるか、そういう仕組みを作るしかない
- もうちょっと勉強してから出直してこよう