

# Webエンジニア フルタイムコース プロを目指すためのRuby入門 第8章

---

- 【問題】モジュールを理解する -



DIVE INTO CODE



## グループワーク 復習

---

チェリー本\_第8章 復習問題に取り組めます。

問題に答えたら、次のスライドに移動してください。次のスライドに答えが載っています。

全部で5問です。  
制限時間は10分です。

最後の2問はチェリー本 p.315 8.9 モジュールに関する高度な話題から出しています。じっくり考えてみてください。



# グループワーク 復習

---

問1

モジュールのインスタンスは作成できますか？

```
module Greeter  
  def Greeter  
    'Hello'  
  end  
end
```

```
greeter = Greeter.new
```

ヒント: チェリー本 p.284 8.2.2



# グループワーク 復習

---

答え

問1

できない。

#=> NoMethodError (undefined method `new' for Greeter:Module)と、エラーが返ってくる。

モジュールにはインスタンスを作成するためのnewメソッドが定義されていない。

**チェリー本 p.284 8.2.2を復習**



## グループワーク 復習

---

### 問2

ほかのモジュールを継承して新しいモジュールを作  
ることはできますか？

```
module AwesomeGreeter < Greeter  
end
```

ヒント: チェリー本 p.284 8.2.2



## グループワーク 復習

答え

問2

できない。シンタックスエラー(文法エラー)が返ってくる。

```
SyntaxError ((irb):1: syntax error, unexpected '<')  
module AwesomeGreeter < Greeter  
                        ^
```

継承できないというエラーではなく、文法エラーが返ってくる。エラーから原因が分かりづらいため、注意が必要。

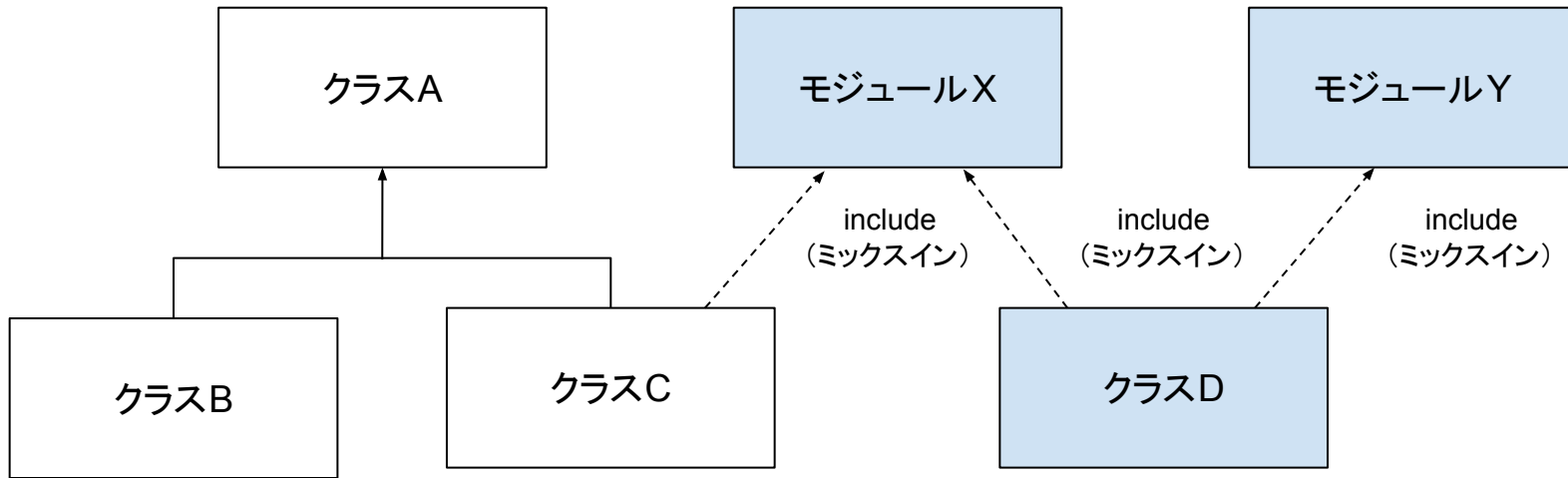
**チェリー本 p.284 8.2.2を復習**



# グループワーク 復習

## 問3

下記の図のように、クラスDはモジュールXとモジュールYを同時にミックスインできますか？  
ヒント:  
チェリー本 p.287 図8-1





## グループワーク 復習

---

答え  
問3  
できる。

ミックスインを利用することで多重継承に似たしくみを実現している。

**チェリー本 p.287 図8-1を復習**





# グループワーク 復習

問4 最後の式の戻り値は何になりますか？

```
module M
  def hoge
    puts "m"
  end
end
class A
  def hoge
    puts "a"
  end
end
class B < A
  include M
  def hoge
    puts "b"
  end
end
B.new.hoge
```

ヒント:チェリー本 p.317 図8-6



## グループワーク 復習

答え

問4

b

=> nil

クラスやモジュールがどの順番でメソッド探索されるかは、ancestorsメソッドで確認できます。includeされたモジュールは、クラスとそのスーパークラスの間に追加されます。

例:

B.ancestors

=> [B, M, A, Object, Kernel, BasicObject]

**チェリー本 p.317 図8-6を復習**



# グループワーク 復習

問5 最後の式の戻り値は何になりますか？

```
module M1 ; end  
module M2 ; end
```

```
class C1  
  prepend M1  
  include M2  
end
```

ヒント:チェリー本 p.319 8.9.3

```
C1.ancestors  
=>
```



## グループワーク 復習

---

答え

問5

C1.ancestors

=> [M1, C1, M2, Object, Kernel, BasicObject]

prependされたモジュールは、ミックスインしたクラスよりも先にモジュールが呼ばれます。

**チェリー本 p.319 8.9.3を復習**

# お疲れ様でした！

